

11-1998

## Development of a Discrete Ordinates Code System for Unstructured Meshes of Tetrahedral Cells, with Serial and Parallel Implementations

Rodney L. Miller

Follow this and additional works at: <https://scholar.afit.edu/etd>



Part of the [Nuclear Engineering Commons](#)

---

### Recommended Citation

Miller, Rodney L., "Development of a Discrete Ordinates Code System for Unstructured Meshes of Tetrahedral Cells, with Serial and Parallel Implementations" (1998). *Theses and Dissertations*. 5124.  
<https://scholar.afit.edu/etd/5124>

This Dissertation is brought to you for free and open access by the Student Graduate Works at AFIT Scholar. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of AFIT Scholar. For more information, please contact [AFIT.ENWL.Repository@us.af.mil](mailto:AFIT.ENWL.Repository@us.af.mil).



Reproduced From  
Best Available Copy

DEPARTMENT OF THE AIR FORCE  
AIR UNIVERSITY  
**AIR FORCE INSTITUTE OF TECHNOLOGY**

Wright-Patterson Air Force Base, Ohio

19990108 043

**DEVELOPMENT OF A DISCRETE ORDINATES  
CODE SYSTEM FOR UNSTRUCTURED MESHES  
OF TETRAHEDRAL CELLS, WITH SERIAL  
AND PARALLEL IMPLEMENTATIONS**

**DISSERTATION**

**RODNEY L. MILLER, CAPTAIN, USAF**

**AFIT/DS/ENP/98-02**

Approved for Public Release; distribution unlimited

2025 RELEASE UNDER E.O. 14176

**DEVELOPMENT OF A DISCRETE ORDINATES  
CODE SYSTEM FOR UNSTRUCTURED MESHES  
OF TETRAHEDRAL CELLS, WITH SERIAL  
AND PARALLEL IMPLEMENTATIONS**

**DISSERTATION**

Presented to the Faculty of the School of Engineering  
of the Air Force Institute of Technology  
Air University in Partial Fulfillment of the  
Requirements for the Degree of  
Doctor of Philosophy

Rodney L. Miller, B.S., M.S.

Captain, USAF


November 1998

Approved for Public Release; distribution unlimited

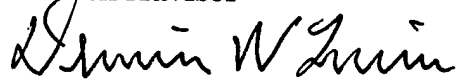
**DEVELOPMENT OF A DISCRETE ORDINATES  
CODE SYSTEM FOR UNSTRUCTURED MESHES  
OF TETRAHEDRAL CELLS, WITH SERIAL  
AND PARALLEL IMPLEMENTATIONS**

Rodney L. Miller, B.S., M.S.  
Captain, USAF

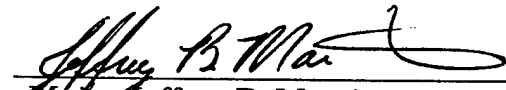
Approved:

  
Kirk A. Mathews  
Associate Professor of Nuclear Engineering  
Research Advisor


17 Nov 1998  
Date

  
Dennis W. Quinn  
Professor of Mathematics

18 NOV 1998  
Date

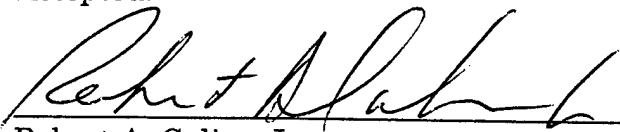
  
Major Jeffery B. Martin  
Assistant Professor of Nuclear Engineering

18 Nov 1998  
Date

  
Meir Pachter  
Professor of Electrical Engineering  
Dean's Representative

18 Nov 1998  
Date

Accepted:

  
Robert A. Calico, Jr.  
Dean, Graduate School of Engineering

## **Dedication**

This dissertation is dedicated in loving memory to my maternal grandfather, Harold V. Wallace, and paternal grandmother, Mary Phyllis Miller, both of whom passed beyond this world during the course of this effort.

## Acknowledgments

Many people and organizations contributed to the success of this research. First and foremost, I thank my wife, Cindy, and my daughters, Christy and Jessica, for their steadfast support and understanding during these three years. Second, I thank Dr. Kirk Mathews for his leadership, support, and encouragement whilst I did battle with the gods of computation. Without him, this research would not be. Additionally, I thank my committee, Maj. Jeff Martin and Dr. Dennis Quinn, for their advice, support, and encouragement. Thanks also to the Air Force Nuclear Weapons Integration Group for sponsoring this research. Thanks to Mr. Rob Kehlet of the Defense Special Weapons Agency for his support of our computational laboratory as well as the Aeronautical Systems Center Major Shared Resource Center for providing us with the computational hardware and software used in this effort. To The Portland Group, especially Larry Meadows, thank you for helping us get TETRAN to run in parallel using HPF. To Dr. Susann Mathews, thanks for the much needed support and encouragement. Lastly, I thank my comrades in arms preceding me in this research: Maj. Dennis Miller, Bryan Minor, Maj. Glenn Sjoden, and Capt. Charlie Brennan. We could not have ventured further without your efforts. Finally, thanks to Maj. Mark DelGrande for his morale support in helping me finish this effort.

## Table of Contents

Dedication .....	iii
Acknowledgments .....	iv
Table of Contents .....	v
List of Figures .....	vii
List of Tables .....	ix
Abstract .....	x
Chapter I: Introduction .....	1
Background .....	1
Monte Carlo Techniques .....	3
Deterministic Methods .....	4
Non-linear Discrete Ordinates Methods .....	9
Other Unstructured Grid Transport Methods .....	12
Purpose .....	12
Goals .....	13
Scope .....	14
Limitations .....	14
Organization .....	15
Chapter II: Derivation of Exponential Characteristic Spatial Quadrature ...	17
The Unit Tetrahedron .....	17
Tetrahedron Splitting .....	20
Characteristic Solution in the Sub-cell Unit Tetrahedron .....	22
Moments Functions .....	22
Derivation of the Exponential Characteristic Method .....	24
EC Source Distribution .....	24
EC Inflow Face Flux Distribution .....	27
Derivation of Sub-Cell EC Spatial Quadrature .....	29
Sub-Cell Conservation Equations .....	36
Linear Characteristic Spatial Quadrature .....	37
Chapter III: TETRAN Implementation .....	44
Parallel Programming Model .....	44
Multigroup Implementation .....	47
Anisotropic Scattering .....	48
TETRAN Data Flow .....	52
Pro/Engineer .....	52
Cross Section Data Format .....	53
Input Module .....	54
Overview of TETRAN .....	55
TETRAN Pseudo-Code .....	57
Chapter IV: Testing .....	62
Convergence and Convergence Rate .....	64
Run-time Profiling .....	69



Cell Aspect Ratio Robustness .....	73
Curvilinear and Inclined-Plane Surfaces .....	79
Coarse Mesh with Optically Thick Cells .....	84
Multi-group, Anisotropic Scattering Performance .....	91
HPF Parallel Demonstration .....	107
Chapter V: Conclusions and Recommendations .....	113
Conclusions .....	113
Recommendations for Future Work .....	115
Appendix A: Exponential Moment Functions — Evaluation and Identities .....	117
Definitions .....	117
Exponential Moment Function Identities .....	118
Identity 1 .....	118
Identity 2 .....	119
Identity 3 .....	119
Identity 4 .....	119
Identity 5 .....	120
Identity 6 .....	120
$M_0(x,y,z,w)$ and $\rho(x,y,z,w)$ Numerical Evaluation .....	120
Appendix B: Evaluation of Source and Inflow Flux Coefficients .....	123
Source Coefficients .....	123
Source System Root Solver .....	126
Source First Guess Algorithm .....	129
Inflow Flux Coefficients .....	136
Inflow Flux System Root Solver .....	137
Inflow Flux First Guess Algorithm .....	138
Appendix C: Stable EC Quadrature Formulations .....	146
IEEE® 754 Overflow and Underflow Errors .....	146
Source Contribution to Cell Flux Moments .....	147
Inflow Flux Contribution to Cell Flux Moments .....	150
Source Contribution to Outflow Face Flux Moments .....	152
Inflow Flux Contribution to the Outflow Face Flux Moments .....	154
Source System and Jacobian Elements .....	156
Inflow Face Flux System and Jacobian Elements .....	159
Appendix D: Derivation of the Surface Cell Algorithm .....	161
Motivation .....	161
Tetrahedron Aspect Ratio .....	162
The Surface Cell Algorithm .....	165
Derivation of Surface Cell Spatial Quadratures .....	167
Linear Characteristic Surface Cell Approximation .....	168
Exponential Characteristic Surface Cell Approximation .....	173
Implementation of the Surface Cell Quadrature .....	177
Bibliography .....	178
Vita .....	185

## List of Figures

Figure 1. Correlation of Production Code Transport Algorithms to Hardware Floating-Point Performance. ....	8
Figure 2. Development timeline of non-linear methods research. ....	10
Figure 3. Tetrahedron Arbitrarily Oriented in XYZ Space.....	17
Figure 4. The Unit Tetrahedron in the UVW Coordinate System. ....	18
Figure 5. The Face Coordinate System.....	19
Figure 6. The Face Local UV Coordinate System. ....	19
Figure 7. Example of a Tetrahedron Cell Split. ....	20
Figure 8. Block diagram for iteration on the scattering source algorithm. ...	48
Figure 9. Data flow for TETRAN. ....	52
Figure 10. Convergence Graph for Exponential Characteristic Method.....	67
Figure 11. Convergence Graph for the Linear Characteristic Method. ....	68
Figure 12. Cell aspect ratio parameters. ....	74
Figure 13. Meshes for Robustness Test. ....	74
Figure 14. Distribution of Cell Aspect Ratios for Fine Mesh.....	76
Figure 15. Coarsest Mesh of Sphere in Tetrahedron (138 cells).....	81
Figure 16. Finest Mesh Sphere in Tetrahedron Mesh (10075 cells). ....	82
Figure 17. Geometry for Coarse Mesh Problem. ....	86
Figure 18. Relative Errors in Source Region Scalar Flux with Respect to EC Fine Mesh Benchmark for Deep Penetration Problem. ....	88
Figure 19. Relative Errors in Shield Region Scalar Flux with Respect to EC Fine Mesh Benchmark for Deep Penetration Problem. ....	88
Figure 20. Relative Error with Respect to Fine Mesh EC Benchmark for $J_{+x}$ using EC and LC.....	90
Figure 21. Geometry for the Water-Iron-Water Problem. ....	92
Figure 22. Group 1 Scalar Flux in Source Region.....	97
Figure 23. Group 2 Scalar Flux in Source Region.....	98
Figure 24. Group 3 Scalar Flux in Source Region.....	98
Figure 25. Group 1 Scalar Flux in Iron Region. ....	99
Figure 26. Group 2 Scalar Flux in Iron Region. ....	100
Figure 27. Group 3 Scalar Flux in Iron Region. ....	100
Figure 28. Group 1 Scalar Flux in Outer Water Region. ....	102
Figure 29. Group 2 Scalar Flux in Outer Water Region. ....	102
Figure 30. Group 3 Scalar Flux in Outer Water Region. ....	103
Figure 31. Group 1 Current Out of +X Face (X=50.0 cm). ....	104
Figure 32. Group 2 Current Out of +X Face (X=50.0 cm). ....	105
Figure 33. Group 3 Current Out of +X Face (X=50.0 cm). ....	105
Figure 34. Demonstration of Speed Up for HPF Parallel TETRAN.....	110
Figure 35. The $(\rho_{UV}, \rho_{VW}, \rho_W)$ Solution Space. ....	125
Figure 36. Mapping of $(\rho_{UV}, \rho_{VW}, \rho_W)$ Phase Space to $(X_s, Y_s, Z_s)$ Coefficient Space. ....	125

Figure 37. Plot of $X_s(\rho_{UV}, \rho_{VW}, \rho_W)$ for $X_s = -100, -10, 0, 10, 100$ (right to left). .....	131
Figure 38. The $(\rho_{UV}^f, \rho_V^f)$ Phase Space.....	139
Figure 39. Plot of $X_f(\rho_{UV}^f, \rho_V^f)$ for $X_f = -100, -10, 0, 10, 100$ (right to left)..	140
Figure 40. Definition of Tetrahedron Cell Aspect Ratio. ....	162
Figure 41. Examples of Unstructured Tetrahedra Meshes with Thin Regions. .....	163
Figure 42. Distribution of Cell Aspect Ratios for Fine Meshed Example Problem. ....	164
Figure 43. Cell and Side View of a Surface Cell.....	165
Figure 44. Surface Cell Approximation. ....	166

## List of Tables

Table 1. Possible Parent Tetrahedron Cell Cases. ....	21
Table 2. Summary of Test Problems .....	64
Table 3. Mesh Parameters for Convergence Rate Problem. ....	65
Table 4. Results for Problem 1. ....	66
Table 5. TETTRAN Run-Time Characteristics.....	70
Table 6. TETTRAN Spatial Quadrature Performance. ....	71
Table 7. Cell Optical Thicknesses for Cell Aspect Ratio Problem. ....	77
Table 8. Results for Large Cell Aspect Ratio Problem. ....	78
Table 9. Mesh Optical Thicknesses for Curvilinear and Inclined-Planes Problem. ....	81
Table 10. Results of Transport Calculation Using EC for Test Problem 2. ...	83
Table 11. Mesh Optical Thicknesses for Coarse Mesh Problem .....	86
Table 12. Region Scalar Fluxes for Coarse Mesh, Deep Penetration Problem .....	87
Table 13. Exiting Face Currents for Deep Penetration Problem.....	89
Table 14. $P_1$ Transport Corrected Cross Sections for Water.....	95
Table 15. $P_3$ Transport Corrected Cross Sections for Water.....	95
Table 16. $P_3$ Transport Corrected Cross Sections for Natural Iron.....	96
Table 17. Group and Mesh Dependent Cell Optical Properties for Problem 6. .....	97
Table 18. HPF Parallel TETTRAN Results. ....	109
Table 19. Asymptotic Solutions to the Source System.....	130
Table 20. Asymptotic Solutions for Inflow Face System.....	141

### Abstract

A numerically stable, accurate, and robust form of the exponential characteristic (EC) method, used to solve the time-independent linearized Boltzmann Transport Equation, is derived using direct affine coordinate transformations on unstructured meshes of tetrahedra. This quadrature, as well as the linear characteristic (LC) spatial quadrature, is implemented in our transport code, called TETRAN. This code solves multi-group neutral particle transport problems with anisotropic scattering and was parallelized using High Performance Fortran and angular domain decomposition. A new, parallel algorithm for updating the scattering source is introduced. The EC source and inflow flux coefficients are efficiently evaluated using Broyden's rootsolver, started with special approximations developed here. TETRAN showed robustness, stability and accuracy on a variety of challenging test problems. Parallel speed-up was observed as the number of processors was increased using an IBM SP computer system.

# DEVELOPMENT OF A DISCRETE ORDINATES CODE SYSTEM FOR UNSTRUCTURED MESHES OF TETRAHEDRAL CELLS, WITH SERIAL AND PARALLEL IMPLEMENTATIONS

## Chapter I: Introduction

We present the development of a parallel unstructured tetrahedra mesh discrete ordinates radiation transport code. This code, TETRAN, solves the linear time-independent Boltzmann transport equation for the angular flux density using the linear and exponential characteristic spatial quadratures. It also uses the standard source iteration algorithm with multi-group cross sections and general anisotropic scattering. The code was made parallel using High Performance Fortran directives. We examine the performance of the code using several benchmark problems and present the results. The exponential characteristic method was found to be a stable, robust coarse mesh method when provided with positive source moments. Several challenging test problems demonstrate EC's performance. Performance of the parallel code is briefly presented.

## Background

The fundamental equation describing neutral particle transport processes is the linear Boltzmann transport equation (BTE) (Lewis and Miller, 1993:1). The time-independent BTE describes the balance of neutral particles in a six-dimensional phase space: three spatial dimensions ( $\vec{r}$ ), two

angular dimensions ( $\hat{\Omega}$ ), and energy ( $E$ ). The BTE relates the rate of change of neutral particle density at a point  $\vec{r}$ , moving with energy  $E$ , in direction  $\hat{\Omega}$ , to the differences between sources and losses:

$$\left[ \hat{\Omega} \cdot \vec{\nabla} + \sigma(\vec{r}, E) \right] \psi(\vec{r}, \hat{\Omega}, E) = S(\vec{r}, \hat{\Omega}, E). \quad (1)$$

where the angular flux density,  $\psi$ , is the flux of particles at point  $\vec{r}$ , moving with energy  $E$ , in direction  $\hat{\Omega}$ . The bracketed term accounts for particle losses due to streaming and collisions.  $S(\vec{r}, \hat{\Omega}, E)$  represents the source of all particles in the  $(\vec{r}, \hat{\Omega}, E)$  phase space cell. The characteristic solution to the BTE is presented in Chapter 2. In Chapter 3, we discuss the multi-group energy discretization of the BTE and the impact of the traditional anisotropic scatter treatment on the parallelization of the source iteration algorithm.

Analytical solutions for the BTE exist for idealized problems (one energy group with isotropic scatter and symmetry boundaries). Recently, researchers at Los Alamos developed several semi-analytic benchmarks that solve multi-dimensional transport problems (Kornreich, 1997). Point kernel techniques are used in some radiation shielding calculations where speed, rather than accuracy, is required to determine dose rates from simple source configurations (Glasstone, 1981). Build-up factors, calculated via the numerical techniques discussed below, often augment point kernel calculations.

Numerical methods are generally used to solve the BTE. The two most popular classes of methods are Monte Carlo and deterministic. Where diffusion processes dominate the transport of particles, simplifying assumptions can be made to the BTE, transforming it into the diffusion equation which is solved by well established algorithms (such as finite difference or conjugate gradient) (Ott, 1983).

### **Monte Carlo Techniques**

The Monte Carlo technique simulates individual particle path histories and accumulates the results. As these histories are accumulated, statistics are formed that estimate the uncertainties in the results. The method is generally suitable to complex three-dimensional geometries. Computational grids are not explicitly needed. Besides its general geometry capability, the Monte Carlo technique can also take advantage of continuous energy nuclear data which is the natural form of cross section data found in the Evaluated Nuclear Data Files (ENDF) (Rose, 1990). It can also use multigroup cross section data that are typically used for deterministic transport algorithms (Briesmeister, 1991). The primary disadvantage of the technique is its need for variance reduction techniques for complicated, deep-penetration problems. Another disadvantage is that the method can not represent the flux in an entire system except at great expense since numerous flux estimators are required. Deterministic methods directly create a flux map. An excellent discussion of the Monte Carlo method is found in Lewis and



Miller, Chapter 7 (Lewis and Miller, 1993), or the MCNP™ 4B (Monte Carlo N-Particle Transport Code System) manual, Chapter 1 (Briesmeister, 1991). MCNP™ is widely used due to its general three-dimensional geometry ability, accuracy, and ease of use; we use it here to generate benchmarks for comparison with TETRAN results.

### **Deterministic Methods**

Deterministic methods solve the BTE using simplifying assumptions and techniques to perform the angular and spatial integrations. The most widely used method to solve the BTE is discrete ordinates (Lewis and Miller, 1993: 116). First utilized for transport calculations in stellar atmospheres (Chandrasekhar, 1960), researchers at Los Alamos National Laboratory adapted this method for neutron transport work in the late 1950's (Carlson: 1958).

In the discrete ordinates method, the BTE is solved for a finite number of angular directions,  $\hat{\Omega}_n$ , to get the angular flux,  $\psi_n$ , for each cell in a computational mesh. After all of the  $\psi_n$  have been computed for a given mesh, numerical quadrature is used to calculate the meaningful integral quantity: the scalar flux,  $\phi$ , in each cell  $i$ ,

$$\phi_i = \sum_{n=1}^N w_n \psi_{i,n}. \quad (2)$$

Usually, the angles,  $\hat{\Omega}_n$ , and weights,  $w_n$ , are chosen to accurately integrate spherical harmonics functions, as is described in detail in Lewis and Miller, Chapters 3 and 4 (Lewis and Miller, 1993).

Many algorithms exist within the discrete ordinates family of methods. One of the oldest, yet still used, is the diamond difference method (Lewis and Miller, 128-131). The diamond difference method makes a finite difference approximation to the differential operator in the BTE and assumes several auxiliary equations. The method is easy to implement and is computationally inexpensive. However, diamond difference suffers from numerical instabilities that produce oscillations in the angular flux (Petrovic, 1996) and generally does not propagate flux in the correct direction (Mathews, 1998). In slab geometry, diamond difference can also produce negative angular fluxes unless the computational cell optical thickness,  $\epsilon$ , is small ( $< 0.1$ ). In two- and three-dimensions, diamond difference is not even conditionally positive unless fix-ups are used. This behavior limits the application of diamond difference to deep-penetration shielding problems since exceedingly fine meshes are required to maintain the positivity and accuracy of the method. Indeed, the diamond difference method or one of its cousins [adaptive weighted diamond difference (AWDD) (Alcouffe, 1993) and theta-weighted diamond difference (Rhoades and Azmy, 1996)] has been the backbone of every major discrete ordinates neutral particle transport code developed in the United States since the inception of the method around 1960.

Figure 1 shows the progression of production codes, the methods they employ, and a backdrop of computational hardware generally available at the time of the method [(Intel Corp., 1998) (Foster, 1995), (Dongarra, 1998), (USDOE ASCI, 1998), (Wareing, 1997), (Lewis and Miller, 1993), (Rhoades, 1995), (Alcouffe, 1995), (Briesmeister, 1991)]. The figure shows that few new (higher order) spatial quadratures have been incorporated into the production codes in the last 30 years. The reason for this is that higher order methods are generally more computationally intensive per spatial cell than the traditional or weighted difference approaches. This is particularly true of the characteristic methods presented in this dissertation. Much of the apprehension in embracing higher order methods stems from Lathrop's seminal paper, "Spatial Differencing of the Transport Equation: Positivity vs. Accuracy" (Lathrop, 1969). In this paper, Lathrop presents what have become the de facto criteria for a "desirable" spatial quadrature:

1. "it should be accurate in the sense that it has small truncation error,
2. it should be simple, which means that it should involve a small number of numerical operations and should involve unknowns within a single mesh cell,
3. it should produce positive fluxes if the source and boundary fluxes are positive,
4. the scheme should be conservative in the sense that a well defined relation exists among flows into and out of the cell and sources and absorptions in the cell, and,
5. the scheme is easily generalized to all (curvilinear) geometries."

Obviously, the above criteria are reasonable, particularly from the vantage point of 1969 when computational hardware constraints were severe. However, the 2<sup>nd</sup> and 5<sup>th</sup> criteria have prevented the acceptance of many higher order methods because they tend to be computationally expensive per cell, not very simple, and are not generalizable to curvilinear geometries. Criterion 5 proves to be a particular problem for characteristic methods since such methods assume particle flow along straight lines in Cartesian space. However, we argue that criteria 2 and 5 are unreasonable given the computational and mesh generating capabilities that exist today. Indeed, for true three-dimensional problems not involving the types of symmetries needed to exploit curvilinear coordinates, an unstructured Cartesian mesh is probably a better choice for performing transport analysis.

### Correlation of Production Code Transport Algorithms to Hardware Floating-Point Performance

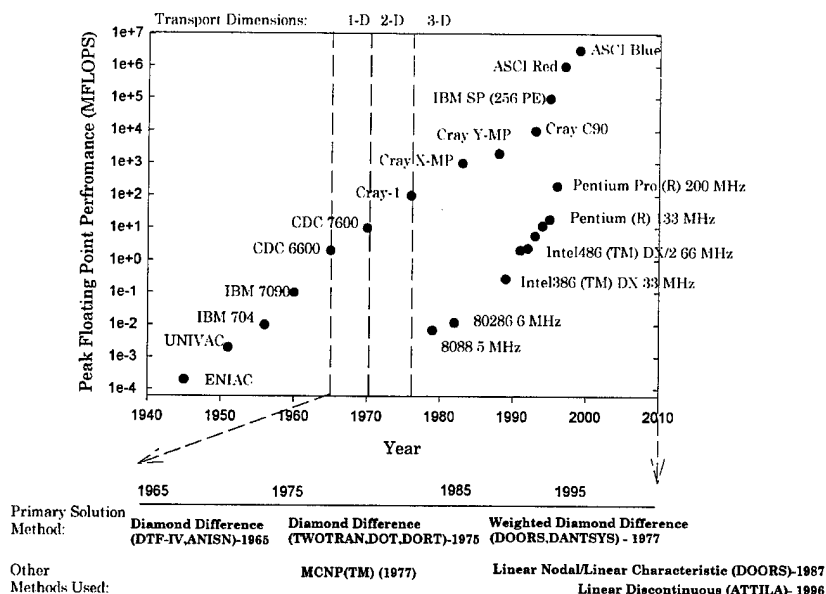


Figure 1. Correlation of Production Code Transport Algorithms to Hardware Floating-Point Performance.

Figure 1 shows the fantastic growth in floating-point performance from the days of the first computer to near term plans for the next generation of supercomputers envisaged by the Department of Energy's Advanced Strategic Computing Initiative (ASCI). Clearly, system performance has grown tremendously since the birth of computational radiation transport circa 1960 (both in processor performance and memory). Meanwhile, transport researchers have derived more accurate transport methods than the traditional or weighted diamond difference approaches which are still the

primary methods of choice for the large discrete ordinates production codes: DANTSYS (Alcouffe, 1995) and DOORS (RSICC, 1998). Additionally, most of the modern discrete ordinates transport codes use Cartesian computational meshes with boxoid (in the case of three-dimensional transport) or rectangular (in the case of two-dimensional transport) computational cells. This approach simplifies meshing a problem where curved surfaces are not involved or one does not want to resolve fine details of the problem. However, the number of cells needed to mesh a problem accurately grows quickly with the detail desired to be refined. It is for this reason that researchers continue to pursue the development of higher order, linear and non-linear discrete ordinates methods for use on unstructured grids.

### **Non-linear Discrete Ordinates Methods**

Over the past ten years, researchers at the Air Force Institute of Technology (AFIT), Texas A&M University, and Los Alamos National Laboratory (LANL) have developed several non-linear characteristic methods to solve the BTE. In these non-linear methods, the needed source or flux coefficients are obtained by root-solving a non-linear system of equations. These methods were pursued due to their inherent positivity given positive source and boundary data and their potential for robust coarse mesh performance. The timeline showing the development of non-linear methods research is shown in Figure 2.

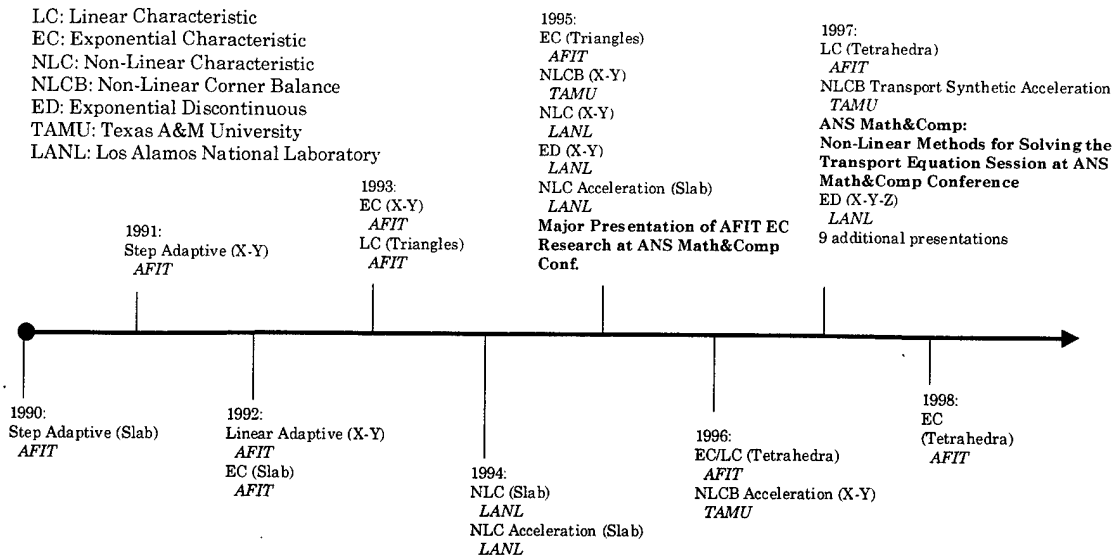


Figure 2. Development timeline of non-linear methods research.

First among this family of methods were the step adaptive (Mathews, 1990) and linear adaptive (Mathews, 1993) methods. Following the development of the adaptive methods was the exponential characteristic (EC) method for slab geometry, developed at both AFIT (termed exponential characteristic or EC)(Mathews, 1994) and LANL (termed non-linear characteristic or NLC)(Walters, 1996). This method takes the average and first spatial moments of the characteristic solution to the BTE and an assumed exponential distribution of the scattering source in a cell to calculate average and first moments of the angular flux. Both implementations exhibited fourth order convergence and excellent deep penetration performance warranting further investigation of the method in higher dimensions. The AFIT implementation differed substantially from

LANL's approach in that the focus of the development was on producing a numerically robust method. The LANL approach had obvious numerical conditioning problems. Both groups continued the development of EC/NLC by implementing the method in two-dimensions. AFIT developed both Cartesian rectangular cell (Minor, 1995) and unstructured triangular mesh (Mathews, 1997) EC algorithms. LANL researchers produced a Cartesian rectangular cell algorithm (Walters, 1995) and convergence acceleration techniques for their methods in slab and XY geometry (Wareing, 1994) (Wareing, 1995). AFIT continued our development by extending the method to three-dimensions, implementing EC on grids of unstructured tetrahedra (Brennan, 1996). The LANL researchers chose to pursue different methods (due to the high computational cost per phase space cell of the NLC method), beginning the development of the exponential discontinuous (ED) method which they have implemented in slab, two- (Wareing, 1995) and three-dimensional Cartesian cells (Wareing, 1996). At Texas A&M, researchers developed the non-linear corner balance (NLCB) method which they have derived for slab and two-dimensional arbitrary grids (Castrianni, 1998). Additionally, this group also developed convergence acceleration techniques (transport synthetic acceleration) for use with their method (Ramone, 1997) (Anistratov, 1996).



## **Other Unstructured Grid Transport Methods**

Besides the above non-linear methods for two- and three-dimensional transport on unstructured grids, researchers developed several linear methods. LANL pioneered the development of the linear characteristic method for unstructured triangles by developing a linear characteristic-nodal method for such grids (Paternoster, 1984). A version of the step characteristic method was extended to two-dimensional arbitrary grids by researchers at Oak Ridge National Laboratory and the Westinghouse Savannah River Laboratory (DeHart, 1994). LANL developed — in concert with several oil and natural gas companies — a three-dimensional unstructured tetrahedral mesh discrete ordinates transport code. This code, ATTILA, performs particle transport using the linear discontinuous method with diffusion synthetic acceleration of the source iteration (Wareing, 1996) and is third order convergent (Wareing, 1998). ). At AFIT, the step and linear characteristic methods were developed for unstructured triangular (Miller, 1993) and Cartesian rectangular (Minor, 1993) cell meshes. Brennan included a development of both the step characteristic and linear characteristic methods for tetrahedral cells (Brennan, 1996).

## **Purpose**

The purpose of this research is to contribute to the development of a high order, high fidelity (with respect to geometry), three-dimensional discrete ordinates transport code system. Currently, only one such code exists

— LANL's ATTILA (although it is not a general distribution code). ORNL's TORT code uses the linear characteristic spatial quadrature (Rhoades, 1995) for deep penetration problems. However, it performs transport on Cartesian grids and is thus poor at modeling geometry in detail. MCNP<sup>TM</sup>, a Monte Carlo code, is widely used for high fidelity, three-dimensional transport calculations. We seek to develop a discrete ordinates code system that performs neutral particle transport on arbitrarily complex three-dimensional geometries with accuracy approaching or exceeding that of MCNP<sup>TM</sup> or other Monte Carlo methods. We envision that such a code system will fill the gap between Monte Carlo and discrete ordinates transport codes, producing accurate high fidelity global flux distributions for coarse meshes and deep penetration problems.

## Goals

The goal of this research is to implement the linear and exponential characteristic methods in a parallel radiation transport code. Specifically, the exponential characteristic method (tetrahedron computational cell) will be derived using the direct affine transformation approach (Mathews, 1998) for tetrahedral cell splitting and moment passing. This development improves on that of prior work (Brennan, 1996). Additionally, we will implement the EC spatial quadrature within a code that is parallel capable with the introduction of High Performance Fortran (HPF) directives.

We begin this development with the long-term vision of fielding a production code system. With that end in mind, TETRAN will be able to perform multi-group, anisotropic scattering problems in a high-performance parallel-computing environment. TETRAN will operate on arbitrarily complicated meshes (limited by the mesh generator and available hardware) and read standard cross section libraries. Additionally, it will produce data in a format that is easily read by state of the art scientific visualization software such as AVS/Express.

### **Scope**

TETRAN was created to be the kernel of a general three-dimensional neutral particle transport code. It solves the time-independent multi-group BTE using the EC and LC spatial quadratures on meshes of tetrahedra. Anisotropic scattering is treated using the traditional spherical harmonics approach. The code operates in parallel on an IBM SP when compiled with the PGHPF 2.4 development compiler.

### **Limitations**

As a first generation code, TETRAN has several limitations due its maturity level. These limitations are:

- 1) No Convergence Acceleration. Adapting or developing convergence acceleration may prove to be straightforward, but issues of efficiency, effectiveness, robustness, and implementation make this suitable as the focus of a follow-on effort.
- 2) No fission source.

- 3) Downscatter only. To be useful, both 2) and 3) would need convergence acceleration. Except for a few unusual applications, adding these features will be straightforward.
- 4) High Performance Fortran (HPF). TETRAN has HPF directives and has been minimally tested using them but current compilers are still under development.
- 5) Only vacuum boundaries and no re-entrant geometries. Other boundaries would unduly complicate parallelism for a first code.
- 6) No external incident beams or fluxes. Same as 5).

Several problems were devised to test the operation of the code and its underlying algorithms. Each problem uses an isotropic emitter uniformly distributed through the volume of one region as the only source of particles in the problem. No incident current problems were tested since the author did not know how to use MCNP™ for this type of problem. Additionally, no investigation of TETRAN performance on thick, diffusive problems was performed, because such problems require convergence acceleration, which is a feature we have yet to develop or adapt to these meshes.

## Organization

The remainder of this document is composed of five chapters. Chapter 2 presents the derivation of the exponential characteristic spatial quadrature using direct affine transformations. Chapter 3 outlines the needed multi-group and anisotropic scattering theory and presents the pseudo-code for TETRAN as well as the parallelization strategy using HPF. We present the results of our testing in Chapter 4. Finally, Chapter 5 presents our

conclusions and recommendations for future work. Four appendices are included. In Appendix A, we present several exponential moment function identities used in this work. Appendix B contains the algorithms used to find the source and inflow face flux coefficients needed for the EC method. Appendix C summarizes the robust algorithms used to calculate the EC spatial quadrature. Finally, Appendix D presents the derivation of a new spatial quadrature, the surface cell algorithm, that was derived but not implemented in this effort because no mesh generators currently support its inclusion in finite element unstructured meshes.

## Chapter II: Derivation of Exponential Characteristic Spatial Quadrature

In this chapter, we derive the exponential characteristic (EC) solution to the Boltzmann Transport Equation (BTE) for a tetrahedron spatial cell. Specifically, the equations for the average and first spatial moments of the angular flux in a tetrahedron cell are derived as well as the needed source and inflow flux systems of equations. The chapter ends with a brief summary of the linear characteristic method, which was implemented prior to the EC quadrature in TETRAN.

