WSRC-MS--92-269

DE92 040165

A DISCRETE ORDINATES APPROXIMATION TO THE NEUTRON TRANSPORT EQUATION APPLIED TO GENERALIZED GEOMETRIES (U)

by

M. D. DeHart

Westinghouse Savannah River Company
Savannah River Site
Aiken, South Carolina 29808

DISCLAIMER

A paper submitted to the Office of Graduate Studies at
Texas A&M University
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY
December, 1992

This report was prepared in connection with work done under Contract No. DE-AC09-89SR18035 with the U.S. Department of Energy.

A DISCRETE ORDINATES APPROXIMATION TO THE NEUTRON TRANSPORT EQUATION APPLIED TO GENERALIZED GEOMETRIES

A Dissertation

by

MARK DAVID DeHART

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 1992

Major Subject: Nuclear Engineering

DISCLAIMER

ABSTRACT

A Discrete Ordinates Approximation to the Neutron Transport Equation Applied to Generalized Geometries. (December 1992)

Mark David DeHart, B.S., Texas A&M University

M.S., Texas A&M University

Chair of Advisory Committee: Dr. Theodore A. Parish

A method for applying the discrete ordinates method for solution of the neutron transport equation in arbitrary two-dimensional meshes has been developed. The finite difference approach normally used to approximate spatial derivatives in extrapolating angular fluxes across a cell is replaced by direct solution of the characteristic form of the transport equation for each discrete direction. Thus, computational cells are not restricted to the traditional shape of a mesh element within a given coordinate system. However, in terms of the treatment of energy and angular dependencies, this method resembles traditional discrete ordinates techniques. Using the method developed here, a general two-dimensional space can be approximated by an irregular mesh comprised of arbitrary polygons.

The method of characteristics, originally developed to eliminate negative fluxes encountered in many finite difference approximations, had been previously applied to regular cell structures in multiple dimensions and various coordinate systems. In such geometries, the geometrical relationships between sides were determined analytically and incorporated directly into the numerical model. However, the present work makes no assumptions about the orientations or the number of sides in a given cell, and computes all geometric relationships between each set of sides in each cell for each discrete direction. A set of non-reentrant polygons can therefore be used to represent any given two dimensional space.

Results for a number of test problems have been compared to solutions obtained from traditional methods, with good agreement. Comparisons include benchmarks against

analytical results for problems with simple geometry, as well numerical results obtained from traditional discrete ordinates methods by applying the ANISN and TWOTRAN computer programs. Numerical results were obtained for problems ranging from simple one-dimensional geometry to complicated multidimensional configurations. These results have demonstrated the ability of the developed method to closely approximate complex geometrical configurations and to obtain accurate results for problems that are extremely difficult to model using traditional methods.

DEDICATION

This work is dedicated to my wife, Leigh. Her patience, understanding and love over the long road leading to this point made the path seem much less rocky; her encouragement and willingness to listen to my problems helped to sustain my efforts; and the daughter she has given us has taught me that miracles are possible.

ACKNOWLEDGEMENTS

I wish to express my sincere and profound gratitude to Ron Pevey and Ted Parish, both of whom provided invaluable support and insight in my endeavors. Their guidance, encouragement, technical assistance and senses of humor were invaluable assets.

I also wish to acknowledge the support given to me by the United States Department of Energy, the Westinghouse Savannah River Company, and most importantly, by several members of my direct management and by my colleagues within the Savannah River Technology Center, who provided support on various levels through the completion of this project. Most notable are Drs. John Menna, Al Garrett, Nick Kuehn, Courtney Apperson, Mel Buckner, and Joe Spencer, each of whom helped to make this project possible. The professionals and staff of the Applied Reactor Technology Group also provided significant feedback; Neal Askew, John Metzger, Gwen Gill and Bruce Hardy were always willing to listen to me think out loud.

For their hands-on assistance in helping me to learn, set up and run various programs, or for performing or assisting in many of the benchmark calculations, I especially wish to thank Chien-Hsiang Chen, Roger Webb, Sam Painter, James Taylor, and Khalid Mansour.

Finally, I wish to acknowledge and thank Drs. Marvin Adams, Erian Baskharone, Wesley Bolch, and Paul Nelson, for their help, encouragement, and valuable comments. I feel privileged to have had a research committee which took a proactive interest in my work, well beyond the minimum requirements of a committee member.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Background

The principal equation used in nuclear reactor analysis is the transport equation, a linearized derivative of the equation developed by Boltzmann for the kinetic theory of gases.[1] This equation may be used to determine the distribution of neutrons and/or photons in a reactor as a function of position, energy, direction and time. Typically, a solution for either a neutron or photon source distribution is sought in a given problem, and the coupling between the two is treated externally. This research will focus on the problem of neutron transport; however, the method will be equally applicable to photon transport problems.

The neutron transport equation may be presented in various forms, and simplifications are often applied to tailor the equation to the requirements of a specific application. In nuclear engineering applications, the transport equation is often written in terms of the angular neutron flux as the dependent variable. The angular neutron flux is defined as the product of the angular neutron density and the neutron velocity. The time-independent form of the linear transport equation is then expressed as[2]

$$\hat{\Omega}\cdot\vec{\nabla}\psi(\vec{r},\hat{\Omega},E)+\sigma_t(\vec{r},E)\psi(\vec{r},\hat{\Omega},E) = \int_{4\pi} d\hat{\Omega}'\int_0^{\infty} dE'\,\sigma(\vec{r},E'\rightarrow E,\hat{\Omega}'\rightarrow\hat{\Omega})\psi(\vec{r},\hat{\Omega}',E')+S(\vec{r},\hat{\Omega},E) \qquad (1)$$

where $\psi \equiv$ angular flux at $\vec{r}$ per unit volume, in direction $\hat{\Omega}$ per unit solid angle and at energy E per unit energy;

$\sigma_t \equiv$ total macroscopic cross-section for interaction at $\vec{r}$ and energy E;

$\sigma(\vec{r},E'\rightarrow E,\hat{\Omega}'\rightarrow\hat{\Omega}) \equiv$ transfer cross-section at $\vec{r}$ from original energy E′ and direction $\hat{\Omega}'$ to final energy E and direction $\hat{\Omega}$ (scattering and fission), per unit direction

and per unit energy; and

$S \equiv$ external neutron source at position $\vec{r}$ per unit volume, in direction $\hat{\Omega}$ per unit solid angle and at energy E per unit energy.

In general, the transport equation can be difficult to apply, and can be solved analytically only for highly idealized cases.[3] Hence, simplifications and numerical approximations are often necessary to apply the equation in engineering applications.

The methods for the numerical solution of the neutron transport equation can be divided into two types: (1) stochastic (Monte Carlo) methods and (2) deterministic techniques.[4] Within the deterministic class of methods, further categorization breaks the approaches into discrete ordinates and integral transport approaches.[5] Integral transport approximations are derived from the integral form of the transport equation, in which the angular dependence of the angular flux is eliminated by integrating out all angular variables, based on an assumed order of scattering.[3,4] This technique can be applied in complicated, heterogeneous geometries in multiple dimensions; however, the method does not treat anisotropic scattering well, since the solution becomes increasingly difficult for each additional order of anisotropic scattering introduced. Additionally, the matrices produced are usually full and thus difficult and computationally expensive to solve.[3-5] Discrete ordinates approaches are derived from the integrodifferential form of the transport equation, where space, time, and energy dependencies are normally treated by the use of a finite difference grid, while angular behavior is treated by considering a number of discrete directions. The angular solution is coupled to spatial scalar flux via some form of numerical integration. Because of the direct angular treatment, angularly dependent distribution functions can be computed, and treatment of anisotropic scattering is straightforward.

In typical reactor problems containing more than a single fuel cell, non-orthogonal geometries such as a hexagonal arrangement of fuel assemblies or curved interfaces are often encountered. Such geometries are difficult to model efficiently with a reasonable

degree of discretization using a traditional discrete ordinates approach. Finite difference schemes are generally limited to orthogonal geometries; in other words, irregular surfaces and other boundaries not conforming to the coordinate system cannot be treated in an exact manner, and must be approximated.[6] A remedy for this shortcoming is to utilize a finite element treatment for spatial variables while maintaining numerical integration in the angular solutions. One such approach has been to develop the differencing equations in terms of a triangular mesh. This allows a better treatment of irregular boundaries, but at the same time complicates the sweeping progression and the solution algorithm.[3,7] A full finite element representation of spatial and angular variables using an even-parity form of the transport equation has been accomplished, but among this method's weaknesses are 1) a limitation to isotropic or linearly anisotropic scattering and 2) as in the spatial finite element method, the complicated and time consuming algorithms for the solutions.[8]

The Method of Characteristics

A discrete ordinates representation of the transport equation can provide detailed information not readily computed by other methods, and can be a valuable tool in reactor analysis. As discussed earlier, however, efficient application of discrete ordinates methods is difficult when dealing with complicated, non-orthogonal geometries, due to the nature of finite difference approximations for spatial derivatives.

An alternative to the discrete representation of the spatial variable is achieved in the method of characteristics, in which the transport equation is solved analytically along characteristic directions within a computational cell. The angular flux ψ is solved along the s-axis, where this axis is oriented along the characteristic direction $\hat{\Omega}$. Since only the angular flux in direction $\hat{\Omega}$ is of concern, then the streaming term can be rewritten as:[9]

$$\hat{\Omega}\cdot\vec{\nabla}\psi(\vec{r},\hat{\Omega},E)=\frac{d\psi(s,E)}{ds}. \qquad (2)$$

The right hand side of the transport equation can be represented as a single total source term,

Q, comprised of the scattering, fission and external neutron sources, as follows:

$$Q(\vec{r},\hat{\Omega},E) = \int_{4\pi} d\hat{\Omega}' \int_0^{\infty} dE'\, \sigma(\vec{r},E'\rightarrow E,\hat{\Omega}'\rightarrow\hat{\Omega})\psi(\vec{r},\hat{\Omega}',E') + S(\vec{r},\hat{\Omega},E) \quad (3)$$

Using Equations 2 and 3, and omitting the energy dependence for clarity, Equation 1 can be reduced to the characteristic form:

$$\frac{d\psi}{ds} + \sigma_t\psi = Q \quad (4)$$

which has a solution of the form:[10]

$$\psi(s) = \psi_0 e^{-\sigma_t s} + e^{-\sigma_t s}\int_0^{s} Qe^{\sigma_t s'}ds' \quad (5)$$

where s is the distance along the characteristic direction, and ψ_0 is the known angular flux at s=0. Methods for the determination of an appropriate value for ψ_0 and for evaluation of the integral term vary in different solution techniques.[3,9,11-13] In numerical approximations, this equation is applied for a finite set of characteristic directions within a computational cell. ψ_0 is given from boundary conditions for known cell sides, and angular fluxes on unknown sides are computed using equation 5.

One of the simplest schemes employing the Method of Characteristics is the Step Characteristic (SC) method developed by Lathrop.[14] In this approach, the source Q and macroscopic total cross-section σ_t are assumed to be constant within a computational cell. Integration of equation 5 can be performed to obtain:

$$\psi(s) = \psi_0 e^{-\sigma_t s} + \frac{Q}{\sigma_t}(1 - e^{-\sigma_t s}). \quad (6)$$

Figure 1 shows a sample computational cell in which the SC method can be applied. For a given characteristic direction $\hat{\Omega}$, the angular flux on any unknown side may be expressed in

terms of a suitable average of fluxes from known sides which contribute to the unknown side. For the characteristic direction $\hat{\Omega}$ shown in Figure 1, the unknown "top" flux ψ_T may be computed as a linearly weighted average of contributions from known sides ψ_B and ψ_L. The fluxes on each of the two known sides are taken to be constant along the length of each side, representing the average angular flux in direction $\hat{\Omega}$, and must be specified from external boundary conditions or from a completed calculation in an adjacent cell.

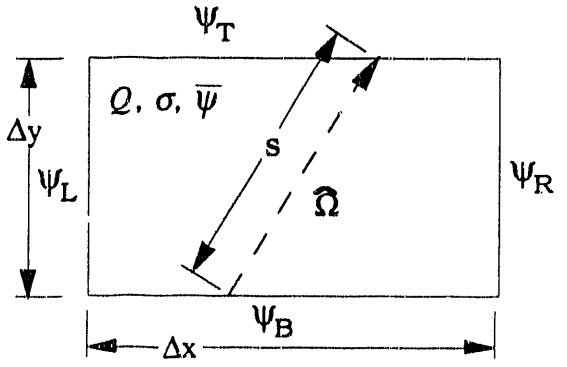


Figure 1. Typical Rectangular Cell Used in the Step Characteristic Approach

The set of characteristic directions is chosen from a quadrature set, so that the resulting angular fluxes may be numerically integrated to obtain a scalar flux. Knowing the lengths of the sides of a rectangular cell (Δx and Δy) and the direction cosines of $\hat{\Omega}$ in the x-y plane (μ and η), a function for the length s can easily be determined. The solution for $\psi(s)$ from equation 6 can then be integrated along the length of each unknown side to determine the unknown side's average angular flux. Once the angular flux is known on all four sides, a neutron balance on the cell can be used to determine the cell's average angular flux.[11]

Although the SC method described above is based on rectangular cells, the derivation of Equation 5 makes no assumptions about the shape of the cell. It merely requires a knowledge of the relationship between cell edges along the direction of the characteristic. Hence, the method is not be restricted to any particular geometry. Many other approaches utilize the method of characteristics, for a variety of different geometries. The following

section will also discuss some of the more significant approaches, and will describe their strengths and limitations. In addition, alternative methods for handling complex geometries will be described.

CHAPTER II

LITERATURE REVIEW

Discrete ordinates methods generally provide a closer approximation to the transport equation than integral transport and diffusion theory based methods, but often provide more information than is necessary in reactor calculations. However, in situations where the angular variation of the neutron flux is not well represented by an isotropic approximation, discrete ordinates methods yield more accurate solutions than can be obtained in other deterministic approaches. Geometrical limitations of traditional discrete ordinates methods have restricted the application of discrete ordinates methods in complicated geometries.[3] However, much recent work has attempted to modify spatial difference equations to eliminate or reduce geometric limits. Most of this work has involved variations in the approximations used in finite difference representations. The method of characteristics, although not originally applied for complicated geometries, has been shown to be effective in solution of the transport equation for numerous geometries. Applicability of characteristics methods has been limited, however, and therefore these methods are somewhat special purpose. This research will develop a discrete ordinates formulation, using the characteristics method, which can be applied in essentially any arbitrary geometry, by using polygonal cells to approximate irregular shapes.

The following section will describe current methods used in applying the discrete ordinates approach to generalized geometries. This will be followed by a review of the development and growth of characteristics methods as a tool in shielding and reactor analyses.

Discrete Ordinates Methods for Generalized Geometries

There exist numerous methods for solving the integrodifferential form of the neutron transport equation.[3,4] A large number of these methods use finite element techniques, primarily because of their capabilities to provide a better representation of irregular

geometries[3]; however, most finite element methods are actually based on a second-order formulation derived from the integrodifferential form. Methods using finite difference concepts have also been successfully applied to general non-orthogonal geometries. Both finite element and finite difference representations for generalized geometry problems will be discussed. The conceptual differences between these two normally distinct approaches can become unclear, however, since low-order functional relationships used in finite element methods can in some cases also be derived from a finite difference perspective. While the following review categorizes finite element and finite difference representations of the transport equation based on the approaches applied in their derivations, it should be recognized that in some cases the distinction is primarily one of semantics.

Finite Element Methods

Finite element approaches are often based on the solution of an even-parity second-order formulation of the transport equation[4]. This form is attractive in a finite element sense because the within-group equation can be formulated variationally to yield sparse and symmetric coefficient matrices, reducing the computational burden often associated with finite element methods. However, methods using the normal first order form of the transport equation have also been successfully applied in various geometries.

The variational formulation of the even-parity transport equation, originally derived by Vladimirov[15], is derived by separating the angular flux into even- and odd-parity components. Manipulations and substitutions result in a second order integrodifferential expression in terms of the even-parity flux. Under the assumption of isotropic scattering and source terms, the odd-parity component of the flux can eliminated, and the form of the equation is relatively simple. The isotropic assumption is not necessary; however, the form of the resulting anisotropic formulation includes both even- and odd-parity components, and is significantly more difficult to apply.[3] However, current ongoing work may have largely overcome this problem.

The even-parity approach typically solves both spatial and angular components within the finite element implementation; although in some approaches the angular components are treated through the use of discrete ordinates or spherical harmonics. Because of the symmetry of the equation, it requires half the angular directions used in traditional discrete ordinates methods for a given degree of resolution. Miller et al[8,16,17] have developed even-parity based finite element representations in one- and two-dimensional Cartesian coordinate systems. The method has been used in irregular geometries, and has also been shown to reduce the "ray effects" problem common to most traditional discrete ordinates approaches. However, this implementation assumes isotropic source and scattering behavior, and is therefore not as flexible as traditional methods. Use of specular reflection boundary conditions common to transport problems also complicates the formulation, and forces an implicit solution in the numerical implementation rather than the preferred iterative solution[3].

Lillie and Robinson[18] have developed an even-parity finite element model which incorporates anisotropic scattering, but the computational efficiency of this method is unattractive relative to simpler and significantly faster finite difference approaches. Because of the time and storage requirements for this approach, it cannot be effectively applied in large complicated problems.

The finite element method can also be applied to the ordinary first order transport equation.[3,4] The approach can be developed solely in terms of the spatial component of the transport equation, or in terms of both space and angle. In a purely spatial representation, angular dependencies are treated external to the spatial solution by integration over discrete angular solutions, as in standard discrete ordinates methods; however, treatment of inhomogenous boundary conditions is difficult, and the sweep order for cells must be carefully selected.[3] Coupled space-angle (phase-space) methods do not require a specific sweep order and treatment of boundary conditions is straightforward; additionally, as with the even-parity method, they have been shown to reduce the ray effect which can cause

problems for traditional methods. Continuous finite element approximations guarantee spatial continuity of the angular flux; however, discontinuous finite elements are usually used to approximate the angular variable, as this simplifies the treatment of boundary conditions. Discontinuous functions may be used for spatial variables to provide for better convergence, but care must be taken to ensure continuity of the angular flux.

The coupled phase-space finite element equations result in asymmetric matrices which do not lend themselves to iterative solutions, and numerical matrix inversion is necessary. In general, this form is less popular than the even-parity approach; spatial representations offer little over finite difference methods other than geometric flexibility, while coupled approaches are difficult and computationally expensive to solve.[3]

Nevertheless, the spatial finite element method has been successfully applied for triangular[19,20] and quadrilateral cells[21,22] in distorted meshes. Solution methods for Cartesian (x-y) and cylindrical (r-z) geometries have been demonstrated. Phase-space methods have also been successfully applied, primarily by French researchers.[3] These approaches are generally limited to x-y geometries, since in curved coordinate systems the orientation of a spatial cell can change for the same angular cell, resulting in instabilities.[22]

Finite Difference Methods

Morel and Larsen[23] have recently introduced a Multiple-Balance (MB) technique for differencing the transport equation. Standard discrete ordinates formulations use whole-cell auxiliary relationships to relate unknowns in the cell balance equations; the diamond difference approximation is one such relationship. In the MB approach, auxiliary equations are chosen which are approximations to balance equations for predefined subregions within each cell. The approximate balance equations for the sub-cells are coupled by the definition of the scattering and inhomogenous source terms. The choice of the source term definitions can be used to develop a variety of different schemes, similar in nature to various whole-cell auxiliary equations. A derivation of source scalar fluxes based on a diffusion limit flux has

been shown to provide second order accuracy and positive solutions. Although the developers have at this point only applied the approach within a one-dimensional slab geometry with isotropic scattering, the method should be readily extendable to anisotropic scattering in general multidimensional geometries.

Adams[24] has used the multiple-balance philosophy to develop a Corner Balance (CB) spatial discretization method for arbitrary cells within X-Y geometries. In this approach, a two-dimensional problem domain is represented by arbitrary polygonal cells. Each cell is divided into quadrilaterals, one for each corner of the polygon, defined by the center points and the included corner of two adjacent lines and a vertex in the interior of the cell. Approximate relationships based on the relationship of cell sides to a given angular direction are then used to determine side-averaged fluxes for each quadrilateral, which are coupled to the sub-cell average angular flux via an approximate balance equation. Closure relationships can then be used to solve the exact whole-cell balance equation. The method has been most heavily tested using rectangular grids, and has been show to perform well. However, analysis has shown that the solution will degrade as the spatial grid is distorted from its rectangular starting point. Ongoing work is studying the necessary closure relationships for distorted cell configurations.

An S_n method using General Quadrilaterals (GQ) in rectangular (x-y) and cylindrical (r-z) geometries has been developed by Alcouffe for radiation transport.[25] This approach is based upon three distinct differencing schemes, depending on the number of quadrilateral sides which can be "seen" from a given discrete direction; hence the differencing scheme changes with respect to the direction of the angular flux component being computed. For a single visible side, a step approximation is used, while one- and two-dimensional diamond-difference approximations are used for two and three sides visible respectively. The method is not guaranteed positive, and corrects negative flux values by setting them to zero. As the quadrilaterals become rectangles, the approach reduces to a simple diamond-difference

approach. The GQ method is simple, fast, and is compatible with popular diffusion synthetic acceleration techniques. However, the approach has not yet been applied to neutron transport problems.

Development of Characteristics-Based Methods

The method of characteristics has been primarily applied in two classes of discrete ordinates problems: deep penetration shielding calculations and reactor calculations . The former, described briefly below, uses the characteristics method to take advantage of the streaming behavior of neutrons in large absorbing regions, but is not efficient in eigenvalue problems or in strongly fissioning or scattering media.[26] Of more interest in this work are methods developed for reactor applications, several of which were based on Lathrop's step characteristic method. Various approaches have been used to extend the SC method: curvilinear and spherical geometries; three dimensional problems; and higher order representation of boundary flux, cell source and cross-section approximations within computational cells. Many of the reactor applications were developed in an effort to guarantee positive fluxes given positive boundary conditions and sources, while others were developed in an effort to obtain more accurate solutions or to extend methods to more complicated geometries.

Characteristics Methods in Shielding Applications

Takeuchi has used the method of characteristics in Cartesian, spherical, and two-dimensional (r-θ) cylindrical geometries to solve the transport equation for deep penetration shielding problems.[27,28] In this approach, fluxes are computed at the vertices of a cell. In a two-dimensional problem, it is assumed that the source is known at all four vertices and that the angular flux is known at three vertices. A sample cell for a problem in Cartesian coordinates is shown in Figure 2, where the angular flux at vertex D is unknown. For each characteristic direction, a line is drawn back from the unknown vertex to intercept the side

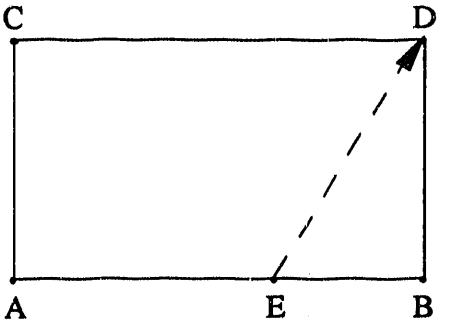


Figure 2. Cartesian Cell Configuration for Takeuchi's Method.[26]

between two known vertices, as shown by line DE in the figure. The flux and source at point E are calculated by linear interpolation between known values as A and B. The source is then assumed to vary linearly between E and D, and direct integration of Equation 5 is possible, to obtain a value for the flux at point D.

Takeuchi's method is simple, and can easily be extended to any geometry or cell arrangement.[9] However, it does not enforce a particle balance within computational cells, limiting the applicability and accuracy of the method.[9,26] Acceleration techniques and eigenvalue solutions used in reactor physics methods often rely on particle conservation to obtain an accurate and efficient solution,[26] hence the method has limited applicability in reactor calculations. However, the method has been shown to be advantageous in shielding calculations involving large absorbing regions, because it treats the streaming operator analytically. The method has been implemented in the PALLAS series of computer codes for shielding applications.[29]

Characteristics Methods in Reactor Applications

The Step Characteristic method of Lathrop, discussed earlier, employs a particle balance in computing the cell average fluxes, making it a considerably more valuable tool in reactor physics applications. This technique was derived in an effort to develop a method which would avoid the problem of negative fluxes, which could result when applying a diamond

difference (DD) type finite difference scheme for spatial derivatives. Negative fluxes which can occur in DD solutions have been shown to slow iteration convergence rates and to sometimes cause the failure af acceleration techniques, and can introduce instabilities in time dependent solutions.[11] The essential aspects of the SC method were described in the previous chapter, and will not be repeated here. This method has been shown to be in good agreement with DD solutions, and the computational time and storage requirements for an individual cell are essentially the same for both methods.[12] Additionally, theoretical cell average errors are $O(\Delta x^2)$ for both methods[30], although numerical experiments indicate that the DD method is slightly more accurate for slab geometries.[12] The order of the SC error has been shown to depend on the direction of the characteristic and the aspect ratio of the cell, i.e., the ratio of cell height to width. Thus, the error term is slightly greater than $O(\Delta x^2)$.[14]

A Linear Characteristic (LC) method has been developed,[13] based on the SC method, by assuming a linearly varying source across the cell. The source is assumed to be represented by the cell average value at the geometric center of each cell, but varies by a linear gradient computed from scalar fluxes at each vertex of the cell. Using a linear expression for Q(s), Equation 5 can be integrated to obtain an analytic expression for $\psi(s)$, and the geometric relationships of the SC method are applied to determine edge and cell average fluxes. This technique has been shown to be extremely accurate, and theoretical cell average flux errors are $O(\Delta x^4)$ (Ref. 30). In application, the error is considerably larger, but the LC method remains more accurate than the DD approach.[13] However, storage requirements per cell are considerably larger than for DD or SC methods, and approximately twice the computational time is required to complete a calculation foı identical mesh sizes.[12] However, because of the increased accuracy larger cell sizes are possible, reducing the storage and computing time costs.