### The Unit Tetrahedron

The derivation of the exponential characteristic method begins with a discussion of the unit tetrahedron coordinate system. This discussion will be brief since the face and tetrahedra cell coordinate systems, affine transformations, and cell splitting are presented in detail in a previously submitted paper (Mathews, 1998). Consider the tetrahedron in Figure 3.

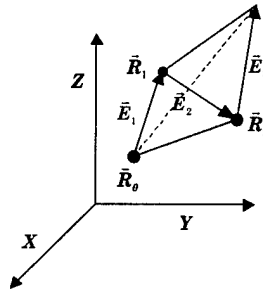


Figure 3. Tetrahedron Arbitrarily Oriented in XYZ Space.

As was pointed out by Brennan (Brennan, 1996), it would be an intractable problem to develop a characteristic method using this global cell coordinate

system because such a method would need to account for any arbitrary streaming direction and cell orientation. With reference to Figure 3, we define an oblique Cartesian (affine) coordinate system with origin at  $\bar{R}_0$  and basis vectors  $\bar{E}_1, \bar{E}_2$ , and  $\bar{E}_3$  for coordinates  $U, V$ , and  $W$ , respectively. In that coordinate system, the tetrahedral cell is (by construction of the coordinate system) always the *unit tetrahedron* shown in Figure 4.

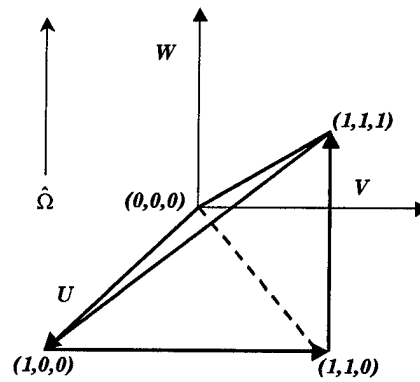


Figure 4. The Unit Tetrahedron in the UVW Coordinate System.

Each cell in the global mesh is mapped to its own unit tetrahedron system. All cell flux moments, source coefficients, and sub-cells are defined based on this coordinate system. After the cell has been split (along the streaming direction  $\hat{\Omega}$ ), the inflow flux is passed from the upstream cell to the downstream cell and is properly transformed into the sub-cell face systems. This involves some moderate complexity and is fully discussed in the referenced paper. In addition to the cell coordinate system, each cell has four faces. Each face has its own coordinate system defined by two edge vectors

whose cross product produces an outward normal vector. This face system is shown in Figure 5.

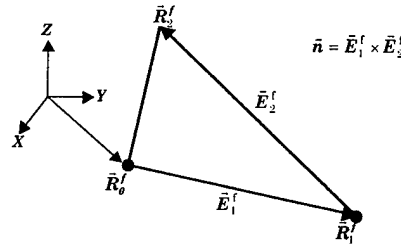


Figure 5. The Face Coordinate System.

With reference to Figure 5, we define an oblique Cartesian (affine) coordinate system with origin at  $\bar{R}_0^f$  and basis vectors  $\bar{E}_1^f$  and  $\bar{E}_2^f$  for coordinates  $U^f$  and  $V^f$ , respectively. In this coordinate system, the tetrahedron face is always the *unit tetrahedron face* shown Figure 6.

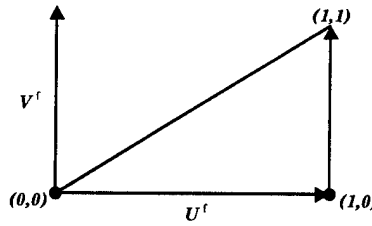


Figure 6. The Face Local UV Coordinate System.

Having presented the needed coordinate systems to pass cell and face flux moments within the transport calculation, we will briefly overview the concept of cases for cell splitting.



## Tetrahedron Splitting

The spatial quadratures derived in this work require that each tetrahedron cell be split along the streaming direction into a number of sub-tetrahedra. It is for these sub-tetrahedra that the quadratures are derived. An example cell split is shown in Figure 7.

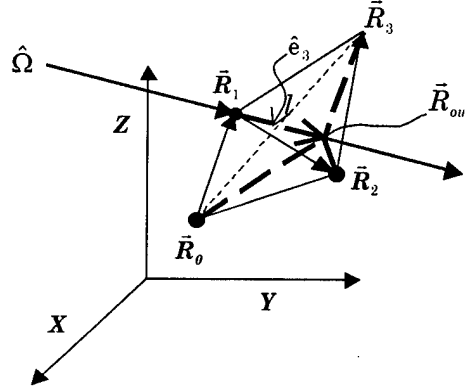


Figure 7. Example of a Tetrahedron Cell Split.

Referring to Figure 7, the splitting line, parallel to  $\hat{\Omega}$ , enters the cell at  $\bar{R}_{\text{in}} = \bar{R}_1$  and is propagated across the cell to  $\bar{R}_{\text{out}}$ , where  $\bar{R}_{\text{out}}$  is the point in the cell that will produce a number of sub-tetrahedra that have exactly one input and one output face. We term these case 1 tetrahedra. In the case above, the cell is split into three sub-tetrahedra with corner nodes  $(\bar{R}_0, \bar{R}_3, \bar{R}_1, \bar{R}_{\text{out}})$ ,  $(\bar{R}_3, \bar{R}_2, \bar{R}_1, \bar{R}_{\text{out}})$ , and  $(\bar{R}_2, \bar{R}_0, \bar{R}_1, \bar{R}_{\text{out}})$  in XYZ space and  $(\bar{r}_0^1, \bar{r}_1^1, \bar{r}_2^1, \bar{r}_3^1)$ ,  $(\bar{r}_0^2, \bar{r}_1^2, \bar{r}_2^2, \bar{r}_3^2)$ , and  $(\bar{r}_0^3, \bar{r}_1^3, \bar{r}_2^3, \bar{r}_3^3)$  in each sub-cell where the lowercase refers to the sub-cell coordinates. The node ordering is chosen to define the sub-cell edge vectors  $\bar{e}_1 = \bar{r}_1 - \bar{r}_0$ ,  $\bar{e}_2 = \bar{r}_2 - \bar{r}_1$ , and  $\bar{e}_3 = \bar{r}_3 - \bar{r}_2$ ,

which obey the relation  $(\bar{e}_1 \times \bar{e}_2) \cdot \bar{e}_3 > 0$ . In this case, the parent cell is a case -3 tetrahedron (3 inflow faces and 1 outflow face) thus producing 3 sub-tetrahedra. Note also that the optical thickness of each sub-cell is the same,  $\varepsilon = \sigma l$ , where  $\sigma$  is the total cross section for the cell material and  $l = |\bar{R}_{\text{out}} - \bar{R}_{\text{in}}|$  is the length of the splitting line ( $\bar{e}_3$  in each subcell) depicted in Figure 7. Only six parent cell configurations for splitting an arbitrarily oriented tetrahedron cell and streaming direction are possible. These cases are outlined in Table 1.

Table 1. Possible Parent Tetrahedron Cell Cases.

Case	Sub-cells	# of inflow faces ( <i>i</i> )	# of outflow faces ( <i>j</i> )
1	1	1	1
2	2	1	2
3	3	1	3
4	4	2	2
-2	2	2	1
-3	3	3	1

Referring to Table 1, the parent cell case is determined by the taking the dot product of the streaming direction and each outward face normal,  $\hat{n} \cdot \hat{\Omega}$ . An outflow occurs for  $\hat{n} \cdot \hat{\Omega} > 0$  and an inflow occurs for  $\hat{n} \cdot \hat{\Omega} < 0$ . For  $\hat{n} \cdot \hat{\Omega}$  close to zero ( $< 10^{-12}$ ), no contribution is made to the case (a no-flow face). Letting  $i$  be the total number of inflow faces and  $j$  the total of outflow faces, the case of a tetrahedron with respect to  $\hat{\Omega}$  is given as

$$\text{Case}(i, j) = 10 - 11i + i^2 - 4j + 5ij. \quad (3)$$

The algorithm used to properly split a tetrahedron into the appropriate number of sub-tetrahedra is presented in the aforementioned paper (Mathews, 1998).

### Characteristic Solution in the Sub-cell Unit Tetrahedron

The characteristic solution of the BTE along the streaming path  $\hat{\Omega}$  (which is parallel to the  $w$  axis—  $\bar{e}_3$ — of the sub-cell) is particularly elegant when we operate on the unit tetrahedron. Integrating the BTE along the streaming line, from the point of entry into the sub-cell, at  $(u, v, 0)$  (where lower case  $u$  and  $v$  are used to indicate sub-cell coordinates), to the point of interest, at  $(u, v, w)$ , the angular flux is

$$\psi(u, v, w) = \psi^{\text{in}}(u, v) e^{-\epsilon w} + l \int_0^w S(u, v, w') e^{-\epsilon(w-w')} dw', \quad (4)$$

where  $S(u, v, w)$  is the source,  $\psi^{\text{in}}(u, v)$  is the inflow flux, and  $\psi(u, v, w)$  is the sub-cell flux. Before proceeding with the derivation of the exponential characteristic method, we will now briefly discuss the integral functions which are central to the EC method: the exponential moments functions.

### Moments Functions

The exponential moment functions (Mathews, 1994) occur in deriving the EC method. They are defined as

$$\mathcal{M}_n(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m) = \int_0^1 dt_1 (1-t_1)^n e^{-\mathbf{x}_1 t_1} \int_0^{t_1} dt_2 e^{(\mathbf{x}_1 - \mathbf{x}_2) t_2} \dots \int_0^{t_{m-1}} dt_m e^{(\mathbf{x}_{m-1} - \mathbf{x}_m) t_m}. \quad (5)$$

They are used to package the numerical difficulty of the EC method and help provide an elegant derivation. Had their pattern not been discerned, this research would have little hope of success. They are related via a multitude of recurrence relations and are always positive (Minor, 1993; Mathews, 1994; Minor, 1995; Mathews, 1997). Appendix A summarizes the exponential moment function identities used in this work.

In addition to the moment functions, various ratios of moment functions are useful. One common moment function ratio is the  $\rho$  function, defined as:

$$\rho(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m) = \frac{\mathcal{M}_1(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m)}{\mathcal{M}_0(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m)}. \quad (6)$$

Note that  $0 < \rho(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m) < 1$ .

Lastly, we introduce a new moment function ratio,  $\mathcal{R}_j$ , which is defined as

$$\mathcal{R}_j(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m) = \frac{\mathcal{M}_0(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m, \mathbf{x}_j)}{\mathcal{M}_0(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m)}, \quad (7)$$

where the  $j$  subscript indicates that the  $j^{\text{th}}$  argument is repeated in the numerator  $\mathcal{M}_0$  function argument list (but not in the denominator  $\mathcal{M}_0$ ). Note

also that  $0 < \mathcal{R}_j < 1$ . The  $\mathcal{R}_j$  function along with Identity 6 (Appendix A) allow for particularly elegant and robust formulations of the EC spatial quadrature.

Key to this research was the development of stable and accurate numerical routines for evaluating one through four argument moment ( $\mathcal{M}_0$ ) and  $\rho$  functions. The one-, two-, and three-argument function algorithms were presented previously (Mathews, 1997). The four-argument algorithm is presented in Appendix A.

## Derivation of the Exponential Characteristic Method

### EC Source Distribution

We begin the derivation of the exponential characteristic method by defining a volume moment operator for the unit tetrahedron exactly as done in the linear characteristic derivation (Mathews, 1998):

$$M[g] \equiv \frac{1}{V} \iiint_{\text{cell}} g(\bar{\mathbf{R}}) d^3 R = 6 \int_0^1 dU \int_0^U dV \int_0^V dW g(U, V, W). \quad (8)$$

The EC method uses an exponential source distribution,

$S(X, Y, Z) = \exp(\alpha + \beta_X X + \beta_Y Y + \beta_Z Z)$ , that has the known central moments

(initialized or known from a previous iteration),  $S_A = M[S(X, Y, Z)]$ ,

$S_X = M[(X - \bar{X})S(X, Y, Z)]$ ,  $S_Y = M[(Y - \bar{Y})S(X, Y, Z)]$ , and

$S_Z = M[(Z - \bar{Z})S(X, Y, Z)]$ . However, neither the moments nor their coefficients

are of any particular interest by themselves. Analogous to the LC derivation,

the moments produced by the angular quadrature can be with respect to any three linearly independent basis vectors; we choose the unit tetrahedron edge vectors. Thus, we seek a function in  $U$ ,  $V$ , and  $W$ , that matches the moments of the source in the oblique (unit tetrahedron) coordinate system. The source distribution then becomes:

$$S(U, V, W) = \exp[A_s + \tilde{B}_u U + \tilde{B}_v V + \tilde{B}_w W]. \quad (9)$$

Operating on  $S(U, V, W)$  with the volume moment operator, we find the following system for the source moments:

$$S_A = M[S(U, V, W)] = 6 \exp[A_s] \mathcal{M}_0(X_s, Y_s, Z_s) \quad (10)$$

$$S_U = M[US(U, V, W)] = S_A (1 - \rho(X_s, Y_s, Z_s)) \quad (11)$$

$$S_V = M[VS(U, V, W)] = S_U - S_A \frac{\mathcal{M}_0(X_s, X_s, Y_s, Z_s)}{\mathcal{M}_0(X_s, Y_s, Z_s)} \quad (12)$$

and,

$$S_W = M[WS(U, V, W)] = S_V - S_A \frac{\mathcal{M}_0(X_s, Y_s, Y_s, Z_s)}{\mathcal{M}_0(X_s, Y_s, Z_s)}, \quad (13)$$

where

$$\begin{aligned} X_s &= -\tilde{B}_u, \\ Y_s &= -(\tilde{B}_u + \tilde{B}_v), \\ Z_s &= -(\tilde{B}_u + \tilde{B}_v + \tilde{B}_w). \end{aligned} \quad (14)$$

Using Identity 6 (Appendix A), equations (10) through (13) can be written in an algebraically efficient form as

$$S_A = 6 \exp[A_s] \mathcal{M}_0(X_s, Y_s, Z_s), \quad (15)$$

$$S_W = S_A \mathcal{R}_3(X_s, Y_s, Z_s), \quad (16)$$

$$S_V = S_W + S_A \mathcal{R}_2(X_s, Y_s, Z_s), \quad (17)$$

and

$$S_U = S_V + S_A \mathcal{R}_1(X_s, Y_s, Z_s). \quad (18)$$

Unlike the LC spatial quadrature, the above system is not directly invertible to find the source coefficients. In order to obtain  $A_s$ ,  $\tilde{B}_u$ ,  $\tilde{B}_v$ , and,  $\tilde{B}_w$ , use of a non-linear root-solver is necessary; we use Broyden's method (Burden, 1993). Although the development of an accurate first guess algorithm and efficient root-solving were key contributions of this research, the details of these algorithms have been relegated to Appendix B in order to preserve the flow of the following derivations.

Once the coefficients are found, direct affine transformations are used to transform the coefficients from the unit tetrahedron cell to the various unit tetrahedron sub-cells for use in the spatial quadrature. The reason for this is as follows. Consider two linear functions,  $l(\bar{u}(\bar{x}))$  and  $l'(\bar{x})$ , where

$l(\bar{u}(\bar{x})) = l'(\bar{x})$  and  $\bar{x}$  is a coordinate vector. Now, suppose that  $l(\bar{u}) = a + \bar{b} \cdot \bar{u}$

and  $\bar{u}(\bar{x}) = \bar{u}_0 + \bar{\bar{J}} \cdot (\bar{x} - \bar{x}_0)$  where  $\bar{u}_0$  is some initial value,  $\bar{\bar{J}}$  is the Jacobian of the transformation, and  $\bar{x}_0$  is a translated origin for the subcell coordinate system. Then we have that  $l(\bar{u}(\bar{x})) = \alpha + \bar{b} \cdot \bar{u}_0 + \bar{b} \cdot \bar{\bar{J}} \cdot \bar{x} - \bar{b} \cdot \bar{\bar{J}} \cdot \bar{x}_0 = \alpha' + \bar{b}' \cdot \bar{x}$  where  $\alpha' = \alpha + \bar{b} \cdot \bar{u}_0 - \bar{b} \cdot \bar{\bar{J}} \cdot \bar{x}_0$  and  $\bar{b}' = \bar{b} \cdot \bar{\bar{J}}$ . In the case of the EC method, because the source coefficients are related linearly in the spatial coordinate as discussed above, exponentiating  $l(\bar{u}(\bar{x}))$  and  $l'(\bar{x})$  has no effect on the transformation of the coefficients. Indeed, as long as the coefficients are maintained in a linear form, one can perform any arbitrary operation,  $f$ , on both  $l(\bar{u}(\bar{x}))$  and  $l'(\bar{x})$  and still perform a linear transformation of the coefficients as presented above.

### EC Inflow Face Flux Distribution

Analogous to the source equation development, we must define an operator to take moments over an inflow face. In the case of the unit face, this operator is

$$M^f[g] \equiv \frac{1}{A_f} \iint_{\text{face}} g(\bar{R}) d^2R = 2 \int_0^1 dU^f \int_0^{U^f} dV^f g(U^f, V^f). \quad (19)$$

As with the source distribution, we assume an exponential distribution of the inflow flux on the face,

$$\Psi_{\text{in}}^f(U^f, V^f) = \exp(A^f + B_u^f U^f + B_v^f V^f), \quad (20)$$

such that the moments over the face are



$$\Psi_A^f = M^f[\Psi_{in}^f(U^f, V^f)] = 2 \exp(A_f) \mathcal{M}_0(X_f, Y_f), \quad (21)$$

$$\Psi_U^f = M^f[U^f \Psi_{in}^f(U^f, V^f)] = \Psi_A^f [1 - \rho(X_f, Y_f)], \quad (22)$$

$$\text{and} \quad \Psi_V^f = M^f[V^f \Psi_{in}^f(U^f, V^f)] = \Psi_U^f - \Psi_A^f \frac{\mathcal{M}_0(X_f, X_f, Y_f)}{\mathcal{M}_0(X_f, Y_f)}, \quad (23)$$

where the coefficients are

$$\begin{aligned} X_f &= -B_u^f \\ Y_f &= -(B_u^f + B_v^f). \end{aligned} \quad (24)$$

Using Identity 6 (Appendix A), equations (21) through (23) can be expressed compactly as

$$\Psi_A^f = 2 \exp(A_f) \mathcal{M}_0(X_f, Y_f), \quad (25)$$

$$\Psi_V^f = \Psi_A^f \mathcal{R}_2(X_f, Y_f), \quad (26)$$

and

$$\Psi_U^f = \Psi_V^f + \Psi_A^f \mathcal{R}_1(X_f, Y_f). \quad (27)$$

As with the source system, the system of equations for the inflow flux moments is non-linear and requires a root-solver (Broyden's method) to obtain the needed coefficients.

## Derivation of Sub-Cell EC Spatial Quadrature

The above source and inflow flux coefficients are rotated/translated into the sub-cell coordinate system using direct affine transformations (Mathews, 1998). This is possible since the coefficients are linear and are only operated on by the exponential. Having obtained the coefficients for the sub-cell, we now define sub-cell volume and outflow face area moment operators and the sub-cell characteristic equation for the flux,  $\psi$ .

Analogous to the cell volume operator, the sub-cell volume moment operator is defined as

$$m[g(\tilde{r})] = 6 \int_0^1 du \int_0^u dv \int_0^v dw g(u, v, w). \quad (28)$$

We use lower case for sub-cell variables and upper case for cell variables. The sub-cell characteristic equation for the sub-cell, obtained by integrating the BTE along the streaming direction from  $(u, v, 0)$  to the point  $(u, v, w)$ , is

$$\psi(u, v, w) = \psi^{\text{in}}(u, v) e^{-\varepsilon w} + l \int_0^w S(u, v, w') e^{-\varepsilon(w-w')} dw', \quad (29)$$

where the  $l$ , the path length across the sub-cell, accounts for treating the source distribution as a point function. This is done to keep the units of the source (particles/cm<sup>3</sup>/s) and flux (particle-cm/ cm<sup>3</sup>/s) consistent regardless of the coordinate system of the independent variable for position ( $\bar{R}$ ,  $\tilde{R}$ , or  $\tilde{r}$ ) (Mathews, 1998). The source distribution is assumed to be an exponential,

$S(u, v, w) = \exp(a_s + b_u^s u + b_v^s v + b_w^s w)$ . Similarly, the inflow flux is also

assumed to have an exponential distribution on the sub-cell face,

$$\psi^{\text{in}}(u, v) = \exp(a_f + b_u^f u + b_v^f v).$$

The derivation of the sub-cell flux moments for EC starts by calculating the average cell flux,

$$\begin{aligned}\psi_A^{\text{subcell}} &= m[\psi(u, v, w)] = \psi_{A,\text{in}}^{\text{subcell}} + \psi_{A,\text{src}}^{\text{subcell}}, \\ \psi_{A,\text{in}}^{\text{subcell}} &= 6 \exp(a_f) \mathcal{M}_0(x_1^f, y_1^f, z_1^f), \\ \psi_{A,\text{src}}^{\text{subcell}} &= 6 l \exp(a_s) \mathcal{M}_0(x_1^s, y_1^s, z_1^s, w_1^s),\end{aligned}\tag{30}$$

where the coefficients for the sub-cell inflow face (obtained via affine transformation from the global face) are

$$\begin{aligned}x_1^f &= -b_u^f, \\ y_1^f &= -(b_u^f + b_v^f), \\ z_1^f &= -(b_u^f + b_v^f) + \varepsilon,\end{aligned}\tag{31}$$

and the source coefficients are

$$\begin{aligned}x_1^s &= -b_u^s, \\ y_1^s &= -(b_u^s + b_v^s), \\ z_1^s &= -(b_u^s + b_v^s) + \varepsilon, \\ w_1^s &= -(b_u^s + b_v^s + b_w^s),\end{aligned}\tag{32}$$

where the source and inflow flux coefficients are denoted by either an s or f superscript, respectively, and  $\varepsilon$  is the cell optical thickness.

Each coefficient above is subscripted with 1 to differentiate it from the coefficients for the outflow flux. This notation promotes compactness. Note that each sub-cell will have its own different set of coefficients.

As with the average flux, the  $u$ -moment of the flux is defined as

$$\begin{aligned}
\psi_u^{\text{subcell}} &= m[u \psi(u, v, w)] = \psi_{u,\text{in}}^{\text{subcell}} + \psi_{u,\text{src}}^{\text{subcell}}, \\
\psi_{u,\text{in}}^{\text{subcell}} &= \psi_{A,\text{in}}^{\text{subcell}} [1 - \rho(\mathbf{x}_1^f, \mathbf{y}_1^f, \mathbf{z}_1^f)] \\
&= \psi_{A,\text{in}}^{\text{subcell}} [\mathcal{R}_1(\mathbf{x}_1^f, \mathbf{y}_1^f, \mathbf{z}_1^f) + \mathcal{R}_2(\mathbf{x}_1^f, \mathbf{y}_1^f, \mathbf{z}_1^f) + \mathcal{R}_3(\mathbf{x}_1^f, \mathbf{y}_1^f, \mathbf{z}_1^f)], \\
\psi_{u,\text{src}}^{\text{subcell}} &= \psi_{A,\text{src}}^{\text{subcell}} [1 - \rho(\mathbf{x}_1^s, \mathbf{y}_1^s, \mathbf{z}_1^s, \mathbf{w}_1^s)] \\
&= \psi_{A,\text{src}}^{\text{subcell}} [\mathcal{R}_1(\mathbf{x}_1^s, \mathbf{y}_1^s, \mathbf{z}_1^s, \mathbf{w}_1^s) + \mathcal{R}_2(\mathbf{x}_1^s, \mathbf{y}_1^s, \mathbf{z}_1^s, \mathbf{w}_1^s) + \\
&\quad \mathcal{R}_3(\mathbf{x}_1^s, \mathbf{y}_1^s, \mathbf{z}_1^s, \mathbf{w}_1^s) + \mathcal{R}_4(\mathbf{x}_1^s, \mathbf{y}_1^s, \mathbf{z}_1^s, \mathbf{w}_1^s)],
\end{aligned} \tag{33}$$

where Identity 6 (for three- and four-arguments) in Appendix A is used to obtain the second relationship for  $\psi_{u,\text{in}}^{\text{subcell}}$  and  $\psi_{u,\text{src}}^{\text{subcell}}$ , respectively. Using the formulation produced by this identity allows us to see that

$$\psi_A^{\text{subcell}} > \psi_u^{\text{subcell}} > \psi_v^{\text{subcell}} > \psi_w^{\text{subcell}}.$$

The derivation of the  $v$ -moment of the flux is a little more complex than the average and  $u$ -moment derivations. In the previous cases, the integral definition for the moment functions follows directly from the application of the moment operators. In the case of the  $v$ -moment, integration by parts is used to evaluate the integrals. Applying the moment operators,

$$\begin{aligned}
\psi_v^{\text{subcell}} &= \mathbf{m}[v \psi(u, v, w)] = \psi_{v, \text{in}}^{\text{subcell}} + \psi_{v, \text{src}}^{\text{subcell}}, \\
\psi_{v, \text{in}}^{\text{subcell}} &= \psi_{u, \text{in}}^{\text{subcell}} - \psi_{A, \text{in}}^{\text{subcell}} \frac{\mathcal{M}_0(\mathbf{x}_1^f, \mathbf{x}_1^f, \mathbf{y}_1^f, \mathbf{z}_1^f)}{\mathcal{M}_0(\mathbf{x}_1^f, \mathbf{y}_1^f, \mathbf{z}_1^f)} \\
&= \psi_{A, \text{in}}^{\text{subcell}} [\mathcal{R}_2(\mathbf{x}_1^f, \mathbf{y}_1^f, \mathbf{z}_1^f) + \mathcal{R}_3(\mathbf{x}_1^f, \mathbf{y}_1^f, \mathbf{z}_1^f)], \\
\psi_{v, \text{src}}^{\text{subcell}} &= \psi_{u, \text{src}}^{\text{subcell}} - \psi_{A, \text{src}}^{\text{subcell}} \frac{\mathcal{M}_0(\mathbf{x}_1^s, \mathbf{x}_1^s, \mathbf{y}_1^s, \mathbf{z}_1^s, \mathbf{w}_1^s)}{\mathcal{M}_0(\mathbf{x}_1^s, \mathbf{y}_1^s, \mathbf{z}_1^s, \mathbf{w}_1^s)} \\
&= \psi_{A, \text{src}}^{\text{subcell}} [\mathcal{R}_2(\mathbf{x}_1^s, \mathbf{y}_1^s, \mathbf{z}_1^s, \mathbf{w}_1^s) + \mathcal{R}_3(\mathbf{x}_1^s, \mathbf{y}_1^s, \mathbf{z}_1^s, \mathbf{w}_1^s) + \\
&\quad \mathcal{R}_4(\mathbf{x}_1^s, \mathbf{y}_1^s, \mathbf{z}_1^s, \mathbf{w}_1^s)],
\end{aligned} \tag{34}$$

where we see that the second equations for  $\psi_{v, \text{in}}^{\text{subcell}}$  and  $\psi_{v, \text{src}}^{\text{subcell}}$  are obtained

by applying the previous definitions of  $\psi_{u, \text{in}}^{\text{subcell}}$  and  $\psi_{u, \text{src}}^{\text{subcell}}$  and Identity 6.

Finally, the  $w$ -moment of the cell flux is derived by applying the moment operator and integration by parts,

$$\begin{aligned}
\psi_w^{\text{subcell}} &= \mathbf{m}[w \psi(u, v, w)] = \psi_{w, \text{in}}^{\text{subcell}} + \psi_{w, \text{src}}^{\text{subcell}}, \\
\psi_{w, \text{in}}^{\text{subcell}} &= \psi_{v, \text{in}}^{\text{subcell}} - \psi_{A, \text{in}}^{\text{subcell}} \frac{\mathcal{M}_0(\mathbf{x}_1^f, \mathbf{y}_1^f, \mathbf{y}_1^f, \mathbf{z}_1^f)}{\mathcal{M}_0(\mathbf{x}_1^f, \mathbf{y}_1^f, \mathbf{z}_1^f)} \\
&= \psi_{A, \text{in}}^{\text{subcell}} \mathcal{R}_3(\mathbf{x}_1^f, \mathbf{y}_1^f, \mathbf{z}_1^f), \\
\psi_{w, \text{src}}^{\text{subcell}} &= \psi_{v, \text{src}}^{\text{subcell}} - \psi_{A, \text{src}}^{\text{subcell}} \frac{\mathcal{M}_0(\mathbf{x}_1^s, \mathbf{y}_1^s, \mathbf{y}_1^s, \mathbf{z}_1^s, \mathbf{w}_1^s)}{\mathcal{M}_0(\mathbf{x}_1^s, \mathbf{y}_1^s, \mathbf{z}_1^s, \mathbf{w}_1^s)} \\
&= \psi_{A, \text{src}}^{\text{subcell}} [\mathcal{R}_3(\mathbf{x}_1^s, \mathbf{y}_1^s, \mathbf{z}_1^s, \mathbf{w}_1^s) + \mathcal{R}_4(\mathbf{x}_1^s, \mathbf{y}_1^s, \mathbf{z}_1^s, \mathbf{w}_1^s)],
\end{aligned} \tag{35}$$

where again the previous moments and the moment function identities are used to arrive at the second form of the equations.

Re-arranging terms in order to minimize the amount of arithmetic associated with the contributions to the cell flux moments due to the source, one obtains the following equations:

$$\begin{aligned}
\psi_{A,src}^{subcell} &= 6l \exp(a_s) \mathcal{M}_0(x_1^s, y_1^s, z_1^s, w_1^s), \\
\psi_{w,src}^{subcell} &= \psi_{A,src}^{subcell} \left( \mathcal{R}_3(x_1^s, y_1^s, z_1^s, w_1^s) + \mathcal{R}_4(x_1^s, y_1^s, z_1^s, w_1^s) \right), \\
\psi_{v,src}^{subcell} &= \psi_{w,src}^{subcell} + \psi_{A,src}^{subcell} \mathcal{R}_2(x_1^s, y_1^s, z_1^s, w_1^s), \\
\psi_{u,src}^{subcell} &= \psi_{v,src}^{subcell} + \psi_{A,src}^{subcell} \mathcal{R}_1(x_1^s, y_1^s, z_1^s, w_1^s).
\end{aligned} \tag{36}$$

Similarly, the following equations are found for the inflow flux contribution to the sub-cell flux moments,

$$\begin{aligned}
\psi_{A,in}^{subcell} &= 6 \exp(a_f) \mathcal{M}_0(x_1^f, y_1^f, z_1^f), \\
\psi_{w,in}^{subcell} &= \psi_{A,in}^{subcell} \mathcal{R}_3(x_1^f, y_1^f, z_1^f), \\
\psi_{v,in}^{subcell} &= \psi_{w,in}^{subcell} + \psi_{A,in}^{subcell} \mathcal{R}_2(x_1^f, y_1^f, z_1^f), \\
\psi_{u,in}^{subcell} &= \psi_{v,in}^{subcell} + \psi_{A,in}^{subcell} \mathcal{R}_1(x_1^f, y_1^f, z_1^f).
\end{aligned} \tag{37}$$

The outflow face flux moments will now be derived. The characteristic equation for the flux on the outflow face is

$$\psi^{out}(u, v) = \psi(u, v, v) = \psi^{in}(u, v) e^{-\varepsilon v} + l \int_0^v S(u, v, w') e^{-\varepsilon(v-w')}. \tag{38}$$

Note that the outflow face equation is  $v^{face} = v = w$  (refer to Figure 4). The outflow face area moment operator is

$$\mathbf{m}^{out}[g(u,v,v)] = 2 \int_0^1 du \int_0^u dv g(u,v,v). \quad (39)$$

The inflow face area moment operator is not explicitly defined because the basis vectors corresponding to the inflow face flux are passed to it from the upstream neighbor faces (outflow faces).

Starting with the average flux on the outflow face,

$$\begin{aligned} \psi_A^{outface} &= \mathbf{m}^{out}[\psi(u,v,v)] = \psi_{A,in}^{outface} + \psi_{A,src}^{outface}, \\ \psi_{A,in}^{outface} &= 2 \exp(a_f) \mathcal{M}_0(\mathbf{x}_2^f, y_2^f), \\ \psi_{A,src}^{outface} &= 2l \exp(a_s) \mathcal{M}_0(\mathbf{x}_2^s, y_2^s, z_2^s), \end{aligned} \quad (40)$$

where the face coefficients are

$$\begin{aligned} \mathbf{x}_2^f &= -\mathbf{b}_u^f \\ y_2^f &= -(\mathbf{b}_u^f + \mathbf{b}_v^f) + \varepsilon \end{aligned} \quad (41)$$

and the source coefficients are

$$\begin{aligned} \mathbf{x}_2^s &= -\mathbf{b}_u^s \\ y_2^s &= -(\mathbf{b}_u^s + \mathbf{b}_v^s) + \varepsilon \\ z_2^s &= -(\mathbf{b}_u^s + \mathbf{b}_v^s + \mathbf{b}_w^s). \end{aligned} \quad (42)$$

Applying the moment operator with respect to  $u$ ,

$$\begin{aligned}
\psi_u^{\text{outface}} &= \mathbf{m}^{\text{out}}[u \psi(u, v, v)] = \psi_{u, \text{in}}^{\text{outface}} + \psi_{u, \text{src}}^{\text{outface}}, \\
\psi_{u, \text{in}}^{\text{outface}} &= \psi_{A, \text{in}}^{\text{outface}} [1 - \rho(\mathbf{x}_2^f, \mathbf{y}_2^f)] \\
&= \psi_{A, \text{in}}^{\text{outface}} [\mathcal{R}_1(\mathbf{x}_2^f, \mathbf{y}_2^f) + \mathcal{R}_2(\mathbf{x}_2^f, \mathbf{y}_2^f)], \\
\psi_{u, \text{src}}^{\text{outface}} &= \psi_{A, \text{src}}^{\text{outface}} [1 - \rho(\mathbf{x}_2^s, \mathbf{y}_2^s, z_2^s)] \\
&= \psi_{A, \text{src}}^{\text{outface}} [\mathcal{R}_1(\mathbf{x}_2^s, \mathbf{y}_2^s, z_2^s) + \mathcal{R}_2(\mathbf{x}_2^s, \mathbf{y}_2^s, z_2^s) + \mathcal{R}_3(\mathbf{x}_2^s, \mathbf{y}_2^s, z_2^s)],
\end{aligned} \tag{43}$$

where again Identity 6 (Appendix A) is used to arrive at the second equation for the  $u$ -moment of the outflow flux.

Finally, applying the outflow face moment operator and integrating by parts, the following equations for the  $v$ -moment of the outflow flux are obtained:

$$\begin{aligned}
\psi_v^{\text{outface}} &= \mathbf{m}^{\text{out}}[v \psi(u, v, v)] = \psi_{v, \text{in}}^{\text{outface}} + \psi_{v, \text{src}}^{\text{outface}}, \\
\psi_{v, \text{in}}^{\text{outface}} &= \psi_{u, \text{in}}^{\text{outface}} - \psi_{A, \text{in}}^{\text{outface}} \frac{\mathcal{M}_0(\mathbf{x}_2^f, \mathbf{x}_2^f, \mathbf{y}_2^f)}{\mathcal{M}_0(\mathbf{x}_2^f, \mathbf{y}_2^f)} \\
&= \psi_{A, \text{in}}^{\text{outface}} \mathcal{R}_2(\mathbf{x}_2^f, \mathbf{y}_2^f), \\
\psi_{v, \text{src}}^{\text{outface}} &= \psi_{u, \text{src}}^{\text{outface}} - \psi_{A, \text{src}}^{\text{outface}} \frac{\mathcal{M}_0(\mathbf{x}_2^s, \mathbf{x}_2^s, \mathbf{y}_2^s, z_2^s)}{\mathcal{M}_0(\mathbf{x}_2^s, \mathbf{y}_2^s, z_2^s)} \\
&= \psi_{A, \text{src}}^{\text{outface}} [\mathcal{R}_2(\mathbf{x}_2^s, \mathbf{y}_2^s, z_2^s) + \mathcal{R}_3(\mathbf{x}_2^s, \mathbf{y}_2^s, z_2^s)].
\end{aligned} \tag{44}$$

Re-arranging terms in order to minimize the algebraic operations, the the outflow face flux moments due to the source are

$$\begin{aligned}
\psi_{A, \text{src}}^{\text{outface}} &= 2l \exp(a_s) \mathcal{M}_0(\mathbf{x}_2^s, \mathbf{y}_2^s, z_2^s), \\
\psi_{v, \text{src}}^{\text{outface}} &= \psi_{A, \text{src}}^{\text{outface}} (\mathcal{R}_2(\mathbf{x}_2^s, \mathbf{y}_2^s, z_2^s) + \mathcal{R}_3(\mathbf{x}_2^s, \mathbf{y}_2^s, z_2^s)), \\
\psi_{u, \text{src}}^{\text{outface}} &= \psi_{v, \text{src}}^{\text{outface}} + \psi_{A, \text{src}}^{\text{outface}} \mathcal{R}_1(\mathbf{x}_2^s, \mathbf{y}_2^s, z_2^s).
\end{aligned} \tag{45}$$



Similarly, the outflow face flux moments due to the inflow flux are

$$\begin{aligned}
\psi_{A,\text{in}}^{\text{outface}} &= 2 \exp(a_f) \mathcal{M}_0(\mathbf{x}_2^f, y_2^f), \\
\psi_{v,\text{in}}^{\text{outface}} &= \psi_{A,\text{in}}^{\text{outface}} \mathcal{R}_2(\mathbf{x}_2^f, y_2^f), \\
\psi_{u,\text{in}}^{\text{outface}} &= \psi_{v,\text{in}}^{\text{outface}} + \psi_{A,\text{in}}^{\text{outface}} \mathcal{R}_1(\mathbf{x}_2^f, y_2^f).
\end{aligned} \tag{46}$$

### Sub-Cell Conservation Equations

The sub-cell conservation equations are derived by applying the volume moment operator to the BTE for each moment, yielding:

$$\text{Average :} \quad 3(\psi_A^{\text{outface}} - \psi_A^{\text{in}}) + \varepsilon \psi_A^{\text{subcell}} = l S_A, \tag{47}$$

$$u\text{-moment:} \quad 3(\psi_u^{\text{outface}} - \psi_u^{\text{in}}) + \varepsilon \psi_u^{\text{subcell}} = l S_u, \tag{48}$$

$$v\text{-moment:} \quad 3(\psi_v^{\text{outface}} - \psi_v^{\text{in}}) + \varepsilon \psi_v^{\text{subcell}} = l S_v, \tag{49}$$

$$w\text{-moment:} \quad 3 \psi_v^{\text{outface}} - \psi_A^{\text{subcell}} + \varepsilon \psi_w^{\text{subcell}} = l S_w. \tag{50}$$

Note in (50) that  $\psi_w^{\text{outface}} = \psi_v^{\text{outface}}$  while  $\psi_w^{\text{inface}} = 0$  thus leading to the correct conservation relation (the  $\psi_A^{\text{subcell}}$  term is obtained as a result of the integration of the BTE). The conservation equations are used in the spatial quadrature to check that the quadrature is numerically accurate. In the event that the sub-cells are not too optically thin, these equations could be used to replace some of the more expensive spatial quadrature formulas

(Brennan, 1996). However, this was not done in TETRAN because it can be numerically ill conditioned.

### Linear Characteristic Spatial Quadrature

Before discussing the implementation of the EC method in our code system, we briefly summarize the linear characteristic (LC) spatial quadrature as developed in (Mathews, 1998). Its performance is compared to that of EC in Chapter 4. The LC method has been implemented in some other codes systems because it is much less expensive computationally than the EC method, so it provides TETRAN with a more mainstream transport capability. Adding LC requires modest extra effort, because all of the cell splitting and transformation algorithms are the same regardless of which characteristic spatial quadrature is used.

LC was the first quadrature implemented into the TETRAN code system. It assumes a linear source and inflow flux distribution as opposed to the exponential assumptions for EC:

$$S(U, V, W) = A_s + \tilde{B}_u U + \tilde{B}_v V + \tilde{B}_w W \quad (51)$$

and

$$\Psi_{in}^f(U^f, V^f) = A^f + B_u^f U^f + B_v^f V^f. \quad (52)$$

Exactly as was done with the EC source function, spatial moments are taken of the LC source distribution (equation (51)). The resulting relationships for the LC source moments are:

$$S_A = M[S(U, V, W)] = A_s + \frac{3}{4}\tilde{B}_u + \frac{1}{2}\tilde{B}_v + \frac{1}{4}\tilde{B}_w, \quad (53)$$

$$S_U = M[US(U, V, W)] = \frac{3}{4}A_s + \frac{3}{5}\tilde{B}_u + \frac{2}{5}\tilde{B}_v + \frac{1}{5}\tilde{B}_w, \quad (54)$$

$$S_V = M[VS(U, V, W)] = \frac{1}{2}A_s + \frac{2}{5}\tilde{B}_u + \frac{3}{10}\tilde{B}_v + \frac{3}{20}\tilde{B}_w, \quad (55)$$

and  $S_W = M[WS(U, V, W)] = \frac{1}{4}A_s + \frac{1}{5}\tilde{B}_u + \frac{3}{20}\tilde{B}_v + \frac{1}{10}\tilde{B}_w. \quad (56)$

Unlike the EC method where the source coefficients ( $A_s, \tilde{B}_u, \tilde{B}_v, \tilde{B}_w$ ) must be obtained by root-solving, the system of equations above can be directly inverted to give the source coefficients:

$$A_s = 16S_A - 20S_U, \quad (57)$$

$$\tilde{B}_u = -20S_A + 40S_U - 20S_V, \quad (58)$$

$$\tilde{B}_v = -20S_U + 40S_V - 20S_W, \quad (59)$$

and  $\tilde{B}_w = -20S_V + 40S_W. \quad (60)$

Similarly, the inflow face flux coefficients are obtained by taking the appropriate moments over the tetrahedron cell's face using (52):

$$\Psi_A^f = M^f[\Psi_{in}^f(U^f, V^f)] = A^f + \frac{2}{3}B_u^f + \frac{1}{3}B_v^f, \quad (61)$$

$$\Psi_U^f = M^f[U^f \Psi_{in}^f(U^f, V^f)] = \frac{2}{3}A^f + \frac{1}{2}B_u^f + \frac{1}{4}B_v^f, \quad (62)$$

and 
$$\Psi_V^f = M^f[V^f \Psi_{in}^f(U^f, V^f)] = \frac{1}{3}A^f + \frac{1}{4}B_u^f + \frac{1}{6}B_v^f. \quad (63)$$

As was done for the source moment equations, the inflow flux moment equations can be directly inverted to solve for the inflow flux coefficients:

$$A^f = 9\Psi_A^f - 12\Psi_U^f, \quad (64)$$

$$B_u^f = -12\Psi_A^f + 24\Psi_U^f - 12\Psi_V^f, \quad (65)$$

and 
$$B_v^f = -12\Psi_U^f + 24\Psi_V^f. \quad (66)$$

Clearly, obtaining the required coefficients to solve the source and inflow flux systems is much less numerically intensive than that for EC. Thus, it should be expected that the LC method would be computationally cheaper than EC.

### **Linear Characteristic Sub-Cell Spatial Quadrature**

Proceeding with the LC development, the spatial quadrature is found analogously to EC by taking spatial moments of (29) (using the operator defined in (28)) with the appropriate source and inflow flux functional forms.

In the case of LC, the source is assumed to be linearly distributed,

$S(u, v, w) = a_s + b_u^s u + b_v^s v + b_w^s w$ . The inflow flux function is assumed to be

linearly distributed,  $\psi^{\text{in}}(u, v) = a_f + b_u^f u + b_v^f v$ .

The cell average flux moment is:

$$\begin{aligned}\psi_A^{\text{subcell}} &= m[\psi(u, v, w)] = \psi_{A,\text{in}}^{\text{subcell}} + \psi_{A,\text{src}}^{\text{subcell}}, \\ \psi_{A,\text{in}}^{\text{subcell}} &= 6(a_f K_{0,0,0}(\varepsilon) + b_u^f K_{1,0,0}(\varepsilon) + b_v^f K_{0,1,0}(\varepsilon)), \\ \psi_{A,\text{src}}^{\text{subcell}} &= 6l(a_s K_{0,0,0}(\varepsilon) + b_u^s K_{1,0,0}(\varepsilon) + b_v^s K_{0,1,0}(\varepsilon) + \\ &\quad b_w^s [K_{0,0,1}(\varepsilon) - K_{0,0,0}(\varepsilon)]),\end{aligned}\tag{67}$$

where the special function,  $K$ , is defined as

$$K_{i_1, i_2, \dots, i_m}(\varepsilon) = \int_0^1 dt_1 \int_0^{t_1} dt_2 \dots \int_0^{t_{m-1}} dt_m t_1^{i_1} t_2^{i_2} \dots t_m^{i_m} e^{-\varepsilon t_m}.\tag{68}$$

$K$  is only a function only of the cell optical thickness,  $\varepsilon$ . Because  $\varepsilon$  is the same for each subcell, the necessary  $K$  function values need only be calculated once, enhancing efficiency. With the introduction of the  $K$ -functions, the LC spatial quadrature is expressed in an elegant form. Algorithms that accurately and efficiently calculate the needed  $K$ -functions for the LC quadrature are discussed in the previously mentioned paper (Mathews, 1998).

The  $u$ -,  $v$ -, and  $w$ -moment of the sub-cell flux are given as

$$\begin{aligned}
\psi_u^{\text{subcell}} &= \mathbf{m}[u \psi(u, v, w)] = \psi_{u, \text{in}}^{\text{subcell}} + \psi_{u, \text{src}}^{\text{subcell}}, \\
\psi_{u, \text{in}}^{\text{subcell}} &= 6(a_f K_{1,0,0}(\epsilon) + b_u^f K_{2,0,0}(\epsilon) + b_v^f K_{1,1,0}(\epsilon)), \\
\psi_{u, \text{src}}^{\text{subcell}} &= 6l(a_s K_{1,0,0,0}(\epsilon) + b_u^s K_{2,0,0,0}(\epsilon) + b_v^s K_{1,1,0,0}(\epsilon) + \\
&\quad b_w^s [K_{1,0,1,0}(\epsilon) - K_{1,0,0,1}(\epsilon)]),
\end{aligned} \tag{69}$$

$$\begin{aligned}
\psi_v^{\text{subcell}} &= \mathbf{m}[v \psi(u, v, w)] = \psi_{v, \text{in}}^{\text{subcell}} + \psi_{v, \text{src}}^{\text{subcell}}, \\
\psi_{v, \text{in}}^{\text{subcell}} &= 6(a_f K_{0,1,0}(\epsilon) + b_u^f K_{1,1,0}(\epsilon) + b_v^f K_{0,2,0}(\epsilon)), \\
\psi_{v, \text{src}}^{\text{subcell}} &= 6l(a_s K_{0,1,0,0}(\epsilon) + b_u^s K_{1,1,0,0}(\epsilon) + b_v^s K_{0,2,0,0}(\epsilon) + \\
&\quad b_w^s [K_{0,1,1,0}(\epsilon) - K_{0,1,0,1}(\epsilon)]),
\end{aligned} \tag{70}$$

and

$$\begin{aligned}
\psi_w^{\text{subcell}} &= \mathbf{m}[w \psi(u, v, w)] = \psi_{w, \text{in}}^{\text{subcell}} + \psi_{w, \text{src}}^{\text{subcell}}, \\
\psi_{w, \text{in}}^{\text{subcell}} &= 6(a_f K_{0,0,1}(\epsilon) + b_u^f K_{1,0,1}(\epsilon) + b_v^f K_{0,1,1}(\epsilon)), \\
\psi_{w, \text{src}}^{\text{subcell}} &= 6l(a_s K_{0,0,1,0}(\epsilon) + b_u^s K_{1,0,1,0}(\epsilon) + b_v^s K_{0,1,1,0}(\epsilon) + \\
&\quad b_w^s [K_{0,0,2,0}(\epsilon) - K_{0,0,1,1}(\epsilon)]).
\end{aligned} \tag{71}$$

Operating on (38) with the outflow face area operator (39) using the linear functions presented at this beginning of this section, the LC outflow face flux moments are

$$\begin{aligned}
\psi_A^{\text{outface}} &= \mathbf{m}^{\text{out}}[\psi(u, v, v)] = \psi_{A, \text{in}}^{\text{outface}} + \psi_{A, \text{src}}^{\text{outface}}, \\
\psi_{A, \text{in}}^{\text{outface}} &= 2(a_f K_{0,0}(\epsilon) + b_u^f K_{1,0}(\epsilon) + b_v^f K_{0,1}(\epsilon)), \\
\psi_{A, \text{src}}^{\text{outface}} &= 2l(a_s K_{0,0,0}(\epsilon) + b_u^s K_{1,0,0}(\epsilon) + [b_v^s + b_w^s] K_{0,1,0}(\epsilon) - b_w^s K_{0,0,1}(\epsilon)),
\end{aligned} \tag{72}$$

$$\begin{aligned}
\psi_u^{\text{outface}} &= \mathbf{m}^{\text{out}}[u \psi(u, v, v)] = \psi_{u,\text{in}}^{\text{outface}} + \psi_{u,\text{src}}^{\text{outface}}, \\
\psi_{u,\text{in}}^{\text{outface}} &= 2(a_f K_{1,0}(\epsilon) + b_u^f K_{2,0}(\epsilon) + b_v^f K_{1,1}(\epsilon)), \\
\psi_{u,\text{src}}^{\text{outface}} &= 2l(a_s K_{1,0,0}(\epsilon) + b_u^s K_{2,0,0}(\epsilon) + [b_v^s + b_w^s] K_{1,1,0}(\epsilon) - b_w^s K_{1,0,1}(\epsilon)),
\end{aligned} \tag{73}$$

and

$$\begin{aligned}
\psi_v^{\text{outface}} &= \mathbf{m}^{\text{out}}[v \psi(u, v, v)] = \psi_{v,\text{in}}^{\text{outface}} + \psi_{v,\text{src}}^{\text{outface}}, \\
\psi_{v,\text{in}}^{\text{outface}} &= 2(a_f K_{0,1}(\epsilon) + b_u^f K_{1,1}(\epsilon) + b_v^f K_{0,2}(\epsilon)), \\
\psi_{v,\text{src}}^{\text{outface}} &= 2l(a_s K_{0,1,0}(\epsilon) + b_u^s K_{1,1,0}(\epsilon) + [b_v^s + b_w^s] K_{0,2,0}(\epsilon) - b_w^s K_{0,1,1}(\epsilon)).
\end{aligned} \tag{74}$$

Although the source and inflow flux moment coefficients are easily obtained for LC as opposed to EC, it is not obvious from the above moments that the method is numerically cheaper than EC. However, what is not shown above is that the various needed  $K$ -functions are related by recurrences such that only a handful of functions need be calculated. The needed functions are obtained from stable recurrences. See (Mathews, 1998) for details. Additionally, the  $K$ -functions are functions of one variable whereas the moments functions needed for EC have two, three, or four arguments.

This chapter began by introducing the notion of characteristic methods. Specifically, the exponential characteristic method was derived assuming an exponential distribution of the source and inflow flux. In order to solve the EC quadrature for a spatial cell, non-linear root-solving must be done to obtain the unknown source and flux parameters. Following the

discussion of the source and inflow flux, we derived the EC spatial quadrature formulas to calculate the average and first spatial moments of the angular flux for a sub-cell which are obtained by splitting the parent cell. The linear characteristic method was also briefly presented along with its spatial quadrature formulas. Several special functions were defined which package the numerical difficulties of both methods and allow for elegant derivations. With this mathematical foundation, the stage is set to discuss the implementation of the EC and LC quadratures in a radiation transport code system.



### **Chapter III: TETRAN Implementation**

TETRAN was designed to be the kernel of a parallel, general radiation transport code for shielding applications. In this chapter, we first present the programming model used to achieve parallel operation. The algorithm used by TETRAN to solve multigroup, anisotropic scattering problems is then discussed. Finally, the pseudo-code for TETRAN is presented to familiarize the reader with the code implementation.

#### **Parallel Programming Model**

The primary objective of performing radiation transport using multiple processors is to analyze large problems in a reasonable period of time. By partitioning the problem space among many processors operating on their own local data and reporting their results as needed to the other processors, we hope to reduce the wall clock time for getting accurate results for the problem. We believe that this is the true measure of quadrature performance. Currently, researchers compare method performance by citing the time to calculate the angular flux values for a phase space cell (results for one spatial cell in one streaming direction for one energy group and iteration). Although computational efficiency is an important attribute of a spatial quadrature, it is not the only criteria by which to judge a method. Using phase space cell performance criteria focuses on the computational efficiency of a quadrature without regard to the methods accuracy. By implementing parallel algorithms, we can ultimately shift the focus of quadrature performance to a

more balanced view of efficiency and accuracy. Thus, the issues of computational cost of relatively expensive spatial quadratures, like EC and LC, will become moot as they are incorporated into parallel algorithms.

The standard von Neumann iteration on the scattering source (Lewis and Miller, 1993) can be thought of as an *embarrassingly parallel* algorithm. In this algorithm, we sweep through the mesh for each streaming direction,  $\hat{\Omega}_n$ , calculating the needed flux moments ( $\psi_A, \psi_u, \psi_v$ , and  $\psi_w$  in the case of EC and LC) for each cell in the mesh. After all of the streaming directions have been computed, numerical quadrature is used to calculate the scalar flux,  $\phi$ , for each cell and check for convergence of this value as compared to the last iteration. If the flux is not converged, the scattering source is updated using the matrix multiplication,  $S_{i,n}^{A,U,V,W} = \sum_{n'} \psi_{i,n'}^{A,U,V,W} \bar{\bar{T}}_{n',n,i}$ , where the subscripts  $i$  and  $n$  refer to the cell and streaming direction, respectively, and  $\bar{\bar{T}}$  is a material (i.e. cell) dependent matrix.  $\bar{\bar{T}}$  maps all of the group to group and angle to angle particle transfers that can occur and is discussed in more detail below. Thus, the algorithm only requires communication between processors to calculate the flux and update the source after an iteration. These reduction operations (dot products for the numerical quadrature and matrix multiplication for the source) require expensive global communications so that all of the processors can report their data to each other. Unfortunately, inter-processor communication affects the parallel

performance of an algorithm because the communication time is generally orders of magnitude slower (Koelbel, 1994) than the compute speed of a processor. Fortunately, by pursuing this strategy, we make parallel the expensive computations required to get the angular flux moments during the mesh sweep. If the scalar flux and source reductions are done efficiently, the algorithm should scale with the number of processors attacking the problem up to the point where inter-processor communications dominates the cost of the problem.

In order to implement the above algorithm, we chose to use High Performance Fortran (HPF). In HPF, data distribution directives (!HPF\$ ALIGN and !HPF\$ DISTRIBUTE) are used at a high level of abstraction to partition data among processors. Additional directives are used to assert the independence of loop iterations (!HPF\$ INDEPENDENT), thus partitioning computation among processors as well. This *coarse grain* approach to parallelism shifts the work in producing parallel code from the programmer to the compiler and allows for highly portable code. In contrast to the HPF approach to parallelism, a *fine grain* approach requires the programmer to code all of the communication and data distribution requirements using either the message passing interface (MPI) (Snir, 1996) or parallel virtual machine (PVM) (Geist, 1994) library calls. The fine grain approach typically produces machine code that is faster than its HPF equivalent given the maturity level of current HPF compilers. However, fine grain codes also take

much longer to write than equivalent HPF code and are generally tuned to specific hardware architectures. For example, using HPF to parallelize TETRAN took only 23 total HPF directives. This was possible because TETRAN was written with strict adherence to the Fortran 90/95 standard. HPF can be considered a parallel version of Fortran 90. Thus, by using Fortran 90 we were quickly able to write a parallel radiation transport code. Additional discussion of the HPF language can be found in *The High Performance Fortran Handbook* (Koelbel, 1994).

### **Multigroup Implementation**

The multigroup implementation of TETRAN is a variation of the standard approach that is readily parallelized. The BTE is discretized in energy, denoted by the  $g$  subscript:

$$[\hat{\Omega}_n \cdot \bar{\nabla} + \sigma_{i,g}] \psi_{i,n,g} = S_{i,n,g}, \quad (75)$$

where  $i$  is the cell index,  $n$  is the streaming direction index,  $g$  is the energy group index, and  $\sigma$  is the total cross section. The energy groups are arranged in order of decreasing energy such that the highest energy group is 1. The solution strategy is as follows. First, group 1 is converged by using the standard von Neumann iteration on the scattering source (Lewis, 1993). This is the within-group problem and the iterations are termed inner iterations. Having obtained the scalar flux for this group and converged angular fluxes, the down scatter contribution of group 1 to the lower energy groups (2

through  $G$ ) is calculated using the scattering tensor  $\overline{\overline{T}}$  (to be discussed below). The algorithm then proceeds to group 2 and repeats the process, where group 2 now has a group 1 contribution to its source particles. This process is repeated until all of the groups have been converged. In the case where fission is present, there is a possibility that lower energy groups will be a source of *upscatter* particles. This is repeated until the group fluxes converge and is known as the outer iteration. However, fission sources and upscatter are not supported in TETRAN making the outer iteration unnecessary. A block diagram describing this process is shown in Figure 8 and more detail regarding the implementation of this algorithm is presented later in this chapter.

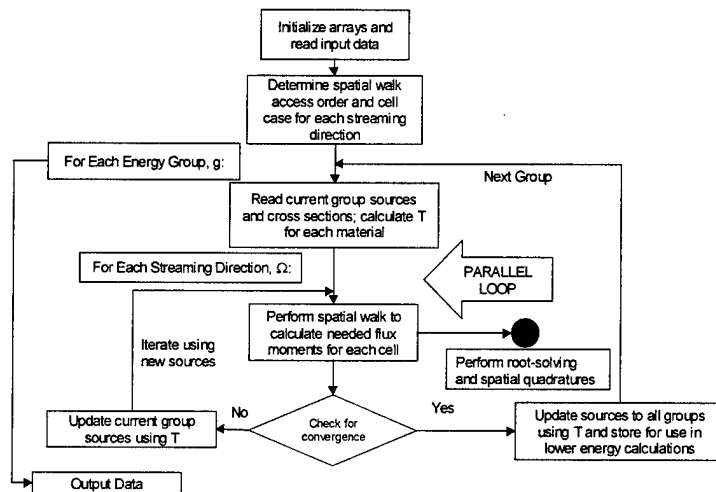


Figure 8. Block diagram for iteration on the scattering source algorithm.

## Anisotropic Scattering

The traditional approach for modeling particle scatter is the spherical harmonics approach, which uses Legendre moments of the scattering cross

sections to model the angular dependence of particle scatter. The accurate treatment of scattering is very important to discrete ordinate calculations because the BTE is solved by decomposing the computational domain in  $\hat{\Omega}$  (equation (2)) and the scalar fluxes are calculated using numerical quadrature. If we use a poor model or bad data, the angularly dependent scattering sources will be wrong and the calculation will converge to the wrong solution. The spherical harmonic expansion of the scattering source is

$$S_g(\vec{r}, \hat{\Omega}_n) = \sum_{l=0}^L \sum_{m=-l}^l Y_{lm}^*(\hat{\Omega}_n) \sum_{g'=1}^G \sigma_{l g g'}(\vec{r}) \phi_{lg'}^m(\vec{r}), \quad (76)$$

where  $S$  is the scattering source,  $Y_{lm}^*$  is the conjugate spherical harmonic,  $\sigma_{l g g'}$  are the Legendre moments of the scattering cross section from group  $g'$  to group  $g$ , and  $\phi_{lg'}^m$  are the coefficients of a spherical harmonic expansion of the group flux,

$$\phi_{lg'}^m(\vec{r}) = \sum_{n'=1}^N w(\hat{\Omega}_{n'}) Y_{lm}(\hat{\Omega}_{n'}) \psi_g(\vec{r}, \hat{\Omega}_{n'}), \quad (77)$$

where the quadrature approximation in (2) is used to calculate the moments. Substituting (77) into (76) and rearrange the summations, one arrives at the following expression,

$$S_g(\vec{r}, \hat{\Omega}_n) = \sum_{n'} \sum_{g'} \psi_{g'}(\vec{r}, \hat{\Omega}_{n'}) T_{g' \rightarrow g}(\hat{\Omega}_{n'} \rightarrow \hat{\Omega}_n, \vec{r}), \quad (78)$$

where the elements of the matrix  $T_{g' \rightarrow g}$  are

$$T_{g' \rightarrow g}(\hat{\Omega}_{n'} \rightarrow \hat{\Omega}_n, \vec{r}) = w(\hat{\Omega}_{n'}) \sum_{l=0}^L \sigma_{sl, g' \rightarrow g}(\vec{r}) \sum_{m=-l}^l Y_{lm}^*(\hat{\Omega}_n) Y_{lm}(\hat{\Omega}_{n'}). \quad (79)$$

Although  $\bar{\bar{T}}$  is conceptually a tensor that maps all particle energy and direction transfers in the momentum space, it is used as a matrix in the following manner. Each material in the problem has a unique  $T_{n' \rightarrow n, g' \rightarrow g, mat}$ , where the *mat* subscript indicates the material index where the total number of materials is *nmat*. During the source update for the within-group problem,  $T_{n' \rightarrow n, g' \rightarrow g, mat}$  is accessed based on the material number of the *i*<sup>th</sup> cell for the current group,  $T_{n' \rightarrow n, g' \rightarrow g, mat(i)}$ . This representation is efficient because it requires that only *nmat* different  $\bar{\bar{T}}$  matrices be stored in memory during the calculation. Thus, it is economical to replicate  $\bar{\bar{T}}$  across processors for a parallel problem, further enhancing data locality. Additionally, because material regions tend to be represented contiguously in the mesh,  $T_{n' \rightarrow n, g' \rightarrow g, mat(i)}$  is used for several cells before a new matrix is needed because of a change in material. This reduces cache misses (non-contiguous memory access) associated with reading in a new  $T_{n' \rightarrow n, g' \rightarrow g, mat(i)}$ .

After the within-group problem has converged, the resulting converged flux moments ( $\psi_A$ ,  $\psi_u$ ,  $\psi_v$ , and  $\psi_w$ ) are used to update the downscatter source for the lower groups. This is accomplished as follows. The

$T_{n' \rightarrow n, g' \rightarrow g+1, mat(i)}$  are calculated using (79). The source is calculated via (78) and added to contributions from other groups (accounting for the summation in (76)). Note also that the source terms are calculated and stored on the processors which access them thereby partitioning the problem in memory among several processors. Thus, we are trading interprocessor communications for storage of large problems. The above process is done for  $g = g + 1$  to  $g = G$ . Then the within-group problem is solved for the next group, and so on.

The above approach is not normally used since it entails storing the angular flux moments for every cell, which is expensive. For example, an  $S_8$  level-symmetric quadrature contains 80 quadrature directions which requires (for LC and EC) storing 80  $\psi_A$ ,  $\psi_U$ ,  $\psi_V$ , and  $\psi_W$  moments (320 total) for each cell to calculate the scattering source moments. However, this algorithm is obviously parallel when the problem is decomposed along quadrature directions ( $\hat{\Omega}$ ). This methodology trades the benefit of decomposing the problem into several sub-problems running independently, each on its own processor (in its own memory space) against the inter-processor communication required to update the source via the global matrix multiplication in (78). For computationally intensive spatial quadratures (such as LC and EC), we anticipate that this trade off should result in a scalable algorithm if the communications required for the matrix multiplications are handled efficiently.



## TETTRAN Data Flow

TETTRAN requires geometric, nuclear, and boundary condition data in order to perform transport. This section describes where the data comes from and how it is assembled for TETTRAN. Figure 9 below shows the general flow of data for a TETTRAN run.

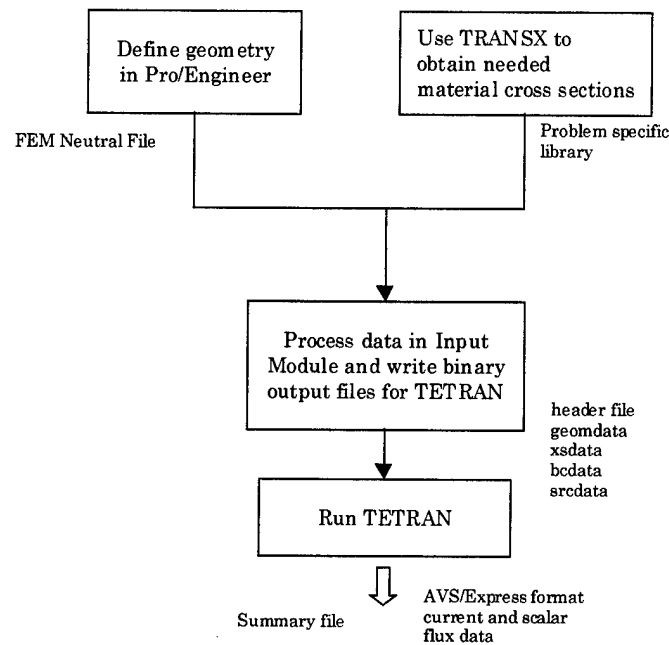


Figure 9. Data flow for TETTRAN.

## Pro/Engineer

Parametric Technology Corporation's Pro/Engineer ® 18.0 (Pro/E) was used to define the various geometries used in this research. Pro/E is a general, high-end computer aided design and manufacturing (CAD/CAM) system, which holds a large fraction of the CAD/CAM market in the United States. Many national and military laboratories as well as defense and

energy contractors use it. Pro/E excels in modeling large, complex assemblies quickly and has an integrated unstructured tetrahedron cell mesh generator for use in structural and thermal finite element analysis.

Pro/E was used to perform several tasks. First, it was used to develop the individual parts that are placed in a larger assembly. At the assembly level, the Pro/Mesh® package was used to add boundary conditions to each assembly surface. Additionally, Pro/Mesh® was used to add mesh control to the assembly so that we could perform mesh refinement and control how individual parts were meshed. After meshing, we analyzed statistics regarding cell aspect ratios and attempt to improve the mesh. Pro/Mesh® allows for mesh file output to many finite element packages, such as NASTRAN (Brennan, 1996). It also has a generic format, the FEM Neutral format (PTC, 1997), which is a generic finite element mesh file giving all of the data needed by any finite element code. We generated version 1 FEM Neutral files to be read by the TETTRAN input module.

### **Cross Section Data Format**

The TETTRAN input module uses TRANSX 2.15, a Los Alamos National Laboratory cross section processing code, to generate the problem specific data libraries for TETTRAN (MacFarlane, 1992). TRANSX is a very powerful cross section processing code that can extract group-weighted, self-shielded cross sections (total and Legendre scattering moments) from a number of different libraries. We used it for TETTRAN to mix cross sections, collapse

energy groups, and generate transport corrected scattering moments in a standard way. The output file is in the standard 6E12.5 format, i.e., six columns of data per row, each with a format of E12.5. Using TRANSX allows us to access and process a large number of standard cross section libraries in a well documented, straightforward manner.

### **Input Module**

The data for TETRAN come from a variety of sources. In order to reduce the complexity of TETRAN and remove input data processing from the transport code, we developed an input module to read the needed data and put it in a consistent format for TETRAN. Thus, support for additional data file formats can be added by revising or replacing the input module. Briefly, it performs the following functions. For the geometry data, it reads the FEM Neutral format file and determines the number of materials and boundary conditions. It then determines the cell connection topology, which is an important (but slow) step whereby the neighbor cells and faces for each cell in the mesh is determined. The connection topology is independent of the angular or spatial quadrature chosen for the transport calculation. It is needed in the transport code to determine the cell access order for each spatial walk, which is dependent on the angular quadrature. Following the generation of the connection topology, the geometry data is written to a binary file to be read by TETRAN. Following the geometry data, the module reads the problem dependent cross section library created by TRANSX and

writes a binary file for input to TETRAN. The input module then reads any user specified multi-group isotropic source files and writes this data to a binary file. Lastly, for each boundary condition specified in the mesh file, the analyst inputs the type of boundary condition (vacuum or incident current) and the file containing multigroup incident current data. This data is written to a binary file as well. In addition to the binary problem data, a header file is written which is a text file that tells TETRAN several key parameters for the run such as the desired spatial and angular quadratures. These parameters are changeable by the user to allow for transport with different angular and spatial quadratures for the same mesh (including the connection topology), boundary conditions, and nuclear data.

TETRAN reads the above data and performs the transport. Following a successful run, the code produces three output text files. The first file is a run summary that contains (among other things) the region average scalar fluxes and boundary currents for the problem. The others contain multigroup scalar flux and vector current data for each cell in the mesh. This data, which is in AVS/Express UCD (Unstructured Cell Data) format (AVS, 1996), is suitable for visualization with AVS/Express (a scientific visualization system).

## Overview of TETRAN

TETRAN was written in ANSI standard Fortran 90, which has been the standard for the Fortran language since 1991. Additionally, in some areas of the code, we have used Fortran 95 syntax (such as the *Pure* keyword) to aid

in the parallelization of TETRAN. We chose to adhere to the Fortran90/95 standard for at least three reasons. First, Fortran 90/95 contains many intrinsic functions such as *DOT\_PRODUCT* and *MATMUL* (general matrix multiplication) that allowed very readable and compact code to be written, aiding in the long-term maintenance of TETRAN. Second, Fortran 90/95 has adopted a superior model for global data storage based upon modules, allowing global data to be passed throughout the code without using the traditional Fortran 77 COMMON block (which was prone to problems). Thirdly, using Fortran 90/95 allowed the quick addition of High Performance Fortran (HPF) directives in order to parallelize the code (Koelbel, 1994). Other reasons include Fortran 90/95's treatment of whole and masked array operations and use of explicit interfaces for error checking. We found the book, *FORTTRAN 90/95 Explained* (Metcalf, 1996) and Digital Electronic Corporation's (DEC) Visual Fortran 5.0 invaluable in implementing TETRAN to the Fortran 90 /95 standard.

TETRAN contains approximately 9400 lines of code in 87 subroutines. The code is intended to be the kernel for a much more capable transport code system based on the linear and exponential characteristic methods for unstructured tetrahedron cell meshes. This version is capable of solving the time-independent BTE using the standard source iteration technique. It is capable of solving multigroup (energy) problems assuming downscatter only (no fission) with general anisotropic scattering based on the spherical

harmonics approach (discussed earlier). Convergence acceleration, a technique that enables the source iteration algorithm to converge more rapidly, is not included in this version. Additionally, High Performance Fortran (HPF) directives are included for a parallel version of the code. The parallelization strategy uses domain decomposition in the angular ( $\hat{\Omega}$ ) dimension. Limited testing was performed on the parallel version of the code due to compiler limitations. The code was developed on a Hewlett Packard Vectra XU 2/200 Dual Pentium Pro 200 MHz workstation using Digital Visual Fortran, Version 5.0C. The code was then ported to the Aeronautical System Center (ASC) Major Shared Resource Center (MSRC) IBM SP (ASC, 1998). We chose to target the IBM SP because of its large per node memory (1 GB), fast inter-processor connection switch, and large amount of per node scratch disk (2 GB). It compiled and ran serially under AIX 4.2 and the xlf90 compiler, version 5, with minor modifications. The HPF version of the code was compiled using The Portland Groups' PGHPF 2.4 development compiler. All of the data presented in this report was produced on one or more nodes of the ASC IBM SP.