A quadratic representation of the source distribution has also been developed for one-

dimensional Cartesian problems.[31] In this Quadratic method (QM), the spatial variation of the source term Q is approximated by a second order polynomial. For a given computational cell, the cell average flux in the cell and in the two adjacent cells are used to determine the coefficients of the polynomial. Then, as in the previous method, an analytical expression for $\psi(s)$ can be derived by integration of Equation 5. This method produces cell average flux errors $O(\Delta x^6)$ for isotropic scattering, but approximations typically introduced in solving polynomials for higher order scattering terms degrade the error to $O(\Delta x^4)$. Storage requirements are approximately 50% greater than for the LC method, but computational cost is approximately the same. However, the solution method becomes very complex in multiple dimensions or more complicated geometries.[30]

While Takeuchi used a solution for fluxes at cell vertices, each of the above methods (SC, LC, and QM) solved for averaged edge fluxes. A method developed by Wagner et al[32] combines both techniques in a three-dimensional Cartesian system. In this approach, a volume is represented by a set of identical three-dimensional rectangular parallelepipeds of width w, height h, and depth d. Characteristic directions are derived from the geometry of the cell, by lines drawn from the geometric center of the cell to the center of each of the six faces and to each of the eight corners or vertices. This set of 14 directions represents a quadrature set, for which a consistent set of weights are derived. The source is assumed constant in each cell, and Equation 6 can be used to solve fluxes at unknown faces and vertices given appropriate boundary conditions. A cell balance is performed to compute the cell averaged flux, thus conservation is enforced. Because all cells are the same size and shape, coefficients are identical for each cell, and storage is small. The method is rapid, but is limited to simple geometries. Also, when scattering is important, cells must be small because of the constant source approximation; for three-dimensional problems, this can result in an extremely large number of cells. Finally, it is limited to a relatively low order angular approximation due to its fixed 14-direction set of characteristics.[9]

Askew[33,9] has successfully applied the method of characteristics in more complicated geometries. In this technique, a mesh of cells is overlaid by a series of parallel lines for a given characteristic direction, with at least one line intersecting each cell. Each cell is assumed to have constant properties Each of the parallel lines has associated with it an area determined by the line spacing; these areas are illustrated in Figure 3 by boxes surrounding each of three characteristic lines for an arbitrary direction. Within each of theses areas, an average flux is computed by integration of Equation 6 along the characteristic line from a known boundary to an unknown boundary. The cell's average flux is then computed by a area-weighted average for each of the areas in the cell. Since the sum of the "characteristic areas" is unlikely to equal the cell area, the areas are modified in terms of the length and weight of the associated characteristic lines The averaging process then completes a neutron balance for the cell. Cells are swept by the solution algorithm in the direction of the characteristic. Although this method has been successfully applied for more complicated geometries than previous methods, it still possesses severe limitations. The set of characteristic directions and line spacing must be carefully selected for a given problem geometry, and the order of a quadrature set is not easily changed. Restrictions resulting from orientations of characteristic lines limit the complexity of the problem geometry.

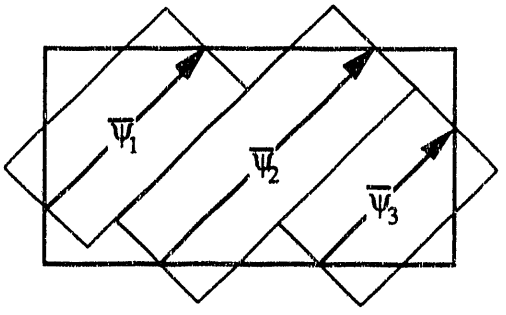


Figure 3. Example Cell and Characteristic Area Used in Askew's Method.[11]

Askew's approach has been included in the CACTUS[34] computer program, which has been used in the United Kingdom in reactor analysis. In a review of characteristic methods

developed by 1981, Alcouffe[9] stated "its geometric flexibility is closer to that of Monte Carlo codes than any other code that has come to our attention." However, the code is also described as being much less efficient than general purpose transport codes, and is recommended only for special applications.

Filippone[35] has independently combined aspects of Askew's approach with a diamond difference approach to take advantage of the strengths of both methods. The angular flux is broken into collided and uncollided flux components. A DD method is used to solve the collided flux within the cell given the source and zero flux boundary conditions, while a characteristic approach similar in nature to that of Askew is used to solve for the uncollided (or streaming) flux in the cell, with specified boundary conditions and no source term. This method is strictly positive, has been shown to give better results than either an SC or DD approach for a given mesh and angular discretization. However, there is substantially more computational overhead in such an approach, and the method is limited to the same orthogonal geometries which limit the generality of DD methods.

CHAPTER III

THEORY

In order to formulate a model for implementation of the method of characteristics in a generalized geometry, approximations must be made in order to reduce the transport equation to a solvable form. Using these approximations, it becomes possible to represent the transport equation is such a way that it can be easily applied in a numerical model. This chapter will introduce the the basic premises upon which the model developed in this work is based, the rationale for each of the approximations made in developing the model, and the limitations imposed by each approximation. Next, the derivation of a method to compute the angular flux on an unknown boundary will be presented. Geometric orientations of the sides of a cell relative to an arbitrary direction vector will be discussed, in order to provide the relationships necessary to complete the computation of boundary fluxes. The mapping between the two-dimensional cell representation and a given three-dimensional characteristic direction will be discussed. Next, a neutron balance will be applied in the development of an expression for a cell averaged angular flux. Finally, a discretized form of the integral relationship between angular fluxes and the cell source term Q will be developed.

Fundamental Concepts and Assumptions

In order to formally describe the derivation of the set of equations used in discretizing the transport equation, basic assumptions must be made in order to lay the groundwork upon which the method can be based. The following paragraphs will define the properties of a computational cell, interrelationships between cells and relationships between cells and the problem domain, and the basis for characteristic directions used. These principles will be used in the derivation of cell-flux relationships presented in the next section of this chapter.

Cell Properties and Geometries

As was discussed in Chapter I, the Step Characteristic method of Lathrop is based on a geometry represented by a set of rectangular computational cells. Within each cell, properties are assumed uniform and constant during an inner iteration in order to solve for the spatial distribution of the angular neutron flux. Based on the assumption of constant properties, the characteristic solution for a cell is then found to be simple and straightforward, as given by equation 6. However, as was noted earlier, this equation makes no assumptions about the shape of the computational cell, and will give an exact solution for the flux at any point s when ψ_0 is known and properties are constant. Thus, the SC method is chosen as a starting point for development of this model. Because it is an extension of the SC approach into generalized cells, the method developed in this work is referred to as the Extended Step Characteristic (ESC) method.

The primary assumption in this model will therefore be that within each computational cell all properties, i.e., σ_t and Q, are uniform. The characteristic formulation for the transport equation (Equation 4) is then a simple non-homogeneous first order differential equation in the characteristic coordinate s. However, in a practical problem, properties are unlikely to remain constant over significant volumes. Thus, this approach, like many other differencing schemes, is a poor approximation when cell volumes become too large; σ_t is a material property and may be constant, but Q, which depends on the neutron flux, will vary with position. However, since the solution would become exact in an infinitesimally small cell, it is expected that the approximation will be reasonable for computational cells in which the change in the flux (and therefore the source) is small over the domain of the cell.

Because properties are assumed constant in a cell, there will be a step change in properties at each cell boundary. Therefore, because the solution to the characteristic equation is along a straight line, one must guarantee that any straight line drawn across a cell remain within the cell for properties to remain constant. Otherwise, the assumptions

inherent in the solution of the characteristic equation are violated. In order to meet this requirement, each cell must be required to be non-reentrant. Furthermore, because a curved boundary which is non-reentrant in a given cell must be re-entrant in any cell sharing that boundary, cell boundaries must be defined by straight lines. Thus, computational cells are restricted to non-reentrant polygons. No restrictions are placed on the number of sides or the shape of the polygon, as long as it remains non-reentrant.

As a result of this geometric configuration, each side of a cell can have one of three possible attributes relative to neutrons travelling in a given characteristic direction, as illustrated in Figure 4: 1) neutrons can enter the cell when crossing a side (as shown by sides D, E and F in the figure); 2) neutrons can exit the cell when crossing a side (sides B and C); or 3) in a special case, neutrons will never cross a given side (side A). Expressed mathematically, these relationships become:

$$\text{Category 1:}\quad \hat{\Omega}_k \cdot \hat{n}_i < 0; \tag{7a}$$

$$\text{Category 2:}\quad \hat{\Omega}_k \cdot \hat{n}_i > 0; \tag{7b}$$

$$\text{Category 3:}\quad \hat{\Omega}_k \cdot \hat{n}_i = 0; \tag{7c}$$

where $\hat{n}_i$ is a unit vector in the cell-outward direction normal to side i, and $\hat{\Omega}_k$ is the kth discrete element of a set of characteristic directions. A category 1 side will be termed an "incoming" side with respect to the direction $\hat{\Omega}_k$, and a category 2 side will be referred to as an "outgoing" side. For simplicity, the definition of equation 7c will be included as a special case of equation 7a for an incoming side. Thus, equation 7 can be rewritten as:

$$\text{Side i is "incoming" with respect to } \hat{\Omega}_k\text{:}\quad \hat{\Omega}_k \cdot \hat{n}_i \le 0; \tag{8a}$$

$$\text{Side i is "outgoing" with respect to } \hat{\Omega}_k\text{:}\quad \hat{\Omega}_k \cdot \hat{n}_i > 0. \tag{8b}$$

In order to be able to solve for fluxes on outgoing sides of a cell, it will be shown that fluxes on all incoming sides must be known. Each incoming side of each cell will be given from a boundary condition or will be the outgoing side of an adjacent cell.

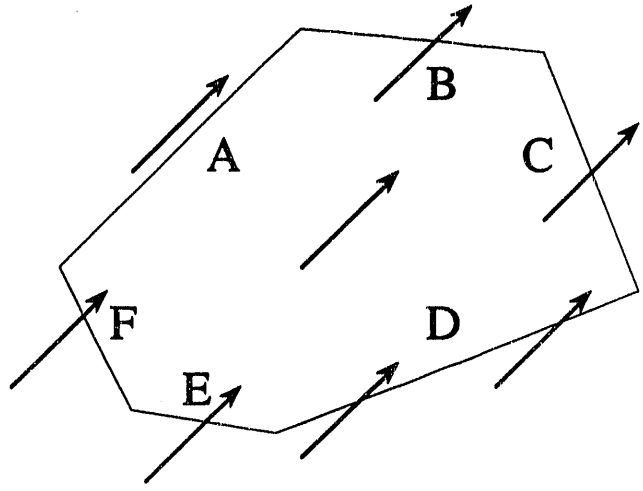


Figure 4. Orientation of the Sides of a Cell With Respect to a Given Direction Vector.

Relationships Between Cells

In the ESC method, the shape of the computational cell and the form of the neutron balance differ from that used in standard discrete ordinates methods. Nevertheless, the relationships between cells are treated essentially as they would be in standard approaches. The entire problem domain is mapped in terms of a set of finite cells. Each side of each cell is adjacent to either an external boundary condition or another cell. For each discrete direction, cells are swept in a predetermined order beginning at a known boundary (from a specified external boundary condition) moving in the given direction. The precise order of sweep is such that as the solution for one cell is obtained, the cell provides sufficient boundary conditions for the solution of an adjacent cell. Hence, cells sharing a given side share the value of the angular flux on that side. Knowledge of the flux on all incoming sides of a cell is sufficient to solve for all outgoing sides. Once the angular flux has been determined for all sides of the cell for the given direction, it is possible to use a neutron balance to compute the average value of the angular flux within the cell.

The sweeping of cells continues for a given direction until all cells have been calculated. The procedure is then repeated for the next direction, until all directions have been computed. At this point, the cell average angular fluxes are known for each cell for each

direction used. Numerical quadrature can then be used to determine the average scalar flux in each cell in the problem domain. The scalar fluxes are used to determine fission and scattering reaction rates in each cell, to update the value of the cell average source, Q. The process is repeated, and the iteration continues until all scalar fluxes converge to within a specified tolerance.

Like standard discrete ordinates methods, this approach can be performed assuming a single energy group, or any number of discretized energy groups. The multigroup approach used in the ESC method is the standard approach used in most methods, and is independent of the shape of each computational cell. Hence, the details of the multigroup formalism will be omitted from this discussion.

The Set of Characteristic Directions

The characteristic solution to the transport equation only gives the angular flux in the direction of the characteristic. In order to compute interaction rates within a cell, scalar fluxes must be computed. In order to easily compute the scalar flux from the set of angular fluxes, it is convenient to choose the set of characteristic directions from an appropriate quadrature set. Then, the set of computed angular fluxes can be combined with appropriate directional weights and summed to obtain a scalar flux solution within a cell. Therefore, it is most appropriate to choose characteristic directions from an established set of base points and weights. Such quadrature sets have been developed and used in numerous earlier discrete ordinates approaches, and will be used in this development. No restriction is placed on the nature or order of the quadrature set, as long as it is sufficient to adequately represent the scalar flux from computed angular fluxes.

Angular Flux at a Cell Boundary

As in the development of the SC method, as well as most finite difference methods, the ESC approach will not explicitly determine the flux distribution as a function of position

along the sides of a computational cell. Instead, the angular flux on each cell side will be represented in terms of the average angular flux along the length of the side. This will be sufficient to determine the net leakage across each cell side, which is needed in order to maintain a cell balance. An average value of the flux for an incoming side must be specified from a boundary condition or from the prior solution of an adjacent cell. The average flux along a given outgoing side can be computed by integrating the flux along the side and dividing by the length of the side. However, in order to perform the integration, the form of the distribution of the angular flux on the side must be known. This distribution can be determined from the properties of the cell and from the average flux on each of the known incoming sides.

Since the characteristic solution (equation 6) allows calculation of the angular flux at any point s in a single cell given an initial condition, the exact value of the flux can be computed at any point on any outgoing side if the flux along each incoming sides is known. As an initial condition, it is assumed that the angular flux in some characteristic direction is known at some starting point, $s = 0$, i.e., $\psi(0) = \psi_0$, on an incoming side. In order to determine the flux at some point on an outgoing side, one need only know the distance s measured back along a characteristic direction to the appropriate incoming side. This method can then be expanded to determine a functional form of the flux for every point on the outgoing side, which can be integrated to produce the average outgoing flux on the side.

In order to develop a mathematical relationship between two arbitrary sides of a cell, first consider two arbitrary coplanar line segments in space whose endpoints each lie on a pair of parallel lines laid in the direction $\hat{\Omega}_k$, as shown in figure 5. Points B_1 and B_2 can be considered to be the "projections" of A_1 and A_2 respectively relative to $\hat{\Omega}_k$. Since s is the distance between a point on segment A and its projection on segment B, it can be seen that s varies linearly in moving from the the "beginning" to the "end" of the pair of segments.

If α is the distance along segment B measured from endpoint B_1 and B has a total length

L, then the distance s between A and B along direction $\hat{\Omega}_k$ can be written as a linear function in terms of the position α:

$$s(\alpha) = s_1 + \left(\frac{s_2 - s_1}{L}\right)\alpha, \tag{9}$$

where s_1 and s_2 are related to the distances along the characteristic direction between A_1, B_1 and A_2, B_2 respectively. (It is important to note that the length s is the same as the distance between the endpoints only when the characteristic vector lies in the plane of the computational cell. This is not necessarily the case, depending on the choice of quadrature directions. This will be discussed in more detail later in this chapter.)

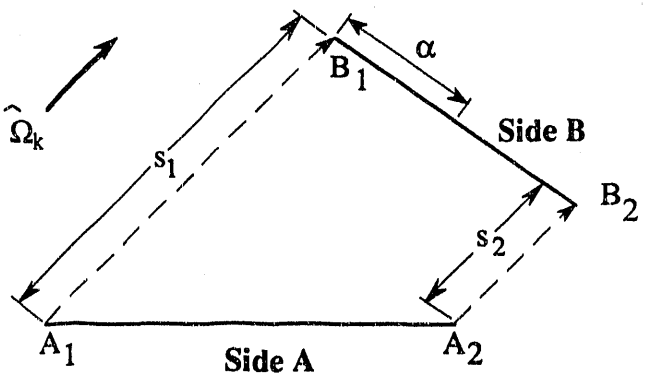


Figure 5. Line Endpoints for Computation of Average Fluxes.

If $\psi(\alpha)$ is the angular flux on B at a distance α from B_1, then $\overline{\psi}_B$, the average value of ψ on B, is given by:

$$\overline{\psi}_B = \frac{\int_{\alpha=0}^{\alpha=L} \psi(s(\alpha))\, d\alpha}{\int_{\alpha=0}^{\alpha=L} d\alpha}. \tag{10}$$

Equation 6, the solution to the characteristic equation in the step approximation, can be

rewritten in terms of the average known angular flux on A as:

$$\psi_B(s) = \left(\overline{\psi}_A - Q/\sigma_t\right)e^{-\sigma_t s} + Q/\sigma_t. \tag{11}$$

Inserting equations 9 and 11 into equation 10 and simplifying yields:

$$\overline{\psi}_B = \frac{1}{L}\int_{\alpha=0}^{\alpha=L}\left[\left(\overline{\psi}_A - Q/\sigma_t\right)\exp\left(-\sigma_t\left(s_1 + \left(\frac{s_2 - s_1}{L}\right)\alpha\right)\right) + Q/\sigma_t\right] d\alpha\,. \tag{12}$$

For the special case where A and B are parallel, $s_1 = s_2$ and the second term in the exponential drops out. Equation 12 can easily be integrated to obtain:

$$\overline{\psi}_B = \left(\overline{\psi}_A - Q/\sigma_t\right)\exp(-\sigma_t s_1) + Q/\sigma_t. \tag{13a}$$

In the more general case, $s_1 \neq s_2$, the result is slightly more complicated:

$$\overline{\psi}_B = \frac{\left(\overline{\psi}_A - Q/\sigma_t\right)}{\sigma_t(s_2 - s_1)}\left[\exp(-\sigma_t s_1) - \exp(-\sigma_t s_2)\right] + Q/\sigma_t. \tag{13b}$$

Equation 13 can also be written in a simplified form:

$$\overline{\psi}_B = \beta_{AB}\overline{\psi}_A + \left(1 - \beta_{AB}\right)\frac{Q}{\sigma_t}, \tag{14}$$

where $$\beta_{AB} = \begin{cases} \dfrac{\left[\exp(-\sigma_t s_1) - \exp(-\sigma_t s_2)\right]}{\sigma_t(s_2 - s_1)} & s_1 \neq s_2 \\ \exp(-\sigma_t s_1) & s_1 = s_2. \end{cases}$$

Thus far, this development has only considered the special case where contributions to side B are only due to the cell internal source and a single incoming side, i. e., side A. For an arbitrarily shaped cell and discrete direction $\hat{\Omega}_k$, it is likely that the outgoing side would receive contributions from two or more incoming sides, as illustrated in Figure 6, for a cell with four incoming sides contributing to the flux on the single outgoing side. In such a situation, the outgoing side can be subdivided into multiple components. Side B of Figure 6 can be represented by four components B_W, B_X, B_Y and B_Z, as illustrated in Figure 7. The

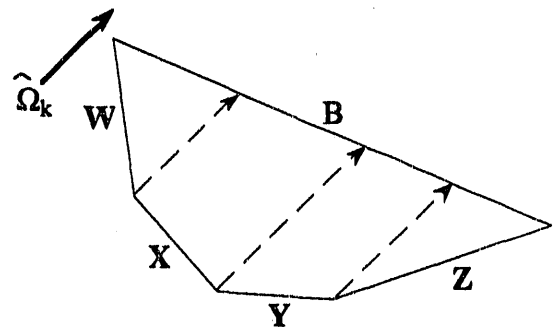


Figure 6. Contributions of Multiple Incoming Sides on an Outgoing Side.

average angular flux $\overline{\Psi}$ can be computed for each component of side B using equation 13; then $\overline{\Psi}_B$, the average flux for the entire length of B, can be calculated by the length-weighted

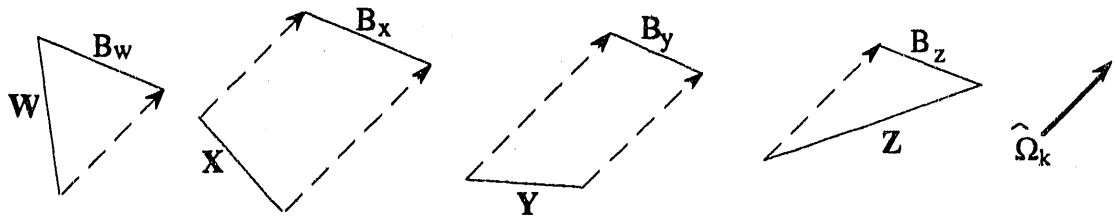


Figure 7. Components of a Side Receiving Contributions from Multiple Sides.

average of each component. In general, for a given side B comprised of n components, the side's average flux is given by:

$$\overline{\Psi}_B = \sum_{i=1}^{n} \frac{\overline{\Psi}_i £_i}{L_B}, \tag{15}$$

where $£_i$ is the length of the projection of the ith side onto B, and $\overline{\Psi}_i$ is the average flux computed for side B due to the flux on side i. Defining the weighting ratio w_{ij},

$$w_{ij} = £_i/L_j \tag{16}$$

where L_j is the total length of the jth side, and using equation 14, this can be written in a more general form as:

$$\bar{\psi}_j = \sum_{i=1}^{n} w_{ij}\left(\beta_{ij}\bar{\psi}_i + (1 - \beta_{ij})\frac{Q}{\sigma_t}\right). \tag{17}$$

Equation 16 can be used to compute the average flux on the jth side of a cell due to contributions from n incoming sides. It will be shown in the following section that in some cases the ith side will not contribute to the flux on the jth side; in this case, w_{ij} will be found to be zero.

The weight term w_{ij} has been kept separate from the geometric attenuation term β_{ij} in order to illustrate the individual features of each term. However, in application the two terms can be combined. Combining terms and noting that the sum of the w_{ij} terms over all i sides is equal to 1.0, Equation 17 can be rewritten as:

$$\bar{\psi}_j = \frac{Q}{\sigma_t} + \sum_{i=1}^{n} \gamma_{ij}\left(\bar{\psi}_i - \frac{Q}{\sigma_t}\right) \tag{18}$$

where $\gamma_{ij} = w_{ij}\beta_{ij}$.

Using equations 13a, 13b and 18, one can compute the average flux on each of the outgoing sides for a given cell, once the angular flux on each incoming side is known. At this point, all variables are known except the distances s_1 and s_2, and the lengths $\pounds_i$ and L. These can be determined from the geometry of the cell and the direction $\hat{\Omega}_k$. The following section will derive general geometric relationships between arbitrary line segments in a two dimensional space, so that the requisite distances can be computed.

Relationships Between Cell Boundaries

In order to represent relationships between any two sides of a computational cell without prior knowledge of the shape of the cell, it is necessary to consider all possible orientations of the two line segments relative to a discrete direction $\hat{\Omega}_k$. Tests can then be performed to categorize the relationship as one of these orientations. Knowing the orientation between two sides, one can determine if the flux at a given side contributes to the flux at the

remaining side. If this is the case, the distances s_1, s_2, ℓ_i and L_j described earlier can be determined. This process can be repeated for each incoming side in order to determine all contributions to a given outgoing side.

Consider two coplanar line segments A and B, where A represents an incoming or "known" side and B is on outgoing or "unknown" side along the boundary of a computational cell. Let segment A represent the ith side of an n-sided cell, while B is the jth side of the same cell. The projection of A on B can take one of six forms, depending on the orientation of $\hat{\Omega}_k$. Figure 8 illustrates the six possible cases. In orientation 1, a fraction of the projection of A lies to the left of B; similarly, in orientation 3 a fraction of the projection of A lies to the right of B. In both cases, segment B will receive only a fraction of its flux contribution from segment A. In orientation 2, segment B lies completely within the projection of A, and the flux at B will depend only on the flux at side A. In orientation 4, the projection of A lies entirely within the length of B. Thus, like orientations 1 and 3, segment B will receive only a fraction of its flux contribution from segment A. In orientation 5, the projection of A lies completely to the left of B, while in 6 the projection lies entirely to the right of B. The net result is the same; the flux on segment A will not contribute to the flux on segment B. The possible case where the projection of A coincides with the endpoints of B is considered to be a special case of orientation 4.