### **TETTRAN Pseudo-Code**

As a concise and precise exposition of the structure of the TETTRAN program, we present the following pseudo-code outline. Note that HPF directives/constructs are denoted by *!HPF\$*.

Begin

Read runtime data in problem data file: names of binary input files, spatial quadrature, angular quadrature, etc.

Read binary data files.

Distribute quadrature directions on to processor grid

!HPF\$ distribute (\*,block) :: streaming directions  $\hat{\Omega}_n$   
(note: all mesh and nuclear data (cross sections) are replicated across processors)

Align transport arrays (fluxes and sources) to  $\hat{\Omega}_n$  (ensures other computational arrays are on the same processor to avoid unnecessary communication)

!HPF\$ align (\*,:) with  $\hat{\Omega}_n$  (\*,:) ::  
 $\psi_A, \psi_u, \psi_v, \psi_w, S_A, S_u, S_v, S_w$   
(each variable partitioned among processors along columns of the array.)

!HPF\$ Independent: For each quadrature direction,  $\hat{\Omega}_n$ , determine the cell access order and case for the spatial walk.

For each of the G groups in the problem (g= 1, 2, ..., G; downscatter only):

Read current group cross sections and generate within group T-matrix for each material.

!HPF\$ Extrinsic (HPF\_LOCAL): Read group source for each cell and direction. (an extrinsic subroutine is required to keep the source data i/o local to the processor)

Read current group boundary data for each boundary cell and direction.

Do until converged (Inner Iteration):

Set all cell angular flux moments to zero.

!HPF\$ Independent: \*Meshsweep: For each direction,  $\hat{\Omega}_n$ , perform spatial walk. (\*presented below) (the fluxes for each quadrature direction are calculated local to their own processor)

Test Convergence: Calculate scalar fluxes and check relative change from last inner iteration. If not converged, update within group source for each cell and direction using T-matrix for the cell's material. (!HPF\$: this is where the global communications occur using dot products and matrix multiplications. Note that we let the compiler determine the communication patterns for the global communications.)

```

         $\phi(1:n_{\text{cells}}) = \text{Matmul}[\psi(1:n_{\text{cells}}, 1:n_{\text{ang}}), \text{weight}(1:n_{\text{ang}})]$ 

        For i=1 to ncells
             $S_{A,u,v,w}(i, 1:n_{\text{ang}}) = \text{Matmul}[\psi_{A,u,v,w}(i, 1:n_{\text{ang}}), T(1:n_{\text{ang}}, 1:n_{\text{ang}}, \text{matid}(i))] +$ 
                 $S_{A,u,v,w}^{\text{Ext}}(i, 1:n_{\text{ang}})$ 

            Next cell

        End do

        Down Scatter Source: Loop over current group downscatter cross
        sections, calculating T-matrices and then the current group
        contribution to lower groups. !HPF$ Extrinsic (HPF_LOCAL): Update
        the source for each lower group and store until needed. (again,
        local routine is used to keep sources local to the processor they
        are used on)

        For groups g'=g+1 to G
            Calculate T-matrices for group g' and materials
            For each cell i
                 $S_{A,u,v,w}(i, 1:n_{\text{ang}}) = \text{Matmul}[\psi_{A,u,v,w}(i, 1:n_{\text{ang}}), T(1:n_{\text{ang}}, 1:n_{\text{ang}}, \text{matid}(i))] +$ 
                     $S_{A,u,v,w}^{\text{Ext}}(i, 1:n_{\text{ang}})$ 

                Next i

                !HPF$ Extrinsic(HPF_Local) Write to local scratch
                disk for later use.

            Next g'

            Write Group Output: Write current group scalar fluxes and
            currents to file for later processing.

        Next Group

        Write Flux, Current, and Summary Data

    End

```

Of course, the above is a top-level view of the overall operation of TETRAN.

The real transport work is done in the *Meshsweep* loop, shown below. Note that all of the routines within *Meshsweep* are *Pure* routines, i.e. they produce no side-effects as is required of *!HPF\$ Independent* loops.



Meshsweep: !HPF\$ Independent: Do sequentially or in parallel for each quadrature direction,  $\hat{\Omega}_n$ .

Set face flux moments to zero and initialize boundary flux values.

For each cell (in the walk sequence for this direction):

Get cell index and determine cell parameters (case, source, etc.)

Determine cell split parameters.

Get source parameters for cell:

LC, direct calculation; EC, generate initial estimates then root solve

Determine cell upstream neighbor faces and calculate transformation matrix to get inflow flux moments.

For each sub-cell:

Get inflow flux coefficients either directly for LC or by root solve for EC. Use direct affine transformation to get coefficients into correct sub-cell coordinate system.

Get source coefficients by direct affine transformation from parent coordinate system.

Perform spatial quadrature using correct sub-cell flux and source coefficients to get  $\psi_A$ ,  $\psi_u$ ,  $\psi_v$ ,  $\psi_w$ ,  $\psi_A^{out}$ ,  $\psi_u^{out}$ , and  $\psi_v^{out}$ .

Check sub-cell conservation.

Next sub-cell

Transform/combine sub-cell flux moments back to parent cell UVW coordinate system.

For each output face, transform/combine outflow flux moments to parent global UV face system.

Next cell in walk

End do

More detail regarding the stable evaluations in *Meshsweep* are shown in

Appendix C.

In this chapter, we discussed our strategy for the implementation of a parallel radiation transport code. Using the spherical harmonics source

equations and the discrete ordinates equation, we developed a new approach to calculate source updates that involves the T-matrix. With this approach to the scattering source updates, the iteration on the scattering source algorithm was implemented using Fortran 90/95 in a parallel-ready way allowing the solution of multigroup, anisotropic scattering problems. High Performance Fortran (HPF) directives were used to decompose the needed transport arrays along their angular dimension, allowing the spatial walks in all streaming directions to be done in parallel. An overview of these algorithms and the role of the spatial quadrature were presented along with the sources of needed input data. With the needed spatial quadratures and algorithms developed, we now move on to Chapter 4 where we test TETRAN's performance on a variety of problems.

## Chapter IV: Testing

In this section, we present the results of our testing of TETTRAN and the exponential characteristic method. Each test was selected to demonstrate important features of TETTRAN's performance. The problems presented here (except the demonstration of parallel code scaling, which was run with 2 to 32 processors) were run on a single node of the ASC IBM SP (known as hpc02) (ASC, 1998). Additionally, the  $S_8$  level-symmetric angular quadrature was used for all problems. The code was compiled using optimizations (-O2; an IBM specific compiler flag) and tuning specific to the processor architecture under the XLF90 version 5 compiler. We compare the scalar flux and exiting current results of each problem with MCNP results using the same input data and geometry except for the coarse mesh, deep penetration problem. Note that the current is presented in units of particles/s through a surface. We do this because MCNP current data is presented in these units. The HPF version of the code was compiled on the ASC IBM SP using The Portland Groups' PGHPF 2.4 development compiler (an alpha version compiler) (Portland Group, 1996). The convergence tolerance was  $10^{-6}$  maximum relative difference between the final and next to last iteration cell scalar fluxes.

Seven problems were chosen to test specific aspects of TETTRAN's performance. Our first problem is used for convergence testing: a uniform cube with a uniformly-embedded isotropic source. Our second problem is

based on problem 1 and is used as the basis for gathering of run-time statistics for TETRAN's operation on the IBM SP. Problems 3, 4, and 5 demonstrate three different aspects of robustness of TETRAN. Problem 3 is a uniform cube of material with a uniformly-embedded isotropic source surrounded by a very thin wall of shield material. This problem demonstrates TETRAN's robust handling of very poorly shaped cells. Problem 4, a spherical source within a tetrahedron, demonstrates TETRAN's ability to perform transport on meshes with curvilinear and inclined plane surfaces and demonstrates a limitation of current mesh generators. Problem 5 demonstrates the thick cell performance of EC as compared to LC. This problem is composed of a uniform cube with a uniformly embedded isotropic source surrounded on three sides by an optically thick shield. Our sixth problem is similar to one found in the literature (Castrianni, 1998), nested cubes of water, iron, and water with a source dissolved in the innermost cube. This problem is used to demonstrate TETRAN's ability to perform multigroup, anisotropic scattering problems. Finally, problem 7 demonstrates scaling of our HPF parallel implementation of TETRAN for the simple cube problem used to gather runtime statistics. Table x summarizes the test problems discussed in this chapter and the code feature stressed by the problem.

Table 2. Summary of Test Problems

Test Problem	Feature Tested
1	Convergence and convergence rate
2	Run-time performance
3	Poorly shaped cell robustness
4	Poorly shaped cell robustness and mesh volume conservation
5	Robustness with optically thick cells
6	Multigroup, anisotropic scattering performance
7	Parallel performance

### Convergence and Convergence Rate

Our first test problem demonstrates the convergence and estimates the convergence rate of the EC and LC methods on simple meshes. The problem is a 1000-cm<sup>3</sup> cube with a volume-distributed isotropically-emitting source of 1.0 particle/(cm<sup>3</sup> - s) and vacuum boundaries. All of the particles are mono-energetic. The total cross section of the cube material is  $\sigma = 1.0 \text{ cm}^{-1}$  with an isotropic scattering cross section,  $\sigma_s = 0.5 \text{ cm}^{-1}$ . TETRAN was run on seven increasingly finer meshes with nearly uniformly shaped tetrahedra. The characteristics of these meshes (number of cells and optical thickness statistics) are shown in Table 3.

The results produced by TETRAN for each mesh are listed in Table 4 below with the corresponding MCNP and MCSN results. The MCNP results are presented to show that the method is converging toward the expected physical solution but are not used to estimate convergence rates because they contain no discretization errors.

Table 3. Mesh Parameters for Convergence Rate Problem.

Mesh	Cells	$\varepsilon_{\min}$	$\bar{\varepsilon}$	$\varepsilon_{\max}$	$\varepsilon_{\max}/\varepsilon_{\min}$
1	6	4.2209	8.67230	17.321	4.1036
2	161	1.2371	2.89800	5.7875	4.6783
3	1292	0.56301	1.44680	3.0510	5.4191
4	4352	0.37680	0.96461	2.0336	5.397
5	10330	0.26177	0.72388	1.5889	6.0698
6	20157	0.20323	0.57922	1.2189	5.9976
7	47794	0.10087	0.43431	1.0050	9.9633

In order to determine the rate of convergence of the methods, Dr. Kirk Mathews developed a Monte Carlo transport code, MCSN, which transports particles along the directions used in discrete ordinates angular quadrature sets. Thus, the angular discretization error of the discrete ordinates angular quadratures is accounted for in MCSN allowing us to perform mesh refinement comparisons with the MCSN benchmark. Because the MCSN benchmark does not contain spatial truncation error (it is continuous in  $R^3$ ), but does contain the angular quadrature error, the relative error between the benchmark and the discrete ordinates solutions for several meshes provides us with our convergence rate graph. Contrast this result with the MCNP result, which does not contain either spatial or angular discretization error because particles are sampled uniformly in both  $R^3$  and  $\hat{\Omega}$ .

Table 4. Results for Problem 1.

Mesh	Scalar Flux [particles/cm <sup>2</sup> - s]	J (x=5.0 cm) [particles/ s]
1	1.511537	40.92211
2	1.540445	38.31644
3	1.547338	37.73447
4	1.549319	37.54896
5	1.549909	37.51102
6	1.550281	37.47919
7	1.550566	37.45456
MCSN	1.55091 (±0.00001)	37.42417 (by conservation)
MCNP	1.554800 (±0.000311)	36.99300 (±0.05179)

Table 4 shows the scalar flux in cube and the current out of the +x face at  $x = 5.0\text{ cm}$ . Clearly, both the flux and current are converging toward a solution that is close to the MCSN solution.

To estimate the order of convergence for the EC method, we use the analog to the rectangular cell case, assuming that average linear cell dimension,  $\Delta x$ , is approximated by  $\bar{\epsilon}$ . As the mesh is refined,  $\bar{\epsilon}$  decreases by a factor  $n = \bar{\epsilon}_1 / \bar{\epsilon}_2$ , where  $\bar{\epsilon}_1 > \bar{\epsilon}_2$ . In the thin cell limit, the mesh refinement should result in a reduction in the solution error by

$$n^p = \frac{e_{r,1}}{e_{r,2}}, \quad (80)$$

where  $e_{r,1}$  and  $e_{r,2}$  are the solution errors relative to a benchmark result for the two meshes and  $p$  is the order of convergence. The absolute relative error for the scalar flux,  $\phi$ , is defined as

$$e_{r,i} = \frac{|\Delta\phi_i|}{\phi_i} = \frac{|\phi_{\text{bench}} - \phi_i|}{\phi_{\text{bench}}}, \quad (81)$$

where  $\phi_{\text{bench}}$  is the scalar flux predicted by MCSN and  $\phi_i$  is the average scalar flux in the cube predicted by TETRAN for the  $i^{\text{th}}$  mesh. The relative errors for both the EC and LC quadratures are plotted in Figure 10 and Figure 11 below versus the average cell optical thickness,  $\bar{\epsilon}$ .

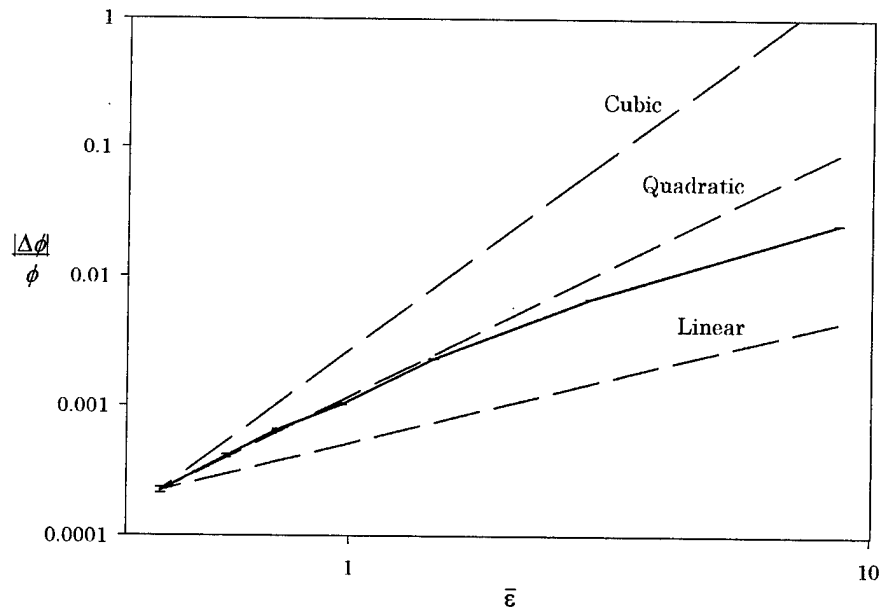


Figure 10. Convergence Graph for Exponential Characteristic Method.



Figure 10 shows that the EC method is a stable coarse mesh performer and that it appears to be at least quadratically convergent as compared to the fan of lines that represent ideal linear, quadratic, and cubic convergence with respect to the benchmark and the finest mesh solution. However, the true convergence rate is obscured because Pro/Mesh® cannot produce self-similar meshes with increasing refinement. The EC and LC methods have been shown to be cubically convergent on such meshes for an optically thin problem (Brennan, 1996). Thus, we conclude that the EC method as implemented in TETRAN is also cubically convergent and the non-uniform discretization is obscuring this behavior.

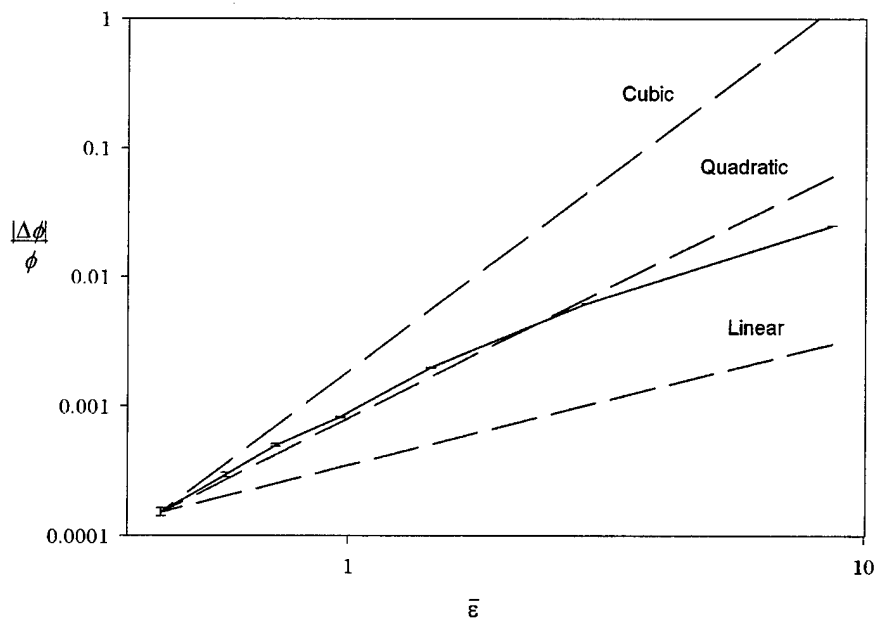


Figure 11. Convergence Graph for the Linear Characteristic Method.

As with EC, the convergence rate for LC is being obscured by the lack of self-similarity in the mesh refinement. Thus, it is clear that future research needs a high quality mesh generator with the ability to refine an unstructured mesh in a self-similar way. The LC results are slightly more accurate than the EC results compared to the benchmark (an effect seen in Brennan's tests as well). This is because this problem has a scattering source and flux that is concave down throughout the cube. The EC method is attempting to model this behavior with an exponential that is concave up, which introduces more approximation error than the LC method, which uses a linear function. Thus, it should be expected that EC will perform less accurately than LC for this problem. EC's coarse mesh capability is demonstrated in a later test problem.

### **Run-time Profiling**

We performed two levels of timing for TETRAN. At the top level of the code, we timed the length of time spent performing input and output, the spatial walk, calculating the scalar flux, testing convergence, and updating the scattering source. The performance figures are for a 162-cell mesh with an  $S_8$  angular quadrature (80 directions) which took 19 iterations to converge. The problem was run on one node of the ASC IBM SP. The results for EC are listed in Table 5. There is no substantial difference between the EC and LC run-time results at this level of profile.

Table 5. TETRAN Run-Time Characteristics.

Action	% of Time Spent During Execution of TETRAN using EC Spatial Quadrature
Read Input Data	0.72%
Generate Walks Lists	0.29%
Spatial Walks	96.51%
Calculate Scalar Flux, Test Convergence, and Update Source	0.65%
Calculate Currents and Output Data	1.39%
Miscellaneous: initialize arrays, allocation, de- allocation, other i/o	0.45%

In addition to the above, we compiled quadrature specific data on the cost to perform transport using EC and LC within TETRAN. This data is presented in Table 6.

The IBM *prof* profiler (IBM, 1993) was used to determine that TETRAN spends about 45% of its execution time evaluating exponentials and moment functions using EC; whereas, the LC method spends a negligible fraction of its execution time evaluating  $K$  functions. The above performance values (Table 5) are for a completely unoptimized code. This is because the design philosophy behind TETRAN's development precluded the use of numerical approximations. Thus, TETRAN serves as a clean baseline for performance enhancements such as using the conservation equations in the spatial quadrature. Currently, the conservation equations are evaluated to check the accuracy of the spatial quadrature and are not used in the transport calculation (we considered confidence in the results more important

at this stage than optimization). Additionally, Fortran 90 compilers are not yet as effective at optimization as Fortran 77 compilers.

Table 6. TETRAN Spatial Quadrature Performance.

<b>Action</b>	<b>Average Time/Phase Space Cell (<math>\mu</math>s/cell-angle-group-iteration) (% of total)</b>	
	<b>EC</b>	<b>LC</b>
Split cell	95 (3%)	95 (10%)
Obtain source coefficients	537 (18%)	34 (4%)
Obtain face coefficients and translate/rotate into sub-cell	822 (27%)	302 (32%)
Spatial quadrature	1099 (36%)	43 (5%)
Re-combine sub-cell moments into parent cell	200 (7%)	200 (21%)
Overhead: cache misses, page faults, resetting face arrays, timing, etc.	260 (9%)	260 (28%)
<b>Total</b>	<b>3005</b>	<b>934</b>

The performance data in Table 6 indicate that EC and LC are moderately expensive spatial quadratures. However, note that in the previous effort (Brennan, 1996), the LC method was approximately 10 times faster than EC for the same mesh. We have reduced this by a factor of three with anisotropic scattering included. Comparing the performance of TETRAN with other codes at this time is not appropriate. TETRAN is not optimized at all. The goal of this research was to develop EC and LC to be robust and

accurate for coarse mesh problems. Brennan pointed out (Brennan, 1996) that the EC method pays for itself when accurate solutions are required on coarse mesh, deep penetration problems. Additionally, our philosophy has always been to prize accuracy, consistency, and robustness over computational speed since today's high performance computing hardware is tomorrow's antique.

TETTRAN, like all unstructured mesh transport codes, suffers from poor cache performance with regard to current high-performance computer (HPC) architectures. Currently, most HPC architectures employ a hierarchy of memory access based on caches. For example, one node of an IBM SP operates with a 135 MHz Power2 Super Chip which has a 128 Kbyte data cache, 500 256-byte cache lines, and 256 translation look-aside buffers (TLB). The TLB is used to translate between the virtual storage address of a page of memory (4096 bytes) and its current real storage address. Data is accessed through the cache lines. There is also an instruction cache, which is 32 Kbytes in size. If the needed data is not located in one of the current cache lines, the data from the oldest cache line is flushed, a cache miss (the data was not in cache) occurs and a new cache line containing the needed data is read in, taking 8 to 12 clock cycles. If the current data is not in the current TLB, then the processor checks through all of its TLB to see if it is there (1Mbyte of data). If it is not, a page fault occurs and the processor must use the operating system to go to real memory and access the data, costing 36 to

56 clock cycles. If the data is stored in arrays of unit stride and accessed in this fashion, then cache misses will occur when the data for a cache line is exhausted and another is needed. This is the typical access pattern for a structured Cartesian-mesh transport code. For an unstructured code, such as TETRAN, this is a performance nightmare. Data is accessed indirectly for each cell and face. Thus, there is little hope that even a small to moderate size problem will use more than one value from a cache line or possibly a page. Thus, the code expends a significant amount of cycle time looking through memory versus performing calculations. We found both the "P2SC Overview" (Chin, 1996) and *High Performance Computing* (Dowd, 1998) very helpful in aiding our understanding of modern computer architectures and the IBM Power2 implementation. Regardless, this issue will be significant for unstructured mesh transport for many years to come.

### **Cell Aspect Ratio Robustness**

We now examine TETRAN's robustness when presented with a poorly conditioned mesh with many large aspect ratio cells, where the aspect ratio is defined as the ratio of the longest cell edge to the shortest distance of a cell vertex to it's opposite face. These parameters are depicted in Figure 12.

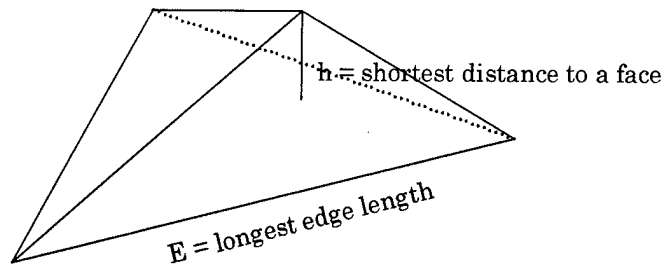


Figure 12. Cell aspect ratio parameters.

To produce a mesh with poor aspect ratios, we used a solid cube of material ( $1 \text{ cm}^3$ ) surrounded on all sides by a thin layer of  $0.01 \text{ cm}$  thickness. Two examples of the resultant meshes are shown in Figure 13.

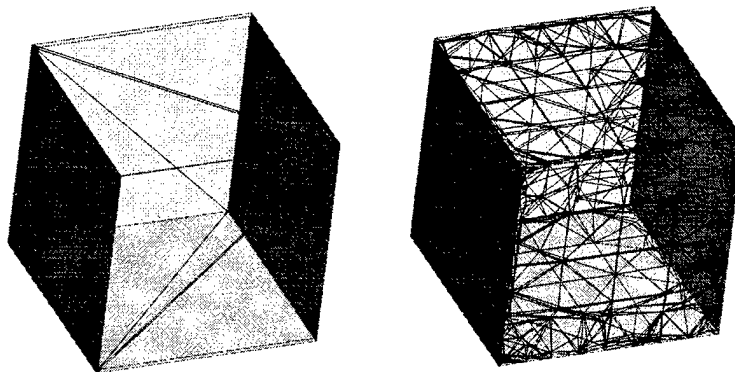


Figure 13. Meshes for Robustness Test.

This problem is chosen since it serves as the basis of a more complex assembly of such thin walled objects. It is also easy to model in MCNP so that we can use the Monte Carlo solution as a reference value. Lastly, the model is representative in proportion to the thin walls that typically surround a

satellite (100 mils about a 1 ft<sup>3</sup> box -> 0.01 cm shield about 1 cm<sup>3</sup> box)

(Hilland, 1998), thus implying military as well as civilian applicability.

The test problem for this demonstration is a 1 cm x 1 cm x 1 cm cube with a constant isotropic source (monoenergetic) of 1.0 particle/s. It has a total cross section of  $\sigma = 1.0 \text{ cm}^{-1}$  and a scattering cross section of  $\sigma_s = 0.5 \text{ cm}^{-1}$ . The shield region is 0.01 cm thick and surrounds the cube. The shield has a total cross section of  $\sigma = 2.0 \text{ cm}^{-1}$  and a scattering cross section of  $\sigma_s = 1.0 \text{ cm}^{-1}$ .

This test case was designed to demonstrate the robustness of both the EC and LC methods. The maximum cell aspect ratio for the coarse mesh problem is 144: the aspect ratio for 80% of the cells in this mesh. The fine mesh has cell aspect ratios distributed as shown in Figure 14. Clearly, refining the mesh reduces the occurrence of large aspect ratio cells. Note that the aspect ratio for an equilateral tetrahedron is 1.2247 indicating that none of the meshes for this problem contained ideally shaped cells. The meshes for this problem (Figure 13) also exhibit a problem with current mesh generators. Notice in the fine mesh on the right that the corners are meshed preferentially versus the interior of the faces. This is presumably because the finite element mesh generator is attempting to produce a good mesh for a thermal or structural response code. We do not need this type of mesh for particle transport. Instead, we need a uniform mesh of well-shaped



tetrahedra. This is an instance where a mesh generator optimized for radiation transport would be useful.

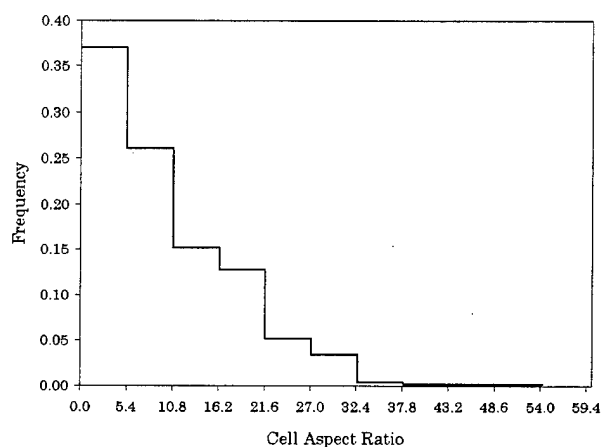


Figure 14. Distribution of Cell Aspect Ratios for Fine Mesh.

The coarse mesh has the minimum number of cells that Pro/Mesh would use for this problem. A finer mesh than the 4902-cell mesh presented could not be generated, because the number of cells required grows enormously with increasing mesh refinement. The cell optical thicknesses for the two meshes analyzed are presented in Table 7. Additionally,  $\epsilon_{\max}/\epsilon_{\min}$  is shown because it reflects the capability of the mesh generator to produce uniform meshes.

Table 7. Cell Optical Thicknesses for Cell Aspect Ratio Problem.

<u>Region</u>	<u>N<sub>cells</sub></u>	<u><math>\epsilon_{\min}</math></u>	<u><math>\bar{\epsilon}</math></u>	<u><math>\epsilon_{\max}</math></u>	<u><math>\epsilon_{\max} / \epsilon_{\min}</math></u>
Source	6	0.42209	0.86392	1.7321	4.10
	1671	0.00108	0.07965	0.51769	479
Shield	36	0.01026	0.02651	0.17321	16.9
	3231	0.00264	0.03736	0.09165	34.7

Cells with large aspect ratios present difficult problems numerically for EC because the source and inflow flux coefficients for these cells are generally large, negative, and close together. This behavior can cause overflows if not handled properly (see Appendix C). Badly shaped cells also cause problems with the cell Jacobian evaluation and rotation matrices. Thus, this test case to a lesser extent shows the robustness of the LC method as well. The successful evaluation of this problem verifies that we have achieved a level of robustness that was not available in the earlier work (Brennan, 1996). Note that this problem does not test the coarse mesh ability of either EC or LC as evidenced by the relatively small optical thicknesses in Table 7.

We demonstrate the performance of the method by comparing the LC and EC results with those obtained from MCNP. This data is shown in Table 8. This data shows that the LC and EC methods are converging toward a discrete ordinates solution that is generally in agreement with the MCNP prediction. Thus, TETRAN provides accurate results even for poor meshes.

Table 8. Results for Large Cell Aspect Ratio Problem.

Method	$N_{\text{cells}}$ (Shield)	$N_{\text{cells}}$ (Cube)	$\Phi_{\text{Shield}}$ [1/(cm <sup>2</sup> -s)]	$\Phi_{\text{Cube}}$ [1/(cm <sup>2</sup> -s)]	$J^+$ (#/s)
MCNP			0.231440 ( $\pm 0.000069$ )	0.403520 ( $\pm 0.000081$ )	0.130500 ( $\pm 0.000104$ )
42 Cells	36	6			
LC			0.239455	0.385757	0.131714
EC			0.240027	0.384161	0.131925
192 Cells	144	48			
LC			0.234940	0.398565	0.131053
EC			0.235534	0.397051	0.131294
4902 Cells	3231	1671			
LC			0.234192	0.400116	0.130974
EC			0.234290	0.399896	0.130997

This problem demonstrated the difficulty that unstructured tetrahedral mesh generators can have with thin regions in a mesh. Clearly, building robust algorithms to handle poor meshes is a way to overcome this problem. Another approach, which was initially pursued in this research, is to develop a spatial quadrature that treats thin regions in a problem with a new spatial quadrature. This approach is discussed in Appendix D and is called the surface cell algorithm. In this algorithm, the thin regions are collapsed to a surface that are treated as computational cells with their own spatial quadrature. Thus, a mesh would have both surface cells and tetrahedra cells: a mixed mesh. This research was abandoned because there were no mesh generators available that produced the needed mesh. Should such a mesh generator become available it would enable the implementation of this approach.

## Curvilinear and Inclined-Plane Surfaces

The problem presented in this section demonstrates TETTRAN's ability to perform transport on complex, multi-region geometries with curvilinear and inclined-plane surfaces. This problem demonstrates the robustness of the algorithms and an issue regarding mesh generation of sources in a complex configuration. We initially believed Brennan's development able to handle such problems. However, it was found that the code was generally not robust enough to handle the few poorly shaped tetrahedra that are produced in such a configuration. Consequently, Brennan's code typically failed with overflow errors (see Appendix C) during execution for simple multi-region problems. EC was made robust by using the implementation presented in Chapter 2.

This problem, a spherical source within a tetrahedron, was chosen to demonstrate the three-dimensional capability of TETTRAN. Indeed, only one other discrete ordinates code, ATILLA, is even capable of performing transport on this configuration with such a high degree of fidelity. MCNP is also capable of performing transport on the configuration and was used as the benchmark for this problem. The geometric configuration is as follows. For the tetrahedron, the corners are located in (X,Y,Z) space at the following nodes: node 0 (5.0, 3.0, -3.0), node 1 (0.0, 0.0, 5.15), node 2 (-5.0, 3.0, -3.0), and node 3 (0.0, -5.66025, -3.0). The sphere is centered at the origin and has a radius of 1.336505 cm (a volume of 10.0 cm<sup>3</sup>). The total cross section for both regions is 0.75 cm<sup>-1</sup> and the scattering cross section is 0.5 cm<sup>-1</sup>. These cross

sections were chosen because they were used by Lathrop (Lathrop, 1971) in his square-in-square problem. We envision this problem as a three-dimensional version of the square-in-square problem (actually, a cube-in-cube is the three-dimensional variant but we wanted a more challenging geometry). A monoenergetic, isotropic source of strength  $1.0 \text{ particles}/(\text{cm}^3\text{-s})$  is located within the sphere.

We examined this transport problem using three meshes: a coarse mesh with 138 cells, a finer mesh with 1209 cells, and the finest mesh with 10075 cells. Table 9 lists the region volumes and optical thicknesses for each region and mesh.

It can be seen from the below data that the source volume for each mesh is not being conserved nor is that of the tet. The actual source volume is  $10.0 \text{ cm}^3$ . Although the overall volume of the problem is conserved, if the source volume is incorrect, the discrete ordinates solution will be incorrect because the source will be in error. In addition to not having the correct volume, none of the meshes have the correct source shape (a sphere) although they are approaching it with each mesh refinement. This is shown in the figures below, where we present the coarse and finest meshes.

Table 9. Mesh Optical Thicknesses for Curvilinear and Inclined-Planes Problem.

Region	Volume (cm <sup>3</sup> )	N <sub>cells</sub>	$\epsilon_{\min}$	$\bar{\epsilon}$	$\epsilon_{\max}$	$\epsilon_{\max}/\epsilon_{\min}$
Source	2.4028	8	0.44200	0.76068	1.5635	3.54
	8.4789	186	0.09247	0.41175	0.87301	9.44
	9.4916	938	0.062168	0.24471	0.60285	9.70
	10.0	Actual	N/A	N/A	N/A	N/A
Tet	115.23	130	0.17931	1.0240	2.7929	15.58
	109.15	1023	0.08732	0.50121	2.0958	24.00
	108.14	9137	0.05553	0.26430	0.62608	11.27
	107.6351	Actual	N/A	N/A	N/A	N/A

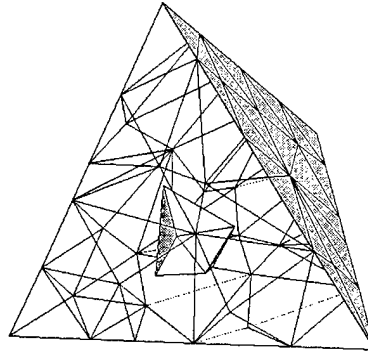


Figure 15. Coarsest Mesh of Sphere in Tetrahedron (138 cells).