To test the location of the projection of a single endpoint of segment A in direction $\hat{\Omega}_k$ relative to segment B, one must find the intersection of two lines. Given that the endpoints of A and B are defined by (x,y) pairs in a Cartesian system, the first line (the projection of an endpoint of A) is defined by the coordinate of the endpoint and the direction $\hat{\Omega}_k$. The second line, the line upon which segment B lies, is defined by the endpoints of B. (If each line is defined in slope-intercept form, i.e., $y = mx + b$, the intersection is defined by the unique point where $x_1 = x_2$ and $y_1 = y_2$. If $|m_1| = |m_2|$ the two lines are parallel, and A does not contribute to B.) The point of intersection of these two lines relative to the endpoints of

B determine whether the projection of the A endpoint lies to the left, right, or on segment B. Performing this test for each endpoint of A will define which of the six possible orientations represents the pair of lines.

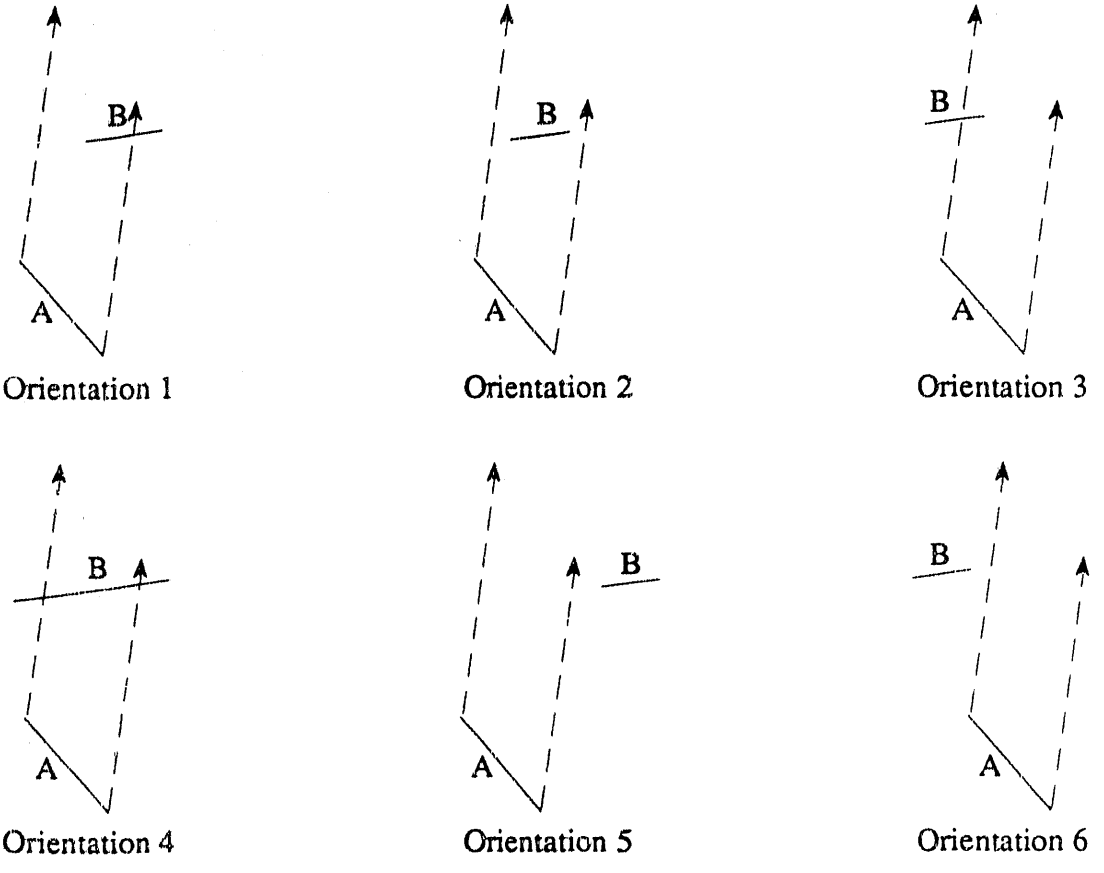


Figure 8. All Possible Orientations of Two Arbitrary Sides and an Arbitrary Direction Vector.

The start and end points for a projection of segment A onto segment B are determined from the orientation of the lines relative to the direction of the angular flux. Obviously, in cases 5 and 6 of Figure 8, there is no projection of side A onto side B. In these cases, the term w_{AB} of equation 16 is zero, and the distances s_1, s_2, $£_A$ and L_B are not needed. In fact, s_1, s_2, and $£_A$ are not even defined for these cases. For the other orientations, however, these values are defined and are necessary to solve equation 13. Figure 9

illustrates the projection of portions (or all) of segment A onto portions (or all) of segment B for the various orientations where A does project onto B. The points A_1, A_2, B_1 and B_2 are not the endpoints of line segments A and B; instead, they represent the endpoints of the regions which are shared by the projection of A onto B. The length $\pounds_A$ is the distance between B_1 and B_2, since it is the length of the projection of A, and the length L_B is the length of the entire line segment B. Given point coordinates in a Cartesian system, these distances are trivial to compute. The distances s_1 and s_2 are proportional to the distance between A_1 and B_1 and A_2 and B_2 respectively. The proportionality is a function of $\hat{\Omega}_k$ and results from the fact that the direction can lie outside of the x-y plane, as shown in Figure 10. The evaluation of the proportionality constant will be discussed later.

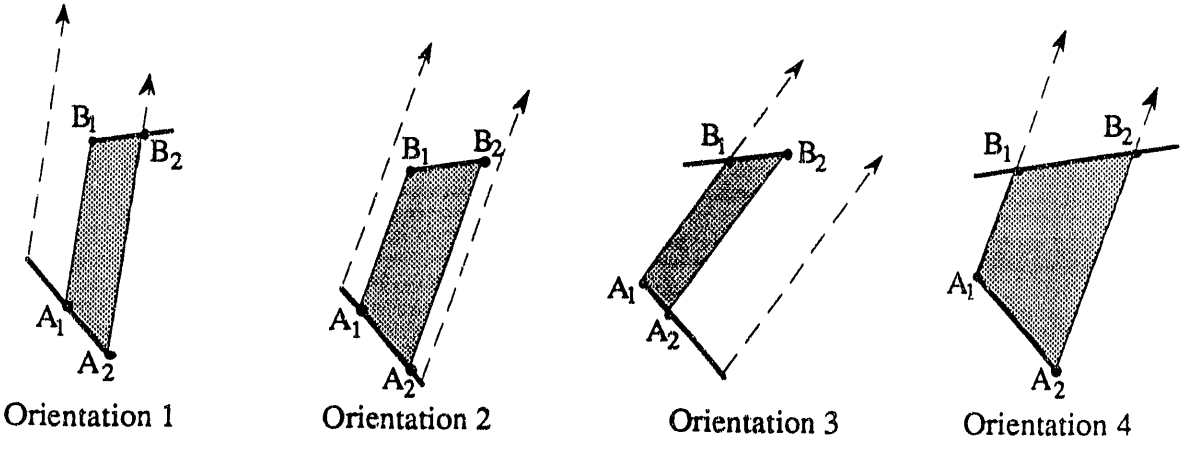


Figure 9. Determination of Projection Start and End Points For Line Segment A Projected onto Line Segment B.

For the case where the projection of A lies entirely within the length of B (orientation 4), the coordinates of B_1 and B_2 are already known from the earlier test of intersection points, and A_1 and A_2 are known as the endpoints of line segment A. The opposite occurs for orientation 2, where B_1 and B_2 are known from the endpoints of segment B. However, in this case, it is necessary to project back in the direction $-\hat{\Omega}_k$ in order to determine points A_1 and A_2. Orientations 1 and 3 are cases where both direct projection and "back-projection"

are necessary to determine two points, while the other two points are known as endpoints, one from each line segment. In orientation 1, A_2 and B_1 are known and A_1 and B_2 are determined from back-projection from B_1 and projection from A_2 , respectively. The opposite approach is used for orientation 3.

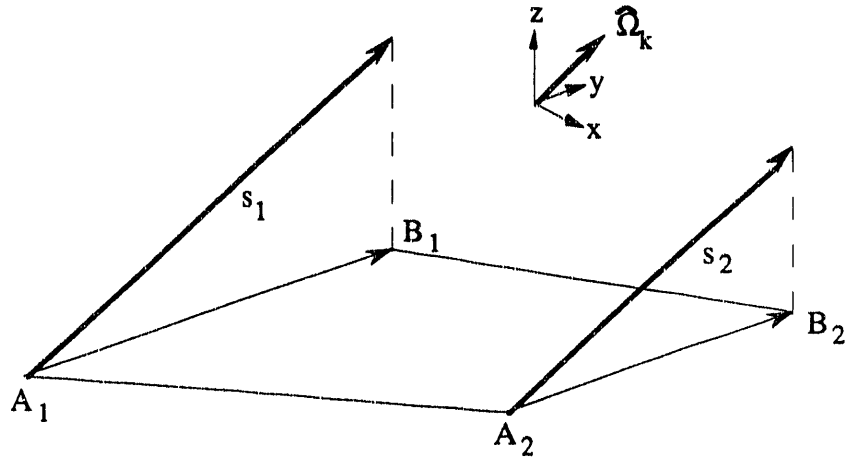


Figure 10. Relationship Between s_1 and s_2 and their Projections in the X-Y Plane.

Note that in orientation 2, the weight term w_{AB} would be equal to 1.0, since the length of the projection of A on B is equal to the length of B. Thus, the angular flux on B would depend only on the flux on A (plus cell internal sources). In the other orientations, the length of the projection of A on B is less than the length of segment B, so that w_{AB} is less than 1.0 and the average flux on B would have components from other cell sides (not shown in Figure 8). These confirm what would be intuitively expected for each configuration.

Mapping a Characteristic Vector into the Two-Dimensional Problem Domain

Even in a two-dimensional x-y system, although the scalar flux is constant with respect to the z axis, the angular flux has components in the z direction. Thus, in order to obtain the scalar flux at a point on the x-y plane, it is necessary to integrate over the unit sphere in all

4π directions of $\hat{\Omega}$.

Recall that the choice of characteristic directions selected for this model were selected to be the same as the set of directions comprising a conventional quadrature set. Quadrature sets specified in the literature[36-38] and used in other discrete ordinates codes[40] are based on a unit sphere, and are usually specified in terms of μ_k and η_k, the respective x and y components of $\hat{\Omega}_k$, where $\hat{\Omega}_k$ is one of a set of discrete directions comprising the quadrature set. Because $\hat{\Omega}_k$ is a unit vector, ξ_k, the z component of the direction, is implicit: $\xi_k = \sqrt{1 - \mu_k^2 - \eta_k^2}$. However, due to the two-dimensional nature of problem, the z component is never explicity used. It is therefore sufficient to evaluate the angular flux at a finite number of points in 4π $\hat{\Omega}$-space in terms of just the μ_k and η_k components of the discrete directions $\hat{\Omega}_k$.

One must recognize, however, that the length of the path travelled by particles moving in a direction out of the x-y plane is always longer than the x-y projection of the path, by a factor of $(\mu^2 + \eta^2)^{-1/2}$. Thus, for any path length s measured in the x-y plane for a given direction $\hat{\Omega}_k$, the true path length travelled is s', where

$$s' = s/\sqrt{\mu^2 + \eta^2}. \tag{19}$$

The distances s_1 and s_2 were stated to be proportional to the distance between projection endpoints. The above factor is the proportionality constant for a particular direction $\hat{\Omega}_k$.

Neutron Balance Within a Computational Cell

Once angular fluxes have been computed for all sides of a cell, it is necessary to compute the cell-averaged angular flux. In order to enforce conservation, a balance condition is applied to the cell. This provides the equation necessary to determine the average flux in the cell. The neutron balance for an arbitrary cell in steady state may be expressed as:

$$\left[\begin{array}{c}\text{net number of}\\ \text{neutrons moving in}\\ \text{direction } \hat{\Omega} \text{ escaping}\\ \text{from the cell}\end{array}\right] + \left[\begin{array}{c}\text{number of neutrons}\\ \text{removed from the cell}\\ \text{or from direction } \hat{\Omega}\\ \text{by interactions}\end{array}\right] = \left[\begin{array}{c}\text{number of}\\ \text{neutrons produced}\\ \text{in the cell moving}\\ \text{in direction } \hat{\Omega}\end{array}\right] \quad (20)$$

or, expressed mathematically,

$$\oint_S \vec{n} \cdot \hat{\Omega}_k \psi dS + \sigma_t \overline{\psi} V = QV \quad (21)$$

where $\vec{n}$ is the outward normal direction at each side of the cell, and V is the two-dimensional volume of the cell. Note that in this context S represents the surface area or perimeter of the cell. Hence, for a cell with n sides, each of the sides having a constant angular flux $\overline{\psi}_i$ and an outward normal direction $\vec{n}_i$,

$$\overline{\psi}_{cell} = \frac{Q}{\sigma_t} - \frac{1}{\sigma_t V} \sum_{i=1}^{n} \overline{\psi}_i \int_{S_i} \vec{n}_i \cdot \hat{\Omega}_k dS_i \quad (22)$$

Since each cell is restricted to be a polygon, each side in the cell will be a straight line, and $\vec{n}_i \cdot \hat{\Omega}_k$ will be constant along the length of the side. Equation 22 can then be simplified to obtain:

$$\overline{\psi}_{cell} = \frac{Q}{\sigma_t} - \frac{1}{\sigma_t V} \sum_{i=1}^{n} \overline{\psi}_i \left(\vec{n}_i \cdot \hat{\Omega}_k L_i\right) \quad (23)$$

where L_i is the length of the ith side, and the term in parentheses represents a leakage coefficient for the side.

Determination of the Source Term

Equation 3 provides an integral relationship between angular fluxes, external sources, and the cell source term, $Q(\mathbf{r},\hat{\Omega},E)$. For the purposes of this research, it will be assumed

that both fission and scattering are isotropic, and therefore have no dependence on the direction $\hat{\Omega}$. Equation 3 is then reduced to the form:

$$Q(\vec{r},\hat{\Omega},E) = \frac{Q(\vec{r},E)}{4\pi} = \int_{E'=0}^{\infty} \frac{\sigma(\vec{r},E'\rightarrow E)}{4\pi} \int_{4\pi} \psi(\vec{r},\hat{\Omega}',E')d\hat{\Omega}'dE' + \frac{S(\vec{r},E)}{4\pi}. \qquad (24)$$

The term σ contains both scattering and fission contributions, and can be written in terms of the two components. Additionally, in order to perform the integration, two approximations are made. First, the continuous energy distribution is represented by a finite set of G constant property energy bands, using the multigroup approach.[3,4,26] Second, it is assumed that the integration of the angular flux can be adequately represented using an N-point three-dimensional quadrature scheme. Additionally, provisions are made for determination of the multiplication eigenvalue, k, for the system. In application, Equation 24 is written in two forms. The first represents the source term for a subcritical system, where no eigenvalue is sought. In this case, an external source is required in order to obtain a non-trivial solution, and Equation 24 is written in the form of Equation 25a. A second form is used when an eigenvalue iteration is to be performed, and is given in Equation 25b. In this form, no external source is used, and k is included in the fission term.

$$Q_g(\vec{r}) = \sum_{g'=1}^{G} \left[\left(\sigma_{s,g'\rightarrow g}(\vec{r}) + \chi_g \nu_{g'} \sigma_{f,g'}(\vec{r})\right) \left(\sum_{n=1}^{N} w_n \psi_g(\vec{r},\hat{\Omega}_n) \right) \right] + S_g(\vec{r}). \qquad (25a)$$

$$Q_g(\vec{r}) = \sum_{g'=1}^{G} \left[\left(\sigma_{s,g'\rightarrow g}(\vec{r}) + \frac{\chi_g \nu_{g'}}{k} \sigma_{f,g'}(\vec{r})\right) \left(\sum_{n=1}^{N} w_n \psi_g(\vec{r},\hat{\Omega}_n) \right) \right]. \qquad (25b)$$

In an eigenvalue search, the multiplication factor k for each step in the outer iteration process is calculated based on the value from the previous outer iteration, based on an initial guess of k = 1.0 for the first iteration. The power iteration method typically applied in multiplication eigenvalue problems[4] is used to determine the value of k for each new

iteration.

This chapter has introduced the basic concepts of the extended step characteristic method, and the necessary relationships between angular fluxes and computational cells have been derived, along with relationships between cells. The following chapter will describe the steps necessary to combine these theoretical relationships into a functional numerical model.

CHAPTER IV

APPLICATION

In the previous chapter, general theoretical concepts and formulations were developed for describing relationships between the cells comprising an arbitrary problem domain, as well as the interrelationships between the sides of a given cell. In order to apply these tools, a numerical model has been developed, and written in terms of a computer program. This computer code, CENTAUR (Characteristic Estimate of Neutron Transport in Arbitrary Uniform Regions), written in FORTRAN-77, has been used to obtain fluxes and reaction rates in multiple energy groups for various problem geometries; the results of some of these calculations are presented in Chapter V. This chapter will present the techniques used in applying the somewhat abstract relationships of the previous chapter within the CENTAUR numerical model.

Rather than delineating the program on a subroutine by subroutine basis, the following sections will provide a more natural development of the numerical tools, starting with the basic problem geometry, moving through intracell relationships, intercell relationships, and then the relevant aspects of the spatial (inner) and energy (outer) iterative schemes for solution of the problem. Finally, memory management techniques included in the code to expand its capabilities, decrease its memory requirements, and increase its efficiency will be discussed.

Geometric Descriptions

Since the ESC method is based on polygonal cells, it is best suited for a Cartesian coordinate system. This does not mean that the use of orthogonal cells is required; it is simply the most straightforward means of describing the spatial location and orientation of the set of line segments of which a problem domain is composed. Any polygon (cell) can then be simply described by the x-y coordinates of its vertices. Using a Cartesian reference,

the following paragraphs describe the method by which a problem geometry is specified as input to CENTAUR, the processing of this data, and the geometric properties of lines and cells computed internally by CENTAUR.

Geometric Input

The concept of a cell is simply a set of lines connected end-to-end to enclose a two-dimensional volume (although a cell is also a sub-domain in which a balance condition is enforced); thus, a cell is defined by its boundaries. Because cells share sides, each line segment is associated with two different cells, or a cell and a boundary condition (BC). Thus, if one were to try to provide a cell-by-cell description of an input geometry, one would have to specify most lines twice, once for each cell. Hence, in order to describe a given geometry, it is most direct to describe the geometry in terms of its constituent line segments, rather than attempting to define each cell based on vertices. This makes the job of preparing input for CENTAUR somewhat simpler, and places the burden of determining the proper cell structure on the programming logic internal to CENTAUR.

In order to describe the method for specifying an input geometry for CENTAUR, it is convenient to use an example. Figure 11 illustrates a cylinder approximated by eight four-sided cells. The cells have been numbered 1-8, and the sides have been numbered 1-24. If one specifies the x-y endpoints of each of the 24 line segments and the two regions associated with each line segment (either two adjacent cells sharing the single line or one cell and its boundary condition), then the problem has been completely defined. For consistency, a numbering scheme convention has been been adopted: each line is specified by the x-y coordinate at a start point and the x-y coordinate at the end point, as well as the cell/BC on the left and the cell/BC on the right, where left and right are relative to the direction moving from start to end. The choice of start or end point is completely arbitrary; however, this choice defines the left and right directions. A CENTAUR description of a

line can then be written in the form:

Line Number	Left Cell Number	Right Cell Number	Start Point x-coordinate	Start Point y-coordinate	End Point x-coordinate	End Point y-coordinate.

Thus, for the above geometry, if the circle radius is unity, CENTAUR input might define line 8 as:

8 8 1 0.000 0.000 0.000 1.000 or 8 1 8 0.000 1.000 0.000 0.000.

For boundary conditions, a boundary condition identifier is used in place of a cell number to indicate the boundary condition type (e.g., vacuum, reflective, white, periodic, etc.). So that the code can distinguish between a cell number and a BC, BC number flags begin at 10000, or are less than 0 for periodic boundary conditions. Thus, side 9 of Figure 11 could be described as:

9 10000 1 0.000 1.000 0.3827 0.9239,

indicating that there is a boundary condition applied to the left of the line and that cell 1 lies to the right of the line. The selection of the origin of the coordinate system is completely arbitrary, since only the relative position of line segments is of consequence. In the above examples, the choice of the center as an origin was merely a convenience.

The difference between an adjacent cell and a boundary condition is transparent within the scope of a cell. Within a cell, all sides with a known flux are boundary conditions for the cell's solution, whether the flux comes from a previously computed adjacent cell or from an externally specified boundary condition.

For the geometry of Figure 11, a total of 24 line descriptions would be necessary. Using this information, algorithms in CENTAUR are used to determine line and cell-based properties. The method used to do this is described below.

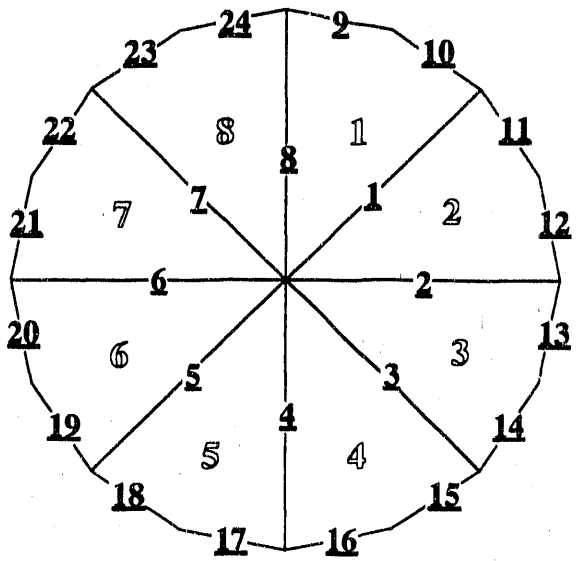


Figure 11. A Numbering Scheme for an Eight Cell, Twenty-Four Side Geometry.

Fundamental Geometric Properties

Once all line segments are specified, CENTAUR possesses all information needed to compute required basic geometric parameters. These properties are saved in memory and are available for later calculations. Computation of additional geometric attributes are performed later, when relationships between sides are used to determine the values s_1, s_2, $\pounds_i$ and L_j, using these fundamental properties.

The basic properties are calculated in eight stages:

1) First, the line data is reduced to a cell based form, where for each cell in the problem, all lines associated with that cell are determined. The number of lines in each cell is saved, and each line is assigned a cell reference number, ranging from 1 to the total number of lines in the cell.

2) The start and end point definitions are rearranged such that the current cell is to the left of each line segment. Thus all line segments associated with a cell will have a direction that is counter-clockwise with respect to the cell. The purpose of this is twofold. First, it simplifies the determination of the "outward" direction, and second, it creates a consistent

convention which simplifies the calculation of the s_1 and s_2 terms.

3) Using the redefined coordinates of the lines within each cell, the directional heading of each line in the cell is determined. All directions and angles are specified in radians, measured counter-clockwise from the x-axis.

4) By subtracting $\pi/2$ from the direction of each line, the outward normal for the line is calculated.

5) Using the input information, the cell/external boundary condition which shares each line as a boundary is saved in the cell data structure.

6) For each line which has another cell as a boundary, the number of the corresponding line within the adjacent cell's line numbering system is determined. This is to be used when information is passed between cells as boundary conditions for a new cell calculation.

7) For each angle in the quadrature set, the outward normal of each line in each cell is compared to the direction of the angle by taking a dot product. If the dot product is less than or equal to zero, the side is marked as as "incoming" side with respect to the angle, as per equation 8a. This is used to determine which sides will become boundary conditions for the cell for each angular direction. Classification of "outgoing" sides is not necessary, since by default any side that is not an incoming side must be an outgoing side

8) For each cell, the cell's area is computed. To compute the area, a point within the cell is determined. For each side in the polygon, the endpoints of the line and a point interior to the cell define a triangle; for an n-sided polygon, this will result in n non-overlapping triangles. The area of the cell is then the sum of the area of the n triangles; the area of a triangle is simple to compute. The coordinates of an interior point can be found by averaging the respective x and y coordinates of each corner of the polygon.

Intracell Relationships

Intracell relationships, or the relationship between the fluxes on the sides of each cell,

are represented by the term γ_{ij} of equation 18. In order to calculate this term, one must first determine the lengths s_1, s_2, $\pounds_i$ and L_j. All but the latter must be computed for each side relative to each of the other sides in the cell, for each direction in the quadrature set. Recall that L is simply the length of a given line segment, and is easily calculated knowing the endpoints of each line segment.

For each cell and each discrete direction $\hat{\Omega}_k$, the contribution of ith side to jth side, γ_{ij}, must be determined for all i and j. Letting A represent a specific side i, and B a particular side j, then the method can be expressed in terms of finding the value γ_{AB}. The four steps necessary to determine γ_{AB} are summarized below, and are described in detail in the following paragraphs.

1) Perform simple tests to determine if side A can contribute to side B. If not, $\gamma_{AB} = 0$, and one can proceed to test A against the next side, if any remain.

2) If the two sides pass the first simple tests, then they must have one of the relative orientations shown in Figure 8, and determination of the orientation of side B relative to side A is necessary. Note that it is still possible that side A does not contribute to side B; if the lines are oriented such that side A does not project onto side B (Orientation 5 or 6), then $\gamma_{AB} = 0$.