Clearly, the mesh in Figure 15 is completely unrealistic for the embedded spherical source. Contrast this with the finest mesh, which is significantly different than the sphere it is modelling with regard to the volumes (9.4916 cm<sup>3</sup> vs. 10.0 cm<sup>3</sup>) (Figure 16).

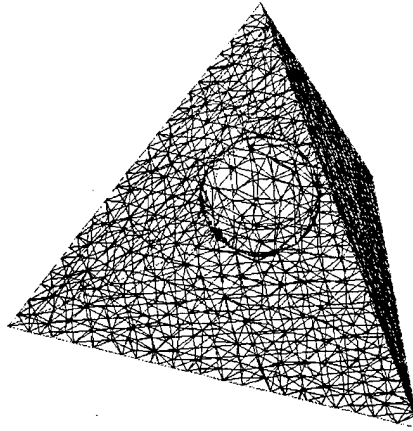


Figure 16. Finest Mesh Sphere in Tetrahedron Mesh (10075 cells).

We cannot rely on the mesh generator to conserve volumes automatically. This feature is critical to nuclear applications because one often deals with volumetric sources. In the case of this problem, even finer meshes are required to get the source right, although we could multiply the region fluxes by a correction factor to account for the error in the source volume in the mesh. Such an approach is plausible for a simple problem but nearly impossible for a complicated source configuration. A volume-conserving mesh generator is needed. The results of the transport calculation (using EC) for each mesh are presented below. Note that each face is numbered according to the node that does not lie on it, e.g. face 0 is comprised of nodes 1, 2, and 3, and so on.

Table 10. Results of Transport Calculation Using EC for Test Problem 2.

$\left(\frac{\text{particles}}{\text{cm}^2 - \text{s}}\right)$			Current Out of Face : (particles/s)			
N <sub>cells</sub>	$\Phi_{\text{Source}}$	$\Phi_{\text{Tet}}$	0	1	2	3
138	0.625647 (2.60382)	0.033598 (0.139830)	0.364600 (1.517396)	0.100795 (0.419490)	0.363604 (1.513250)	0.230081 (0.957552)
1209	0.979378 (1.155076)	0.096365 (0.113653)	1.168384 (1.377990)	0.377978 (0.445786)	1.152992 (1.359837)	1.073805 (1.266444)
10075	1.017173 (1.071656)	0.104814 (0.110429)	1.308251 (1.378325)	0.424641 (0.447386)	1.306480 (1.376459)	1.204846 (1.269381)
MCNP	1.034600 ( $\pm 0.000207$ )	0.108750 ( $\pm 0.000022$ )	1.382700 ( $\pm 0.000968$ )	0.446760 ( $\pm 0.000536$ )	1.383300 ( $\pm 0.000968$ )	1.274700 ( $\pm 0.000892$ )

Notice that if we multiply the scalar fluxes and currents by the source volume ratio (10.0/source volume) , the transport solutions become remarkably more accurate with respect to the Monte Carlo solution. The corrected values, shown in parenthesis, are tabulated in Table 10. Note that correcting the source flux produces a worse estimate of the scalar flux for the coarse mesh than for the other meshes. This is because the source region in the coarse mesh doesn't have the shape of a sphere. Thus, it should be expected that the volume correction will not improve the flux prediction in this case.

Clearly, correcting for the missing source due to the mesh improves the accuracy of the discrete ordinates solution with respect to the MCNP solution. Note, however, that in most complex source configurations it will not be easy to perform such a correction. The better approach for this type of problem is to mesh the problem correctly such that volumes are conserved. This eliminates any need for such corrections and frees us from worrying about such problems.



The above table shows that the source correction works remarkably well for some aspects of the problem. Rather than under-predicting the flux and currents, we over-predict and converge toward the MCNP solution. This behavior is far more appealing for shielding calculations than the under-predicting behavior that occurs when the wrong source is used. This problem shows that work needs to be done in improving mesh generation for applications that need good control over the mesh volumes. Without such an ability, future use of unstructured mesh codes will be limited since the primary benefit of such methods is to model curved geometries. The issue of volume conservation, coupled with the finite element artifacts discussed with respect to the last test, more than drive home the need for a transport specific mesh generator.

With evidence that TETRAN is robust and accurate (given the characteristics of the meshes used) for complicated multi-region problems, we proceed to our next test, a problem that demonstrates TETRAN ability to solve coarse mesh, deep penetration problems.

### **Coarse Mesh with Optically Thick Cells**

Our last test of TETRAN's robustness centers on its ability to solve problems with cells that are optically thick. EC has been shown superior to LC for these types of problems. The LC method, a coarse mesh linear method, is reliably accurate for cell optical thicknesses on the order of 2.0 or less. The difference methods (diamond difference et al.) require much finer meshes. We

demonstrate here EC's performance for cells with optical thicknesses up to 34. Because this problem is a stressing deep penetration problem, we are unable to use MCNP to obtain a bench mark solution to the problem. The thickness of the shield is on the order of 20 mean free paths. Rather than experiment with variance reduction techniques, we use the most converged EC solution as our benchmark for this problem.

This problem is composed of two regions. The inner region is a homogeneous cube (dimensions 10 cm x 10 cm x 10 cm) which contains a homogeneously distributed source of monoenergetic particles emitted isotropically with a strength of 1000 particles/cm<sup>3</sup>. The total and scattering cross sections of the source cube material are  $\sigma = 0.25 \text{ cm}^{-1}$  and  $\sigma_s = 0.0833 \text{ cm}^{-1}$ , respectively. The outer region is a 20 cm x 20 cm x 20 cm cube with  $\sigma = 2.0 \text{ cm}^{-1}$  and  $\sigma = 0.5 \text{ cm}^{-1}$ . The source material is located in one corner of the larger cube, as is shown in Figure 17. We chose cubes to ensure that the volume of the source region was represented exactly.

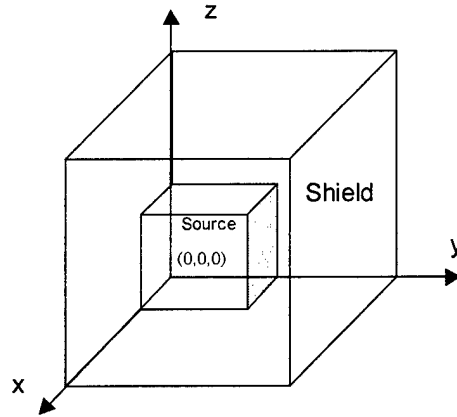


Figure 17. Geometry for Coarse Mesh Problem.

The region scalar fluxes and exiting face currents were calculated for four meshes using EC and LC. The coarsest mesh had 48 cells while the finer meshes had 348, 3063, and 20165 cells, respectively. The mesh optical characteristics are presented in Table 11.

Table 11. Mesh Optical Thicknesses for Coarse Mesh Problem

Region	N <sub>cells in region</sub>	$\epsilon_{\min}$	$\bar{\epsilon}$	$\epsilon_{\max}$	$\epsilon_{\max}/\epsilon_{\min}$
Source:	6	1.0552	2.1763	4.3301	4.10
	48	0.48656	1.0825	2.1663	4.45
	381	0.22113	0.54329	1.0836	4.90
	2047	0.11241	0.30941	0.69405	6.17
Shield:	42	8.0038	17.376	34.641	4.33
	336	3.5912	8.6710	17.321	4.82
	2682	1.7139	4.3362	9.0774	5.30
	18118	0.56092	2.2600	5.1109	9.11

Table 11 shows that the meshes examined contain extremely thick cells in the case of the coarse mesh (up to  $\epsilon \sim 34$ ) to only moderately thick cells in the case of the fine mesh (but still thick enough to reduce LC's accuracy). The

source region is much less optically thick than the shield region. The results for the region average scalar fluxes are listed in Table 12.

Table 12. Region Scalar Fluxes for Coarse Mesh, Deep Penetration Problem

<u>Mesh:</u>	$\phi_{\text{source}}$ $(\frac{\text{particles}}{\text{cm}^2 - \text{s}})$		$\phi_{\text{shield}}$ $(\frac{\text{particles}}{\text{cm}^2 - \text{s}})$	
	<u>EC</u>	<u>LC</u>	<u>EC</u>	<u>LC</u>
coarse (48 cells)	2.8239E+03	2.8000E+03	2.3757E+01	2.4895E+01
less coarse (384 cells)	2.8993E+03	2.8816E+03	2.3167E+01	2.3709E+01
finer (3063 cells)	2.9185E+03	2.9101E+03	2.3041E+01	2.3322E+01
finest (20165 cells)	2.9229E+03	2.9199E+03	2.3005E+01	2.3102E+01

From the above table, it can be seen that both EC and LC are converging to the same result for the finest mesh. The EC method is more accurate than LC for all of the meshes. The relative errors with respect to the finest mesh EC scalar flux for both regions are plotted in Figure 18 and Figure 19.

Figure 18 shows that the EC method is more accurate than the LC method even in the optically thin source region. However, LC is still doing a very good job of getting the scalar flux in this region because the cells are not optically thick.

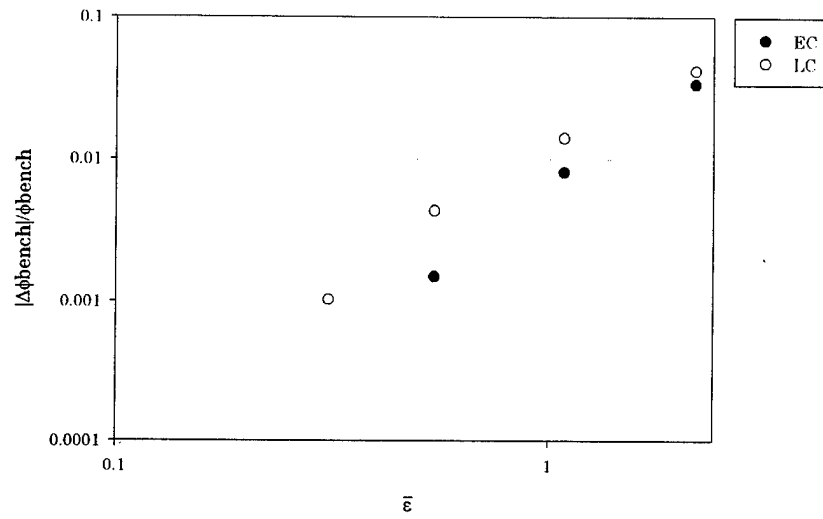


Figure 18. Relative Errors in Source Region Scalar Flux with Respect to EC Fine Mesh Benchmark for Deep Penetration Problem.

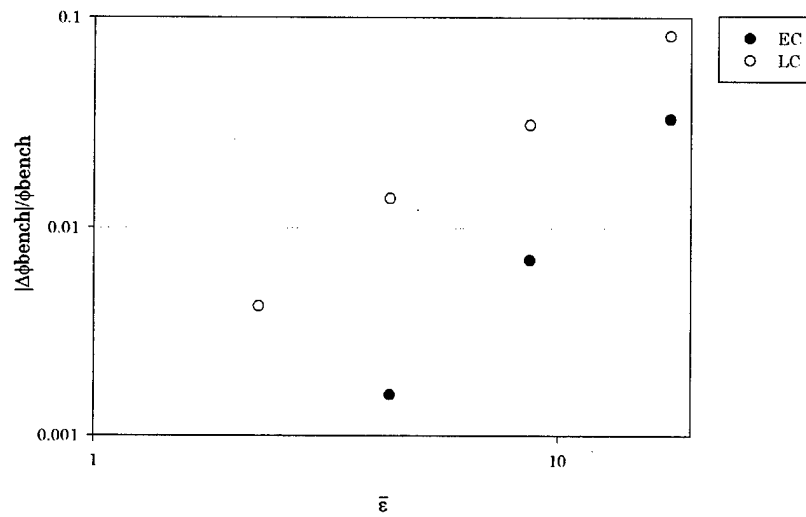


Figure 19. Relative Errors in Shield Region Scalar Flux with Respect to EC Fine Mesh Benchmark for Deep Penetration Problem.

In the case of the shield region (Figure 19), the LC method is significantly in error with respect to EC. This should be expected, however, because the region is optically very thick. LC has little hope of matching the highly exponential behavior of the flux in outer reaches of the shield whereas EC is doing an excellent job in this regard as evidenced by the scalar flux and currents.

Lastly, the exiting face currents for this problem are listed in Table 13.

Table 13. Exiting Face Currents for Deep Penetration Problem.

<u>Mesh</u>	<u>J<sub>+x</sub></u> (particles/s)		<u>J<sub>+y</sub></u> (particles/s)		<u>J<sub>+z</sub></u> (particles/s)	
	<u>EC</u>	<u>LC</u>	<u>EC</u>	<u>LC</u>	<u>EC</u>	<u>LC</u>
48	1.1105E-04	-1.8645E+02	9.1345E-05	-1.8316E+02	9.1895E-05	-1.8531E+02
384	4.1948E-05	7.2448E+00	4.2087E-05	3.9853E+00	4.0641E-05	4.8650E+00
3063	3.5524E-05	-6.6754E-04	3.5078E-05	1.2391E-03	3.5282E-05	4.0023E-04
20165	3.3825E-05	1.3047E-05	3.3890E-05	1.2830E-05	3.3825E-05	1.3599E-05

<u>Mesh</u>	<u>J<sub>-x</sub></u> (particles/s)		<u>J<sub>-y</sub></u> (particles/s)		<u>J<sub>-z</sub></u> (particles/s)	
	<u>EC</u>	<u>LC</u>	<u>EC</u>	<u>LC</u>	<u>EC</u>	<u>LC</u>
48	9.3634E+04	9.1195E+04	9.3127E+04	9.0643E+04	9.3127E+04	9.0644E+04
384	9.1186E+04	9.0434E+04	9.1248E+04	9.0092E+04	9.1091E+04	9.0242E+04
3063	9.0545E+04	8.9966E+04	9.0554E+04	9.0071E+04	9.0542E+04	9.0057E+04
20165	9.0434E+04	9.0280E+04	9.0429E+04	9.0242E+04	9.0427E+04	9.0256E+04

From Table 13, we see that the exiting face current prediction for the far faces of the shield (+x, +y, and +z) show the deep penetration performance of EC versus LC. For the coarsest mesh, LC produces returns negative currents through all of the shield faces. This is nonsense and we expect it because of the coarseness of the mesh. The EC calculation predicts currents that are only a factor of three higher than the most converged solution. Note that the

symmetry of the problem implies that  $J_{+x} = J_{+y} = J_{+z}$  (and  $J_{-x} = J_{-y} = J_{-z}$ ).

This behavior is clearly occurring as the mesh is refined. The LC current prediction is obviously completely in error until the finest mesh and then it is still considerably off because the mesh in the shield is still optically thick ( $\bar{\epsilon} = 2.26$ ) for LC. Further mesh refinement should bring the LC result in line with EC's. The relative errors (with respect to the most converged EC solution) in  $J_{+x}$  are plotted in Figure 20 versus  $\bar{\epsilon}$  for the shield as it is the shield's optical characteristics that are effecting the computation of  $J_{+x}$ .

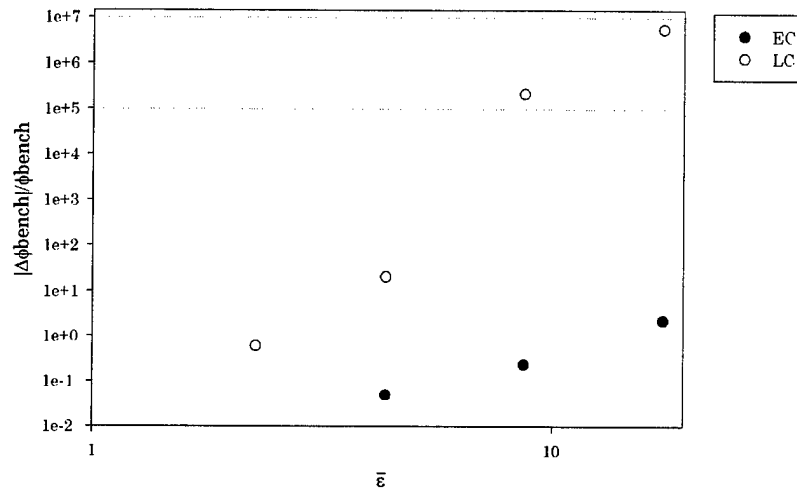


Figure 20. Relative Error with Respect to Fine Mesh EC Benchmark for  $J_{+x}$  using EC and LC.

Notice in Figure 20 that the current predicted by the 384 cell mesh using EC is more accurate than the LC results on the finest mesh (20165), a factor of

roughly 52. This coarse mesh accuracy more than makes up for the factor of 3 difference in computational cost between the EC and LC quadratures.

Lastly, we remark on some issues encountered in producing our coarse mesh results. In initial attempts to solve this problem, the EC method failed to converge. This type of error had been encountered before in exploring set-to-zero fix-ups for the source moments. However, there are no set-to-zero fix-ups in TETRAN. We examined the root-solver and found that the error was occurring because of an inconsistent implementation of the transition from the asymptotic solution to the source system (requiring no root-solves) to the first guess algorithm and the root-solver (discussed in Appendix B). This caused the source to be calculated inconsistently between iterations and prevented convergence. The problem was fixed by changing the break points to be moderately more conservative regarding the use of the asymptotic formulas. Thus, the root-solver is invoked more often in these extreme cases, producing consistently accurate source moments. At some point in the future, we will have to return to this issue and re-evaluate the region of applicability of the asymptotic solutions. However, the performance penalty for the more conservative approach is minimal and thus does not warrant a high priority for future work.

### **Multi-group, Anisotropic Scattering Performance**

Raising the level of complexity of our test problems higher, we examine TETRAN's ability to solve multi-group problems with and without anisotropic



scattering. The problem examined is similar to one found in the literature (Castrianni, 1997). We call it the water-iron-water problem.

The water-iron-water problem is comprised of three regions. Region 1 is a 20 centimeter cube of water centered at the origin with a uniformly distributed source of particles emitted isotropically. Region 2 is a hollow iron cube that surrounds the source and has a thickness of 10 centimeters. Finally, region 3 is a hollow cube filled with water that surrounds the iron region and has a thickness of 30 centimeters. This geometry is depicted in Figure 21.

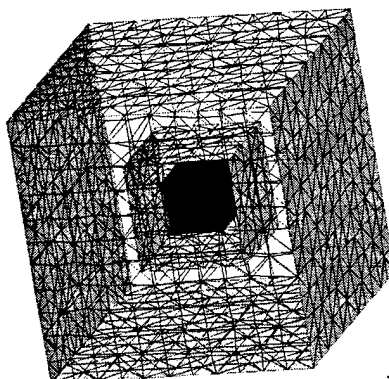


Figure 21. Geometry for the Water-Iron-Water Problem.

This problem is driven by a mono-energetic source of strength 1.0 particles/sec emitted in group one. Vacuum boundary conditions are assumed. Two anisotropic scattering approximations were used. For the EC method, we used  $P_1$  scattering cross sections that were corrected using the diagonal transport approximation (MacFarlane, 1992). Thus, we get  $P_1$  accuracy from an isotropic cross section calculation. These cross sections were also used for

an LC calculation. Additionally, we solved the problem using LC and  $P_3$  scattering moments that were corrected using the diagonal transport approximation to investigate the effect and performance of our anisotropic scattering treatment. We attempted to solve the  $P_3$  problem using EC but were unable to do so because using Legendre moments and spherical harmonics to update the scattering source produces negative sources. EC requires positive source data in order that we may calculate the source coefficients. This is especially true of water where the scattering off the hydrogen atom is particularly forward peaked because neutrons are only forward scatter in the laboratory frame of reference. Attempting to use Legendre polynomials to fit such a function invariably produces negative values. Additionally, narrow energy groups can cause negative cross sections to be produced for certain materials. Alternative cross section representations will be needed in order to eliminate negative cross sections. This is a subject for further research (DelGrande, 1998). The first three neutron energy groups (12.0 MeV to 17.0 MeV) in the MATXS10 cross section library (MacFarlane, 1992) were used for this problem. They are presented in Table 14, Table 15, and Table 16.

The MATXS10 cross sections were selected for this research for two reasons. They are readily available in the TRANSX package and contain many common engineering materials; they include up through  $P_4$  scattering moments; and they are based on the ENDF/B-VI evaluations (the most

current nuclear data files). Also, MATXS10's energy structure is identical to that of the MCNP multi-group cross section library, which is based upon the MENDF5 library used at Los Alamos for high energy transport calculations. However, there are substantial differences between the MATXS10 cross sections and their MCNP equivalent. First, the MATXS10 library is based on ENDF/B-VI while the MCNP version is derived from ENDF/B-V data. More importantly, in order to be incorporated into MCNP, the multi-group  $P_0$  through  $P_4$  data were processed into equally-probable cosine bins to treat the scattering using the MCNP data structure (MCNPXS, 1997). The data must also have been processed to eliminate the possibility of negative cross sections, which is as intolerable for the Monte Carlo method as it is to EC. Thus, although the cross sections appear to be derived from a common parent, they are at least partly different because of the data requirements of the codes for which they are intended. Because of the substantial differences between multi-group and continuous energy cross sections, we compare TETRAN's results with MCNP's multi-group results. This at least minimizes the differences due to the cross section representations. Of course, the MCNP results do not have the angular and spatial discretization errors that are inherent in the TETRAN results.

Table 14. P<sub>1</sub> Transport Corrected Cross Sections for Water.

Water		$\sigma_{s,g' \rightarrow g}(\text{cm}^{-1})$		
g	$\sigma_g(\text{cm}^{-1})$	g' = 1	g' = 2	g' = 3
1	0.074512	2.07399E-03		
2	0.079409	1.31405E-02	1.75144E-03	
3	0.081262	5.74009E-03	1.42416E-02	2.85886E-03
Natural Iron		$\sigma_{s,g' \rightarrow g}(\text{cm}^{-1})$		
g	$\sigma_g(\text{cm}^{-1})$	g' = 1	g' = 2	g' = 3
1	0.132931	1.08255E-02		
2	0.137572	1.38718E-02	1.26268E-02	
3	0.138529	1.41291E-03	1.34612E-02	1.74764E-02

Table 15. P<sub>3</sub> Transport Corrected Cross Sections for Water.

l=0		$\sigma_{sl,g' \rightarrow g}(\text{cm}^{-1})$		
g	$\sigma_g(\text{cm}^{-1})$	g' = 1	g' = 2	g' = 3
1	0.084473	1.20352E-02		
2	0.089661	1.31405E-02	1.20036E-02	
3	0.092259	5.74009E-03	1.42416E-02	1.38556E-02
l=1		$\sigma_{sl,g' \rightarrow g}(\text{cm}^{-1})$		
g		g' = 1	g' = 2	g' = 3
1		9.96124E-03		
2		8.13291E-03	1.02522E-02	
3		2.69041E-03	7.40822E-03	1.09968E-02
l=2		$\sigma_{sl,g' \rightarrow g}(\text{cm}^{-1})$		
g		g' = 1	g' = 2	g' = 3
1		6.67593E-03		
2		3.61894E-03	7.21434E-03	
3		2.54424E-03	2.18526E-03	6.84057E-03
l=3		$\sigma_{sl,g' \rightarrow g}(\text{cm}^{-1})$		
g		g' = 1	g' = 2	g' = 3
1		3.18798E-03		
2		2.60006E-03	3.58607E-03	
3		2.01574E-03	2.35980E-03	3.02045E-03

Table 16. P<sub>3</sub> Transport Corrected Cross Sections for Natural Iron.

$l=0$		$\sigma_{sl,g' \rightarrow g}(\text{cm}^{-1})$		
$g$	$\sigma_g(\text{cm}^{-1})$	$g' = 1$	$g' = 2$	$g' = 3$
1	0.162578	4.04721E-02		
2	0.173704	1.38718E-02	4.87585E-02	
3	0.180036	1.41291E-03	1.34612E-02	5.89831E-02
$l=1$		$\sigma_{sl,g' \rightarrow g}(\text{cm}^{-1})$		
$g$		$g' = 1$	$g' = 2$	$g' = 3$
1		2.96467E-02		
2		3.08452E-03	3.61317E-02	
3		1.58912E-04	9.05030E-04	4.15068E-02
$l=2$		$\sigma_{sl,g' \rightarrow g}(\text{cm}^{-1})$		
$g$		$g' = 1$	$g' = 2$	$g' = 3$
1		1.81862E-02		
2		1.00274E-03	2.24502E-02	
3		1.79372E-04	1.03198E-03	2.75191E-02
$l=3$		$\sigma_{sl,g' \rightarrow g}(\text{cm}^{-1})$		
$g$		$g' = 1$	$g' = 2$	$g' = 3$
1		8.75387E-03		
2		7.57219E-04	1.12032E-02	
3		9.07334E-07	4.47590E-04	1.39011E-02

The results of TETRAN's multi-group transport calculations are presented below. The data presented are for three different tetrahedral meshes: 528, 8849, and 17369 cells. Table 17 presents the group dependent average cell optical thickness by region. This is followed by the results presented by region.

Table 17. Group and Mesh Dependent Cell Optical Properties for Problem 6.

<u>Region</u>	<u>Cells</u>	$\bar{\epsilon}_g$		
		<u>g=1</u>	<u>g=2</u>	<u>g=3</u>
Source:	48	0.64718	0.68971	0.70581
	162	0.41489	0.44216	0.45248
	161	0.42145	0.44915	0.45963
Iron:	144	1.4830	1.5348	1.5454
	931	0.78105	0.80832	0.81394
	1350	0.68926	0.71332	0.71828
Water:	336	1.6522	1.7608	1.8019
	7756	0.57544	0.61326	0.62757
	15858	0.44951	0.47906	0.49023

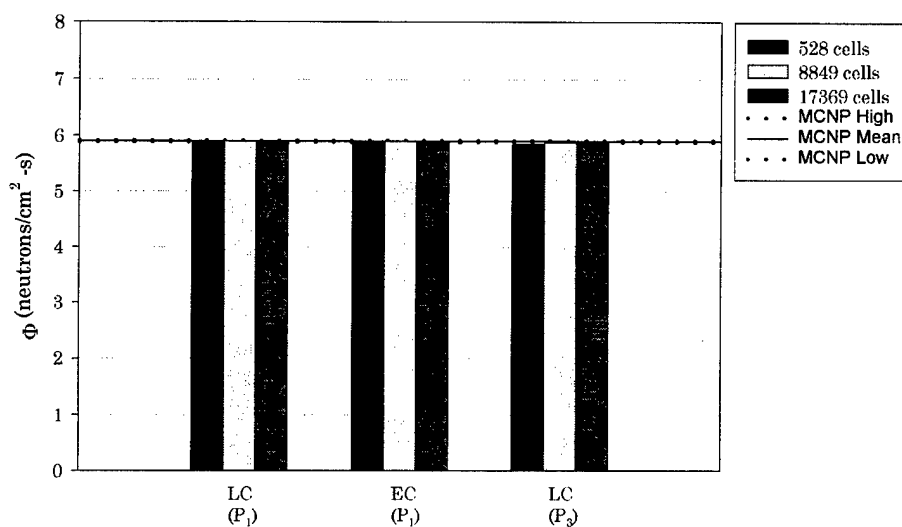


Figure 22. Group 1 Scalar Flux in Source Region.

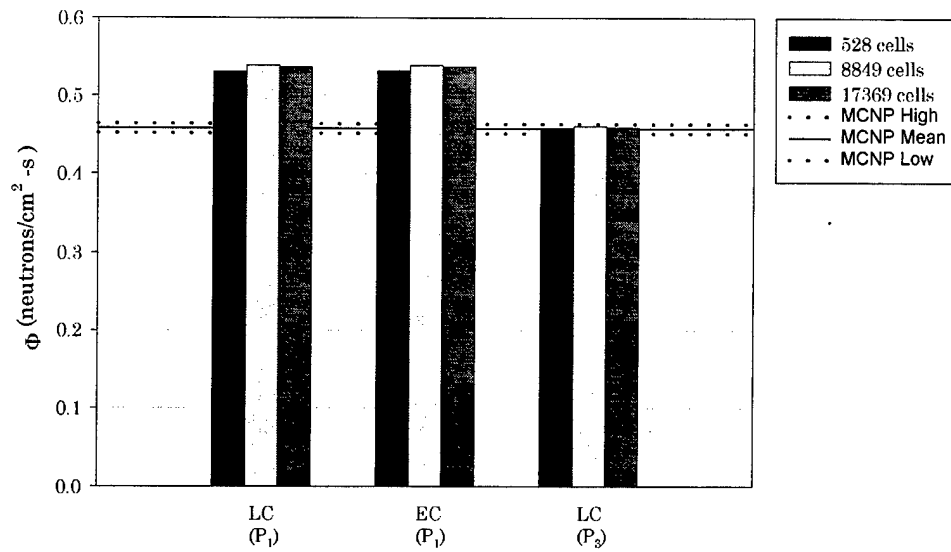


Figure 23. Group 2 Scalar Flux in Source Region.

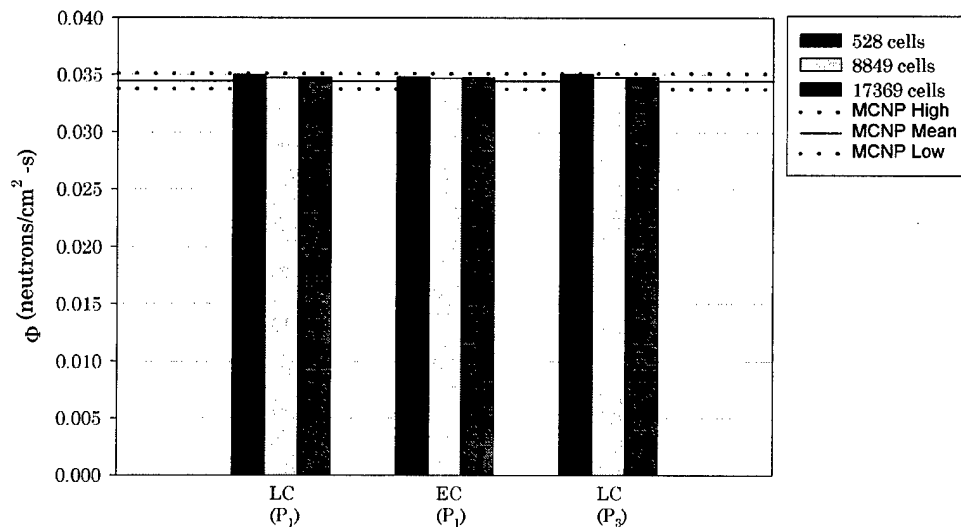


Figure 24. Group 3 Scalar Flux in Source Region.

The above figures show that there is a substantial difference in results between using the  $P_1$  and  $P_3$  scattering approximations. They also show good agreement between the LC  $P_3$  and the MCNP  $P_4$  calculation (indeed, the LC  $P_3$  cross sections are modified  $P_4$  cross sections). The group 1 flux shows excellent agreement with MCNP. However, we should expect this because group 1 is the highest group for this problem. It is in the downscatter groups that we expect to see the effect of the different treatments of anisotropic scattering. Indeed, we see that both group 2 and group 3 scalar fluxes are substantially different from MCNP for the  $P_1$  scattering cross sections whereas the LC  $P_3$  calculation is close to the MCNP results for all three groups. We now present the iron region results.

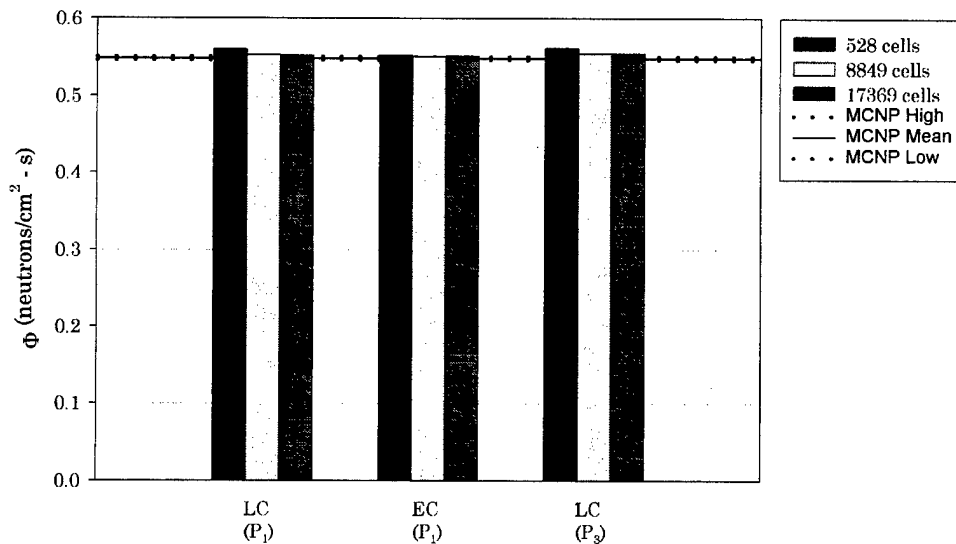


Figure 25. Group 1 Scalar Flux in Iron Region.



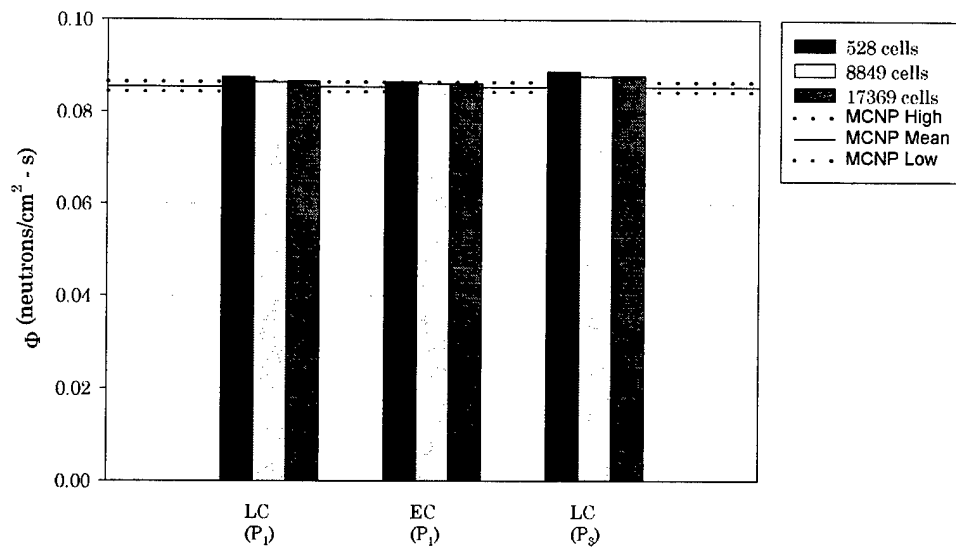


Figure 26. Group 2 Scalar Flux in Iron Region.

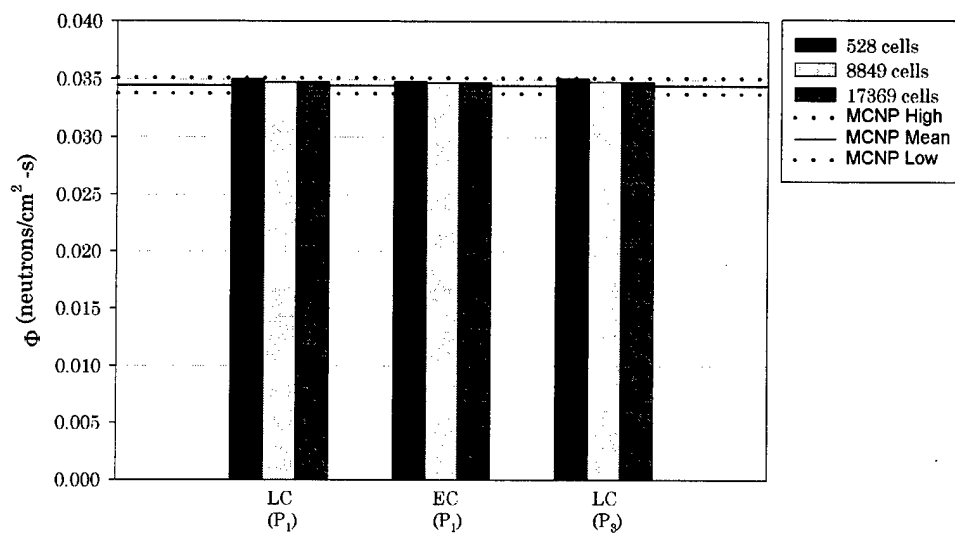


Figure 27. Group 3 Scalar Flux in Iron Region.

The discrete ordinates results for the scalar flux in the iron region show excellent agreement with the MCNP results for both the  $P_1$  and  $P_3$  calculations. This is because neutron scattering in iron is more isotropic than in water due to its greater nuclear mass. Another interesting aspect of the iron region is that the EC calculation is more accurate than the LC solution as compared to MCNP for all three energy groups. The coarse mesh LC calculation is measurably off for the fluxes but converges for the finer meshes. This is because the average optical thickness of a cell in the iron region is 1.52 (over all groups) for the coarse mesh and only 0.80 and 0.71 for the two finer meshes, respectively. Additionally, the maximum cell optical thicknesses range from 2.7 for the coarse mesh to 1.7 for the fine mesh (for all energy groups). Thus, we see again the thick cell performance of EC as compared to LC. We now will examine the scalar flux results in the outer water region.

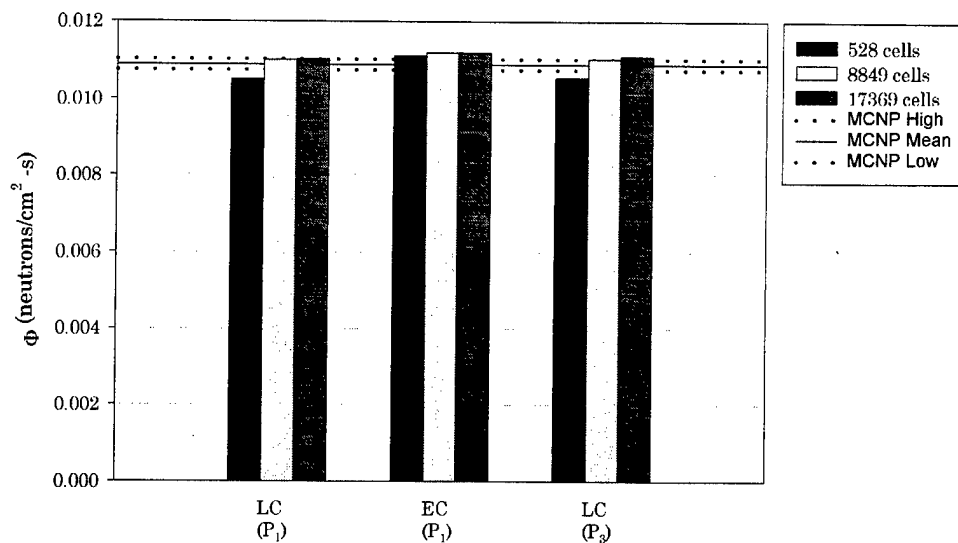


Figure 28. Group 1 Scalar Flux in Outer Water Region.

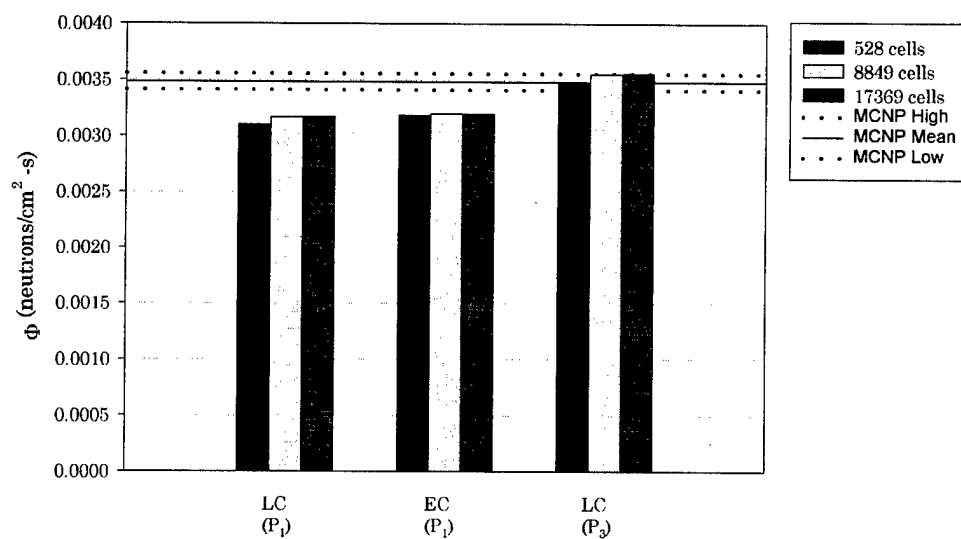


Figure 29. Group 2 Scalar Flux in Outer Water Region.

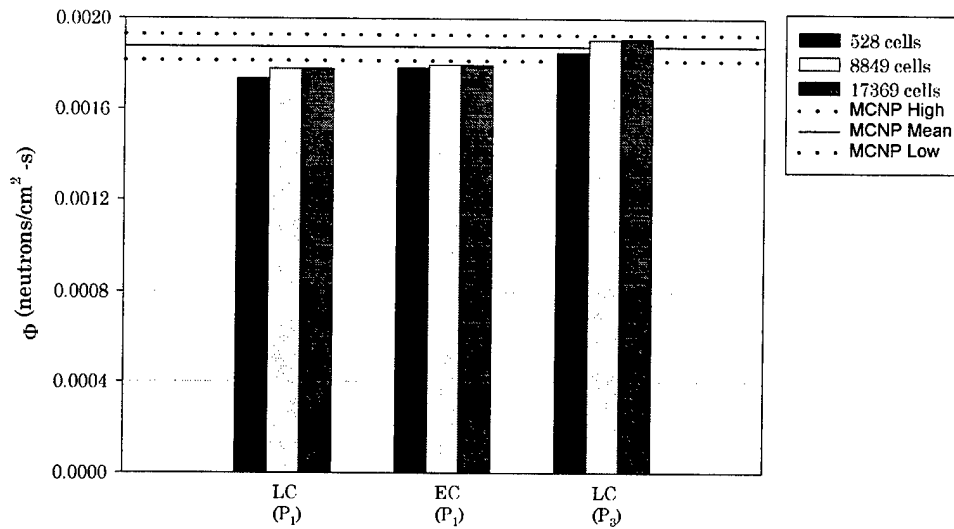


Figure 30. Group 3 Scalar Flux in Outer Water Region.

We see the same type of behavior for the water region that was seen for the source region. Both  $P_1$  calculations are substantially different from the LC  $P_3$  calculation. However, the LC  $P_3$  calculation compares very well to the MCNP calculation. However, this should be expected because water is a pronouncedly anisotropic scatterer of neutrons. Notice also the coarse mesh performance of EC compared to LC. In this instance, the average cell optical thickness (over all groups) is 1.74 for the coarse mesh and only 0.61 and 0.47 for the finer meshes. The maximum optical thicknesses range from 3.5 for the coarse mesh to 1.0 for the fine mesh (nearly constant over all groups). Notice also that the group 1 fluxes compare favorably with the MCNP results and the variations occur in the lower groups as expected.

Finally, we examine the group currents emerging from the +X face at  $X=50.0$  cm. The behavior of the current at the boundary for this problem gives a measure of the impact of the various methods and Legendre scattering order ( $P_1$  or  $P_3$ ) on the particle leakage.

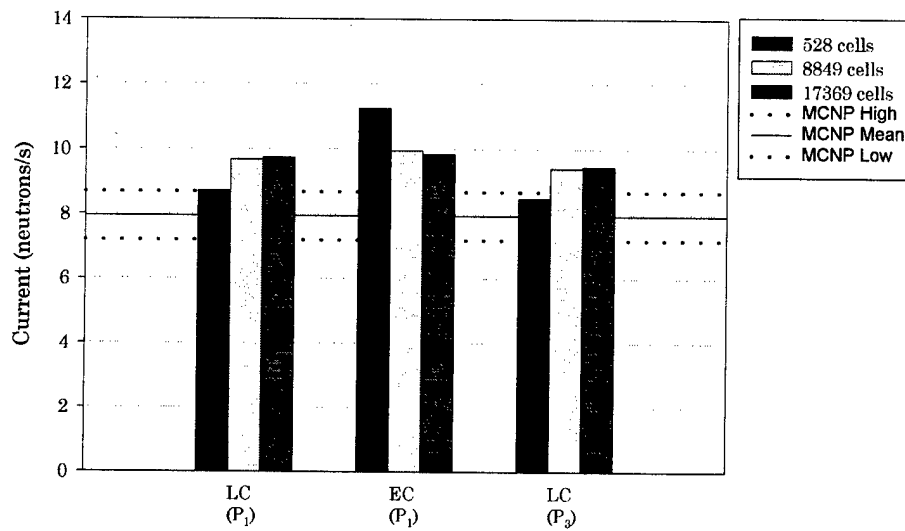


Figure 31. Group 1 Current Out of +X Face ( $X=50.0$  cm).

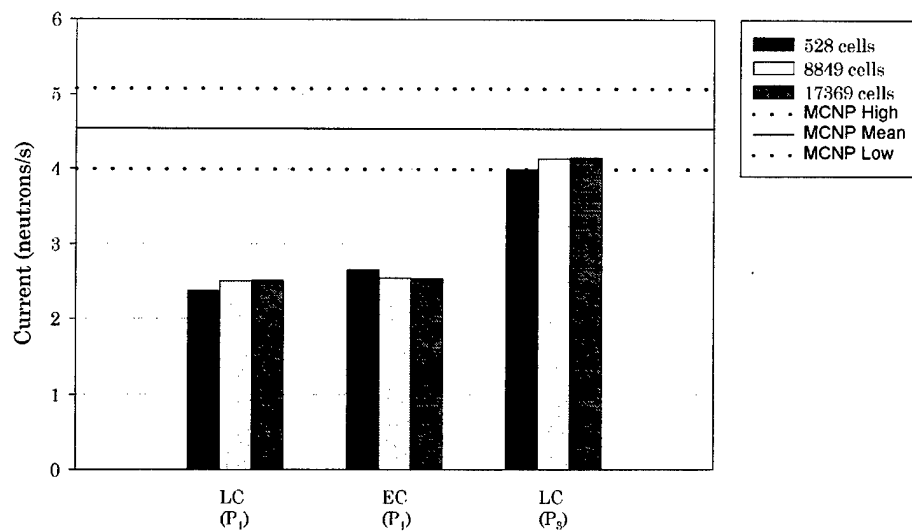


Figure 32. Group 2 Current Out of +X Face (X=50.0 cm).

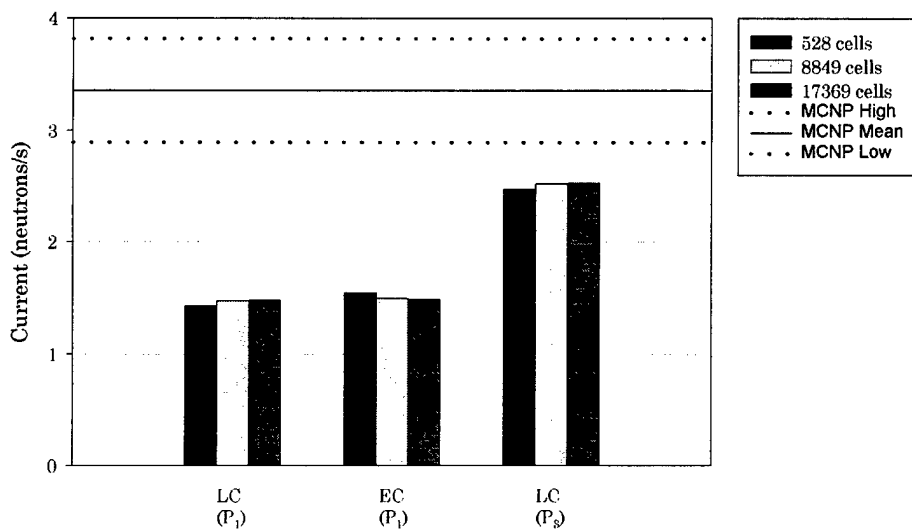


Figure 33. Group 3 Current Out of +X Face (X=50.0 cm).

From the above figures, it is obvious that the current is particularly sensitive to the scattering approximation. The  $P_1$  calculations are not very good for any group in this case. The  $P_3$  calculation is much better. In order to make sure that the algorithm was working correctly, particle balance calculations were done, which showed that particles are being conserved. This forces us to conclude that the differences between the current calculation are due to the nuclear data and the methods (Monte Carlo vs. discrete ordinates vs. EC/LC approximation) used. This conclusion could be further validated if a Monte Carlo code existed which forced particles to travel along discrete ordinates angular quadrature directions and was multigroup capable. Such a code would produce results which contained angular and energy discretization errors but would not have spatial discretization error, making it a superior benchmarking code for discrete ordinates results. The initial results of such a code, MCSN, were presented during testing of the convergence of EC and LC in problem one. Currently, no such capability exists for multigroup, anisotropic scattering problems. Dr. Kirk Mathews (AFIT) is developing an upgrade to MCSN to make it multigroup capable and able to use the T-matrix data structure (Mathews, 1998). It will be interesting to see how EC performs once positive T-matrix data becomes available for this problem (DelGrande, 1998).

The above figures show TETRAN is performing multi-group, anisotropic scattering calculations accurately (given the correct data). The

optical thickness of the regions between the 8849 cell mesh and the 17369 cell meshes are not very different. This is true of the source region as well. These trends are indicative of other problems that current mesh generators have with regard to mesh refinement. The mesh generator used in this work produced meshes that were inconsistent with increasing refinement, i.e. finer meshes were produced, but not all of the regions in the problem were refined consistently. This behavior along with the lack of volume conservation raises concerns regarding the use of common CAD/CAM codes for transport calculations. The T-matrix implementation for the group to group scattering source appears to function well, as does the iteration approach for each energy group. LC is theoretically capable of handling any order of anisotropic scattering while EC will need positive cross sections before it will be usable for problems requiring anisotropic scattering.

### **HPF Parallel Demonstration**

In this last test, we demonstrate the parallel scaling performance and accuracy of our HPF parallel implementation of TETRAN. For this problem, we use the 162 cell mesh used in the convergence rate problem presented earlier. The results are for both EC and LC. We chose to use this simple problem because the HPF compiler, PGHPF 2.4 (development compiler), produced code that runs between 6 (EC) and 8 (LC) times slower than the IBM xlf90 serial code used in the tests presented earlier. We chose a relatively small mesh because ASC's IBM SP proved to have problems with



its message passing library that prevented us from using more than two processors for any larger problem.

For our initial evaluation of TETRAN's parallel performance, we look at the consistency of the answers and the scaling of the code with increasing processors. We look at consistency to ensure that the HPF code produces the same results regardless of the number of processors used (deterministic). The speed up,  $S$ , is used to determine the parallel scaling of the code. Speed up is the ratio of the time to run the problem on one processor to that for multiple processors. No further investigation was done because the compiler has too many performance issues to make any further discussions useful. Future investigations should look at compiling with more than one HPF compiler if available.

The HPF code was found to provide the same results regardless of the number of processors used as required by the HPF standard (Koelbel, 1994). For LC, the scalar flux was 1.541295 particles/(cm<sup>3</sup>-s). The EC calculation for the scalar flux was 1.540445 particles/(cm<sup>3</sup>-s). The parallel speed-up ( $S$ ) and efficiency ( $E$ ) are presented in Table 18.

Table 18. HPF Parallel TETRAN Results.

$N_{\text{proc}}$	Run Time (s)		Speed-up, S		Efficiency, E (S/N)	
	LC	EC	LC	EC	LC	EC
1	882.93	2833.8	1	1	1.000	1.000
2	451.35	1435.6	1.9562	1.974	0.978	0.987
4	256.43	752.87	3.4431	3.764	0.861	0.941
8	154.73	424.31	5.7063	6.6786	0.713	0.835
10	114.92	321.99	7.6831	8.8009	0.768	0.880
16	110.96	327.43	7.9571	8.6547	0.497	0.541
32	170.32	248.33	5.1838	11.412	0.162	0.357
1 (xlf90)	107.74	489.31	N/A	N/A	N/A	N/A

We see in Table 18 that the HPF code is scaling to about 8 processors for LC and about 10 processors for EC before saturating. Beyond 16 processors, both methods appear to saturate as evidenced by the drop in the parallel efficiency shown in Table 18. Using 32 processors is markedly worse for LC than using 8 processors. The EC method performs better than the LC method, presumably because the processors are kept busy performing the expensive quadrature as was hoped. The saturation is probably due to the interprocessor communications required in the source and scalar flux updates or a demonstration of Amdahl's Law which basically says that the part of a program that can't be optimized (in our case, the source updates) will eventually dominate the runtime for a code (Dowd, 1998). Because the compiler has performance issues, we can't be sure. Clearly, much further investigation is required. A plot of the speed up versus the number of processors is shown in Figure 34.

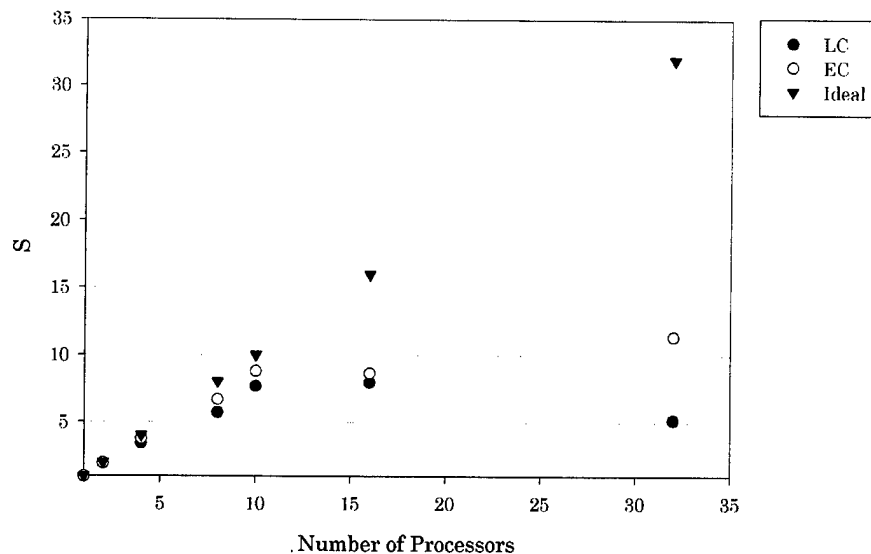


Figure 34. Demonstration of Speed Up for HPF Parallel TETRAN.

Regardless of the above issues, it is encouraging that the code compiles and runs and that the EC method appears to scale better than LC. This validates our design methodology of developing TETRAN using standard Fortran 90 and then inserting HPF directives as needed. We worked very closely with The Portland Group (Larry Meadows) to work out the bugs that prevented the compilation of TETRAN. Portland Group is now working on further enhancing their compiler to get the performance that is readily available with the serial Fortran 90 compiler available on the IBM SP. They believe that the problems with the compiler are in their implementation of the Fortran 90 intrinsic, MATMUL. If this is the case, then the performance results we report are preliminary because we use the MATMUL intrinsic in

every computational subroutine in TETRAN. It is also at the core of the source update routines. We expected that the maturity of HPF compilers would be better than was found (PGHPF was supposed to be mature). However, since HPF is still a developing compiler area, providing transport-related challenges to the compiler developers will hopefully result in better HPF compilers for the future. Lastly, in addition to the compiler issues, we found that the IBM SP has problems running the HPF version of TETRAN for large problems. This problem manifests itself as message-passing errors. The staff at the ASC MSRC is working on this issue as other users have complained of it as well. Although future efforts should consider using the IBM SP because of its capability to handle large amounts of data, other capable machines are available at the MSRC. Most noteworthy is the SGI ORIGIN 2000. This machine is ASC's largest shared memory machine (ASC, 1998). It is possible that the HPF version of TETRAN will perform better on such a machine versus the IBM SP. This performance difference has been observed by other researchers using different codes (Little, 1998).

This chapter presented TETRAN test results. Each problem tested critical features of the spatial quadratures and the overall implementation of the code. The results show that TETRAN is performing as expected. Both spatial quadratures converge to the same solution for the same problem and appear to be accurate compared to MCNP calculations. The EC quadrature proved to be extremely robust as evidenced by the problems it solved. The T-

matrix approach accurately solved the multigroup, anisotropic scatter problem and aided in the parallelization of TETRAN. Three major issues (beyond the scope of this research) were uncovered during this testing regarding mesh generation, positive nuclear data, and HPF compilers. However, TETRAN proved to be a strong performer on a variety of difficult problems.

## Chapter V: Conclusions and Recommendations

### Conclusions

We derived a new form of the EC spatial quadrature that is both elegant and illuminating with regard to its numerical implementation. This new form of the EC quadrature relies on the use of direct affine transformations to pass cell and flux moments within a mesh. EC and LC were implemented into a transport code, TETRAN. Both methods are numerically robust, convergent, and accurate. Several numerically accurate and robust algorithms (for  $\mathcal{M}_0(x, y, z, w)$  and the  $K$ -functions) were developed, enabling accurate computation of the EC (Appendix A) and LC quadratures. We tested the method's performance on a variety of problems, demonstrating convergence and robustness for poorly shaped and optically thick computational cells. The EC spatial quadrature is computationally expensive, costing approximately 3 ms/phase space cell while LC costs about a third of that at 1 ms/phase space cell.

A new root-solver was developed that uses a very accurate first guess algorithm based upon the characteristics of the solution space (Appendix B). The capability to perform standard multi-group calculations with anisotropic scattering was demonstrated for a difficult problem with great success. Our multi-group implementation uses the T-matrix approach over the traditional angular flux moments ( $\phi_{lm}$ ) approach for updating the scattering source. However, use of EC on greater than first order anisotropic scattering

problems will have to wait until research is complete on a positive cross section generator. Standard cross sections, using spherical harmonics with Legendre moments of the scattering cross section, produce negative sources which are intolerable (and physically wrong) for EC. The LC method did not have any problems using the standard data since the LC source assumptions do not require positivity of the source moments. Finally, we demonstrated the parallel operation of an HPF version of TETRAN on a simple problem. Both EC and LC scale with increasing processors, the EC method scaling better than LC. Future work in this area will have to wait for a more mature compiler.

In summary, this effort developed a parallel, unstructured tetrahedral mesh radiation transport code, TETRAN. This code solves the BTE using either the LC or the EC method using direct affine transformations to calculate the needed source coefficients. It can solve multigroup, anisotropic scattering problems using a new approach to calculating the scattering source — the T-matrix algorithm. New findings regarding the nature of the source and inflow flux equations allowed the implementation a new non-linear root-solving strategy. HPF directives made the code parallel on the IBM SP. Substantial effort was made to implement the EC method such that it was robust and accurate. Seven test problems demonstrate this robustness and accuracy as discussed earlier.

## Recommendations for Future Work

Several potential areas of future research were uncovered during this research: mesh generation, parallel algorithms, and benchmarking.

If unstructured tetrahedral mesh transport codes are to succeed, they must have good meshes on which to operate. We recommend that a new mesh generator, optimized for radiation transport, be developed. It should

- conserve volumes,
- minimize aspect ratios of cells,
- provide users control of optical thicknesses,
- combine triangular cells on selected surfaces with tetrahedral cells that align with them, and
- provide optional self-similar mesh refinement where feasible.

We recommend that MCSN be extended to multigroup, anisotropic scatter using T-matrices, in order to provide more useful benchmark results.

Current computer architectures are dependent upon locality of reference for effective use of caches. We recommend development of a regular-hexahedral-mesh code analogous to TETRAN.

Various enhancements are needed to make TETRAN more useful, including:

- generating positive T-matrices, and
- adding convergence acceleration.

Other enhancements might improve the speed of the code, such as:



- using balance equations to eliminate expensive moment functions where well-conditioned, and
- exploring alternatives for indirect addressing of fluxes.

HPF and Fortran 90/95 compilers are not yet mature. Many are still in early stages of development. We recommend testing different hardware platforms and compilers. Ultimately, it may be necessary to recode using MPI and/or using various coding tricks to squeeze performance from a specific platform and compiler. We recommend that this be postponed until everything else is done, both to avoid wasted effort and because it may prove to be unnecessary.

## Appendix A: Exponential Moment Functions — Evaluation and Identities

The EC method requires the evaluation of a number of exponential moment functions. This appendix presents several moment function identities used in the derivation of EC. The algorithm for numerically evaluating four argument exponential moment functions is outlined.

### Definitions

As previously presented, the exponential moment functions are defined as

$$\mathcal{M}_n(x_1, x_2, \dots, x_m) = \int_0^1 dt_1 (1-t_1)^n e^{-x_1 t_1} \int_0^{t_1} dt_2 e^{(x_1-x_2)t_2} \dots \int_0^{t_{m-1}} dt_m e^{(x_{m-1}-x_m)t_m}. \quad (82)$$

Note that  $\mathcal{M}_n(x_1, x_2, \dots, x_m) > 0$ . Additionally, the moments functions are orderless with respect to their arguments, e.g.

$\mathcal{M}_n(x_1, x_2, \dots, x_m) = \mathcal{M}_n(x_2, x_1, \dots, x_m)$  and so on. The multi-argument  $\mathcal{M}_0$

function is most often used in the EC method. The ratio

$\mathcal{M}_1(x_1, x_2, \dots, x_m) / \mathcal{M}_0(x_1, x_2, \dots, x_m)$  also appears frequently with respect to

EC. Thus, we define

$$\rho(x_1, x_2, \dots, x_m) = \frac{\mathcal{M}_1(x_1, x_2, \dots, x_m)}{\mathcal{M}_0(x_1, x_2, \dots, x_m)}. \quad (83)$$

In this case,  $0 < \rho(x_1, x_2, \dots, x_m) < 1$ , which follows from the definition of

$\mathcal{M}_1$  and  $\mathcal{M}_0$ .

Lastly, we define a new exponential moment function ratio,  $\mathcal{R}_j$ , given as

$$\mathcal{R}_j(x_1, x_2, \dots, x_m) = \frac{\mathcal{M}_0(x_1, x_2, \dots, x_m, x_j)}{\mathcal{M}_0(x_1, x_2, \dots, x_m)}, \quad (84)$$

where the  $j$  subscript indicates that the  $j^{\text{th}}$  argument ( $j=1, 2, \dots$  or  $m$ ) is repeated in the numerator. Note that the numerator of  $\mathcal{R}_j$  is a function of  $m+1$  coefficients whereas the denominator is a function of  $m$  terms and that  $0 < \mathcal{R}_j(x_1, x_2, \dots, x_m) < 1$ .

### Exponential Moment Function Identities

Many exponential moment function identities are used in the EC spatial quadrature derivation. These identities are used to derive the quadrature and cast it in a stable form. Most were presented in (Mathews, 1997). All of the presented identities were verified symbolically using Mathematica® 3.0. Additional detail regarding exponential moment functions can be found in (Minor, 1993).

#### Identity 1

$$\mathcal{M}_n(x_1, x_2, \dots, x_m) = \frac{\mathcal{M}_n(x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_m) - \mathcal{M}_n(x_2, \dots, x_{k-1}, x_{k+1}, \dots, x_m)}{x_j - x_k} \quad (85)$$

$x_j \neq x_k$

The divided difference identity is perhaps the oldest of our identities. It was initially proven in Minor's dissertation (Minor, 1993). The above identity is used to stably evaluate  $\mathcal{R}_j(x_1, x_2, \dots, x_m)$  and  $\rho(x_1, x_2, \dots, x_m)$ .

### Identity 2

$$\mathcal{M}_0(x_1, x_2, \dots, x_m) = \exp(-x_1) \mathcal{M}_0(-x_1, x_2 - x_1, \dots, x_m - x_1) \quad (86)$$

Identity 2 is used to cast  $\mathcal{R}_j(x_1, x_2, \dots, x_m)$  in a numerically stable form. It is valid for all values of  $m$ .

### Identity 3

$$\frac{\partial}{\partial x_k} \mathcal{M}_n(x_1, \dots, x_k, \dots, x_m) = -\mathcal{M}_n(x_1, \dots, x_k, x_k, \dots, x_m) \quad (87)$$

Identity 3 is used in the evaluation of the source system and inflow face flux system Jacobians as well as in the derivation of the EC spatial quadrature. It was verified for all  $n$  and  $m$ .

### Identity 4

$$\mathcal{M}_0(0, x_1, x_2, \dots, x_m) = \mathcal{M}_1(x_1, x_2, \dots, x_m) \quad (88)$$

Identity 4 is used with Identity 2 to recast  $\mathcal{R}_j(x_1, x_2, \dots, x_m)$  into a numerically stable form involving  $\rho(x_1, x_2, \dots, x_m)$ . The identity was verified for all values of  $m$ .

### Identity 5

$$\mathcal{M}_0(x_1, x_2, \dots, x_m) = \frac{\mathcal{M}_0(x_1, x_2, \dots, x_{m-1}) - \exp(-x_1) \mathcal{M}_0(x_2 - x_1, \dots, x_m - x_1)}{x_m} \quad (89)$$

$x_m \neq 0$

Identity 5 is used in the numerical evaluation of the moment functions. It was confirmed by direct symbolic evaluation of the integral functions using Mathematica® for  $m=2$  to 5. This identity was first introduced in (Mathews, 1996).

### Identity 6

$$\rho(x_1, x_2, \dots, x_m) + \sum_{j=1}^m \mathcal{R}_j(x_1, x_2, \dots, x_m) = 1 \quad (90)$$

Identity 6 is new to this research. It is used to recast the EC spatial quadrature into a numerically stable form without subtractions. It was confirmed by direct symbolic evaluation of the integral functions using Mathematica® for  $m=2$  to 4.

### $\mathcal{M}_0(x, y, z, w)$ and $\rho(x, y, z, w)$ Numerical Evaluation

The development of stable and accurate routines to evaluate the exponential moment functions and their ratios was an important contribution of this research. Previously, Mathews developed streamlined routines to evaluate  $\mathcal{M}_0(x)$  and  $\rho(x)$ ,  $\mathcal{M}_0(x, y)$  and  $\rho(x, y)$ , and  $\mathcal{M}_0(x, y, z)$  and  $\rho(x, y, z)$ .

The algorithms are presented in (Mathews, 1997). For this effort, we

extended the algorithm used for three-argument case to the four-argument case. Although the four-argument algorithm is not as elegant as the three-argument one, it is stable and accurate to at least 10 digits. The pseudo-code for the four-argument function algorithm is presented below.

Sort  $x, y, z$ , and  $w$  into ascending order -  $w \geq z \geq y \geq x$ . We can do this because the exponential moment functions are orderless with respect to their arguments.

If  $x \geq 38.0$  then all arguments are large enough that

$M_0(x, y, z, w) = \frac{1}{xyzw}$  and  $\rho(x, y, z, w) = 1 - \frac{1}{x} - \frac{1}{y} - \frac{1}{z} - \frac{1}{w}$ . This approximation is accurate to at least 12 digits.

If  $\text{abs}(x)$  and  $\text{abs}(w) \leq 0.025$ , (all arguments small) then we use a Horner nested MacLaurin series where we expand both  $M_0(x, y, z, w)$  and  $\rho(x, y, z, w)$  to 5<sup>th</sup> order terms in  $x, y, z$ , and  $w$  and retain all terms up to 5<sup>th</sup> order.

Let  $\chi = \frac{y-x}{w-x}$  and  $\omega = \frac{z-x}{w-x}$ .

If  $\chi < 0.1$  or  $\chi > 0.9$  or  $\omega < 0.1$  or  $\omega > 0.9$  or  $(\omega - \chi) < 0.1$  or  $(\omega - \chi) > 0.9$  or  $\text{abs}(w-x) < 0.1 \text{ Max}\{\text{abs}(x), \text{abs}(w)\}$  then one or more arguments are close together but not near zero. In this case, we use identity 5 (above) to get :

$$M_0(x, y, z, w) = \frac{M_0(x, y, z) - \exp(-x)M_0(y-x, z-x, w-x)}{w}$$

$$\rho(x, y, z, w) = \frac{M_0(x, y, z)\rho(x, y, z) - M_0(x, y, z, w)}{w M_0(x, y, z, w)}$$

(Note that if  $\text{abs}(x) > \text{abs}(w)$ , we switch  $x$  and  $w$  for better conditioning.)

Else, we use the general divided difference formulation given as:

$$M_0(x, y, z, w) = [(1-\chi)(\omega-\chi)(1-\omega)M_0(x) - (1-\omega)\omega M_0(y) + \chi(1-\chi)M_0(z) - \chi\omega(\omega-\chi)M_0(w)] / [(w-x)^3 (1-\chi)\chi(\omega-\chi)(1-\omega)\omega]$$

$$\begin{aligned} \rho(x, y, z, w) = & [(1-\chi)(\omega-\chi)(1-\omega)\rho(x)\mathcal{M}_0(x) - (1-\omega)\omega\rho(y)\mathcal{M}_0(y) + \\ & (1-\chi)\chi\rho(z)\mathcal{M}_0(z) - \chi\omega(\omega-\chi)\rho(w)\mathcal{M}_0(w)] / \\ & [(1-\chi)(\omega-\chi)(1-\omega)\mathcal{M}_0(x) - (1-\omega)\omega\mathcal{M}_0(y) + \\ & (1-\chi)\chi\mathcal{M}_0(z) - \chi\omega(\omega-\chi)\mathcal{M}_0(w)] \end{aligned}$$

The last equations are obtained by applying Identity 1 to  $\mathcal{M}_0$  and  $\mathcal{M}_1$ , rearranging terms, and applying the definitions for  $\chi$  and  $\omega$ .

## Appendix B: Evaluation of Source and Inflow Flux Coefficients

The exponential characteristic method requires that source and inflow flux coefficients be computed and used in the sub-cell spatial quadrature. In this appendix, we present the algorithm used to calculate these coefficients.

### Source Coefficients

We first discuss the equations and algorithms needed to get the unit tetrahedron source coefficients. Recalling the source system equations,

$$S_A = 6 \exp(A_s) \mathcal{M}_0(X_s, Y_s, Z_s), \quad (91)$$

$$S_U = S_A [1 - \rho(X_s, Y_s, Z_s)], \quad (92)$$

$$S_V = S_U - S_A \mathcal{R}_1(X_s, Y_s, Z_s), \quad (93)$$

and

$$S_W = S_V - S_A \mathcal{R}_2(X_s, Y_s, Z_s), \quad (94)$$

where  $S_A$ ,  $S_U$ ,  $S_V$ , and  $S_W$  are the cell source average and first moments (known from the previous iteration), the  $\mathcal{R}_j$  functions were previously defined in Appendix A, and the coefficients are,

$$\begin{aligned} X_s &= -\tilde{B}_u, \\ Y_s &= -(\tilde{B}_u + \tilde{B}_v), \\ Z_s &= -(\tilde{B}_u + \tilde{B}_v + \tilde{B}_w). \end{aligned} \quad (95)$$



This system of equations is easily converted to a non-linear system of three equations and three unknowns by dividing equations (92) through (94) by  $S_A$  and letting  $\rho_{UV} = \frac{S_U - S_V}{S_A}$ ,  $\rho_{VW} = \frac{S_V - S_W}{S_A}$ , and  $\rho_W = \frac{S_W}{S_A}$ . This substitution and the application of Identity 6 (Appendix A), results in the following system of equations:

$$\rho_{UV} = \rho_U - \rho_V = \mathcal{R}_1(X_s, Y_s, Z_s), \quad (96)$$

$$\rho_{VW} = \rho_V - \rho_W = \mathcal{R}_2(X_s, Y_s, Z_s), \quad (97)$$

and

$$\rho_W = \mathcal{R}_3(X_s, Y_s, Z_s), \quad (98)$$

where  $0 < \rho_{\text{sum}} = \rho_{UV} + \rho_{VW} + \rho_W < 1$ .