3) If A does project onto B, then use the specific orientation type to determine the distances s_1, s_2, $\pounds_A$ and L_B.

4) Calculate the weighted contribution of A onto B, γ_{AB}.

Four elementary tests are possible to determine if side A has the potential to contribute to any other side B for a given direction $\hat{\Omega}_k$ in a given cell. These tests do not require extra calculations relative to the geometry of the two sides, and are therefore performed as a preliminary screening prior to determination of relative orientations. For the first test, if i = j, then side B is the same as side A, and the contribution γ_{AA} is considered to be zero. Second, if a specific side A is not an "incoming" side for neutrons travelling in direction $\hat{\Omega}_k$,

then it cannot contribute to other sides, and $\gamma_{AB} = 0$ for all other sides in the cell. In the previous section of this chapter, the classification of all sides as either "incoming" or "outgoing" was discussed. Since all "incoming" sides in a cell are known prior to this test, the determination is trivial. Third, if sides A and B are both "incoming" sides, then A cannot contribute to B, and $\gamma_{AB} = 0$ for the particular side B. If side A passed the second test then it is already known to be "incoming", and it is only necessary to check side B to test this condition. Finally, side A is checked to see if it is parallel to direction $\hat{\Omega}_k$. If so, then A cannot contribute any other side, and $\gamma_{AB} = 0$ for all other sides. This is the special case of Equation 8a discussed earlier. Directions for all line segments have already been computed and stored, so only a simple subtraction is necessary to establish whether the two directions are parallel.

If sides A and B pass all of the above tests, then the two sides must possess one of the relative orientations of Figure 8. In order to determine the exact type of orientation, it is necessary to project the endpoints of side A along the direction $\hat{\Omega}_k$ onto the line containing side B. The projection of each endpoint of A is given by the intersection of a line passing through the endpoint in direction $\hat{\Omega}_k$ and the line defined by side B. For the first line, the slope of the line is the arctangent of $\hat{\Omega}_k$; knowing the slope m_1 and a point (x_1,y_1) on the line, the y-intercept of the first line, y^0_1, can be computed:

$$y^0_1 = y_1 - m_1 x_1. \tag{26}$$

For the line containing side B, the slope m_2 is defined by the two endpoints $\left(x_2^{start},y_2^{start}\right)$ and $\left(x_2^{end},y_2^{end}\right)$:

$$m_2 = \frac{y_2^{start} - y_2^{end}}{x_2^{start} - x_2^{end}}. \tag{27}$$

Using either endpoint, equation 26 can be used to obtain y^0_2. The point of intersection (x_{int},y_{int}) is then given by:

$$x_{int} = \frac{y^0_2 - y^0_1}{m_1 - m_2}, \tag{28}$$

and

$$y_{int} = m_2 x_I + y^0_2. \tag{29}$$

Equations 26-29 are applied to each endpoint of side A to find their projections in $\hat{\Omega}_k$. Projections of the endpoints of side B onto the line containing side A are also required. The same procedure is used to find these projections.

Letting B_s and B_e represent the start and end points of side B, and $\hat{A}_s$ and $\hat{A}_e$ represent the projection of the start and end points of side A along $\hat{\Omega}_k$ onto the line containing side B, as illustrated in Figure 12. Tests can then be performed to determine which orientation is

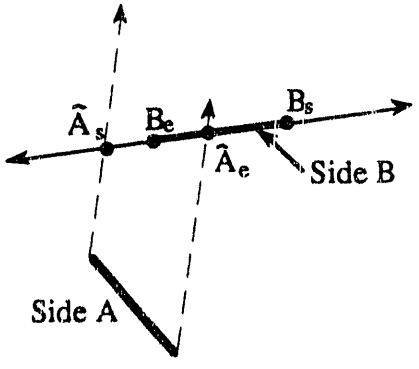


Figure 12. Projection of Side A onto the Line Containing Side B.

represented by the two sides. The flowchart shown in Figure 13 illustrates the logic required to determine the relative orientation of two sides based on the positions of points $\hat{A}_s$, $\hat{A}_e$, B_s, and B_e on the line defined by side B. Because all sides have endpoints arranged such that travel from start to end is in a counter-clockwise direction relative to the cell center, several orientations of these points cannot occur, and only four tests are necessary to determine the relative positioning of the two sides.

The test to determine if a given point lies "between" two other points is trivial for three colinear points: it is sufficient to compare the x-coordinate of each point to find the intermediate value, although comparison of the y-coordinates is also necessary when the line is vertical. The term "between" is extended to include points having the same x (or y for vertical lines) coordinate. In this sense, a point b is between points a and c on a non-vertical

line when $x_a \leq x_b \leq x_c$ or $x_a \geq x_b \geq x_c$. Similarly, when the points lie on a vertical line, b is between points a and c when $y_a \leq y_b \leq y_c$ or $y_a \geq y_b \geq y_c$.

If the two lines are classified as being of orientation type 5 or 6, then the flux on side A cannot contribute to that on side B, and $\gamma_{AB} = 0$. Otherwise, the orientation type must be used to determine distances s_1, s_2 and $\pounds_A$, so that γ_{AB} can be computed.

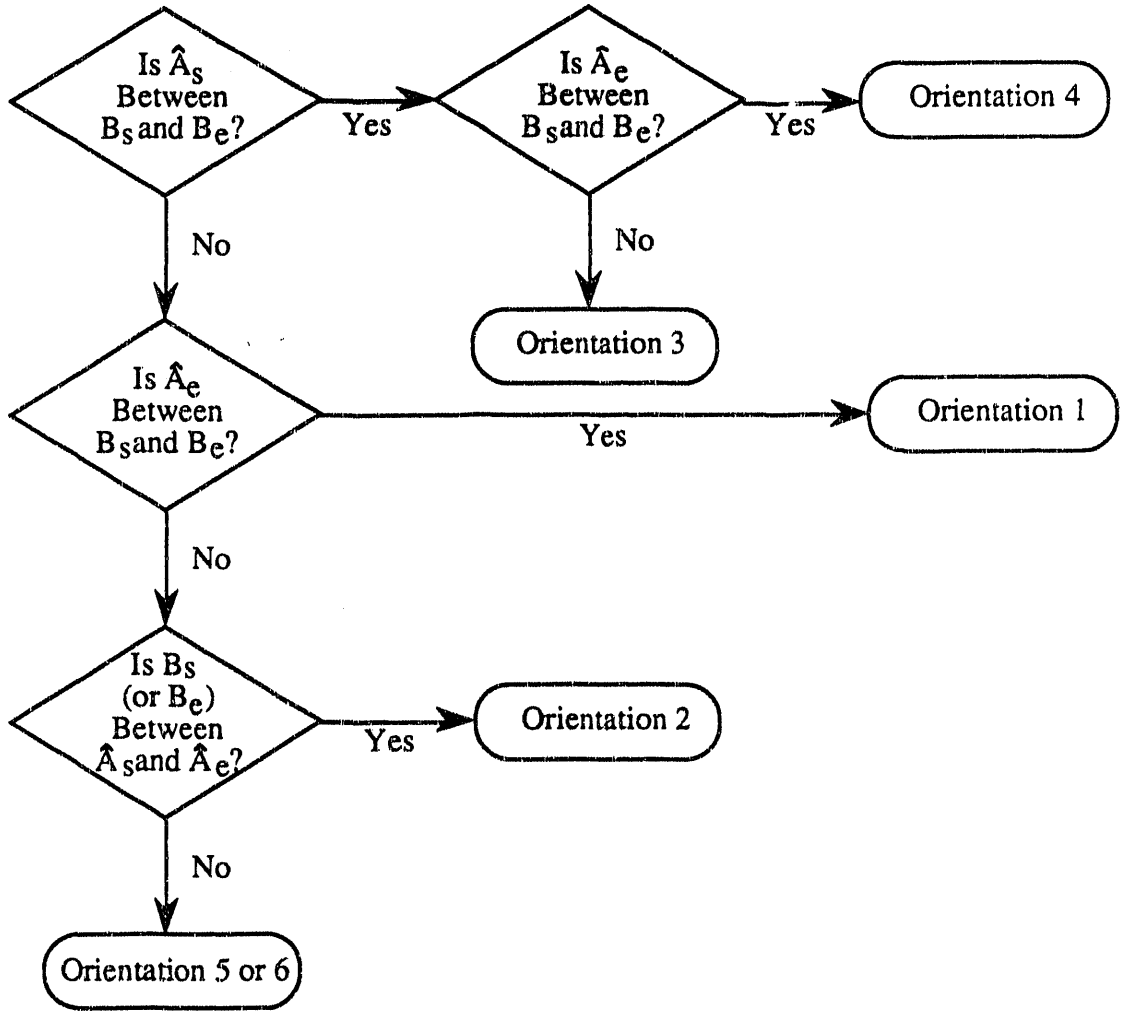


Figure 13. Testing Scheme for Determination of Relative Orientation.

As discussed in the previous chapter, the distances s_1 and s_2 are defined by $\hat{\Omega}_k$ and the distances between A_1 - B_1 and A_2 - B_2, respectively, and $\pounds_A$ is the distance between B_1 and B_2 (see Figure 9). Each of the points A_1, A_2, B_1, and B_2 defined by one of the points A_s,

A_e, $\hat{A}_s$, $\hat{A}_e$, B_s, B_e, $\hat{B}_s$, or $\hat{B}_e$. Further, A_1 can only take the values A_s or $\hat{B}_e$, A_2 the values A_e or $\hat{B}_s$, B_1 the values B_e or $\hat{A}_s$, and B_2 the values B_s or $\hat{A}_e$, depending on the type of orientation. Based on Figure 9, Table 1 gives the definition of s_1, s_2, and $£_A$ in terms of the distances between endpoints and projected endpoints for each of the four contributing orientations, where $f_k = \left(\mu_k^2 + \eta_k^2\right)^{-1/2}$.

Orientation Type	s_1	s_2	$£_A$
1	$f_k \cdot \lvert\hat{B}_e - B_e\rvert$	$f_k \cdot \lvert A_e - \hat{A}_e\rvert$	$\lvert\hat{A}_e - B_e\rvert$
2	$f_k \cdot \lvert\hat{B}_e - B_e\rvert$	$f_k \cdot \lvert\hat{B}_s - B_s\rvert$	$\lvert B_s - B_e\rvert$
3	$f_k \cdot \lvert A_s - \hat{A}_s\rvert$	$f_k \cdot \lvert\hat{B}_s - B_s\rvert$	$\lvert B_s - \hat{A}_s\rvert$
4	$f_k \cdot \lvert A_s - \hat{A}_s\rvert$	$f_k \cdot \lvert A_e - \hat{A}_e\rvert$	$\lvert\hat{A}_e - \hat{A}_s\rvert$

Table 1. Computation of Distances s_1, s_2, and $£_A$.

Knowing s_1, s_2, and $£_A$ from Table 1, γ_{AB} can be calculated using equations 14, 16, 18, and 19. For computational efficiency, these equations are combined into a single equation for each of the two forms of β_{AB} from equation 14. For the general case where lines A and B are skewed,

$$\gamma_{AB} = \frac{£_A}{\lvert B_s - B_e\rvert}\left(\frac{\exp(-\sigma_t s_1) - \exp(-\sigma_t s_2)}{\sigma_t(s_2 - s_1)}\right), \tag{30a}$$

and for the special case where A is parallel to B, i.e., $s_1 = s_2$,

$$\gamma_{AB} = \frac{£_A}{\lvert B_s - B_e\rvert}\exp(-\sigma_t s_1). \tag{30b}$$

This completes the description of the calculation of γ_{AB} for any two sides A and B, for a given $\hat{\Omega}_k$. If a set of N directions are used in a quadrature set, then the coefficient γ must be computed for each direction k, where k ranges from 1 to N. The calculation must also be

performed for each side i in a cell, relative to all other sides j in the cell, for each discrete direction, $\hat{\Omega}_k$. Thus, a total of $n \times n \times N$ coefficients, γ_{ijk}, are required for each cell.

The multigroup treatment used to treat energy dependence has not yet been discussed. However, it is worth noting at this point that the total cross section σ_t will vary with energy; hence each weighted coefficient γ will have an energy dependence, and must be computed for each energy group. For G energy groups, a coefficient γ_{ijkg} is required for each g, where g = 1 to G.

Once the coefficient γ_{ijkg} has been computed for each side in a cell relative to all other sides in the cell, for all directions, for all energy groups, and for all cells in the problem, one can begin the iterative process to calculate the fluxes in all cells using equations 18 and 22. In CENTAUR, all γ's are precomputed and saved prior to beginning the iterative solution, so that a rapid sweep of all cells may be performed. However, in order to begin the calculation for all cells, the sweep direction required for each $\hat{\Omega}_k$ must be determined. The following section will address the manner in which this is accomplished.

Order of Cell Solution

The proper cell ordering, or sweep, must be predetermined for each $\hat{\Omega}_k$ such that upon entering a cell calculation, the flux on each incoming side is known from either an external boundary condition or the earlier calculation of an adjacent cell. In order for such a sweep to work, at least one cell must have all incoming sides specified from external boundary conditions; this will be the first cell in the sweep ordering. This concept is no different than the sweep ordering schemes used in traditional finite difference based discrete ordinates methods in x-y geometries; however, for an arbitrary mesh the sweep pattern is not as intuitively evident as it is for regularly spaced rectangular cells. Hence, the determination of the appropriate sweep order for each characteristic direction is performed by programming internal to CENTAUR based on the input geometry.

Because the identification of a specific side of a cell as either an incoming or outgoing side is dependent on the direction $\hat{\Omega}_k$ (equation 8), the sweep ordering is likely to be different for each discrete direction. Thus, the sweep order must be independently determined for each direction. The process by which the sweep order is determined for a given direction is outlined below.

1. Create a list of "remaining" cells, containing all cell numbers.
2. Determine the sides in all cells which have external boundary conditions, and label these sides as "known" within the cell
3. Step through the list of "remaining" cells until a cell is found in which all incoming sides are "known."
4. Add the cell number to the end of the sweep order list. (This list is initially empty.)
5. Delete the cell number from the "remaining" cell list
6. For each outgoing side in the cell, label as "known" the equivalent incoming side on the adjacent cell sharing the side.
7. Go to step 3 and repeat, until all cells are removed from the "remaining" cell list.

This process is then repeated for all remaining directions in the angular set. This will complete all information required in order to perform calculations for all edge fluxes in all cells for all directions.

Leakage Coefficients

During the iteration process, it is necessary to use equation 23 to determine cell center fluxes, so that cell averaged scalar fluxes may be determined. Although this equation will be applied within an iterative loop, it is possible to precompute the leakage coefficients used in the equation and save them for use during the iteration process. From equation 23, the leakage coefficient for the ith side relative to the kth direction, λ_{ik}, is given by:

$$\lambda_{ik} = \vec{n}_i \cdot \hat{\Omega}_k L_i \,. \tag{31}$$

Since the direction components of $\hat{\Omega}_k$ are known and the direction $\vec{n}_i$ was computed earlier, the dot product is easily computed. As before, L_i is the length of the ith side, and is the distance between the endpoints of the side. λ_{ik} is then saved for each side of each cell for use during the iteration process. Equation 23 is then simplified and written in terms of the kth direction and the gth energy group as:

$$\bar{\psi}_{cell,kg} = \frac{Q_g}{\sigma_{t,g}} - \frac{1}{V}\sum_{i=1}^{n} \bar{\psi}_{ikg}\lambda_{ik}. \tag{32}$$

Iterative Solution

As mentioned previously, the iterative scheme utilized within CENTAUR strongly resembles that described in the literature[3,4,26] and used in other production codes.[39] Therefore, this chapter will not detail the derivation and assumptions of the multigroup formalism, nor will it expound on the details of the iterative scheme applied in CENTAUR. Both are based on traditional approaches, and are not intended to be the emphasis of this work. Instead, the remainder of this section will provide only an overview of these approaches, and will focus on the relevant portions of the CENTAUR implementation of such schemes.

The multigroup approximation can be applied to equation 4, where ψ and σ_t are appropriately averaged over discrete energy bands, and Q_g is an energy dependent term which can receive contributions from all energy groups in a given discretization scheme. For an assumed spatial (cell by cell) distribution for Q_g for a each of a complete set of G energy groups, equations 18 and 32 together with the methods described in the previous sections of this chapter can be used to determine the spatial distribution of the cell average angular fluxes for the energy group. Using equation 25, the source term Q_g can be reevaluated based on the computed fluxes for each of the G groups. Iteration can be performed on the spatial solutions until Q_g and ψ_g are consistent for all energy groups. The

energy iteration is termed an "outer" iteration, because each single energy iteration requires a complete spatial solution for all energy groups. As indicated earlier in this chapter, iteration is required within each energy group in order to obtain a converged spatial distribution. This spatial iteration is referred to as the "inner" iteration, since a fully converged set of inner spatial iterations is performed for all energy groups for each step of the outer energy iteration scheme. Outer iterations are only required when higher energy groups receive contributions from lower energy groups, i.e., for problems with fission or upscattering.

The following section describes the steps involved in each outer iteration, followed in the succeeding section by a description of how inner spatial iterations interact with the outer iteration loop.

Outer Iterations

In order to begin a calculation, a guess is made for the average angular fluxes within each cell. These are used to compute a new scalar flux for each cell for each energy group. This results in a non-zero source distribution, $Q_g(\mathbf{r})$, which in turn is used as a first guess to generate a new spatial angular flux distribution for each energy group. These results are used to update the initial scalar flux guess, and iteration is performed until all cell averaged angular fluxes have converged for all energy groups.

Given a complete set of angular fluxes for all cells within an energy group, whether from an initial guess or a previous iteration, equation 25 provides the necessary relationship to determine the cell averaged source. The outer iteration is not concerned with boundary conditions or the values of the angular fluxes on the sides of the cell; all that is required is the cell average value of the fluxes in each cell for all energy groups. Equation 25 is applied on a cell by cell basis for each energy group and each cell. Since there is no coupling of cells in this equation, the order in which cell sources are computed is completely arbitrary. In fact, because of the decoupling of space and energy in the outer iteration, the shape of the

cell is irrelevant at this point. Only the solution for the fluxes in the inner iterations provides the proper spatial relationships for each energy group, while the outer iteration is insensitive to the method by which the cell fluxes are computed. For this reason, the outer iteration scheme differs very little from the schemes used in other deterministic transport codes.

Prior to calculating the group source terms Q_g, the cell average angular fluxes are tested against their values from the previous outer iteration. If the change in all fluxes for all energies over the last set of inner iterations is within a specified convergence criterion, then the outer iterations are taken to be converged, and iterations are terminated. If not, a new value of Q_g is determined for each energy group, and a new set of inner iterations is begun.

Inner Iterations

The fundamental equation that is used to calculate the spatial distribution of the angular flux on each face of each cell is given by equation 18. Rewritten in terms of a specific direction k and an energy group g, this equation becomes:

$$\bar{\psi}_{jkg} = \frac{Q_g}{\sigma_{t,g}} + \sum_{i=1}^{n} \gamma_{ijkg} \left(\bar{\psi}_{ikg} - \frac{Q_g}{\sigma_{t,g}} \right). \tag{33}$$

Given the total macroscopic cross section $\sigma_{t,g}$ and source term Q_g for a given energy group, along with boundary conditions (incoming fluxes) for each cell, the flux can be determined for each jth unknown or outgoing side. All terms needed for the computation of the outgoing flux on each side are available upon entering a cell: the relational coefficient γ_{ijkg} is computed before iterations begin; all incoming sides are guaranteed to be known based on the predetermined sweeping order; and Q_g is supplied from the calculation in the beginning of the current outer iteration. (Q_g remains constant for all inner iterations within a single outer iteration, and as with $\sigma_{t,g}$ is independent of the direction $\hat{\Omega}_k$.) The cell face fluxes are only intermediate values used to determine cell average angular fluxes; once $\bar{\psi}_{jkg}$ has been determined for each of the n sides in the cell, equation 32 can be used to determine the cell

averaged angular flux for the kth direction, $\overline{\psi}_{cell,kg}$. One can then proceed to the next cell.

As was discussed earlier, a specific sweep order is required for each direction $\hat{\Omega}_k$ in the quadrature set. For each consecutive cell in the sweep ordering, equation 33 is applied for each side in the cell, to determine the contribution to the side from all other sides. Rather than trying to keep track of only the sides that contribute to the jth side, the summation is applied for all n sides in the cell. If γ_{ijkg} was computed correctly, it will have a value of zero when i = j or when the ith side does not contribute to the jth side. Note that in equation 33 there are no angular variables. Only γ_{ijk} has an angular dependence, and it has already been computed and stored. Thus, although cells must be solved in a predetermined sequence for a given direction, the choice of the direction is arbitrary, and the sweep can be applied for any $\hat{\Omega}_k$ in any order. In fact, because of this independence, potential exists for parallel calculations on multiple processors for each direction in the direction set for each sweep across the problem domain; however, this capability has not been studied in this work.

Although each sweep across the problem domain is independent of angle, angular interdependencies can exist at external boundaries for certain types of boundary conditions. CENTAUR supports reflective, white, periodic, and vacuum boundary conditions. All but the latter provide a coupling between angular fluxes at the boundary, thus iteration is required after completing calculations for all directions. When all external boundary conditions are vacuum, spatial iteration is not required.

Memory Management

The previous sections of this chapter have described the implementation of the extended step characteristic method. This information is sufficient to develop a rudimentary computational tool. However, additional memory management techniques are desirable in order to advance the capabilities of the CENTAUR program; storage of all precomputed

terms for all cells and energy groups can require extensive amounts of memory. For example, a relatively small problem, using 500 five-sided cells and a relatively low S_8 quadrature (40 directions), with only five energy groups, will require 20 megabytes of memory just to save the γ coefficients in double precision. A reasonably large problem using higher order S_{16} quadrature and 40 energy groups could require more that 2 gigabytes of core memory. (However, it is worth noting that all of the coefficients need not be in core memory at the same time.) Clearly, given current hardware technology, this amount of memory is unacceptably large, and some method must be used in order to reduce this excess. This section will describe two features of CENTAUR which alleviate much of the memory overhead of the ESC method.

Two options were initially considered for reducing CENTAUR's memory requirement. The first was to optimize the coefficient storage. The second was to forego storage of relational coefficients and compute side to side relationships as they are needed, as is done for traditional finite difference methods. Although the latter method reduces the storage required, it greatly increases the computational work, requiring numerous and relatively elaborate mathematical operations for each side of each cell to be repeated during each inner iteration. Thus, it was decided to attempt to minimize the storage of coefficients based on the knowledge of the system rather than to recompute them as they are needed.

It was recognized that for any given system, at least half of the γ coefficients would be zero, since for the directions in any symmetric quadrature set a given side would be an outgoing side relative to half of the quadrature directions. In addition, it can be heuristically argued that as the number of sides in a cell is increased, the number of relational coefficients for each side with a value of zero relative to all other sides will increase. Hence, there will be a net savings in storage space if storage of coefficients with a value of zero (or less than some very small number) were to be eliminated. Replacing a double precision array of at least 50% zeros with an array of non-zero terms and an integer array of pointers to the non-

zero terms will result in a net savings of at least 25%, since on most computer systems the size of an integer is at most half that of a double precision floating point number. For example, if a set of ten coefficients were stored in a linear array, and 50% of the terms were zero, the block of memory could be represented as:

0.051	0.025	0.000	0.000	0.000	0.104	0.084	0.025	0.000	0.000

If the zeroes were to be eliminated, and an array of pointers used to indicate the original position of each non-zero term, the two arrays would appear as follows:

0.051	0.025	0.104	0.084	0.025	1	2	6	7	8

Note that the storage requirement of each integer pointer is half that of each double precision coefficient. Also, as the number of zeros increases, the space savings will also increase.

Even with elimination of all of the zeros, a storage reduction to less than 10% of the unreduced space is unlikely, and substantially more storage reduction would still be necessary. It was recognized that in order to prevent recalculation of all relational coefficients, it would be necessary to write the coefficients to one or more external files. In order to avoid constantly locating and reading data as it is needed, it was decided to write data files by energy group, so that during each set of inner iterations for a given energy group all coefficients needed would be readily available. This method appears to be the most convenient way to save data and minimize I/O overhead. In addition, memory requirement for the γ coefficients would not change as the number of energy groups is increased.

To implement this approach, CENTAUR writes the group-dependent γ coefficients as they are computed to a scratch file. At the beginning of a given outer iteration, the data for the first energy group is read and used in a set of inner iterations until the spatial solution for the first group has converged. The old coefficients are overwritten when the coefficients for the next energy group are read in, and the inner iterations for the second group is initiated.

This continues for all energy groups, and is repeated for each outer iteration.

Like any other computer code, the model in CENTAUR is of little value until it has been verified against analytical results or the results of other computer codes. The following chapter describes some of the results obtained during comparison of CENTAUR to both analytical solutions and independent discrete ordinates numerical solutions.

CHAPTER V

RESULTS

In order to establish the feasibility and validity of the Extended Step Characteristic approach, it is appropriate to compare results of CENTAUR calculations to those obtained by other methods. Specifically, comparisons will be made with other traditional finite-difference based discrete ordinates methods, and with analytical solutions of the transport equation for simple geometries. However, comparisons will emphasize the performance of CENTAUR relative to existing discrete ordinates codes; analytical solutions are available only for rather simple geometries and are therefore of little value in demonstrating the capabilities of the ESC method. More importantly, the ESC method is intended primarily to provide an extension of the applicability of discrete ordinates approaches, and not necessarily to improve the accuracy of the discrete ordinates approximation to the transport equation.

Two computer codes were used for finite-difference based calculations: ANISN[40] and TWOTRAN.[39] ANISN is a one-dimensional program for slab, cylindrical, and spherical coordinates, while TWOTRAN is a two-dimensional program, and can handle Cartesian, cylindrical, and spherical coordinates. For problems which have only one-dimensional properties, both codes are used in comparison to CENTAUR, but in more complicated two-dimensional problems only TWOTRAN is used.

Each of the subsequent sections of this chapter addresses comparisons of results for both analytical and numerical solutions for various geometries. These results are intended to demonstrate not only the accuracy of the ESC method, but also the flexibility and advantages of the method.

Case 1: Two Region, Two Group Slab

Case 1 was created in order to demonstrate that the CENTAUR solution is accurate for a

simple one-dimensional slab case. An infinite series of infinitely tall slabs of fuel separated by moderator is used, to demonstrate the capability to handle multiple materials, with reflective boundary conditions, in a Cartesian coordinate system. Figure 14a illustrates a portion of the geometry modelled. Because of symmetry, it is only necessary to model a half fuel-half moderator cell, using a reflective boundary condition at the center of each, as shown in Figure 14b.

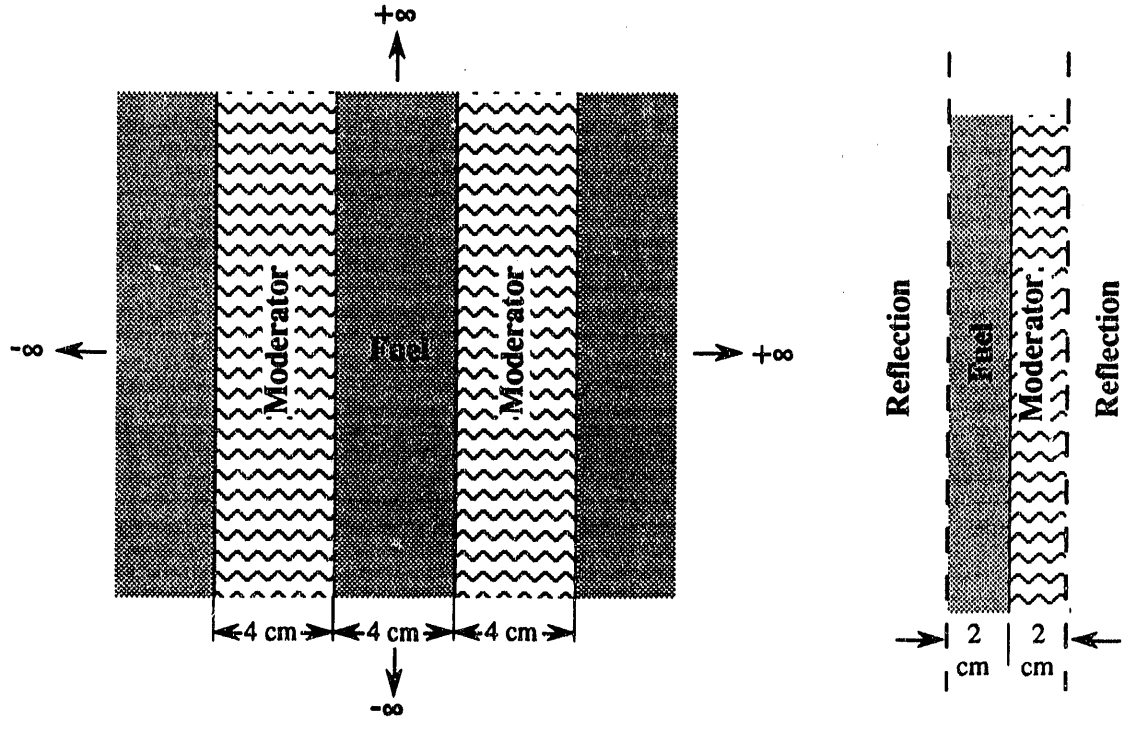


Figure 14. Slab Geometry of Case 1.

Since the ESC method was derived in terms of a Cartesian coordinate system, and will reduce to the SC method discussed in Chapter I when rectangular cells are used, one would expect good agreement with finite difference results. The purpose of this test is to demonstrate that this is the case.

CENTAUR does not possess the ability to perform purely one-dimensional calculations, because of its two-dimensional polygon-based input. However, by using rectangular cells

which are very tall relative to their width and by applying reflective boundary conditions at the top and bottom ends of the cells, any end effects can be diminished. Thus, the CENTAUR model consists of 20, 0.2x10.0 cm rectangular cells, 10 in each region. A TWOTRAN model was created using the same cell structure using x-y geometry, and an ANISN model was created using 20, 0.2 cm with cells, 10 in each region assuming slab geometry. For material properties, two-group macroscopic cross-sections were chosen which could represent the behavior of fuel and moderator materials. These properties are given for each material in Table 2. Scattering was assumed to be isotropic in the laboratory system. The fission distribution for each energy group is given by χ_g. For this and all the following calculations, the convergence criterion was set at 0.0001.

Material/ Energy Group	$\sigma_{a,g}$ (cm^{-1})	$\nu\sigma_{f,g}$ (cm^{-1})	$\sigma_{t,g}$ (cm^{-1})	$\sigma_{s,1->g}$ (cm^{-1})	$\sigma_{s,2->g}$ (cm^{-1})	χ_g
Fuel:						
g=1	0.235	0.2125	0.685	0.400	0.000	0.95
g=2	0.550	0.8750	0.900	0.050	0.350	0.05
Moderator:						
g=1	0.005	0.0000	0.270	0.015	0.000	–
g=2	0.015	0.0000	0.365	0.250	0.350	–

Table 2. Fuel and Moderator Macroscopic Cross-Sections for Case 1.

The results of this set of calculations are presented in Table 3, which gives the multiplication factor, k_∞, and regional absorption probabilities, P_a, for each energy group, for all three codes. Agreement between all three methods is good; CENTAUR and TWOTRAN are in closest agreement, although both are close to the values computed by ANISN. Error relative to the TWOTRAN solution is given in parentheses below each value; TWOTRAN was chosen as a reference to illustrate how close the CENTAUR

solution is to that of TWOTRAN, relative to the ANISN solution.

Both TWOTRAN and CENTAUR solutions were based on the same three-dimensional S_{16} quadrature set, while the ANISN solution was based on a one-dimensional S_{16} symmetric set. The different quadrature sets are the most likely cause of the differences in the solutions.

	ANISN	TWOTRAN	CENTAUR
k_∞	1.15065 (0.0626%)	1.14993 (0.000%)	1.14984 (0.00783%)
P_a, Fuel, g = 1	0.5557 (-0.305%)	0.5574 (0.000%)	0.5576 (0.0359%)
P_a, Fuel, g = 2	0.4076 (-0.394%)	0.4060 (0.000%)	0.4058 (-0.0493%)
P_a, Moderator, g = 1	0.0055 (1.852%)	0.0054 (0.000%)	0.0054 (0.000%)
P_a, Moderator, g = 2	0.0312 (0.320%)	0.0313 (0.000%)	0.0312 (0.320%)

Table 3. Calculated Values of k_∞ and P_a (and Error Relative to TWOTRAN) for Case 1.

Figure 15 illustrates the close agreement between the scalar fluxes computed using the various methods. In this figure, the scalar flux is plotted as a function of position for each energy group. The average error between CENTAUR and TWOTRAN is less than 0.04%, while the average error between CENTAUR and ANISN is less than 0.25%. The shape of the solutions are as one would expect, i.e., group 1 (fast) fluxes are peaked in the fuel, and depressed in the moderator, while the reverse is true of the group 2 (thermal) fluxes. Both curves have zero slope at the left and right boundaries, as required by the symmetry boundary conditions.

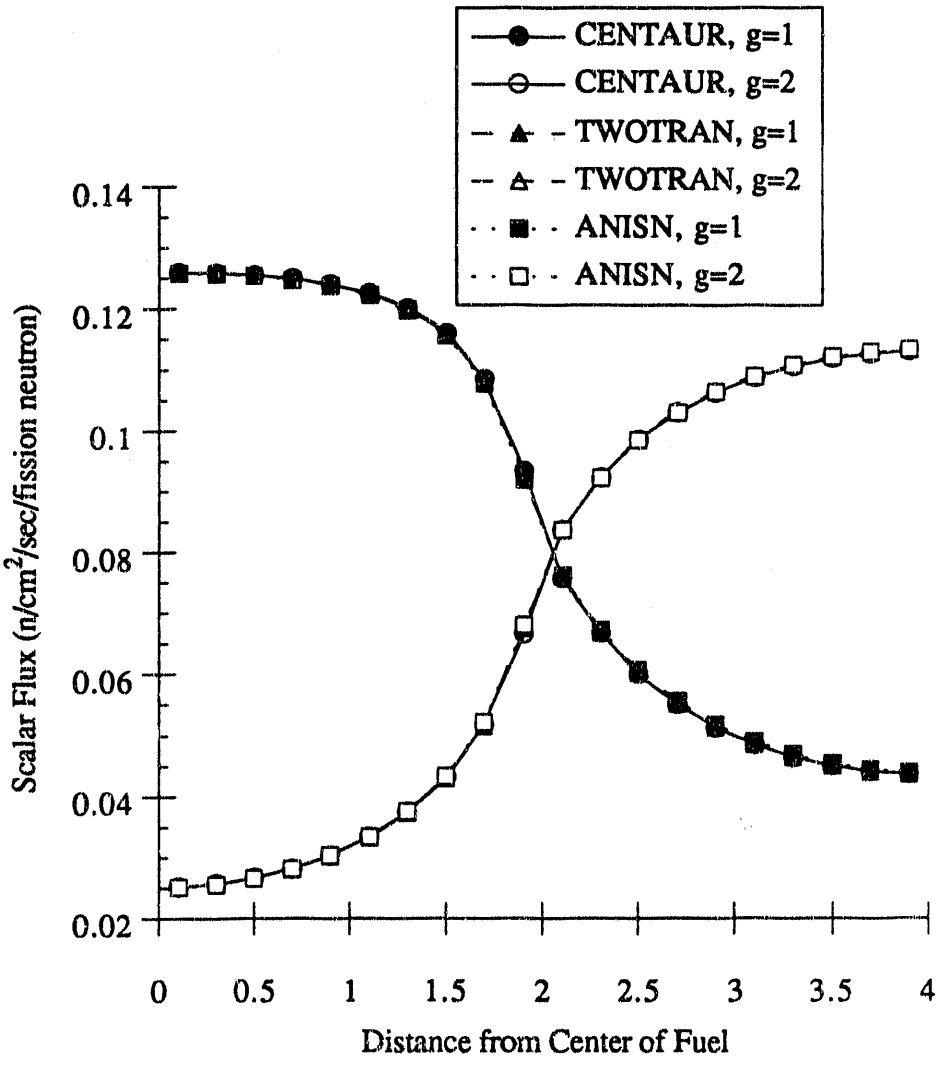


Figure 15. Comparison of Scalar Flux Solutions for Case 1.

Case 2: Ray Effects in a Square Non-Multiplying Medium with a Localized Source

One of the shortcomings of discrete ordinates methods is a class of phenomena termed "ray effects," in which anomalies in the shape of the flux solution can occur in regions dominated by strongly absorbing media, or at distances beyond a localized driving source region; such effects usually diminish with higher order quadrature. Ray effects are entirely

due to the inability of a quadrature set to accurately integrate the angular flux to obtain a scalar flux, even if the angular flux solution is exact for each discrete direction.[4] It is most dominant, therefore, in problems where significant streaming occurs, such that there is a strong angular variation in the neutron flux. In an infinitely fine spatial grid, where spatial error terms are close to zero, the ray effect will be essentially the same for any two methods which use the same quadrature set for angular integration, regardless of the form of the spatial approximation. As the size of the spatial grid increases, numerical diffusion will artificially introduce some degree of "scattering" into the streaming solution, which will tend to flatten the angular distribution of the neutron flux. This will reduce the magnitude of anomalies present in the scalar flux solution. Hence, an increase in spatial truncation error will artifically attenuate ray effects.

Case 2 demonstrates ray effects for such a situation in the ESC method, relative to TWOTRAN calculations. The geometry of Case 2, taken from Reference 4, is shown in Figure 16a; because of symmetry, the problem can be reduced to the form illustrated in Figure 16b. In order to assess the error in the two discrete ordinates solutions due to ray effects, it is necessary to have an "exact" solution for comparison. Although the geometry is too complicated to obtain an exact analytical solution, it lends itself to a simple Monte Carlo solution. A single energy group was assumed, with the following macroscopic cross-sections used in all regions: $\sigma_s = 0.50$ and $\sigma_a = 0.25$. An external source density of 1 n/cm^2-sec was used within the source region. Monte Carlo, TWOTRAN, and CENTAUR solutions were obtained using a 30x30 square grid within the 2cm x 2cm area shown in Figure 15b. Discrete ordinates results were generated using S_4, S_{10}, and S_{16} quadrature sets; the Monte Carlo solution was obtained using 2×10^8 histories.

Figures 17, 18, and 19 illustrate the discrete ordinates solutions for both methods, for each order of quadrature, relative to the Monte Carlo solution. The error associated with each point in the Monte Carlo solution is small ($\leq 0.38\%$); for illustrative purposes this

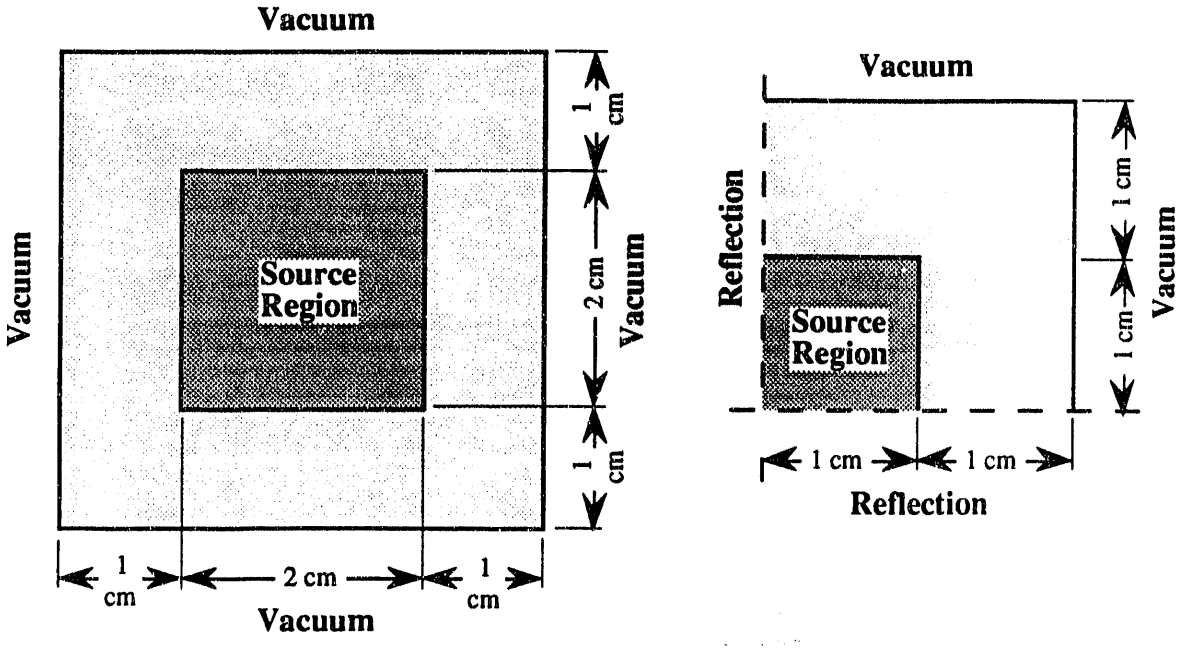


Figure 16 Localized Source Geometry of Case 2.

solution is considered exact, and error bars were omitted. In each figure, the scalar flux moving vertically along the right hand side of the volume (at x = 1.9667 cm) is plotted. Figure 17 demonstrates ray effects when using an S_4 quadrature set. As would be expected, both methods suffer from ray effects, although the magnitude of the error introduced in the ESC solution of CENTAUR is less than that suffered by the finite difference solution of TWOTRAN. Although it would appear that the CENTAUR solution is a better approximation than the TWOTRAN solution, the previous discussion of ray effects suggests that the reduced fluctuations in the spatial distribution of the scalar flux results from a larger truncation error in the CENTAUR solution.

Figures 18 and 19 exhibit the same behavior for S_{10} and S_{16} quadrature sets respectively; as the order of the quadrature increases, the magnitude of the error decreases for both methods, as one would presume. The figures serve to illustrate that the ESC method is in good agreement with the finite differenced TWOTRAN solution, but indicate that the truncation error of the spatial approximation in the ESC method is larger than that of

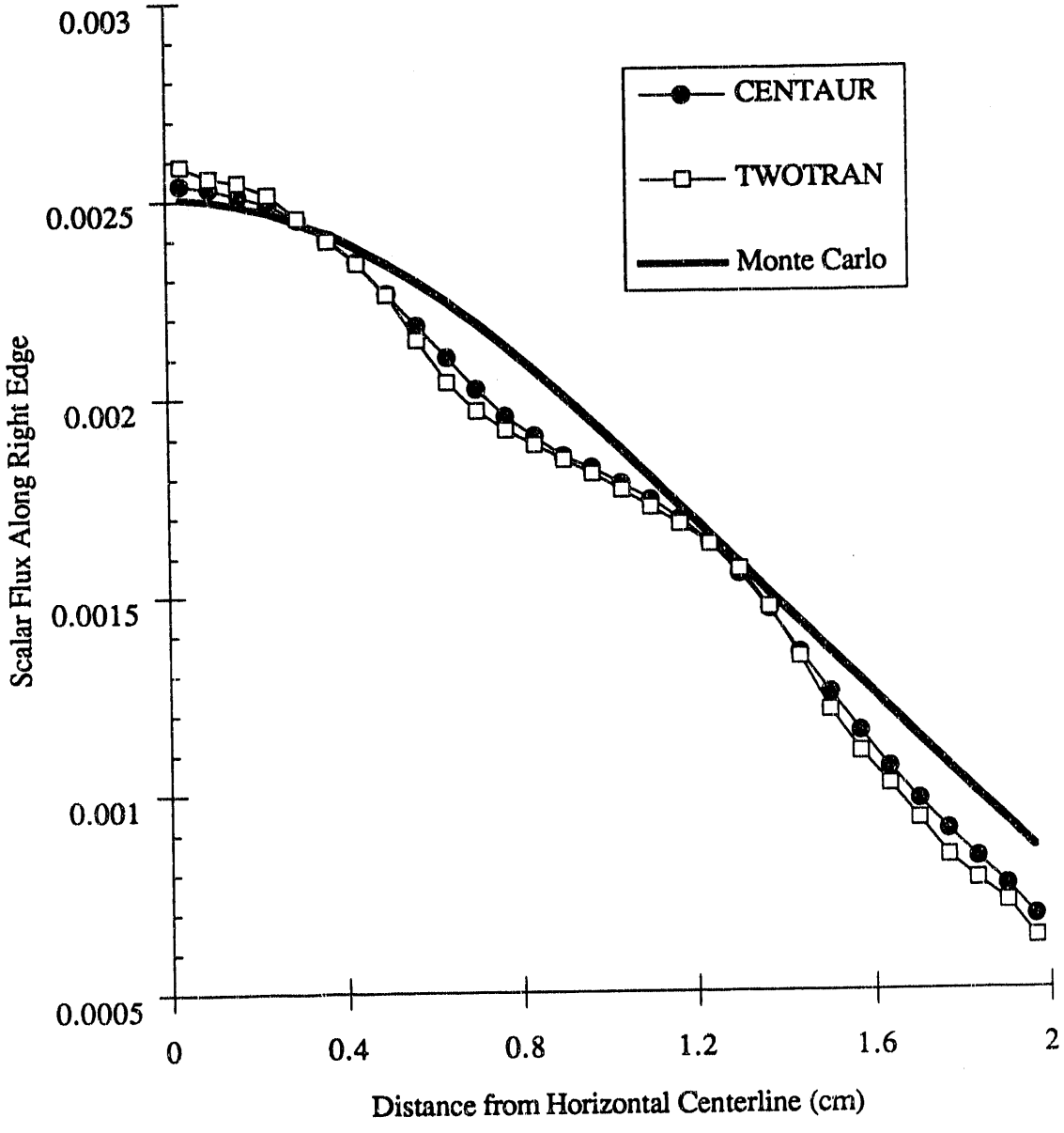


Figure 17. Ray Effects Using S_4 Quadrature.

the diamond difference approximation for the spatial derivatives. This behavior is not unexpected; as was discussed in Chapter II, the truncation error associated with the SC approach is slightly greater than that of the DD approximation. Since in rectangular cells the ESC approximation reduces to the simpler SC method, the truncation error for the ESC approach will be greater than that of the DD method of TWOTRAN. This is expected to

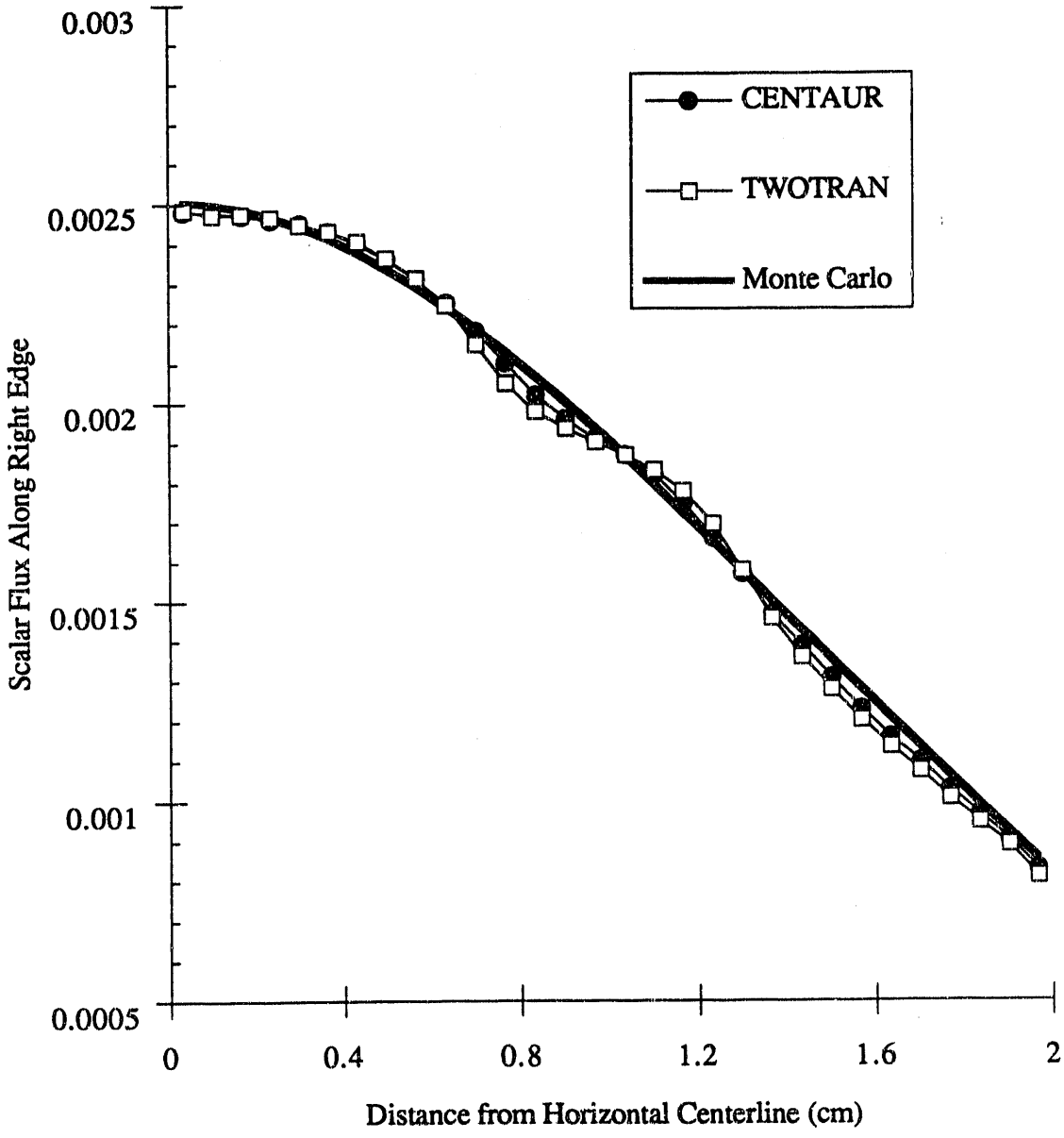


Figure 18. Ray Effects Using S_{10} Quadrature.

also be true for non-rectangular cells, although a direct comparison of non-rectangular cells cannot be made, since finite difference methods are restricted to rectangular cells in the x-y coordinate system. The effect of a larger truncation error in a model comprised of non-rectangular cells is addressed later in this chapter, in Case 4.