The solution to the above source equations is contained within a phase space bounded by the tetrahedron depicted below.

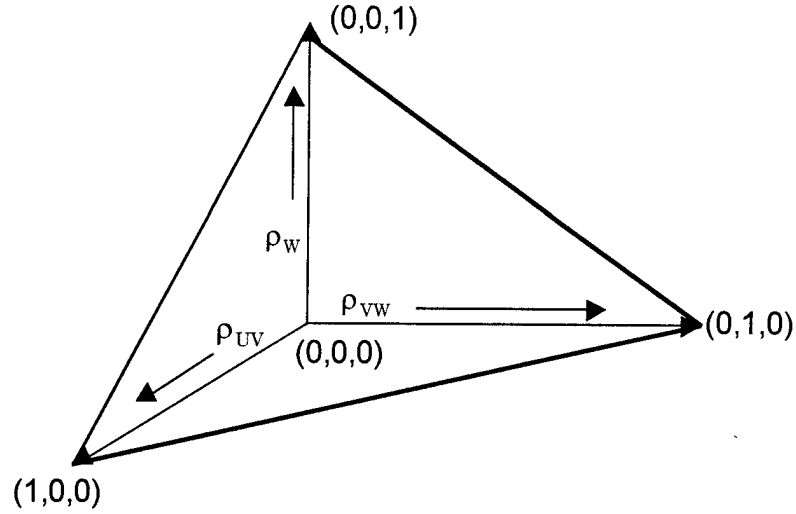


Figure 35. The  $(\rho_{UV}, \rho_{VW}, \rho_W)$  Solution Space.

The above phase space maps  $(\rho_{UV}, \rho_{VW}, \rho_W)$  to the  $(X_s, Y_s, Z_s)$  coefficient space according to the mapping below.

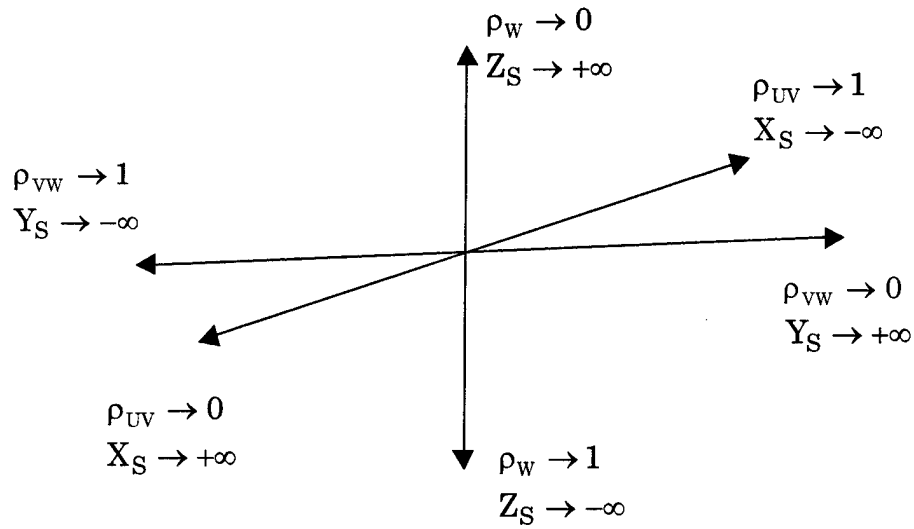


Figure 36. Mapping of  $(\rho_{UV}, \rho_{VW}, \rho_W)$  Phase Space to  $(X_s, Y_s, Z_s)$  Coefficient Space.

From Figure 36, we see that the solution space maps the input data in a tetrahedron within a volume bound between 0 and 1 to all of  $(X_s, Y_s, Z_s)$  space. Additionally, the solutions  $X_s(\rho_{UV}, \rho_{VW}, \rho_W)$ ,  $Y_s(\rho_{UV}, \rho_{VW}, \rho_W)$ , and  $Z_s(\rho_{UV}, \rho_{VW}, \rho_W)$ , are expected to vary rapidly with the input data but there should be no discontinuities that would cause bad behavior with the evaluation of the Jacobian that will be necessary for the root solver.

After obtaining  $X_s$ ,  $Y_s$ , and  $Z_s$ , the source coefficients are found to be:

$$\begin{aligned}\tilde{B}_u &= -X_s, \\ \tilde{B}_v &= X_s - Y_s, \\ \tilde{B}_w &= Y_s - Z_s,\end{aligned}\tag{99}$$

and

$$A_s = \log_e \left( \frac{S_A}{6 \mathcal{M}_0(X_s, Y_s, Z_s)} \right).\tag{100}$$

The root solve strategy used in TETTRAN to obtain the source coefficients is presented in the next section.

### Source System Root Solver

The source coefficients presented above are obtained by root-solving the system of equations embodied in (96), (97), and (98). Broyden's method (Burden, 1993), which requires the evaluation of the Jacobian of the system of equations, is used. We chose to use Broyden's method because it requires only one evaluation of the system's Jacobian matrix (which is computationally expensive) followed by iterative updates that are computationally

inexpensive. This approach is taken with penalties of only superlinear convergence (versus quadratic for Newton's method) and the need for an accurate first guess generator (like Newton's method) to start the solver. A first guess is necessary. Naively using (0,0,0) as the first guess, Broyden's method failed to solve the source system after 20+ iterations. Thus, a good first guess for the method is necessary. However, it is shown below that the cost to evaluate the Jacobian dominates the cost of the root-solving algorithm including the first guess algorithm.

The source system of equations, recast in the normal form for root-solving, is

$$f_1(X_s, Y_s, Z_s) = \mathcal{R}_1(X_s, Y_s, Z_s) - \rho_{UV} = 0, \quad (101)$$

$$f_2(X_s, Y_s, Z_s) = \mathcal{R}_2(X_s, Y_s, Z_s) - \rho_{VW} = 0, \quad (102)$$

and 
$$f_3(X_s, Y_s, Z_s) = \mathcal{R}_3(X_s, Y_s, Z_s) - \rho_W = 0. \quad (103)$$

The Jacobian for the above system is

$$J^{\text{Source}} = \begin{pmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{pmatrix}, \quad (104)$$

where the elements are

$$\begin{aligned}
J_{11} &= \frac{\partial f_1}{\partial X_s} = \mathcal{R}_1(X_s, Y_s, Z_s) [\mathcal{R}_1(X_s, Y_s, Z_s) - 2\mathcal{R}_1(X_s, X_s, Y_s, Z_s)], \\
J_{12} &= \frac{\partial f_1}{\partial Y_s} = \mathcal{R}_2(X_s, Y_s, Z_s) [\mathcal{R}_1(X_s, Y_s, Z_s) - \mathcal{R}_1(X_s, Y_s, Y_s, Z_s)], \\
J_{13} &= \frac{\partial f_1}{\partial Z_s} = \mathcal{R}_3(X_s, Y_s, Z_s) [\mathcal{R}_1(X_s, Y_s, Z_s) - \mathcal{R}_1(X_s, Y_s, Z_s, Z_s)], \\
J_{21} &= \frac{\partial f_2}{\partial X_s} = J_{12}, \\
J_{22} &= \frac{\partial f_2}{\partial Y_s} = \mathcal{R}_2(X_s, Y_s, Z_s) [\mathcal{R}_2(X_s, Y_s, Z_s) - 2\mathcal{R}_2(X_s, Y_s, Y_s, Z_s)], \\
J_{23} &= \frac{\partial f_2}{\partial Z_s} = \mathcal{R}_3(X_s, Y_s, Z_s) [\mathcal{R}_2(X_s, Y_s, Z_s) - \mathcal{R}_2(X_s, Y_s, Z_s, Z_s)], \\
J_{31} &= \frac{\partial f_3}{\partial X_s} = J_{13}, \\
J_{32} &= \frac{\partial f_3}{\partial Y_s} = J_{23}, \\
J_{33} &= \frac{\partial f_3}{\partial Z_s} = \mathcal{R}_3(X_s, Y_s, Z_s) [\mathcal{R}_3(X_s, Y_s, Z_s) - 2\mathcal{R}_3(X_s, Y_s, Z_s, Z_s)].
\end{aligned} \tag{105}$$

In evaluating the elements of the Jacobian, we used Identity 3 (Appendix A) as well as the quotient rule for differentiation.

Clearly, the computation of the source system's Jacobian is expensive. It is efficient to re-use the double repeated argument functions which are calculated in the evaluation of the system ( $f_1$ ,  $f_2$ , and  $f_3$ ) before the evaluation of  $J$ . However, this still leaves 6 five-argument/three-argument ratios ( $\mathcal{R}_j$  with repeated arguments) to compute accurately and robustly. If we use the standard Newton's method to invert this system,  $J$  would have to be evaluated for each iteration of the root solver. Broyden's method (Burden, 1993) avoids this requirement by requiring only one evaluation of  $J$  followed

by comparatively cheap iterative updates. Broyden requires 1 vector addition, 3 matrix multiplies, 1 transposition, 1 dot product, and the evaluation of  $f_1$ ,  $f_2$ , and  $f_3$ . This approach provides coefficients that are accurate to 9 digits with no more than 5 iterations and generally 2 or 3. Key to using Broyden's method was the development of a first guess algorithm, presented next. The algorithm used to numerically evaluate the source system and its Jacobian is presented in Appendix C.

### Source First Guess Algorithm

Broyden's Method may converge slowly or not at all if it is started too far from the true root of the system it is solving. To ensure that this not happen in TETRAN, we developed a fairly accurate first guess algorithm based upon the behavior of the solution space. Indeed, Brennan developed a similar algorithm based upon the asymptotic behaviors of  $X_s(\rho_{UV}, \rho_{VW}, \rho_W)$ ,  $Y_s(\rho_{UV}, \rho_{VW}, \rho_W)$ , and  $Z_s(\rho_{UV}, \rho_{VW}, \rho_W)$  for the solution in the corners of the  $(\rho_{UV}, \rho_{VW}, \rho_W)$  phase space (Brennan, 1996). A simple interpolation spliced these solutions together, providing a guess to the solution in the interior of the space. This approach worked well but did not take advantage of the nature of the solution space. Our approach, although perhaps not optimal, is based upon the behavior of  $X_s(\rho_{UV}, \rho_{VW}, \rho_W)$ ,  $Y_s(\rho_{UV}, \rho_{VW}, \rho_W)$ , and  $Z_s(\rho_{UV}, \rho_{VW}, \rho_W)$  in the interior of the solution space.

Our first guess strategy is built around the asymptotic behavior of the solution to the system. These solutions are presented in Table 19.

Table 19. Asymptotic Solutions to the Source System.

	$\rho_{UV} \rightarrow 1$ $\rho_{VW} \rightarrow 0$ $\rho_W \rightarrow 0$ $\rho_{Sum} \rightarrow 1$	$\rho_{UV} \rightarrow 0$ $\rho_{VW} \rightarrow 1$ $\rho_W \rightarrow 0$ $\rho_{Sum} \rightarrow 1$	$\rho_{UV} \rightarrow 0$ $\rho_{VW} \rightarrow 0$ $\rho_W \rightarrow 1$ $\rho_{Sum} \rightarrow 1$	$\rho_{UV} \rightarrow 0$ $\rho_{VW} \rightarrow 0$ $\rho_W \rightarrow 0$ $\rho_{Sum} \rightarrow 0$
$X_S$	$\frac{1}{1 - \rho_{Sum}}$	$\frac{1}{\rho_{UV}} - \frac{1}{1 - \rho_{Sum}}$	$\frac{1}{\rho_{UV}} - \frac{1}{1 - \rho_{Sum}}$	$\frac{1}{\rho_{UV}}$
$Y_S$	$\frac{1}{\rho_{VW}} - \frac{1}{1 - \rho_{Sum}}$	$-\frac{1}{1 - \rho_{Sum}}$	$\frac{1}{\rho_{VW}} - \frac{1}{1 - \rho_{Sum}}$	$\frac{1}{\rho_{VW}}$
$Z_S$	$\frac{1}{\rho_W} - \frac{1}{1 - \rho_{Sum}}$	$\frac{1}{\rho_W} - \frac{1}{1 - \rho_{Sum}}$	$-\frac{1}{1 - \rho_{Sum}}$	$\frac{1}{\rho_W}$

In the case where any of  $\rho_{UV}$ ,  $\rho_{VW}$ , or  $\rho_W$  are greater than or equal to 0.95 and  $\rho_{Sum} \geq 0.99$  or  $\rho_{Sum} \leq 0.01$ , the solutions in Table 19 are accurate to at least 9 digits requiring no root solving.

The more interesting case occurs when  $(\rho_{UV}, \rho_{VW}, \rho_W)$  is not located in one of the corners of the phase space tetrahedron. In this case, we developed an algorithm based upon the geometric form of the solutions in the space. Using AVS/Express, the solution to the system for many random values of  $(\rho_{UV}, \rho_{VW}, \rho_W)$  was visualized. Looking at the form of the  $X_s(\rho_{UV}, \rho_{VW}, \rho_W)$  solution, for example, an interesting result emerged. It is plotted in Figure 37 below.

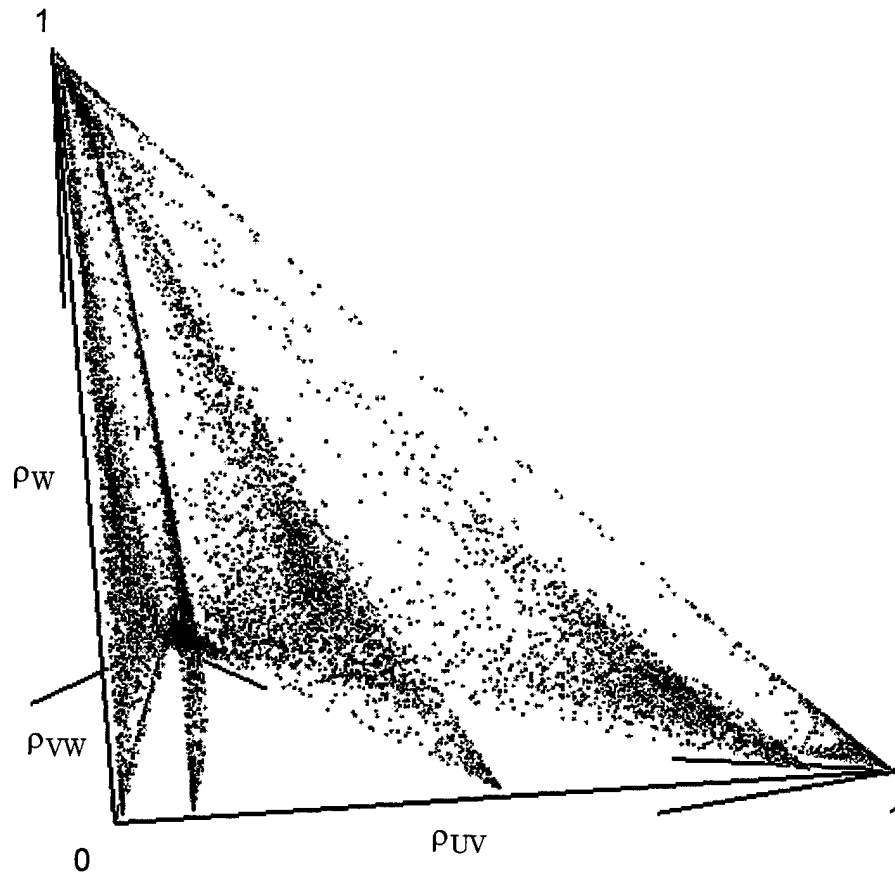


Figure 37. Plot of  $X_s(\rho_{UV}, \rho_{VW}, \rho_W)$  for  $X_s = -100, -10, 0, 10, 100$  (right to left).

Figure 37 shows that  $X_s(\rho_{UV}, \rho_{VW}, \rho_W)$  lies on constant surfaces that sweep the  $\rho_{UV}$  axis. These surfaces are fairly planar for negative  $X_s$  and tend to bend (like the pages of a book) for positive  $X_s$ . What is important to see is that  $X_s(\rho_{UV}, \rho_{VW}, \rho_W)$  appears to vary in a one-dimensional fashion



as  $\sim X_s(\rho_{UV})$ . Due to the symmetry of the system of equations,

$Y_s(\rho_{UV}, \rho_{VW}, \rho_W)$  and  $Z_s(\rho_{UV}, \rho_{VW}, \rho_W)$  behave in a similar manner.

The algorithm used to generate our first guess is based upon the behavior in Figure 37. Treating the problem as quasi-one-dimensional, we treat each input coordinate  $(\rho_{UV}, \rho_{VW}, \rho_W)$  separately. The algorithm is developed for the  $\rho_{UV}$  case because there is exchange symmetry among  $\rho_{UV}$ ,  $\rho_{VW}$ , and  $\rho_W$ . The algorithm is as follows.

First, point data was generated for three constant  $X_s$  surfaces:

$X_s = -5$ ,  $X_s = 0$ , and  $X_s = 5$ . These surfaces are chosen since it is over this range that the asymptotic behavior breaks down. Each set of points was fit using TableCurve® 3D to generate an estimate of the surface equations for the three cases. This was done once and put into a Fortran module. These equations and their statistics are presented below:

$$\rho_{UV}^{X=5}(\rho_{VW}, \rho_W) = 0.191232898 + \rho_{VW} * (-0.00879861 + -0.17756583 * \rho_{VW}) + \rho_W * (-0.00919211 + -0.1772146 * \rho_W) + -0.31946783 * \rho_{VW} * \rho_W \quad (106)$$

$$R^2 = 0.9986184864,$$

$$\text{Standard Error} = 0.0021141815,$$

$$\begin{aligned}
\rho_{UV}^{X=0}(\rho_{VW}, \rho_W) &= 0.5 - 0.5 * \rho_{VW} - 0.5 * \rho_W, \\
R^2 &= 1.0, \\
\text{Standard Error} &= 1.16187e - 10
\end{aligned} \tag{107}$$

and

$$\begin{aligned}
\rho_{UV}^{X=5}(\rho_{VW}, \rho_W) &= 0.808632414 + \rho_{VW} * (-0.98752806 + 0.17397106 * \rho_{VW}) + \\
&\quad \rho_W * (-0.98678519 + 0.173236422 * \rho_W) + 0.309475303 * \rho_{VW} * \rho_W \\
R^2 &= 0.9999311819, \\
\text{Standard Error} &= 0.0020301284.
\end{aligned} \tag{108}$$

The above equations are used to locate the input  $\rho_{UV}$  coordinate with respect to each surface. The exchange symmetry is such that  $\rho_{VW} \rightarrow \rho_W$  and  $\rho_W \rightarrow \rho_{UV}$  for  $Y_s$ , and  $\rho_{VW} \rightarrow \rho_{UV}$  and  $\rho_W \rightarrow \rho_{VW}$  for  $Z_s$  where  $\rho_{UV}$ ,  $\rho_{VW}$ , and  $\rho_W$  are the variables in equations (106) through (108).

Locating  $\rho_{UV}$  with respect to the above surfaces, the value of  $X_s$  is approximated in one of four ways. For  $\rho_{UV}$  to the left of the  $\rho_{UV}^{X=5}$  surface ( $\rho_{UV}^{X=5} - \rho_{UV} > 0$ ),  $X_s = \frac{1}{\rho_{UV}} - \frac{1}{1 - \rho_{Sum}}$ . Note that this is the same form as the asymptotic form in Table 19. The reason for this is that the asymptotic solution is a very good approximation in this range (accurate to three or more digits). When  $\rho_{UV}$  is between the  $X_s = 5.0$  and  $X_s = 0.0$  surfaces, the following fit is used to obtain  $X_s$ :

$$\begin{aligned}
x_{fit}(\rho_{VW}) &= -\log_e(\rho_{VW}) \\
y_{fit}(\rho_{Sum}) &= -\log_e(1 - \rho_{Sum}) \\
X_s(\rho_{VW}, \rho_{Sum}) &= 1.305400076 + x_{fit} * (-4.39382576 + x_{fit} * (7.230625174 + \\
&\quad x_{fit} * (-2.82155197 + 0.532512991 * x_{fit}))) + \\
&\quad y_{fit} * (1.123135173 + y_{fit} * (-4.28162193 + \\
&\quad y_{fit} * (1.623314683 + y_{fit} * (-0.31139215 + \\
&\quad -0.01494551 * y_{fit})))),
\end{aligned} \tag{109}$$

$$R^2 = 0.9994254009,$$

$$\text{Standard Error} = 0.0363020055.$$

In the case where  $\rho_{UV}$  is between the  $X_s = 0.0$  and  $X_s = -5.0$  surfaces, the fit below is used to calculate  $X_s$ :

$$\begin{aligned}
x_{fit}(\rho_{VW}) &= -\log_e(\rho_{VW}) \\
y_{fit}(\rho_{Sum}) &= -\log_e(1 - \rho_{Sum}) \\
X_s(\rho_{VW}, \rho_{Sum}) &= -1.10819185 + x_{fit} * (-1.18202335 + \\
&\quad x_{fit} * (4.431595918 + x_{fit} * (-1.74308591 + \\
&\quad x_{fit} * (0.341900634 + 0.012610457 * x_{fit})))) + \\
&\quad y_{fit} * (3.989171814 + y_{fit} * (-6.98404183 + \\
&\quad y_{fit} * (2.779136284 + -0.53138969 * y_{fit}))) ,
\end{aligned} \tag{110}$$

$$R^2 = 0.9994254009,$$

$$\text{Standard Error} = 0.0363020055.$$

Finally, if the value of  $\rho_{UV}$  is to the right of the  $X_s = -5.0$  surface

( $\rho_{UV}^{X=-5} - \rho_{UV} < 0$ ) the appropriate asymptotic solution (Table 19) is used

because these solutions are generally good to many digits in this range.

The source system root-solving algorithm is summarized as follows.

First, we evaluate the source ratios  $\rho_{UV}$ ,  $\rho_{VW}$ , and  $\rho_W$  using source data

from the previous iteration. If this data falls in the range of accuracy of the asymptotic limit for the solution to the system, we use the asymptotic solution and exit the algorithm. Otherwise, we use the input data to generate a first guess for  $X_s$ ,  $Y_s$ , and  $Z_s$  using the first guess algorithm which uses either the asymptotic solutions or a fit to the data within several regions in the solution space. Since the system is symmetric, we use the same fit equations for each coefficient and exchange variables in a consistent way. After obtaining the guess for the coefficients, Broyden's method is used to invert the system. With the coefficients in hand, we back out the needed coefficients ( $\tilde{B}$  and  $A_s$ ) and proceed with the transport algorithm for the cell. As discussed in the Chapter 4, a problem was found with the break point between the asymptotic solution and the root-solver. The solution was to change the break point to force root-solving to occur more often. The previous algorithms embody this strategy.

It is probably possible do a lot better than the above algorithm. The fit equations, although picked from the best available, could be better. The logarithmic versions were used since they seem to better replicate the behavior of the solution of this exponential system. Additionally, we could have broken the space up into more regions and fitted these with better functions. Finally, with more time, a family of parametric curves might have been found that governs the solution space behavior, perhaps yielding good enough solutions to eliminate root-solving entirely. Regardless, the above

guess generator is good enough to allow Broyden's method to converge on the average of 2 or 3 iterations and no more than 5, satisfying our need for a good guess generator.

### Inflow Flux Coefficients

The coefficients for the inflow flux moments are obtained in a manner similar to the source coefficients. The equations for the inflow flux moments are

$$\Psi_A = 2\exp(A_f)\mathcal{M}_0(X_f, Y_f), \quad (111)$$

$$\Psi_U^f = \Psi_A [1 - \rho(X_f, Y_f)], \quad (112)$$

and 
$$\Psi_V^f = \Psi_U - \Psi_A \mathcal{R}_1(X_f, Y_f). \quad (113)$$

As before with the source moments, equations (112) and (113) are divided by

(111) and define  $\rho_{UV}^f = \frac{\Psi_U^{\text{in}} - \Psi_V^{\text{in}}}{\Psi_A^{\text{in}}}$  and  $\rho_V^f = \frac{\Psi_V^{\text{in}}}{\Psi_A^{\text{in}}}$ . Then, using Identity 6

(Appendix A) the face system becomes

$$\rho_{UV}^f = \mathcal{R}_1(X_f, Y_f) \quad (114)$$

and 
$$\rho_V^f = \mathcal{R}_2(X_f, Y_f). \quad (115)$$

The above system is the two-dimensional analog of the three-dimensional system for the source. It is non-linear and requires root solving. The solution approach for the coefficients is the same as the source system,

i.e. use the asymptotic behavior of  $X_f(\rho_{UV}^f, \rho_V^f)$  and  $Y_f(\rho_{UV}^f, \rho_V^f)$  as the basis for a first guess algorithm and employ Broyden's method as the root-solver.

After obtaining  $X_f$  and  $Y_f$ , the needed coefficients are calculated just as was done for the source coefficients (except there may be up to 3 inflow faces). The coefficients are

$$\begin{aligned}\tilde{B}_u^f &= -X_f, \\ \tilde{B}_v^f &= X_f - Y_f,\end{aligned}\tag{116}$$

and 
$$A^f = \log_e \left( \frac{\Psi_A}{2\mathcal{M}_0(X_f, Y_f)} \right).\tag{117}$$

The following sections present the inflow flux root-solver equations and the first guess algorithm.

### **Inflow Flux System Root Solver**

As with the source, we chose to use Broyden's method to invert the inflow flux system of equations. Thus, these equations must be cast into the appropriate format along with the system's Jacobian.

The appropriately cast system for the above strategy is given as

$$\begin{aligned}f_1(X_f, Y_f) &= \mathcal{R}_1(X_f, Y_f) - \rho_{uv} = 0, \\ f_2(X_f, Y_f) &= \mathcal{R}_2(X_f, Y_f) - \rho_v = 0.\end{aligned}\tag{118}$$

The Jacobian for the above system,  $J^f$ , is

$$\mathbf{J}^f = \begin{pmatrix} J_{11}^f & J_{12}^f \\ J_{21}^f & J_{22}^f \end{pmatrix} \quad (119)$$

where

$$\begin{aligned} J_{11}^f &= \frac{\partial f_1}{\partial X_f} = \mathcal{R}_1(X_f, Y_f) [\mathcal{R}_1(X_f, Y_f) - 2\mathcal{R}_1(X_f, X_f, Y_f)], \\ J_{12}^f &= \frac{\partial f_1}{\partial Y_f} = \mathcal{R}_2(X_f, Y_f) [\mathcal{R}_1(X_f, Y_f) - \mathcal{R}_1(X_f, Y_f, Y_f)], \\ J_{21}^f &= \frac{\partial f_2}{\partial X_f} = J_{12}^f, \\ J_{22}^f &= \frac{\partial f_2}{\partial Y_f} = \mathcal{R}_2(X_f, Y_f) [\mathcal{R}_2(X_f, Y_f) - 2\mathcal{R}_2(X_f, Y_f, Y_f)]. \end{aligned} \quad (120)$$

As with the source system, the evaluation of  $\mathbf{J}^f$  is computationally expensive.

In this case, it is efficient to re-use the three-argument to two-argument moment function ratios since they are needed as part of the evaluation of  $f_1$  and  $f_2$ . The evaluation of the repeated argument  $\mathcal{R}_j$  functions is unavoidable. Thus, Broyden's method was chosen to avoid multiple evaluations of  $\mathbf{J}^f$ . The first guess algorithm for this system is presented below. The algorithm used to numerically evaluate the inflow face flux system and its Jacobian is presented in Appendix C.

### **Inflow Flux First Guess Algorithm**

The first guess algorithm for the inflow face flux system of equations is the two-dimensional analog of the three-dimensional version used for the source system. Indeed, the inflow face flux exhibits the same type of solution

layering shown for the three-dimensional version. In the case of the face flux, the phase space is a unit triangle instead of a tetrahedron (Figure 38).

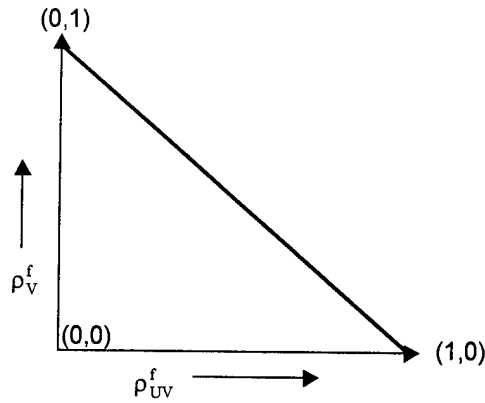


Figure 38. The  $(\rho_{UV}^f, \rho_V^f)$  Phase Space.

As before with the source system, the solutions to the face system lie on well-defined curves of constant  $X_f(\rho_{UV}^f, \rho_V^f)$  (and  $Y_f(\rho_{UV}^f, \rho_V^f)$  by symmetry). This behavior is depicted in Figure 39.



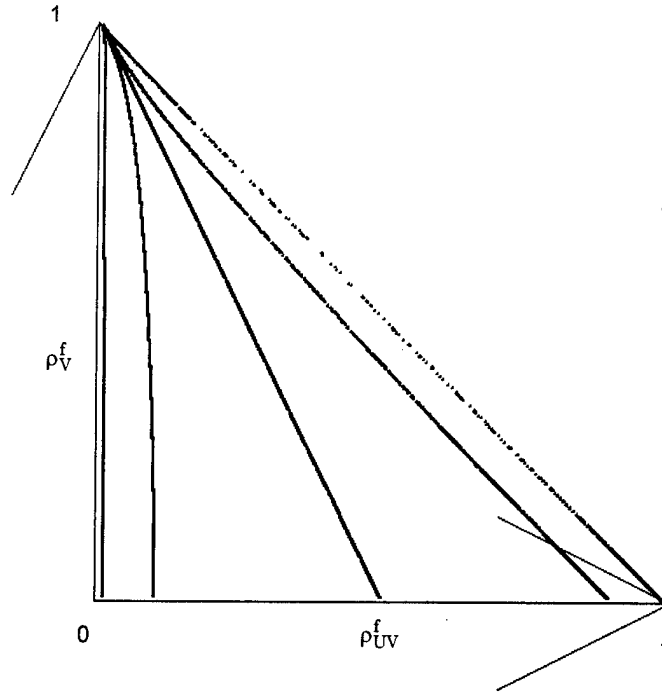


Figure 39. Plot of  $X_f(\rho_{UV}^f, \rho_V^f)$  for  $X_f = -100, -10, 0, 10, 100$  (right to left).

Just as was observed for the source, the above solution space maps the  $\rho^f$  coordinates, each of which is constrained to lie between 0 and 1, to the entire set of real numbers for  $X_f$  and  $Y_f$ .

The first guess strategy is essentially the same as for the source system. The inflow flux parameters  $\rho_{UV}^f$  and  $\rho_V^f$  are treated in a quasi-one-dimensional way by treating each coordinate independently. This can be done because the solutions,  $X_f(\rho_{UV}^f, \rho_V^f)$  and  $Y_f(\rho_{UV}^f, \rho_V^f)$ , seem to vary more

strongly with one coordinate than the other. The algorithm for obtaining  $X_f(\rho_{UV}^f, \rho_V^f)$  is presented because the same formulas apply for  $Y_f(\rho_{UV}^f, \rho_V^f)$  by letting  $\rho_{UV}^f = \rho_V^f$  in the equations below.

The algorithm begins with the asymptotic solutions to the face system in the cases where both  $\rho_{UV}^f$  and  $\rho_V^f$  are approaching 0 or one or the other is approaching 1. As before,  $0 < \rho_{Sum}^f = \rho_{UV}^f + \rho_V^f < 1$ .

Table 20. Asymptotic Solutions for Inflow Face System.

	$\rho_{UV}^f \rightarrow 1$ $\rho_V^f \rightarrow 0$ $\rho_{Sum}^f \rightarrow 1$	$\rho_{UV}^f \rightarrow 0$ $\rho_V^f \rightarrow 1$ $\rho_{Sum}^f \rightarrow 1$	$\rho_{UV}^f \rightarrow 0$ $\rho_V^f \rightarrow 0$ $\rho_{Sum}^f \rightarrow 0$
$X_f$	$-\frac{1}{1 - \rho_{Sum}^f}$	$\frac{1}{\rho_{UV}^f} - \frac{1}{1 - \rho_{Sum}^f}$	$\frac{1}{\rho_{UV}^f}$
$Y_f$	$\frac{1}{\rho_V^f} - \frac{1}{1 - \rho_{Sum}^f}$	$-\frac{1}{1 - \rho_{Sum}^f}$	$\frac{1}{\rho_V^f}$

The asymptotic relations in the first two columns of Table 20 are accurate to at least  $10^{-9}$  when  $\rho_{Sum}^f \geq 0.99$ . Column three is at least this accurate when  $\rho_{Sum}^f \leq 0.01$ . Thus, the relations in Table 20 are used as the solutions to the face system when they are applicable. Otherwise, the first guess algorithm is used.