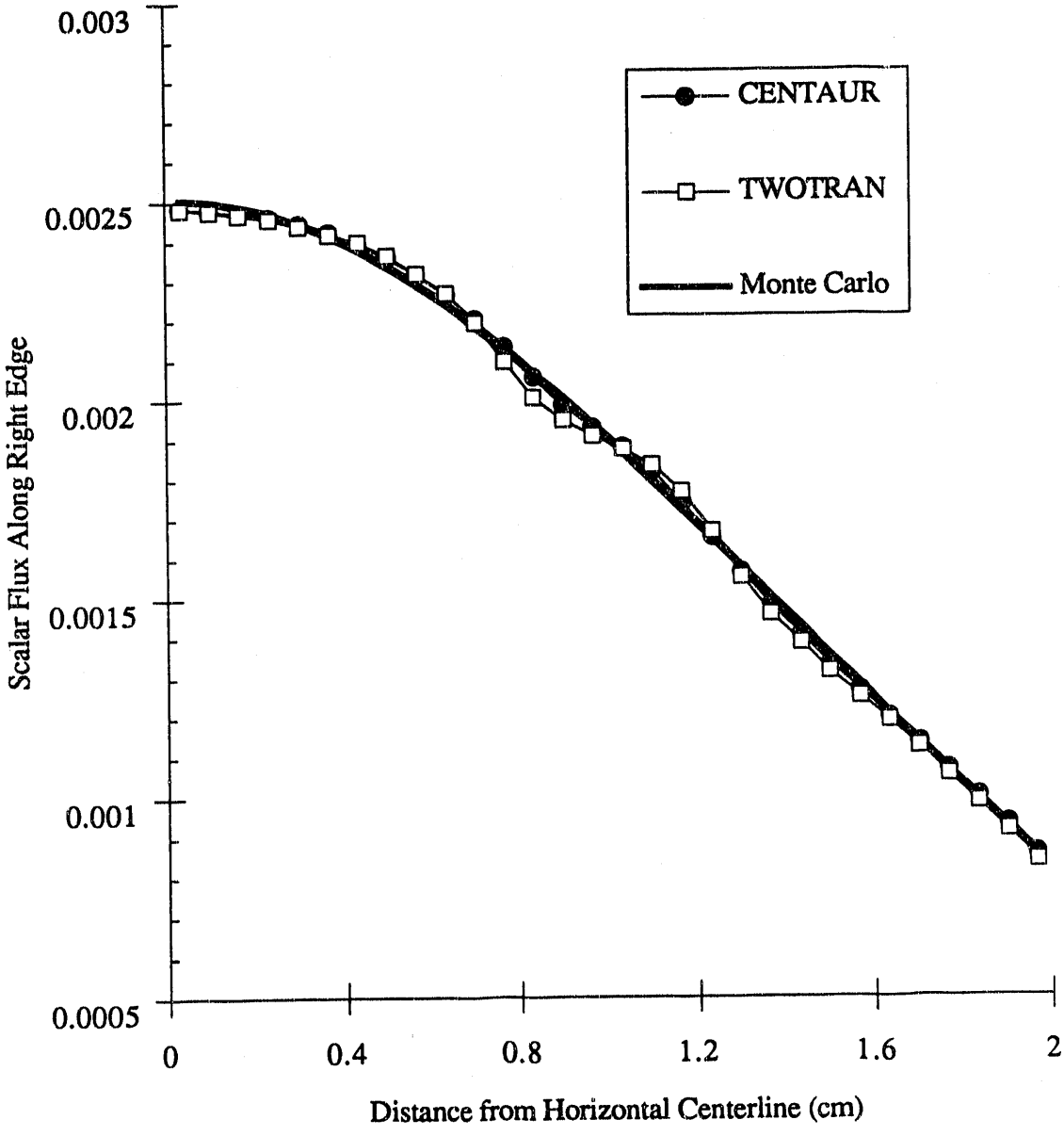


Figure 19. Ray Effects Using S_{16} Quadrature.

Case 3: Escape Probability for an Infinite Absorbing Medium

The purpose of Case 3 was to investigate the results from the ESC method when used for cylindrical geometry. An analytical solution has been derived[41] for the escape probability for monoenergetic neutrons in an infinite cylinder comprised of a uniform source

and a purely absorbing medium. This geometry was represented within CENTAUR as an approximate quarter-cylinder as illustrated in Figure 20, using 6, 12, and 24 azimuthal divisions and 10 radial divisions. Both ANISN and TWOTRAN can represent the geometry exactly in cylindrical coordinates. For the same discretization, ANISN would use 10 radial divisions in cylindrical geometry; TWOTRAN would also use 10 radial divisions, but would be restricted to a fixed number of azimuthal divisions, as the finite difference method does not allow for variability in azimuthal distributions. For the purposes of this comparison, 24 azimuthal divisions were used. For all runs, S_{16} quadrature was used. A constant radius of 1.0 cm was used for all calculations, and the absorption cross-section was varied to obtain a varying effective radius in terms of mean free paths. Because a unit radius was used, the absorption cross-section may also be considered to be the cylinder radius in units of mean free paths.

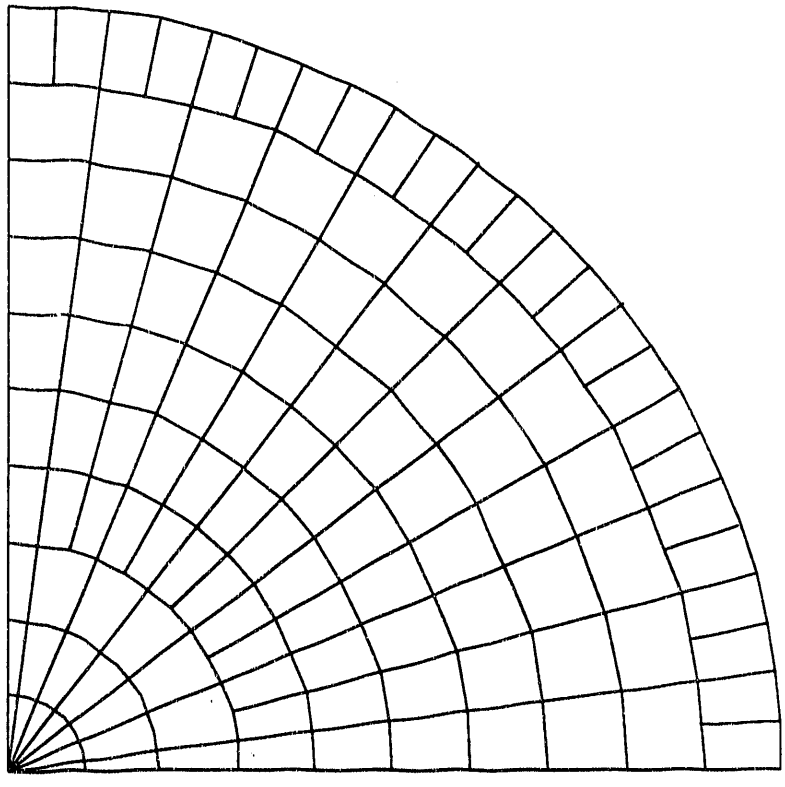


Figure 20. CENTAUR Computational Grid for Case 3.

Table 4 summarizes the results of the calculations, giving the escape probability as a function of absorption cross-section for various cross-sections. These results indicate that although ANISN is in closest agreement with the exact solution, CENTAUR results are significantly closer to the true solution than those of TWOTRAN. The reason for this is not clear, although it is possible that approximations made in coupling the angular and spatial derivative terms in the two-dimensional TWOTRAN model may introduce error not present in the ANISN and CENTAUR models.

	Absorption Cross-Section (cm^{-1})					
	0.5	1.0	1.5	2.0	3.0	4.0
Analytic Solution	0.40405	0.59285	0.69843	0.76355	0.83714	0.87654
ANISN	0.40429 (0.059%)	0.59284 (-0.002%)	0.69828 (-0.021%)	0.76334 (-0.028%)	0.83689 (-0.030%)	0.87629 (-0.029%)
TWOTRAN	0.40620 (0.488%)	0.59422 (0.231%)	0.69941 (0.140%)	0.76431 (0.100%)	0.83766 (0.062%)	0.87693 (0.044%)
CENTAUR	0.40457 (0.129%)	0.59262 (-0.014%)	0.69789 (-0.077%)	0.76296 (-0.077%)	0.83666 (-0.057%)	0.87622 (-0.037%)

Table 4. Escape Probabilities for an Infinite Absorbing Cylinder of Case 3.

Further numerical experiments were performed using a finer azimuthal grid. The solution was not found to significantly improve with increased angular discretization; on the contrary, for lower quadrature orders, the accuracy of the solution was found to degrade

slightly as the azimuthal "width" of the cells became smaller. This problem, related to the shape of a cell, is discussed in detail in Case 4.

Case 4: Solution in a Composite Cylinder with Azimuthal Symmetry

Case 4 is an extension of Case 3, in which additional material regions have been introduced in order to test the robustness of the method. It will be seen that this problem also shows one of the limitations of CENTAUR and the ESC method; this will be described in more detail in following paragraphs.

In this case, as in the previous one, a cylinder is modelled by a set of polygons approximating the curved surfaces. However, for the fourth problem the cylinder is comprised of alternating rings of fissionable fuel and a non-multiplying moderator, similar to that of Case 1. In Case 4, the outer region consists of moderator material, with a white boundary condition on the outermost side of the region. Figure 21a shows the full two-dimensional problem, while Figure 21b shows how symmetry was used to reduce the size of the problem actually modelled. Again, while the one-dimensional ANISN calculation does not allow azimuthal cells, both CENTAUR and TWOTRAN are two-dimensional codes and need a more complicated representation.

A two-group solution is sought, using the properties given in Table 5. Results are summarized in Table 6, which gives the multiplication factor for the problem and regional absorption probabilities for each energy group. All results are compared in terms of the percentage difference relative to the TWOTRAN solution. The TWOTRAN solution does not necessarily represent the closest approximation to the exact solution; however, in the absence of an exact solution, it was chosen as the reference to be consistent with the comparison of Table 3.

The multiplication factor for the problem was determined to be approximately 1.305 by both ANISN and TWOTRAN; the CENTAUR calculation gave a value of approximately

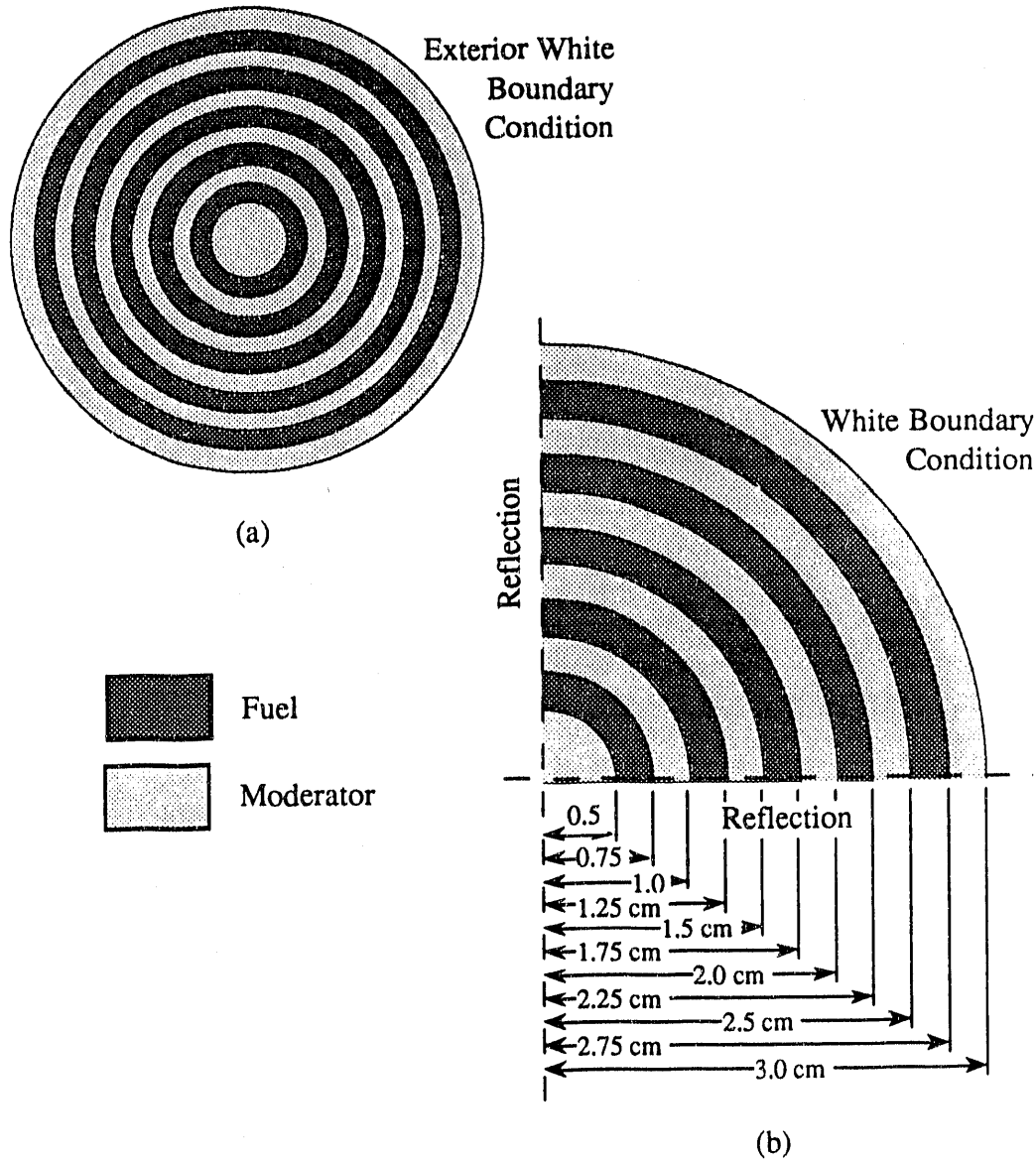


Figure 21. Nested Ring Cylindrical Geometry of Case 4.

1.303. This is a larger difference than was expected; the reason for this discrepancy is not certain. However, when ANISN was run for the same problem using a step (forward differenced) differencing approximation instead of the linear (central differenced) differencing used in previous examples, the computed value of the multiplication factor was 1.30220. Thus, since the error for the linear approximation is $O(\Delta h^2)$ while for the step approximation the error is $O(\Delta h)$, where h represents the mesh size,[3] one might conclude

Material/ Energy Group	$\sigma_{a,g}$ (cm)	$\nu\sigma_{f,g}$ (cm)	$\sigma_{t,g}$ (cm)	$\sigma_{s,1->g}$ (cm)	$\sigma_{s,2->g}$ (cm)	χ_g
Fuel:						
g=1	0.235	0.2125	0.435	0.080	0.000	0.95
g=2	0.550	0.8750	0.900	0.120	0.350	0.05
Moderator:						
g=1	0.005	0.0000	0.270	0.015	0.000	–
g=2	0.015	0.0000	0.365	0.250	0.350	–

Table 5. Fuel and Moderator Macroscopic Cross-Sections for Case 4.

	ANISN	TWOTRAN	CENTAUR
k_∞	1.30541 (0.0299%)	1.30502 (0.000%)	1.30304 (-0.152%)
P_a, Fuel, g = 1	0.3447 (-0.174%)	0.3453 (0.000%)	0.3463 (0.290%)
P_a, Fuel, g = 2	0.6247 (0.112%)	0.6240 (0.000%)	0.6221 (-0.304%)
P_a, Moderator, g = 1	0.0084 (5.00%)	0.0080 (0.000%)	0.0084 (5.00%)
P_a, Moderator, g = 2	0.0222 (-0.448%)	0.0223 (0.000%)	0.0224 (0.448%)

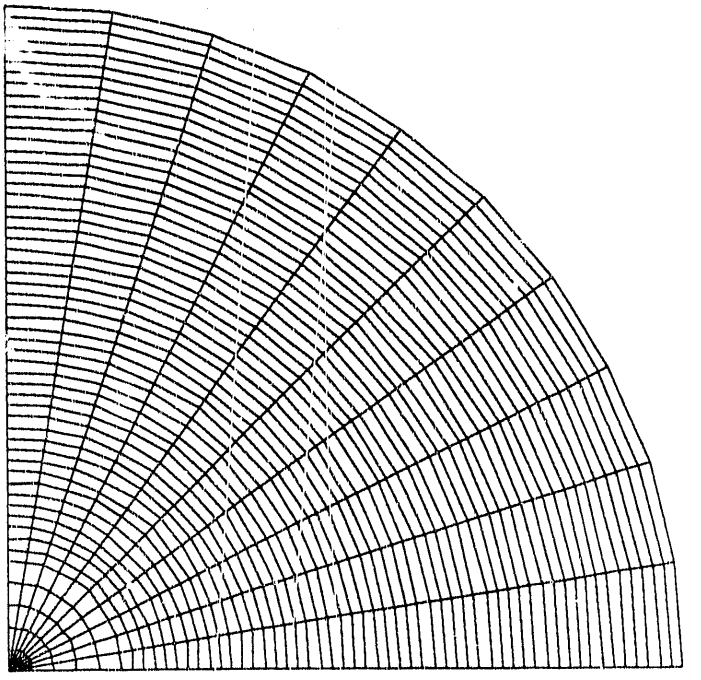
Table 6. Calculated Values of k_∞ and P_a (and Error Relative to TWOTRAN) for Case 4.

that the computed value of the multiplication factor is reduced for this geometry when the order of the error term increases. As discussed in Chapter II, the error of the Step Characteristic method is less than $O(\Delta h^2)$ in rectangular geometries, and increases as the height to width ratio changes from 1.0. Because of the complicated geometrical

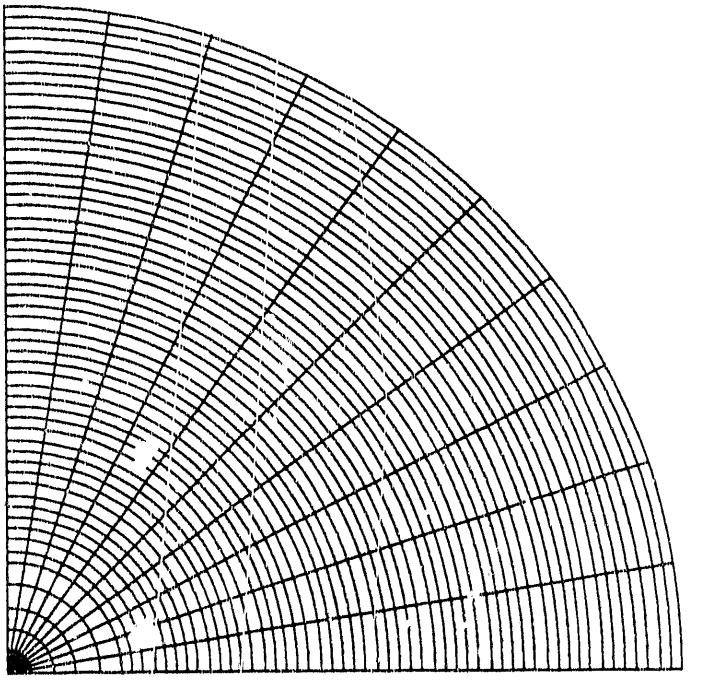
considerations involved in the ESC method, no error analysis has been performed. However, one might reasonably expect that the error would increase by the same mechanism when the height to width ratio is significantly larger than 1.0 for cells which approximate long rectangular cells, as are found in the model used for this case. Hence the error would be greater than that of the linearly differenced discrete ordinates approach. If the multiplication factor is indeed sensitive to this error term, then this would explain the lower value computed by CENTAUR. Figure 22 illustrates the grid structure used in the CENTAUR and TWOTRAN models, and shows the elongated nature of the quasi-rectangular cells in the CENTAUR grid. Five radial cells were used in each region, as were used in the TWOTRAN and ANISN models. For both TWOTRAN and CENTAUR, the quadrant was subdivided into 10 azimuthal zones; however, it should be noted that because of the azimuthal symmetry, only one azimuthal division was necessary for TWOTRAN.

Total absorption rates, P_a, computed for each region type are in good agreement, although in the fuel regions the difference between the CENTAUR and TWOTRAN is approximately twice that between TWOTRAN and ANISN. A plot of the fluxes calculated by all three methods for each energy group are shown in Figures 23 and 24. These figures demonstrate that the CENTAUR solution is in good agreement overall with the results of the other two codes, and show the strong radial variation in the scalar flux. In the group 1 (fast) solution, the three solutions are in good agreement for small radii; however, as the radius increases, the deviation of the CENTAUR solution from the other two solutions increases. This would tend to support the earlier proposition that the accuracy of the CENTAUR solution is limited by long, thin cells, since the length to width ratio increases as the radius gets larger.

In further numerical experiments, as the number of azimuthal cells in the CENTAUR model was increased, the multiplication factor moved toward the value of ~1.305 predicted by the finite difference based codes. For an arrangement using 10, 20, and 40 azimuthal



(a)



(b)

Figure 22. CENTAUR (a) and TWOTRAN (b) Computational Grids Used in Case 4.

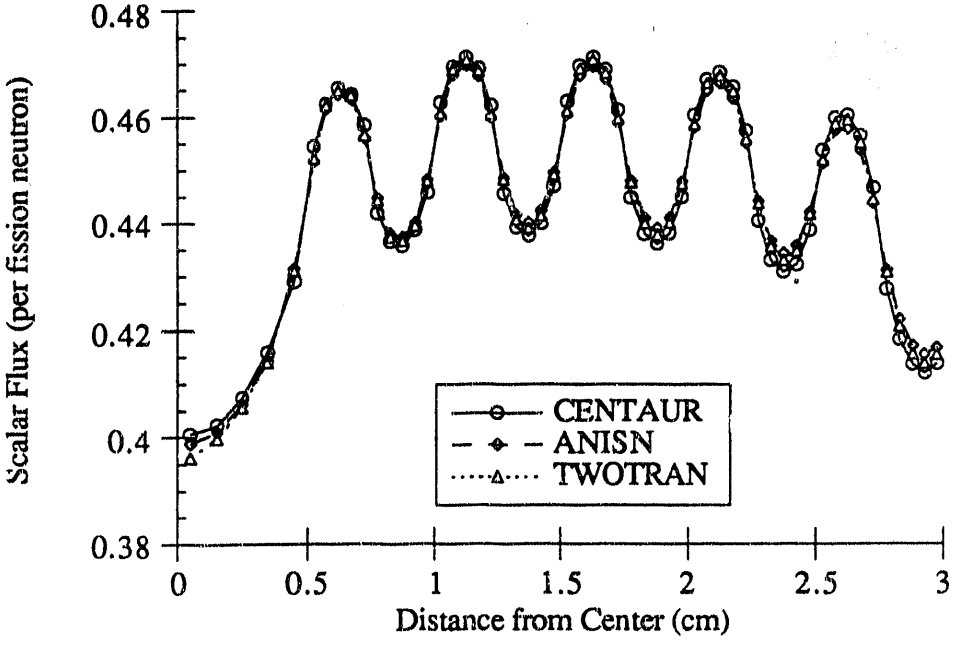


Figure 23. Fast Group Scalar Fluxes from Case 4.

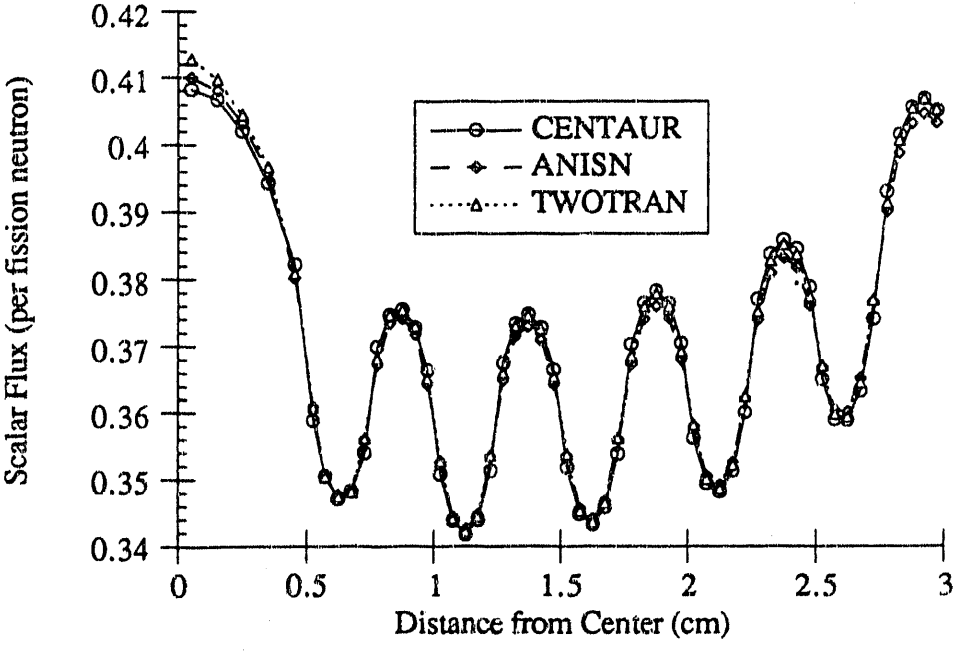


Figure 24. Thermal Group Scalar Fluxes from Case 4.

divisions for rings 1-3, 4-7, and 8-11 respectively, a multiplication factor of 1.30412 was computed. It was felt that perhaps the strong radial variation in the flux due to the structure of the problem might require further radial subdivisions to more accurately resolve the radial distribution. Unfortunately, a finer radial grid would also require a finer azimuthal grid, which would exceed the memory limitations of CENTAUR. Thus, finer cell structures were not studied. However, additional work with cylindrical geometries and simpler structures has indicated that accuracy is improved (in terms of agreement with ANISN and TWOTRAN) when cells are constructed of approximately equal-length sides. The cells are not required to have a square shape in order to obtain this improvement; triangles and pentagons do not appear to cause problems, as long as they are not severely distorted.

While investigating the error described above, a second anomaly was discovered in the CENTAUR solution for this problem. It was found that the azimuthal flux profile, which should remain constant for this problem, actually had a significant variation with angle. The magnitude of the variation was found to decrease with increasing order of quadrature. Figure 25 illustrates the azimuthal variation of the flux in a fuel ring for this problem geometry. The amount of the deviation from the average value decreases with increasing order of quadrature, although the average remains approximately constant. The maximum deviation from average is as large as 1.40% for the S_4 quadrature to 0.35% for S_{16} quadrature. In addition, the variation is symmetric about any multiple of 45°.

From the behavior shown in Figure 25 and from numerical experiments, the following observations were made. The behavior is a result of the combined effects of the shape and orientation of computational cells and of the quadrature approximation of exact integration . It occurs when cells which are significantly larger in one dimension than in others are used, and is therefore related to the error described earlier in this section. It also results from the rotation of these cells; if the orientation of the single long dimension is the same for all cells, the phenomenon is not observed. Use of product-angular quadrature sets,[42] in which the

azimuthal spacing of the component of each direction in the x-y plane is constant, also diminishes the effect. (Unfortunately, the overall accuracy of the solutions obtained using this type of quadrature is reduced.)

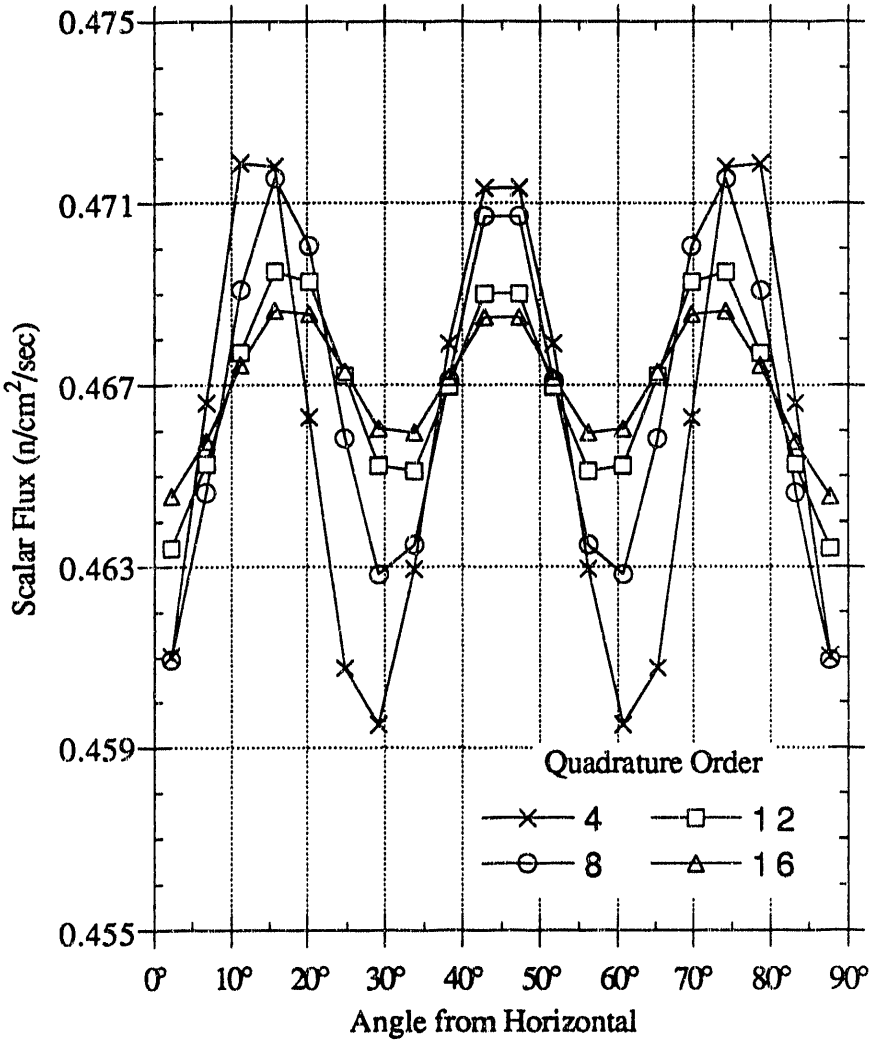


Figure 25. Azimuthal Variation in the Scalar Flux.

The problem is due to a ray effect of a slightly different nature than that normally observed in discrete ordinates methods. The relative weight (expressed in terms of relational coefficients) of one side to an opposite side in a four sided polygon for the set of characteristic directions varies according to the orientation of the polygon. As the polygon

becomes more square, the weight of one side relative to the opposite side becomes less direction dependent, and the orientation will have less effect on the solution. The more elongated a given dimension becomes, the more sensitive the solution becomes to the quadrature directions, requiring a higher order quadrature to diminish the effect.

Both of the problems described in the previous paragraphs illustrate a weakness in the ESC method; however, the cause of these problems is related to the shape of the computational cell. While these limitations indicate that completely arbitrary and unconstrained cells structures can cause problems, this is a problem with all discrete ordinates methods, and it appears that there remains considerable latitude in the shape and structure of a cell.

Case 5: Application in a Non-Orthogonal Geometry

Case 5 was designed to show the advantage of the ESC approach over conventional approaches for a non-orthogonal geometry. For this case, an annular fuel rod was placed in an infinitely repeating hexagonal lattice, as shown in Figure 26. In this geometry, a rectangular region of symmetry can be used to simplify the problem to one which could be reasonably approximated by TWOTRAN, as represented by the rectangular outline in Figure 26. The rectangular region is required so that TWOTRAN can apply reflective boundary conditions; a large number of small square cells were used to approximate the cylindrical regions of the fuel rod and enclosed moderator. The computational grid representing the rectangular region is shown in Figure 27. This model required a 30 by 52 cell computational grid using 0.1 by 0.1 cm cells, for a total of 1560 cells, and provides only a coarse approximation of the rounded surfaces of the annulus. The CENTAUR model for the same region takes advantage of its flexible cell structure to give a much closer approximation to the curved surfaces, as shown in Figure 28. In addition, this model used only 564 computational cells, of various sizes and shapes.

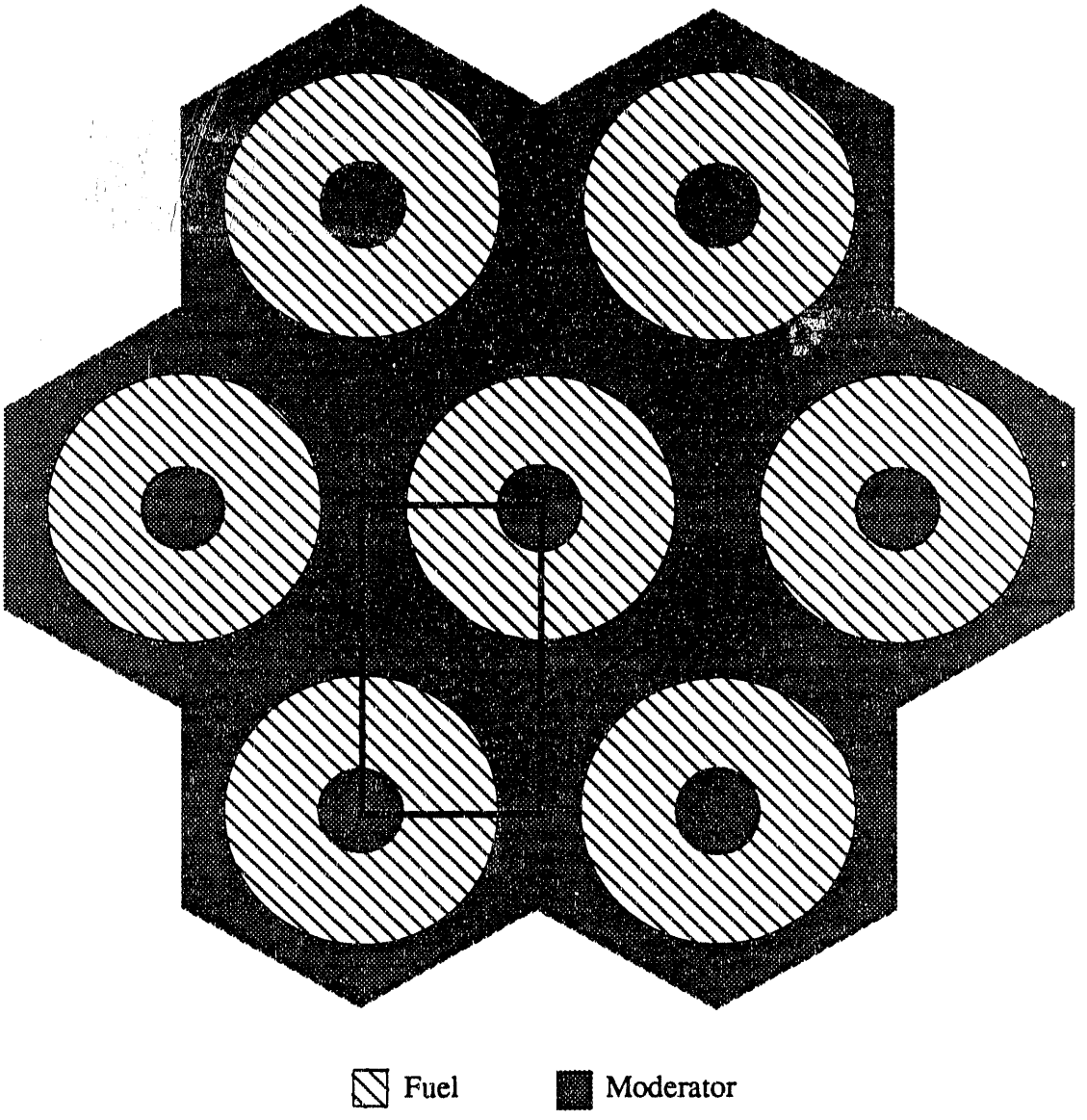


Figure 26. Non-Orthogonal Hexagonal Geometry Used in Case 5.

The problem geometry was intentionally formulated such that it could be approximated by a fine rectangular mesh. Additional fine structure, such as thin cladding around the fuel material, would be difficult to model using TWOTRAN without increasing the number of cells by at least an order of magnitude, while the CENTAUR model could easily accommodate such features with little additional overhead.

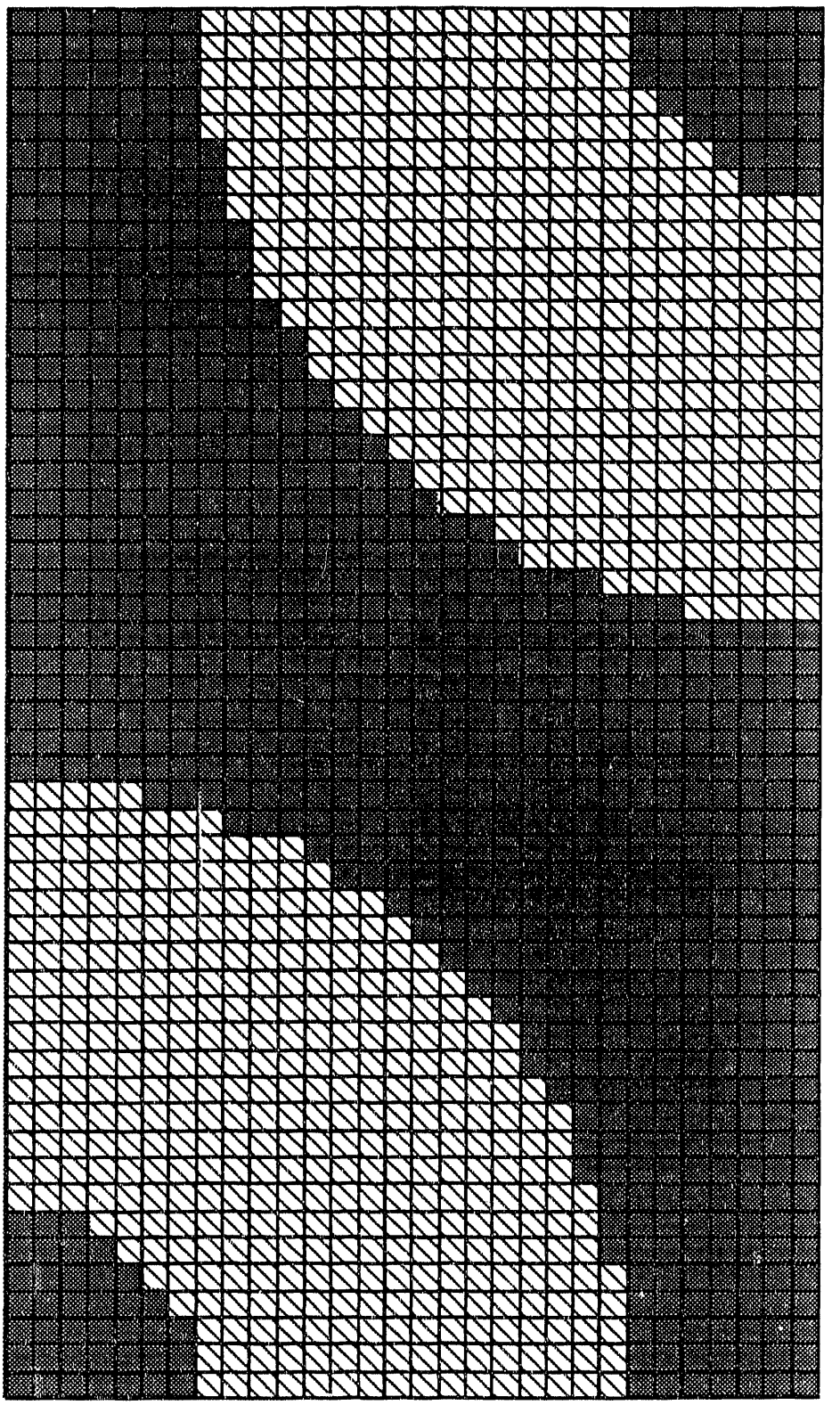


Figure 27. Computational Grid Used by TWOTRAN in the Case 5 Geometry.

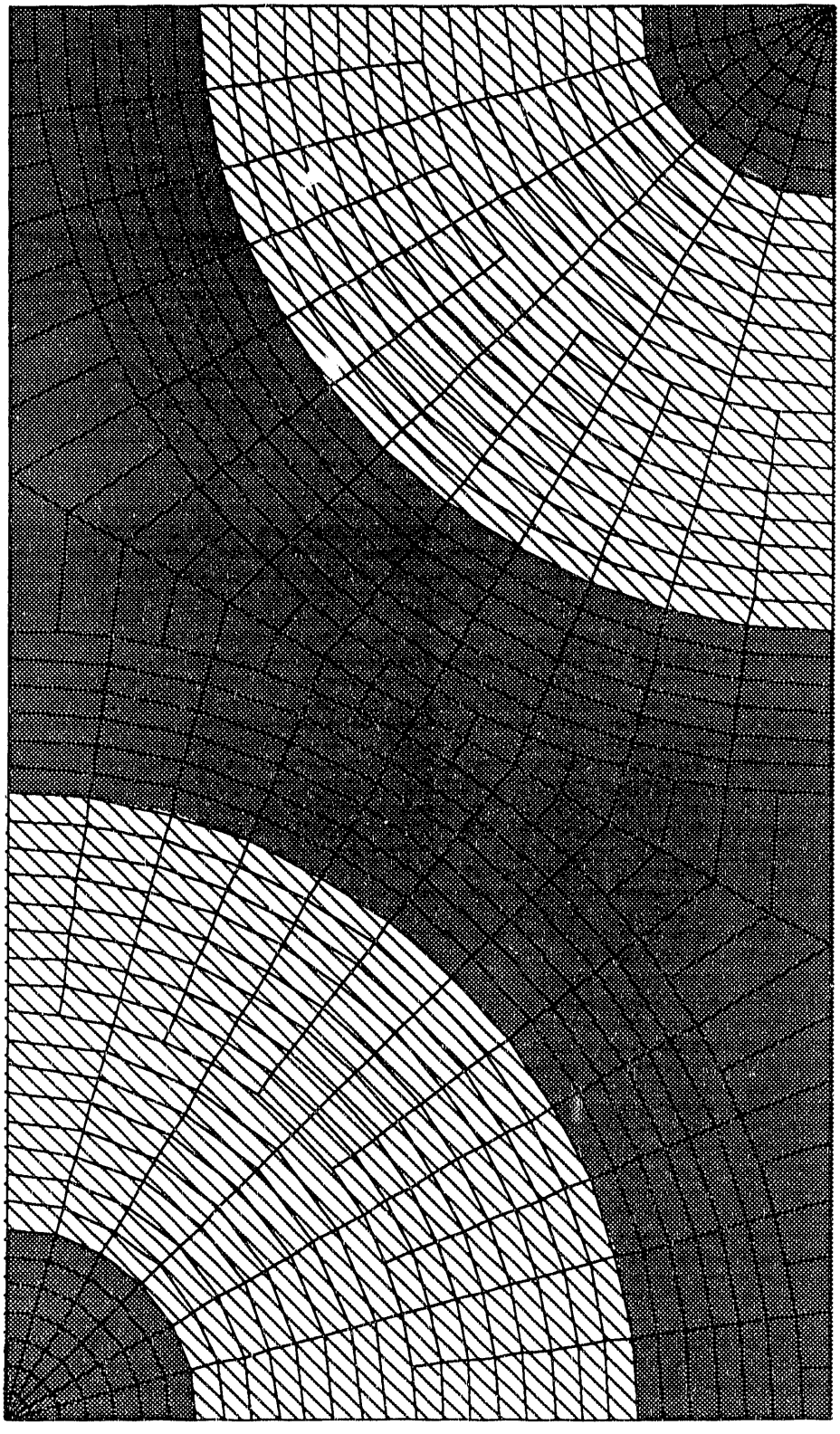


Figure 28. Computational Grid Used by CENTAUR in the Case 5 Geometry.

The geometry of Figure 26 is described by an annular fuel rod of inner diameter 1.4 cm, outer diameter 4.6112 cm, and a center to center pitch of 6.0 cm between fuel rods. Because of the crude approximation resulting from the use of square cells to represent curved surfaces in the TWOTRAN model, it was not possible to match both volumes and surface areas for each region. Thus, it was decided to match the actual volume of the fuel region rather than the surface area of the region. Even with CENTAUR, while it was possible to more closely approximate the surface area of a fuel rod, there was a difference between the actual surface area and the area used in the model. Table 7 shows the volume and surface area of a fuel rod for both the CENTAUR and TWOTRAN models relative to the exact values. The values in brackets give the error in each approximation relative to the exact dimensions. Because the total volume of fuel plus moderator is fixed by the boundaries, matching fuel volumes in each model with the exact volume will also result in exact moderator volumes. However, no attempt was made force the volumes for each moderator region to be constant.

	Actual	CENTAUR	TWOTRAN
Volume	15.1607	15.1607	15.16
(cm^3/unit depth)		[0.0%]	[-0.005%]
Inner Surface Area	4.3982	4.3856	5.6
(cm^2/unit depth)		[-0.29%]	[27%]
Outer Surface Area	14.4865	14.4410	18.4
(cm^2/unit depth)		[-0.31%]	[27%]

Table 7. Fuel Volume and Surface Areas in CENTAUR and TWOTRAN Models

It is worth noting that the CENTAUR model could have been based on a single hexagonal cell, as shown by the hexagonal outline in Figure 26. Instead of reflective

boundary conditions, periodic boundary conditions could be applied for opposing sides. As with traditional discrete ordinates codes, reflection in CENTAUR is limited to surfaces with horizontal or vertical orientations, because of the $\pi/2$ rotational symmetry of the quadrature set used. Thus, while reflection could have been used along the two vertical sides of the hex, it could not be used along the hex's other four sides. However, if a quadrature set with $\pi/3$ rotational symmetry were used, reflective boundary conditons could be used on all sides.

Using the material properties from Case 4, given in Table 5, the problem was solved as a two-group criticality problem. Agreement between the multiplication factors was found to be much closer than had been expected, based on the Case 4 results. The multiplication factor and the group absorption probabilities computed by each code are given in Table 8.

	TWOTRAN	CENTAUR	% Difference
k_∞	1.25557	1.25551	-0.0047
P_a (fuel + moderator), g=1	0.420	0.419	-0.24
P_a (fuel + moderator), g=2	0.580	0.581	0.17

Table 8. Calculated Values of k_∞ and P_a for Case 5.

Excellent agreement is seen between the two codes. The scalar flux distributions were also found to be in close agreement, as shown in Figure 29. This plot depicts the flux profile moving horizontally from the center of the moderator region located inside a fuel element to the centerline of the moderator space between two fuel elements, i.e., along the bottom boundary of the rectangular region.

Note that in Figure 28 the shape of the computational cells, although more rectangular than square, are not as severely distorted as the outermost cells in the previous test case. The close agreement with the multiplication factor computed by TWOTRAN would indicate

that either this amount of distortion has very little effect on the accuracy of the solution, or that both the TWOTRAN and CENTAUR solutions are in error to approximately the same degree. The latter is unlikely, however, unless the agreement in results is very coincidental. Additionally, the radial variation of the flux, as evidenced in Figure 29, is not as severe as in the previous case.

It might be argued that if the crude TWOTRAN model is capable of such accuracy, then there is no clear advantage in using the ESC method in non-orthogonal geometries. While this is true for this particular example, it should be realized that this problem was selected such that a TWOTRAN solution could be obtained. As discussed earlier, additional details such as cladding would make the TWOTRAN model significantly more complicated. In addition, if the hexagonal lattice shown in Figure 26 had consisted of a safety rod element surrounded by six dissimilar fuel elements, the problem would have strained the capabilities of TWOTRAN and other finite-difference based codes to a much higher degree.

The five cases which have been presented in this chapter represent only a fraction of the cases which have been run to test CENTAUR. These cases where chosen for discussion here because they were particularly effective in highlighting the advantages and limitations of the ESC method in modelling various geometrical configurations. Hopefully, these examples have served not only to demonstrate the accuracy of the CENTAUR code, but also to highlight the advantages of the ESC approach.

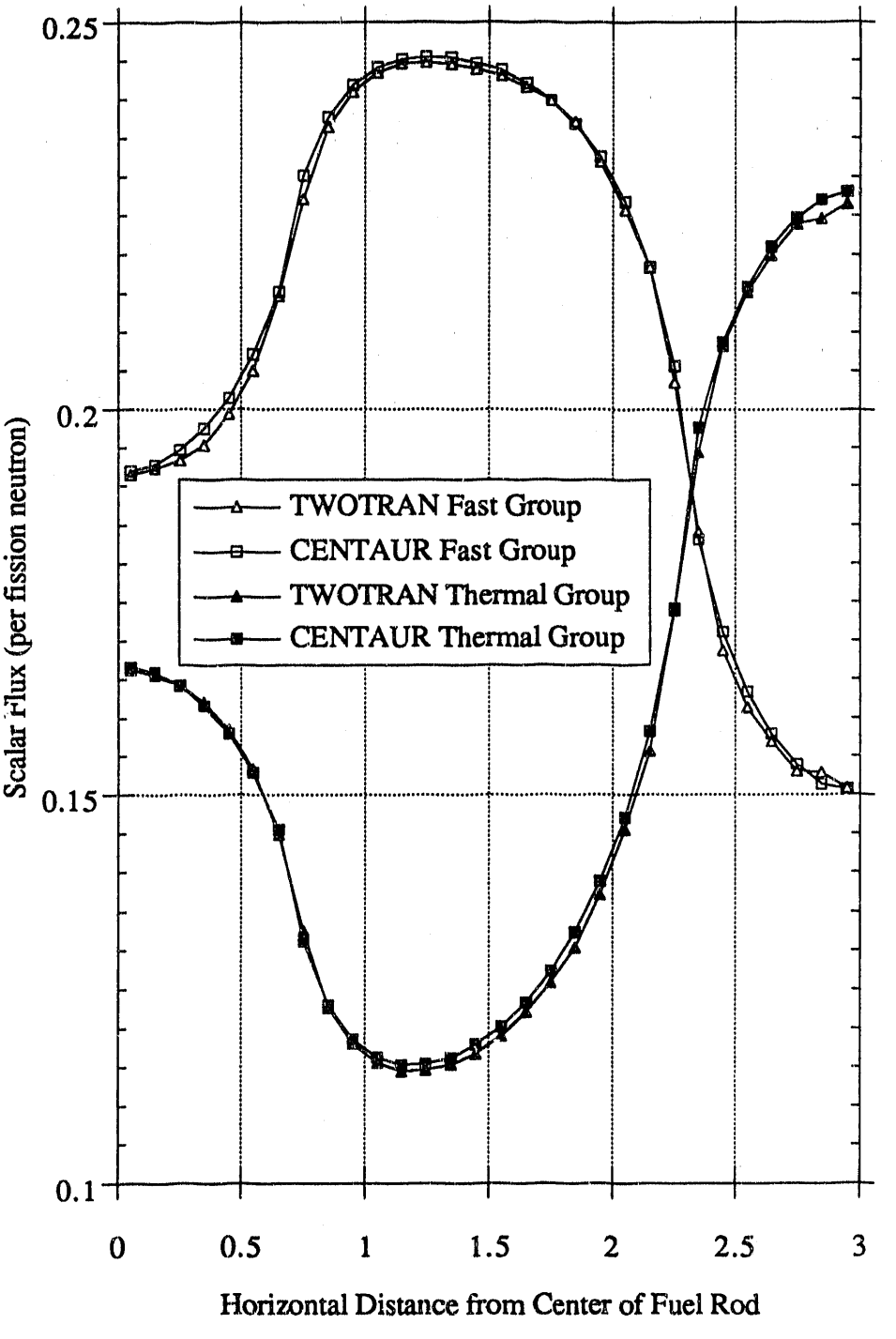


Figure 29. Fast and Thermal Fluxes from the Results of Case 5 Calculations.

CHAPTER VI

CONCLUSIONS

This dissertation has demonstrated the theory and application of the Extended Step Characteristic method, an extension of Lathrop's Step Characteristic method to irregularly shaped polygons. This method was incorporated into the CENTAUR computer program, and was validated with comparisons to the analytical solution for a simple cylinder and the numerical results of other, finite difference based discrete ordinates programs. These results demonstrated that the ESC method is in good agreement with existing discrete ordinates methods, and has the capability to use irregular cell structures in order to approximate the general shapes of arbitrary problem areas. This chapter will discuss some of the features and limitations of the method, and will explore some of the potential avenues for future work in both the theoretical and applied numerical aspects of the method.

Strengths and Weaknesses of the Extended Step Characteristic Method

The CENTAUR code has been developed as a proof-of-principle tool for demonstrating the feasibility of the ESC method when translated into a numerical model. As such, it has not been designed to compete directly with existing discrete ordinates codes, and does not contain all of the features or capabilities of other production class programs. However, in order to solve general problems, it has been written as a versatile tool capable of analyzing multiregion, multigroup problems with a diverse set of boundary conditions. The five cases presented in the previous chapter utilize many of features of the CENTAUR program, and served to validate both CENTAUR and the theory developed in the derivation of the Extended Step Characteristic method. The following paragraphs will summarize the important features of each case, based on the results and discussions of Chapter V.

The first case demonstrated that CENTAUR (and therefore the ESC method) is in good agreement with the finite difference results from TWOTRAN and ANISN for one-dimensional slab geometry. However, this is to be expected; the step characteristic equation

(Equation 6) can be shown to reduce to a forward (or step) differenced finite difference approximation as the distance s becomes small. The ESC method is not as efficient as a finite difference method in this geometry; however, this is not the intended use of the method. It was recognized from the beginning that the ESC method would not be competitive with existing methods when applied to problems easily described with orthogonal coordinates. However, problems with orthogonal geometries provide the opportunity to benchmark CENTAUR results against existing methods.

The second case demonstrated similar agreement for a slightly more complicated two-dimensional problem. It also demonstrated that the ESC approach is subject to a slightly larger error than that of the diamond-differenced approach in TWOTRAN, as evidenced by reduced ray effects due to numerical diffusion. This result was expected, since in the rectangular geometry of this case, the ESC approximation reduces to the simpler Step Characteristic approach; SC is known to have a slightly higher error component than DD for a given rectangular grid. The larger error is also expected to be present in non-rectangular geometries.

The third case studied actually represents the first test of a non-orthogonal geometry for CENTAUR. Although the cylindrical geometry of this case can be modelled exactly using the form of the transport equation developed for cylindrical coordinates, it is only an approximation for CENTAUR. The ESC method is based on the characteristic form of the transport equation applied to straight-sided cells in an x-y specified geometry; thus the representation of the curved surfaces of a cylinder is one of the many "non-orthogonal" geometries for which the method was developed. Results showed not only a good agreement to the analytical solution for various material properties, but also better agreement to the exact results than could be obtained by the TWOTRAN code.

The complexity of the geometry used in Case 4 served to demonstrate a weakness of the ESC method. For this case, CENTAUR was unable to match ANISN and TWOTRAN

solutions for the same geometry as closely as in other tests. The differences were relatively small, but were most pronounced in the calculation of the multiplication factor for the problem. It was postulated that the error resulted a reduced order of accuracy due to the fact that computational cells were significantly elongated in a specific dimension relative to other dimensions. Although other numerical experiments seem to confirm this hypothesis, it has not been rigorously verified. However, such behavior would be consistent with the known behavior of the Step Characteristic method applied in rectangular geometries. In addition, it appeared that the solution also suffered from insufficient radial discretiztion due to a rapidly changing radial scalar flux profile. Limitations in the ability of CENTAUR to handle significantly smaller mesh sizes due to memory constraints precluded further study of this problem. (Memory constraints resulted from hardware limitations in the IBM RS/6000 workstation used in these analyses, and not from any theoretical limitation.)

A second problem, also related to the elongated cells, was identified as a ray effect resulting from the use of discrete directions to represent the angular distribution. This effect resulted from the fact that the major axis of the elongated cells was rotated relative to the set of fixed characteristic directions for various cells used to represent the geometry. Nevertheless, even with some inaccuracy, CENTAUR has demonstrated its applicability in what is again a non-orthogonal geometry for the method. This geometry would have been extremely difficult to model using using a finite difference approach with rectangular cells in Cartesian coordinates, because of the large number of cells which would be required.

The final case presented the results of a truly non-orthogonal geometry. The geometry was simplified in order to accommodate a TWOTRAN solution with a reasonable mesh size, since a solution for a more complicated geometry would be of little value without a parallel calculation to which it could be compared. Results were found to be in very good agreement, much more than was expected due to the coarseness of the TWOTRAN approximation. The CENTAUR solution required approximately one-third of the total

number of cells required by TWOTRAN, and still gave a closer approximation to the actual problem geometry. Additional details, such as cladding, gaps, or ribs, could have been easily introduced into the CENTAUR model, while such details would require an enormous increase in the number of cells needed in the TWOTRAN model.

With the exception of the Monte Carlo solution of Case 2, numerical comparisons made in all cases were limited to ANISN and TWOTRAN calculations. However, these two computer codes can be considered representative of practically all production class finite difference discrete ordinates codes relative to the limitations of cell structures. Although both finite difference and finite element methods based on other cell shapes have been derived, these have been principally limited to simple three and four sided cells, and have not exhibited the generality found in the ESC approach. Thus, the Extended Step Characteristic method has added more generality to the theory of discrete ordinates.

Limitations Within CENTAUR

While the preceding chapter demonstrated the applicability of the ESC method in generalized geometries, as well as some of the apparent limitations in the method's accuracy, little was said of the limitations inherent in CENTAUR itself. The purpose of this work was to provide the development of the theory and application of the Extended Step Characteristic approach; CENTAUR was merely a tool used to apply the method. Thus, the practical limitations of CENTAUR itself have not been addressed in detail. However, it is worth noting at this point that substantial limitations exist in the current form of CENTAUR. Some of these problems were anticipated prior to beginning the code development, but were not considered to be a detriment for the purposes of this research. Others were discovered as the development proceeded, and were addressed to the extent necessary to complete this work.

The primary limitation in CENTAUR is the time required to compute large, detailed problems. As mentioned earlier, most of the development and testing of the code was

performed on an IBM RS/6000 class workstation. Because of the rapid scalar processing capabilities of this machine, the largest problems could be completed in less than an hour. However, similar ANISN and TWOTRAN calculations would complete in only a few minutes. The time required for CENTAUR to set up internal data structures and relational coefficients for a given geometry was small, less than approximately 5% of the total execution time; the bulk of the time was spent in the iteration process.

Two factors contribute to CENTAUR's sluggish calculational cadence during the iteration process. First, the data structures used to store cell data were not optimally designed in terms of the computational algorithm when the code was developed. Thus, much indirect memory access and logical testing is performed during a calculation, reducing the program's computational efficiency. This problem could be addressed in a rewrite of key subroutines to take advantage of the now much better defined computational logic. However, the key factor which impedes the speed of a calculation is disk I/O, due to the large amount of data saved as relational coefficients. Since each set of coefficients is written to disk for each energy group in a multigroup problem, the same data must be read back in once every outer iteration. Even though the data is written in binary format, the sheer size of each file can take 10-30 seconds to read for each energy group in each iteration. Because CENTAUR currently uses no acceleration techniques, several outer iterations are necessary for problems with fission or upscattering. Thus, CENTAUR can become I/O bound. This overhead time could be reduced by the use of high speed hardware such as solid state disks; however, a more practical solution would be to reduce the amount of storage required, to reduce or eliminate disk access time.

Because of the nature of CENTAUR input, based on a line segment by line segment description of the problem geometry, it is generally very cumbersome to assemble any but the simplest of problems "by hand." Thus, for most of the problems analyzed, and for all of the cases discussed in the previous chapter, it was necessary to write a specialized

program to create an input deck for each new geometry type. Such codes were written with as much generality as possible, so that it was possible to change radial and azimuthal grids and material properties with little effort. However, in the development of such codes, it was difficult to determine if the candidate CENTAUR decks correctly represented the desired geometry. Thus, as a companion to the input processing codes, a graphical preprocessor named CENTART was written to read a CENTAUR input deck and provide a "plot" of the geometry represented by the input. The program, written in C and using the windowing and graphics capability of the Macintosh II class personal computer, is independent of the problem geometry, and can scale and plot any valid CENTAUR input file. The output can be transferred to other Macintosh applications via cut and paste operations. (Figures 20, 22, and 27 were created using CENTART. Only Figure 27 was modified, to include regional shading.) CENTART was found to be an invaluable debugging tool for the development of CENTAUR input.

Similar in nature to the complexity of the input is the means by which to understand the code output. In its current form, CENTAUR is capable of providing in its output various properties (average fluxes and reaction rates for primary reaction types) on both a cell by cell and a zone basis, where a zone is defined in the input as a specific set of cells. For most applications, this is sufficient. However, when trying to identify trends or anomalies in a solution, especially when debugging an input deck, it is often useful to look at various spatial profiles. Because of the arbitrary nature of the cells in a CENTAUR model, it is often difficult to present such distributions in a meaningful way. A recent addition to the CENTAUR package is a post-processing code named LEONARDO, which uses geometrical information and fluxes provided in an output file at job completion to provide a color contour plot of two-dimensional flux distributions for each energy group. The magnitude of the flux in each cell is represented by a color selected from a color range scaled to the range of the flux values. This approach is similar to methods used in many other analysis

packages, especially in finite element codes. While such a plotting capability does not provide numerical values, it does provide insight into the behavior of the neutron population for any given geometry and set of boundary conditions, and can be used to identify mistakes in input specifications. LEONARDO led to the discovery of the rotated cell ray effect described in the previous chapter.

Possibilities for Future Development

Considerable potential exists for future development and enhancement of both the CENTAUR code and the ESC method itself. Beyond basic improvements in the code and pre- and post-processing capabilities alluded to in the previous section, potential exists for growth in the basic theory itself. However, certain key problems must first be addressed prior to moving forward.

The most significant problems facing CENTAUR are memory utilization and speed. Many options exist for reducing the storage overhead; a primary consideration would be to check for repeated relational coefficients. Because of the similarity of the shape of many cells with respect to a given characteristic direction and the rotational symmetry of the quadrature sets currently used, it is likely that the same relational coefficient, within a given level of tolerance, is repeated many times in memory. Substantial memory savings could be obtained if repeated values were replaced with a single value and a set of integer pointers. Also, although CENTAUR does not save zero terms, it does save terms which are very small but non-zero. It may be possible to set these values to zero and save the memory space. However, this is unlikely to amount to a significant savings. The most benefit is likely to be seen by running the code on computers with large amounts of core memory. Traditionally, memory requirements of computer codes have pushed the limit of existing technology, providing the incentive for more and more memory. Current trends indicate that large amounts of core memory will be available on most systems in the very near future.

Some of the options for increasing the speed of CENTAUR have been discussed

previously. However, the most significant gain in performance may be realized by taking advantage of the potential for parallelization of the method. As was indicated earlier, the solution for each characteristic direction is independent of all other directions during the sweep process of each inner iteration; it is therefore possible to solve each direction in parallel. This could result in a time savings factor on the order of the number of directions in the quadrature set (144 directions are used for a S_{16} quadrature set). If a given parallel machine has available relatively large amounts of memory for each CPU, then angle dependent data can be saved with each direction on the CPU where it is needed; this would reduce the amount of storage required per CPU by the same quadrature-order dependent factor.

Once memory and speed limitations have been overcome, the next logical step would be to extend the method to a three-dimensional representation. The general theory remains essentially the same, except that side-to-side relationships in two dimensions become face-to-face representations in the three-dimensional case, as polygons are extended to polyhedrons. This will provide the capability to perform deterministic calculations for complex and completely arbitrary volumes. A need for such capabilities has already been identified in the medical field, where boron-neutron capture therapy is used for the treatment of brain tumors, and a reasonable estimate of dose distributions inside the exposed individual's head is required.[43]

An additional avenue for potential development is in elimination of the "step" approximation of constant source and boundary fluxes in each cell in favor of a linear approximation for these values. This would essentially be an extension of the Linear Characteristic method discussed in Chapter II. Since the LC approach has been shown to mitigate some of the problems inherent in the SC method, an derivation of an ESC-like approach based on the LC approximation would be likely to result in an improvement of the accuracy of the solution. However, as with the LC method, such an Extended Linear

Characteristic approach would require additional computational overhead in terms of storage requirements and the number of operations performed per cell per iteration.

Even in the current two dimensional ESC implementation, additional graphical preprocessing software is desirable. Custom programs used to generate input decks are useful when the geometry modelled is of a regular shape, even though it may be non-orthogonal. However, to be able to model completely arbitrary shapes, a tool which could enable a modeler to graphically position and define line segments or cells using a "click and drag" interface would be extremely useful. Such software exists in abundance, and would merely need to be tailored to the specific needs of CENTAUR. Another useful tool would be an "intelligent" preprocessor which could automatically fill a defined boundary with an optimum set of cells, and then reduce the information into the form accepted by CENTAUR. Moving into the domain of possible three-dimensional representations, one would need to construct versatile tools for both pre- and post-processing. Much of this work has been completed and has been successfully applied in three-dimensional finite element structural analyses. Again, it would be necessary to tailor existing algorithms and packages to the specific needs of CENTAUR.

The Extended Step Characteristic method has added a new dimension to discrete ordinates methods. With some additional development, CENTAUR could be put into use as a production code, modelling problems not easily undertaken using discrete ordinates methods. Such practical use will be necessary to more fully realize the advantages and limitations of the method, and would also help to define the user's needs in terms of working with the input and output of the code. It is expected that the ESC approach will provide significant contributions in both the nuclear reactor field for which it was originally conceived, and in non-reactor domains such as shielding and medical applications, where arbitrary shapes often need to be modelled.

REFERENCES

1. G. I. Bell and S. Glasstone, *Nuclear Reactor Theory*, Van Nostrand Reinhold Company, New York (1970).

2. J. J. Duderstadt and L. J. Hamilton, *Nuclear Reactor Analysis*, John Wiley & Sons, New York (1976).

3. R. Sanchez and N. J. McCormick, *Nucl. Sci. Eng.*, v 80, 481 (1982).

4. E. E. Lewis and W. F. Miller, Jr., *Computational Methods of Neutron Transport*, John Wiley & Sons, New York (1984).

5. A. Kavenoky, "Status of Integral Transport Theory," *Proc. Int Topl. Mtg. Advances in Mathematical Methods for the Solution of Nuclear Engineering Problems*, Munich, Germany, April 27-29, 1981, European Nuclear Society (1981).

6. S. Nakamura, *Computational Methods in Engineering and Science*, John Wiley & Sons, New York (1977).

7. W. H. Reed and T. R. Hill, "Triangular Mesh Methods for the Neutron Transport Equation," *Proc. Conf. Mathematical Models and Computational Techniques for the Analysis of Nuclear Systems*, Ann Arbor, Michigan, April 9-11, 1973, U. S. Atomic Energy Commission (1974).

8. W. F. Miller, E. E. Lewis, and E. C. Rossow, "The Application of Phase-Space Finite Elements to the One-Dimensional Neutron Transport Equation,"*Nucl. Sci. Eng.*, v 51, 145 (1973).

9. R. E. Alcouffe and E. W. Larsen, "A Review of Characteristic Methods Used to Solve the Linear Transport Equation," *Proc. Int Topl. Mtg. Advances in Mathematical Methods for the Solution of Nuclear Engineering Problems*, Munich, Germany, April 27-29, 1981, American Nuclear Society (1981).

10. F. B. Hildebrand, *Advanced Calculus for Applications*, Prentice-Hall, Inc., New Jersey (1976).

11. K. D. Lathrop, "Spatial Differencing of the Transport Equation: Positivity vs. Accuracy," *J. Comput. Phys.*, v 4, 475 (1969).

12. R. E. Alcouffe, E. W. Larsen, W. F. Miller, and B. R. Wienke, *Nucl. Sci. Eng.*, v 71, 111 (1979).

13. E. W. Larsen and R. E. Alcouffe, "The Linear Characteristic Method for Spatially Discretizing the Discrete Ordinates Equations in (x,y)-Geometry," *Proc. Int Topl. Mtg. Advances in Mathematical Methods for the Solution of Nuclear Engineering Problems*, Munich, Germany, April 27-29, 1981, American Nuclear Society (1981).

14. K. D. Lathrop, "Spatial Differencing of the Two-Dimensional Transport Equation," Report GA-8746, Gulf General Atomic (1968).

15. V. S. Vladimirov, "Mathematical Problems in the One-Velocity Theory of Particle Transport," Atomic Energy of Canada Ltd., Ontario, 1963; translated from *Trans V. A.*

Steklov Mathematical Institute, v 61 (1961).

16. W. F. Miller, E. E. Lewis, and E. C. Rossow, "The Application of Phase-Space Finite Elements to the Two-Dimensional Neutron Transport Equation in X-Y Geometry,"*Nucl. Sci. Eng.*, v 52, 12 (1973).

17. E. E. Lewis, W. F. Miller, and T. P. Henry, "A Two-Dimensional Finite Element Method for Integral Transport Calculations, *Nucl. Sci. Eng.*, v 58, 203 (1975).

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

19. W. H. Reed, T. R. Hill, F. W. Brinkley, and K. D. Lathrop, "TRIPLET: A Two-Dimensional, Multigroup, Triangular Mesh, Planar Geometry, Explicit Transport Code," Report LA-5428-MS, Los Alamos National Laboratory (1973).

20. T. J. Seed, W. F. Miller, and F. W. Brinkley, "TRIDENT: A Two-Dimensional, Multigroup, Triangular Mesh, Discrete Ordinates, Explicit Neutron Transport Code," Report LA-6735-M, Los Alamos National Laboratory (1977).

21. M. Mordant, "Some Efficient Lagrangian Mesh Finite Elements Encoded in ZEPHYR for Two-Dimensional Transport Calculations," *Ann. Nucl. Energy*, v8, 657 (1981).

22. G. Samba, "Discontinuous Finite Element Method for Solving the Two-Dimensional Neutron Transport Equation on Lagrangian Meshes," *Nucl. Sci. Eng.*, v 92, 197 (1986).

23. J. E. Morel and E. W. Larsen, "A Multiple Balance Approach for Differencing the S_n Equations," *Nucl. Sci. Eng.*, v 105, 1 (1990).

24. M. L. Adams, "A New Transport Discretization Scheme for Arbitrary Spatial Meshes in X-Y Geometry," *Proc. Int. Topl. Mtg. Advances in Mathemetics, Computations, and Reactor Physics*, Pittsburg, Pennsylvania, April 28-May 2 (1991).

25. R. E. Alcouffe, "A Diffusion-Accelerated S_n Transport Method for Radiation Transport on a General Quadrilateral Mesh," *Nucl. Sci. Eng.*, v 105, 191 (1990).

26. R. D. O'Dell and R. E. Alcouffe, "Transport Calculations for Nuclear Analysis: Theory and Guidelines for Effictive Use of Transport Codes," Report LA-10983-MS, Los Alamos National Laboratory (1987).

27. K. Takeuchi, "A Numerical Method for Solving the Neutron Transport Equation in Finite Cylindrical Geometry," *J. Nucl. Sci. Tech.*, v 6, 466 (1969).

28. K. Takeuchi, "Numerical Solution to Space-Angle Energy-Dependent Neutron Integral Transport Equation," *J. Nucl. Sci. Tech.*, v 8, 141 (1971).

29. K. Takeuchi and N. Sasamoto, "Direct Integration Method for Solving the Multigroup Neutron Transport Equation in Three-Dimensional Geometry," *Nucl. Sci. Eng.*, v 80, 536 (1982).

30. E. W. Larsen and W. F. Miller, "Convergence Rates of Spatial Difference Equations for the Discrete-Ordinates Neutron Transport Equations in Slab Geometry," *Nucl. Sci.*

Eng., v 73, 76 (1980).

31. D. V. Gopinath, A. Natarajan, and V. Sundararaman, *Nucl. Sci. Eng.*, v 75, 181 (1980).

32. M. R. Wagner, D. A. Sargis, and S. C. Cohen, "A Numerical Method for the Solution of Three-Dimensional Neutron-Transport Problems," *Nucl. Sci. Eng.*, v 41, 14 (1970).

33. J. R. Askew, "A Characteristics Formulation of the Neutron Transport Equation in Complicated Geometries," Report AEEW-M 1108, United Kingdom Atomic Energy Authority (1972).

34. M. J. Halsull, "CACTUS - a Characteristic Solution to the Neutron Transport Equation in Complicated Geometries," Report AEEW-R 1291, United Kingdom Atomic Energy Authority (1980).

35. W. L. Filippone, "Particle Transport Calculations with the Method of Streaming Rays," *Nucl. Sci. Eng.*, v 77, 119 (1981).

36. B. G. Carlson, "Transport Theory: Discrete Ordinates Quadrature Over the Unit Sphere," Report LA-4554, Los Alamos Scientific Laboratory (1970).

37. K. D. Lathrop and B. G. Carlson, "Discrete Ordinates Angular Quadrature of the Nuetron Transport Equation," Report LA-3186, Los Alamos Scientific Laboratory (1965).

38. C. E. Lee, "Discrete S_N Approximation to Transport Theory," Report LA-2595-MS, Los Alamos Scientific Laboratory (1962).

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

40. W. W. Engle, Jr., "A Users Manual for ANISN: A One Dimensional Discrete Ordinates Transport Code with Anisotropic Scattering," Report K-1693, Oak Ridge National Laboratory (1973).

41. K. M. Case, F. de Hoffman, and G. P. Placzek, "Introduction to the Theory of Neutron Diffusion," U.S. Government Printing Office, Washington, D. C. (1953)

42. I. K. Abu-Shumays, "Compatible Product Angular Quadrature for Neutron Transport in X-Y Geometry" *Nucl. Sci. Eng.*, v 64, 299 (1977).

43. D. W. Nigg, et al, "Radiation Physics Aspects of Boron Neutron Capture Therapy," *Proc. Topl. Mtg. New Horizons in Radiation Protection and Shielding*, Pasco, Washington, April 26 - May 1, American Nuclear Society (1992).

VITA

The author, Mark David DeHart, was born at James Connally Air Force Base, McLennan County, Texas, on July 17, 1960. His parents are Major (Ret.) Donald D. DeHart and Judy M. DeHart, who reside in North Richland Hills, Texas.

Mr. DeHart graduated from Edward W. Clark High School in Las Vegas, Nevada in June, 1978. After attending Iowa State University for a year, he left school to work for a year and a half. He transferred to Texas A&M University in January 1981. In May, 1984, he graduated Cum Laude with a Bachelor of Science degree in Nuclear Engineering. He began his graduate studies in June of the same year, and was appointed a Fellow within the Department of Energy's Nuclear Engineering Fellowship Program at Texas A&M in September, 1984. He earned a Master's degree in Nuclear Engineering in December 1986. He joined the staff of the University as a Lecturer in September, 1988, and remained at Texas A&M University until April 1989.

Mr. DeHart left Texas A&M University to accept a full time position as an Engineer with the Westinghouse Savannah River Company at the Department of Energy's Savannah River Site in Aiken, South Carolina. He is currently a Senior Engineer in the Nuclear Engineering Section where, with the support of DOE and WSRC, he completed his dissertation research.

Mr. DeHart, his wife, Leigh, and daughter Kaitlyn reside in South Carolina. His permanent mailing address is 1616 Georgia Avenue, North Augusta, South Carolina, 29841-3026.

DATE
FILMED
10/9/92