The first guess algorithm uses fits to a family of constant  $X_f$  curves to determine the location of the  $\rho_{UV}^f$  input coordinate (exchange symmetry is used to get  $\rho_V^f$  using the same fits). In the case of the face system, we fit the

$X_f = 10, 1, -1$ , and  $-5$  curves using TableCurve® 2D. This was done once and the results placed in a Fortran module in TETRAN. These curves are used to bracket  $\rho_{UV}^f$  in the solution space. The equations of the fit curves are shown below:

$$\begin{aligned}\rho_{UV}^{f,X=10}(\rho_V^f) &= 0.100241456 + \rho_V^f * (-0.01458791 + \\ &\quad \rho_V^f * (0.139338267 + \rho_V^f * (-0.74015265 + \\ &\quad \rho_V^f * (1.114373468 + -0.59888171 * \rho_V^f))), \\ R^2 &= 0.9999588005, \\ \text{Standard Error} &= 0.0001883807,\end{aligned}\tag{121}$$

$$\begin{aligned}\rho_{UV}^{f,X=1}(\rho_V^f) &= 0.4185662 + \rho_V^f * (-0.35932296 + -0.05794379 * \rho_V^f), \\ R^2 &= 0.9999873181, \\ \text{Standard Error} &= 0.0004786178,\end{aligned}\tag{122}$$

$$\begin{aligned}\rho_{UV}^{f,X=-1}(\rho_V^f) &= 0.581439345 + \rho_V^f * (-0.64121428 + 0.058551053 * \rho_V^f), \\ R^2 &= 0.9999943081, \\ \text{Standard Error} &= 0.0004508165,\end{aligned}\tag{123}$$

and

$$\begin{aligned}\rho_{UV}^{f,X=-5}(\rho_V^f) &= 0.807850555 + \rho_V^f * (-0.98086771 + 0.165769953 * \rho_V^f), \\ R^2 &= 0.9999370731, \\ \text{Standard Error} &= 0.002139227.\end{aligned}\tag{124}$$

After using the above to determine where  $\rho_{UV}^f$  lies in the solution space, we use a fit equation appropriate to the region  $\rho_{UV}^f$  is in. For  $\rho_{UV}^f$  to the left of  $\rho_{UV}^{f,X=10}$ ,  $\rho_{UV}^{f,X=10} - \rho_{UV}^f \leq 0$ , we use the asymptotic solution

$X_f = \frac{1}{\rho_{UV}^f} - \frac{1}{1 - \rho_{Sum}^f}$ , which is accurate to several digits in this range. If  $\rho_{UV}^f$

lies between  $\rho_{UV}^{f,X=10}$  and  $\rho_{UV}^{f,X=1}$ , the following fit function is used:

$$\begin{aligned}
 x &= \frac{1}{\rho_{UV}^f} - \frac{1}{1 - \rho_{Sum}^f}, \\
 y &= \frac{1}{\rho_V^f} - \frac{1}{1 - \rho_{Sum}^f}, \\
 z1 &= x * (1.208399138 - 0.01536180 * x), \\
 z2 &= 0.5 * (1.0 + \text{erf}((y + 1.31619805) / (\sqrt{2} * 2.080453277))), \\
 X_f &= -0.56584199 + z1 + 1.024171722 * z2, \\
 R^2 &= 0.9986085942, \\
 \text{Standard Error} &= 0.0995880886,
 \end{aligned} \tag{125}$$

where  $\text{erf}(x)$  is the error function, which is evaluated efficiently using a standard algorithm or a built-in function. The above function closely resembles a CDF (cumulative distribution function) to which  $X_f(\rho_{UV}^f, \rho_V^f)$  is similar as evidenced by Figure 39. If  $\rho_{UV}^f$  lies between  $\rho_{UV}^{f,X=1}$  and  $\rho_{UV}^{f,X=-1}$ , the fit function below is used to estimate  $X_f$ :

$$\begin{aligned}
x &= \frac{1}{\rho_{UV}^f} - \frac{1}{1 - \rho_{Sum}^f}, \\
y &= \frac{1}{\rho_V^f} - \frac{1}{1 - \rho_{Sum}^f}, \\
z1 &= 0.5 * (1.0 + \\
&\quad \text{erf}((x + 0.5219783834650574) / (\sqrt{2} * 5.368613963271195))), \\
z2 &= 0.5 * (1.0 + \\
&\quad \text{erf}((y + 1.341292214090703) / (\sqrt{2} * 2.93611702967799))), \\
X_f &= -7.313430618887315 + 13.57575241380947 * z1 - \\
&\quad 3.4969015342974 * z2 + 6.489645999251979 * z1 * z2, \\
R^2 &= 0.999932859649556, \\
\text{Standard Error} &= 0.004964030286314623,
\end{aligned} \tag{126}$$

where we have again employed the error function, erf(x). In the case where  $\rho_{UV}^f$  lies between  $\rho_{UV}^{f,X=-1}$  and  $\rho_{UV}^{f,X=-5}$ , the last fit function is used, which is given as :

$$\begin{aligned}
x &= \frac{1}{\rho_{UV}^f} - \frac{1}{1 - \rho_{Sum}^f}, \\
y &= \frac{1}{\rho_V^f} - \frac{1}{1 - \rho_{Sum}^f}, \\
z1 &= x * (1.528744066931101 + 0.06995285682140669 * x), \\
z2 &= 0.5 * (1.0 + \\
&\quad \text{erf}(-(y + 4.09440766373177) / (\sqrt{2} * 2.970727912877624))), \\
X_f &= -0.02622091422930943 + z1 + 0.8776311335141779 * z2, \\
R^2 &= 0.99422295436887, \\
\text{Standard Error} &= 0.09313013094057546.
\end{aligned} \tag{127}$$

Lastly, when  $\rho_{UV}^f$  lies to the right of  $\rho_{UV}^{f,X=-5}$ , the asymptotic behavior is used

as a guess to the solution of the system:  $X_f = \frac{1}{\rho_{UV}^f} - \frac{1}{1 - \rho_{Sum}^f}$ .

The inflow face root-solving algorithm is summarized as follows. First, the inflow face flux moment ratios are computed for the current face:  $\rho_{UV}^f$  and  $\rho_V^f$ . If this data falls within the region of applicability of the asymptotic solutions to the system,  $X_f$  and  $Y_f$  are calculated directly and the next input face is evaluated. Otherwise, this data is passed to a first guess algorithm to estimate the needed  $X_f$  and  $Y_f$  using either the fits or the asymptotic values. These coefficients are used to start a Broyden's method root-solver. After successfully obtaining accurate  $X_f$  and  $Y_f$ , the needed coefficients are calculated and the next inflow face is evaluated.

The above approach is perhaps non-optimal. It requires the evaluation of  $\text{erf}(x)$ , which is efficiently evaluated using standard algorithms. Using  $\text{erf}(x)$  probably requires fewer floating-point operations than the evaluation of  $J^f$  for each face and solver iteration. However, this algorithm is efficient and ensures the convergence of the Broyden method in less than 5 iterations and generally between 2 and 3 for a solution accurate to at least 9 digits. Further work could be done in this area to better take advantage of the nature of the solution space as time warrants.

## Appendix C: Stable EC Quadrature Formulations

The development of stable algorithms to evaluate the flux moments in Chapter 2 was essential to the implementation of the EC method. This appendix outlines the algorithms used to calculate  $\psi_{A,src}^{subcell}$ ,  $\psi_{A,in}^{subcell}$ ,  $\psi_{u,src}^{subcell}$ ,  $\psi_{u,in}^{subcell}$ ,  $\psi_{v,src}^{subcell}$ ,  $\psi_{v,in}^{subcell}$ ,  $\psi_{w,src}^{subcell}$ ,  $\psi_{w,in}^{subcell}$ ,  $\psi_{A,src}^{outface}$ ,  $\psi_{A,in}^{outface}$ ,  $\psi_{u,src}^{outface}$ ,  $\psi_{u,in}^{outface}$ ,  $\psi_{v,src}^{outface}$ , and  $\psi_{v,in}^{outface}$ . Similar techniques are used to evaluate the source root solve system and its Jacobian elements and the inflow face flux system and its Jacobian elements found in Appendix B. Note that throughout the discussion below we refer to the exponential moment function identities found in Appendix A.

### IEEE® 754 Overflow and Underflow Errors

Overflow and underflow errors are produced because computers use finite precision arithmetic. An IEEE overflow is one where the resultant number is greater than  $1.7976931348623158 \times 10^{308}$  (normalized) for 64-bit (double precision) arithmetic. This corresponds approximately to  $\exp(709.78)$ .

An underflow occurs if the resultant number is less than

$2.2250738585072013 \times 10^{-308}$  (normalized) which corresponds to

approximately  $\exp(-708.4)$ . Either case is intolerable to TETRAN. In the overflow case, an error is produced and the code terminates. More insidious, the underflow case will gracefully truncate the value to 0.0 (or some other

small number) without user intervention. The EC quadratures require that  $\psi_A^{\text{subcell}} > \psi_u^{\text{subcell}} > \psi_v^{\text{subcell}} > \psi_w^{\text{subcell}}$  so that if an underflow occurs in one of the moment evaluations, care must be taken to ensure that the other moments correspondingly underflow to 0.0 as well. In some cases, a number may underflow to the un-normalized value of  $4.94065645841246544 \times 10^{-324}$  and not to 0.0. This must be prevented from occurring during the transport calculations.

### Source Contribution to Cell Flux Moments

The source contribution to the cell flux moments are the most complicated formulae; their treatment will be discussed first. In this case,  $\psi_{u,\text{src}}^{\text{subcell}}$ ,  $\psi_{v,\text{src}}^{\text{subcell}}$ , and  $\psi_{w,\text{src}}^{\text{subcell}}$  are functions of  $\mathcal{R}_j(x,y,z,w)$ , requiring the evaluation of multiple five-argument exponential moment functions. Additionally, care must be taken to prevent potential overflow errors because in some cases (particularly for poorly shaped tetrahedra) the coefficients will produce exponents much greater than 709 in the moment function routines. However, it is known from empirical evidence that  $0 < \mathcal{R}_j(x,y,z,w) < 1$ . Indeed, it is the ratio function  $\mathcal{R}_j(x,y,z,w)$  which enables the stable and robust evaluation of the EC quadrature. The pseudo-code algorithm for the evaluation of the source contribution to the flux moments is presented below.

Sort arguments  $x$ ,  $y$ ,  $z$ , and  $w$  into ascending order: the new list is  $x_1$ ,  $x_2$ ,  $x_3$ , and  $x_4$ , where  $x_1 \leq x_2 \leq x_3 \leq x_4$ . Note that the original argument list is retained.



If  $x_1 \geq 700$  and  $a_s \leq -700$  then  $\exp(a_s)$  is very small as are  $\mathcal{R}_1(x, y, z, w), \mathcal{R}_2(x, y, z, w), \mathcal{R}_3(x, y, z, w),$  and  $\mathcal{R}_4(x, y, z, w)$ .

$$\psi_{A,src}^{subcell}, \psi_{u,src}^{subcell}, \psi_{v,src}^{subcell}, \psi_{w,src}^{subcell} \text{ are } 0$$

Else, if  $x_1 \geq 38.0$  and  $|a_s| < 700$  then use the asymptotic value of  $\mathcal{R}_j(x, y, z, w)$  and the  $\exp(a_s)$  will not overflow or underflow.

$$\mathcal{R}_1(x, y, z, w) = \frac{1}{x}, \quad \mathcal{R}_2(x, y, z, w) = \frac{1}{y}, \quad \mathcal{R}_3(x, y, z, w) = \frac{1}{z},$$

$$\mathcal{R}_4(x, y, z, w) = \frac{1}{w}$$

$$\mathcal{M}_0(x, y, z, w) = \frac{1}{xyzw}$$

$$\text{Calculate } \psi_{A,src}^{subcell}, \psi_{u,src}^{subcell}, \psi_{v,src}^{subcell}, \psi_{w,src}^{subcell}$$

Else, if  $x_1 \geq 38.0$  and  $a_s \geq 700$  then an overflow can occur in evaluating  $\psi_{A,src}^{subcell}$  as  $\exp(a_s)$  will overflow. In this case:

$$\mathcal{R}_1(x, y, z, w) = \frac{1}{x}, \quad \mathcal{R}_2(x, y, z, w) = \frac{1}{y}, \quad \mathcal{R}_3(x, y, z, w) = \frac{1}{z},$$

$$\mathcal{R}_4(x, y, z, w) = \frac{1}{w},$$

$$\mathcal{M}_0(x, y, z, w) = \frac{1}{xyzw}$$

$$\psi_{A,src}^{subcell} = \exp(a_s + \log_e(6! \mathcal{M}_0(x, y, z, w)))$$

$$\text{Calculate } \psi_{u,src}^{subcell}, \psi_{v,src}^{subcell}, \psi_{w,src}^{subcell} \text{ using quadrature formulas.}$$

Else, if  $x_1 > -700$  and  $x_4 < 700$  and  $x_1 - x_4 > -700$  and  $a_s > -700$  then use Identities 2 and 4 (Appendix A) to evaluate the  $\mathcal{R}_j$  functions using  $\rho$  functions that are calculated using our four-argument moment function routines.

$$\mathcal{R}_1(x, y, z, w) = \rho(-x, y-x, z-x, w-x)$$

$$\mathcal{R}_2(x, y, z, w) = \rho(x-y, -y, z-y, w-y)$$

$$\mathcal{R}_3(x, y, z, w) = \rho(x-z, y-z, -z, w-z)$$

$$\mathcal{R}_4(x, y, z, w) = \rho(x-w, y-w, z-w, -w)$$

Calculate  $\psi_{A,src}^{subcell}$ ,  $\psi_{u,src}^{subcell}$ ,  $\psi_{v,src}^{subcell}$ ,  $\psi_{w,src}^{subcell}$

Else, if  $x_1 > -700$  and  $x_4 < 700$  and  $x_1 - x_4 > -700$  and  $a_s \leq -700$  then use Identities 2 and 4 (Appendix A) to evaluate the  $\mathcal{R}_j$  functions using  $\rho$  functions that are calculated using our four-argument moment function routines. The exponential,  $\exp(a_s)$ , will underflow so use the logarithmic formulation presented previously for  $\psi_{A,src}^{subcell}$ .

$$\mathcal{R}_1(x, y, z, w) = \rho(-x, y-x, z-x, w-x)$$

$$\mathcal{R}_2(x, y, z, w) = \rho(x-y, -y, z-y, w-y)$$

$$\mathcal{R}_3(x, y, z, w) = \rho(x-z, y-z, -z, w-z)$$

$$\mathcal{R}_4(x, y, z, w) = \rho(x-w, y-w, z-w, -w)$$

$$\psi_{A,src}^{subcell} = \exp(a_s + \log_e(6l M_0(x, y, z, w)))$$

Calculate  $\psi_{u,src}^{subcell}$ ,  $\psi_{v,src}^{subcell}$ ,  $\psi_{w,src}^{subcell}$  using quadrature formulas.

Else, use a stable IEEE 754 compliant routine to evaluate

$\mathcal{R}_1(x, y, z, w)$ ,  $\mathcal{R}_2(x, y, z, w)$ ,  $\mathcal{R}_3(x, y, z, w)$ , and  $\mathcal{R}_4(x, y, z, w)$ . In this algorithm, we apply the appropriate set of identities for the input argument order of which there are 24 different cases. This is because the argument order is sorted in ascending order but the original arguments must be known in order to use the correct algorithm. Thus, the sorted argument lists could be one of the following:  $xyzw$ ,  $xywz$ ,  $xzyw$ ,  $xzwy$ ,  $xwyz$ ,  $xwzy$ ,  $yxzw$ ,  $yxwz$ ,  $yzxw$ ,  $yzwx$ ,  $ywxz$ ,  $ywzx$ ,  $zxyw$ ,  $zxwy$ ,  $zyxw$ ,  $zywx$ ,  $zwxy$ ,  $zwyx$ ,  $wxyz$ ,  $wxzy$ ,  $wyxz$ ,  $wyzx$ ,  $wzxy$ , and  $wzyx$ . It is from one of the above argument lists that the needed  $\mathcal{R}_j$  functions are calculated. The example algorithm for the  $xyzw$  case is presented below. This algorithm can be permuted for any of the above given cases.

Using Identities 2 and 4 (Appendix A):

$$\mathcal{R}_1(x, y, z, w) = \rho(-x, y-x, z-x, w-x)$$

Using Identities 1 and 2 (Appendix A):

$$\mathcal{R}_2(x, y, z, w) = \frac{M_0(-x, y-x, y-x, z-x) - M_0(y-x, y-x, z-x, w-x)}{M_0(-x, y-x, z-x) - M_0(y-x, z-x, w-x)}$$

$$\mathcal{R}_3(x, y, z, w) = \frac{M_0(-x, y-x, z-x, z-x) - M_0(y-x, z-x, z-x, w-x)}{M_0(-x, y-x, z-x) - M_0(y-x, z-x, w-x)}$$

$$\mathcal{R}_4(x, y, z, w) = \frac{M_0(-x, y-x, z-x, w-x) - M_0(y-x, z-x, w-x, w-x)}{M_0(-x, y-x, z-x) - M_0(y-x, z-x, w-x)}$$

Using Identity 2 (Appendix A):

$$\psi_{A,src}^{subcell} = 6l \exp(a_s - x) \mathcal{M}_0(-x, y - x, z - x, w - x)$$

Calculate  $\psi_{u,src}^{subcell}$ ,  $\psi_{v,src}^{subcell}$ ,  $\psi_{w,src}^{subcell}$  using quadrature formulas.

The above IEEE 754 compliant algorithm uses Identity 1 to avoid the need for a general five-argument exponential moment function routine ( $\mathcal{M}_0$  and  $\rho$ ). If such a routine were available, the above algorithm would be much less complicated because one would only worry about which coefficient was x1.

### Inflow Flux Contribution to Cell Flux Moments

The algorithm for calculating  $\psi_{A,in}^{subcell}$ ,  $\psi_{u,in}^{subcell}$ ,  $\psi_{v,in}^{subcell}$ , and  $\psi_{w,in}^{subcell}$  is analogous to the algorithm above except that  $\mathcal{R}_j(x, y, z)$  is the needed function.

Sort arguments  $x$ ,  $y$ , and  $z$  into ascending order: the new list is  $x_1$ ,  $x_2$ , and  $x_3$ , where  $x_1 \leq x_2 \leq x_3$ . Note that the original argument list is retained.

If  $x_1 \geq 700$  and  $a_f \leq -700$  then  $\exp(a_f)$  is very small as are  $\mathcal{R}_1(x, y, z)$ ,  $\mathcal{R}_2(x, y, z)$ , and  $\mathcal{R}_3(x, y, z)$ .

$$\psi_{A,in}^{subcell}, \psi_{u,in}^{subcell}, \psi_{v,in}^{subcell}, \psi_{w,in}^{subcell} \text{ are } 0$$

Else, if  $x_1 \geq 38.0$  and  $|a_f| < 700$  then use the asymptotic value of  $\mathcal{R}_j(x, y, z)$  and the  $\exp(a_f)$  will not overflow or underflow.

$$\mathcal{R}_1(x, y, z) = \frac{1}{x}, \quad \mathcal{R}_2(x, y, z) = \frac{1}{y}, \quad \mathcal{R}_3(x, y, z) = \frac{1}{z}$$

$$\mathcal{M}_0(x, y, z) = \frac{1}{xyz}$$

$$\text{Calculate } \psi_{A,in}^{subcell}, \psi_{u,in}^{subcell}, \psi_{v,in}^{subcell}, \psi_{w,in}^{subcell}$$

Else, if  $x_1 \geq 38.0$  and  $a_f \geq 700$  then care must be taken in evaluating  $\psi_{A,in}^{subcell}$  as  $\exp(a_f)$  will overflow. In this case, use an alternate but equivalent form.

$$\mathcal{R}_1(x,y,z) = \frac{1}{x}, \quad \mathcal{R}_2(x,y,z) = \frac{1}{y}, \quad \mathcal{R}_3(x,y,z) = \frac{1}{z}$$

$$\mathcal{M}_0(x,y,z) = \frac{1}{xyz}$$

$$\psi_{A,in}^{subcell} = \exp(a_f + \log_e(6 \mathcal{M}_0(x,y,z)))$$

Calculate  $\psi_{u,in}^{subcell}$ ,  $\psi_{v,in}^{subcell}$ ,  $\psi_{w,in}^{subcell}$  using quadrature formulas.

Else, if  $x_1 > -700$  and  $x_3 < 700$  and  $x_1 - x_3 > -700$  and  $a_f > -700$  then use Identities 2 and 4 (Appendix A) to evaluate the  $\mathcal{R}_j$  functions using  $\rho$  functions that are calculated using the three-argument moment function routines.

$$\mathcal{R}_1(x,y,z) = \rho(-x, y-x, z-x)$$

$$\mathcal{R}_2(x,y,z) = \rho(x-y, -y, z-y)$$

$$\mathcal{R}_3(x,y,z) = \rho(x-z, y-z, -z)$$

Calculate  $\psi_{A,in}^{subcell}$ ,  $\psi_{u,in}^{subcell}$ ,  $\psi_{v,in}^{subcell}$ ,  $\psi_{w,in}^{subcell}$

Else, if  $x_1 > -700$  and  $x_3 < 700$  and  $x_1 - x_3 > -700$  and  $a_f \leq -700$  then use Identities 2 and 4 (Appendix A) to evaluate the  $\mathcal{R}_j$  functions using  $\rho$  functions that are calculated using three-argument moment function routines. The exponential,  $\exp(a_f)$ , will underflow so use the logarithmic formulation presented previously for  $\psi_{A,in}^{subcell}$ .

$$\mathcal{R}_1(x,y,z) = \rho(-x, y-x, z-x)$$

$$\mathcal{R}_2(x,y,z) = \rho(x-y, -y, z-y)$$

$$\mathcal{R}_3(x,y,z) = \rho(x-z, y-z, -z)$$

$$\psi_{A,in}^{subcell} = \exp(a_f + \log_e(6 \mathcal{M}_0(x,y,z)))$$

Calculate  $\psi_{u,in}^{subcell}$ ,  $\psi_{v,in}^{subcell}$ ,  $\psi_{w,in}^{subcell}$  using quadrature formulas.

Else, use a stable IEEE 754 compliant routine to evaluate

$\mathcal{R}_1(x,y,z), \mathcal{R}_2(x,y,z),$  and  $\mathcal{R}_3(x,y,z)$ . In this algorithm, use the appropriate set of identities for the input argument order of which we are only interested in three cases:  $x_1 = x$ ,  $x_1 = y$ , or  $x_1 = z$ . This is different from above because one had to know the argument order to use

Identity 1 effectively. Here, because we have a general four-argument moment function routine, no concern is required about the order following  $x_1$ . The example algorithm for  $x_1 = x$  is present below. This algorithm can be permuted for the other two cases.

Using Identities 2 and 4 (Appendix A):

$$\mathcal{R}_1(x, y, z) = \rho(-x, y - x, z - x)$$

Using Identities 1 and 2 (Appendix A):

$$\mathcal{R}_2(x, y, z) = \frac{\mathcal{M}_0(-x, y - x, y - x, z - x)}{\mathcal{M}_0(-x, y - x, z - x)}$$

$$\mathcal{R}_3(x, y, z) = \frac{\mathcal{M}_0(-x, y - x, z - x, z - x)}{\mathcal{M}_0(-x, y - x, z - x)}$$

Using Identity 2 (Appendix A):

$$\psi_{A, \text{in}}^{\text{subcell}} = 6 \exp(a_f - x) \mathcal{M}_0(-x, y - x, z - x)$$

Calculate  $\psi_{u, \text{in}}^{\text{subcell}}$ ,  $\psi_{v, \text{in}}^{\text{subcell}}$ ,  $\psi_{w, \text{in}}^{\text{subcell}}$  using quadrature formulas.

## Source Contribution to Outflow Face Flux Moments

The algorithm used to determine  $\psi_{A, \text{src}}^{\text{outface}}$ ,  $\psi_{u, \text{src}}^{\text{outface}}$ , and  $\psi_{v, \text{src}}^{\text{outface}}$  is

identical to that for the inflow flux contribution to the cell flux moments. The only difference is the requirement for three flux moments and the equation

for  $\psi_{A, \text{src}}^{\text{outface}}$ .

Sort arguments  $x$ ,  $y$ , and  $z$  into ascending order: the new list is  $x_1$ ,  $x_2$ , and  $x_3$ , where  $x_1 \leq x_2 \leq x_3$ . Note that the original argument list is retained.

If  $x_1 \geq 700$  and  $a_s \leq -700$  then  $\exp(a_s)$  is very small as are

$\mathcal{R}_1(x, y, z)$ ,  $\mathcal{R}_2(x, y, z)$ , and  $\mathcal{R}_3(x, y, z)$ .

$\psi_{A, \text{src}}^{\text{outface}}$ ,  $\psi_{u, \text{src}}^{\text{outface}}$ , and  $\psi_{v, \text{src}}^{\text{outface}}$  are 0

Else, if  $x_1 \geq 38.0$  and  $|a_s| < 700$  then use the asymptotic value of

$\mathcal{R}_j(x, y, z)$  and the  $\exp(a_s)$  will not overflow or underflow.

$$\mathcal{R}_1(x, y, z) = \frac{1}{x}, \quad \mathcal{R}_2(x, y, z) = \frac{1}{y}, \quad \mathcal{R}_3(x, y, z) = \frac{1}{z}$$

$$\mathcal{M}_0(x,y,z) = \frac{1}{xyz}$$

Calculate  $\psi_{A,src}^{outface}$ ,  $\psi_{u,src}^{outface}$ , and  $\psi_{v,src}^{outface}$

Else, if  $x_1 \geq 38.0$  and  $a_s \geq 700$  then care must be taken in evaluating  $\psi_{A,src}^{outface}$  as  $\exp(a_s)$  will overflow. In this case, use an alternate but equivalent form.

$$\mathcal{R}_1(x,y,z) = \frac{1}{x}, \quad \mathcal{R}_2(x,y,z) = \frac{1}{y}, \quad \mathcal{R}_3(x,y,z) = \frac{1}{z}$$

$$\mathcal{M}_0(x,y,z) = \frac{1}{xyz}$$

$$\psi_{A,src}^{outface} = \exp(a_s + \log_e(2l \mathcal{M}_0(x,y,z)))$$

Calculate  $\psi_{u,src}^{outface}$ , and  $\psi_{v,src}^{outface}$  using quadrature formulas.

Else, if  $x_1 > -700$  and  $x_3 < 700$  and  $x_1 - x_3 > -700$  and  $a_s > -700$  then use Identities 2 and 4 (Appendix A) to evaluate the  $\mathcal{R}_j$  functions using  $\rho$  functions that are calculated using the three-argument moment function routines.

$$\mathcal{R}_1(x,y,z) = \rho(-x, y-x, z-x)$$

$$\mathcal{R}_2(x,y,z) = \rho(x-y, -y, z-y)$$

$$\mathcal{R}_3(x,y,z) = \rho(x-z, y-z, -z)$$

Calculate  $\psi_{A,src}^{outface}$ ,  $\psi_{u,src}^{outface}$ , and  $\psi_{v,src}^{outface}$

Else, if  $x_1 > -700$  and  $x_3 < 700$  and  $x_1 - x_3 > -700$  and  $a_s \leq -700$  then use Identities 2 and 4 (Appendix A) to evaluate the  $\mathcal{R}_j$  functions using  $\rho$  functions that are calculated using the three-argument moment function routines. The exponential,  $\exp(a_s)$ , will underflow so use the logarithmic formulation presented previously for  $\psi_{A,src}^{outface}$ .

$$\mathcal{R}_1(x,y,z) = \rho(-x, y-x, z-x)$$

$$\mathcal{R}_2(x,y,z) = \rho(x-y, -y, z-y)$$

$$\mathcal{R}_3(x,y,z) = \rho(x-z, y-z, -z)$$

$$\psi_{A,src}^{outface} = \exp(a_s + \log_e(2l \mathcal{M}_0(x,y,z)))$$

Calculate  $\psi_{u,src}^{outface}$ , and  $\psi_{v,src}^{outface}$  using quadrature formulas.

Else, use a stable IEEE 754 compliant routine to evaluate

$\mathcal{R}_1(x,y,z)$ ,  $\mathcal{R}_2(x,y,z)$ , and  $\mathcal{R}_3(x,y,z)$ . In this algorithm, apply the appropriate set of identities for the input argument order of which we are interested in only three cases:  $x_1 = x$ ,  $x_1 = y$ , or  $x_1 = z$ . This is different from above because one had to know the argument order to use Identity 1 effectively. Here, because there is a general four-argument moment function routine, there is no concern about the order following  $x_1$ . The example algorithm for  $x_1 = x$  is present below. This algorithm can be permuted for the other two cases.

Using Identities 2 and 4 (Appendix A):

$$\mathcal{R}_1(x,y,z) = \rho(-x, y-x, z-x)$$

Using Identities 1 and 2 (Appendix A):

$$\mathcal{R}_2(x,y,z) = \frac{\mathcal{M}_0(-x, y-x, y-x, z-x)}{\mathcal{M}_0(-x, y-x, z-x)}$$

$$\mathcal{R}_3(x,y,z) = \frac{\mathcal{M}_0(-x, y-x, z-x, z-x)}{\mathcal{M}_0(-x, y-x, z-x)}$$

Using Identity 2 (Appendix A):

$$\psi_{A,src}^{outface} = 2l \exp(a_s - x) \mathcal{M}_0(-x, y-x, z-x)$$

Calculate  $\psi_{u,src}^{outface}$ , and  $\psi_{v,src}^{outface}$  using quadrature formulas.

## Inflow Flux Contribution to the Outflow Face Flux Moments

As before, this case is just a lower dimensional version of the preceding cases. This time, the quadrature formulas depend on  $\mathcal{R}_j(x,y)$ . The flux

moments calculated are  $\psi_{A,in}^{outface}$ ,  $\psi_{u,in}^{outface}$ , and  $\psi_{v,in}^{outface}$ .

Sort arguments  $x$ , and  $y$  into ascending order: the new list is  $x_1$  and  $x_2$ , where  $x_1 \leq x_2$ . Note that the original argument list is retained.

If  $x_1 \geq 700$  and  $a_f \leq -700$  then  $\exp(a_f)$  is very small as are  $\mathcal{R}_1(x,y)$  and  $\mathcal{R}_2(x,y)$ .

$$\psi_{A,in}^{outface}, \psi_{u,in}^{outface}, \text{ and } \psi_{v,in}^{outface} \text{ are } 0$$

Else, if  $x_1 \geq 38.0$  and  $|a_f| < 700$  then we can use the asymptotic value of  $\mathcal{R}_j(x,y)$  and the  $\exp(a_f)$  will not overflow or underflow.

$$\mathcal{R}_1(x,y) = \frac{1}{x}, \quad \mathcal{R}_2(x,y) = \frac{1}{y}$$

$$\mathcal{M}_0(x,y) = \frac{1}{xy}$$

Calculate  $\psi_{A,in}^{\text{outface}}$ ,  $\psi_{u,in}^{\text{outface}}$ , and  $\psi_{v,in}^{\text{outface}}$ .

Else, if  $x_1 \geq 38.0$  and  $a_f \geq 700$  then care must be taken in evaluating  $\psi_{A,in}^{\text{outface}}$  as  $\exp(a_f)$  will overflow. In this case, use an alternate but equivalent form.

$$\mathcal{R}_1(x,y) = \frac{1}{x}, \quad \mathcal{R}_2(x,y) = \frac{1}{y}$$

$$\mathcal{M}_0(x,y) = \frac{1}{xy}$$

$$\psi_{A,in}^{\text{outface}} = \exp(a_f + \log_e(2\mathcal{M}_0(x,y)))$$

Calculate  $\psi_{u,in}^{\text{outface}}$ , and  $\psi_{v,in}^{\text{outface}}$  using quadrature formulas.

Else, if  $x_1 > -700$  and  $x_2 < 700$  and  $x_1 - x_2 > -700$  and  $a_f > -700$  then use Identities 2 and 4 (Appendix A) to evaluate the  $\mathcal{R}_j$  functions using  $\rho$  functions that are calculated using our two-argument moment function routines.

$$\mathcal{R}_1(x,y) = \rho(-x, y-x)$$

$$\mathcal{R}_2(x,y) = \rho(x-y, -y)$$

Calculate  $\psi_{A,in}^{\text{outface}}$ ,  $\psi_{u,in}^{\text{outface}}$ , and  $\psi_{v,in}^{\text{outface}}$ .

Else, if  $x_1 > -700$  and  $x_2 < 700$  and  $x_1 - x_2 > -700$  and  $a_f \leq -700$  then use Identities 2 and 4 (Appendix A) to evaluate the  $\mathcal{R}_j$  functions using  $\rho$  functions, which are calculated using the two-argument moment function routines. The exponential,  $\exp(a_f)$ , will underflow so use the logarithmic formulation presented previously for  $\psi_{A,in}^{\text{outface}}$ .

$$\mathcal{R}_1(x,y) = \rho(-x, y-x)$$

$$\mathcal{R}_2(x,y) = \rho(x-y, -y)$$

$$\psi_{A,in}^{\text{outface}} = \exp(a_f + \log_e(2\mathcal{M}_0(x,y)))$$

Calculate  $\psi_{u,in}^{\text{outface}}$ , and  $\psi_{v,in}^{\text{outface}}$  using quadrature formulas.



Else, use a stable IEEE 754 compliant routine to evaluate  $\mathcal{R}_1(x,y)$  and  $\mathcal{R}_2(x,y)$ . In this algorithm, apply the appropriate set of identities for the input argument order of which we are only interested in two cases:  $x_1 = x$  or  $x_1 = y$ . The example algorithm for  $x_1 = x$  is presented below. This algorithm can be permuted for the other case.

Using Identities 2 and 4 (Appendix A):

$$\mathcal{R}_1(x,y) = \rho(-x, y-x)$$

Using Identities 1 and 2 (Appendix A):

$$\mathcal{R}_2(x,y) = \frac{\mathcal{M}_0(-x, y-x, y-x)}{\mathcal{M}_0(-x, y-x)}$$

Using Identity 2 (Appendix A):

$$\psi_{A,in}^{\text{outface}} = 2 \exp(a_f - x) \mathcal{M}_0(-x, y-x)$$

Calculate  $\psi_{u,in}^{\text{outface}}$ , and  $\psi_{v,in}^{\text{outface}}$  using quadrature formulas.

## Source System and Jacobian Elements

For the sake of brevity, we will not present the algorithm for evaluating source system functions. The source system is comprised of  $\mathcal{R}_j(x,y,z)$  functions whose evaluation was previously discussed in the section, *Source Contribution to Outflow Face Flux Moments*.

As for the Jacobian elements, an approach similar to that discussed in *Source Contribution to Cell Flux Moments* was used. In this case, we have five-argument moment functions over three-argument function ratios that are functions of only three distinct arguments with more than one repeat coefficient. These functions are products of  $\mathcal{R}_j$  functions. Briefly, the needed Jacobian elements are:

$$\begin{aligned}
J_{11} &= \mathcal{R}_1(x, y, z) [\mathcal{R}_1(x, y, z) - 2\mathcal{R}_1(x, x, y, z)], \\
J_{12} &= \mathcal{R}_2(x, y, z) [\mathcal{R}_1(x, y, z) - \mathcal{R}_1(x, y, y, z)], \\
J_{13} &= \mathcal{R}_3(x, y, z) [\mathcal{R}_1(x, y, z) - \mathcal{R}_1(x, y, z, z)], \\
J_{21} &= J_{12}, \\
J_{22} &= \mathcal{R}_2(x, y, z) [\mathcal{R}_2(x, y, z) - 2\mathcal{R}_2(x, y, y, z)], \\
J_{23} &= \mathcal{R}_3(x, y, z) [\mathcal{R}_2(x, y, z) - \mathcal{R}_2(x, y, z, z)], \\
J_{31} &= J_{13}, \\
J_{32} &= J_{23}, \\
J_{33} &= \mathcal{R}_3(x, y, z) [\mathcal{R}_3(x, y, z) - 2\mathcal{R}_3(x, y, z, z)].
\end{aligned}$$

To evaluate the Jacobian elements that are products of  $\mathcal{R}_j$  functions, we use an algorithm similar to that used in the section, *Source Contribution to Cell Flux Moments*. The pseudo-code for this algorithm is shown below.

Sort arguments  $x$ ,  $y$ , and  $z$  into ascending order: the new list is  $x_1$ ,  $x_2$ , and  $x_3$ , where  $x_1 \leq x_2 \leq x_3$ . Note that the original argument list is retained.

If  $x_1 \geq 38.0$  then we use the asymptotic solutions for the needed ratios presented below.

$$\begin{aligned}
\mathcal{R}_1(x, y, z) \mathcal{R}_1(x, x, y, z) &= \frac{\mathcal{M}_0(x, x, x, y, z)}{\mathcal{M}_0(x, y, z)} = \frac{1}{x^2} \\
\mathcal{R}_2(x, y, z) \mathcal{R}_1(x, y, y, z) &= \frac{\mathcal{M}_0(x, x, y, y, z)}{\mathcal{M}_0(x, y, z)} = \frac{1}{xy} \\
\mathcal{R}_3(x, y, z) \mathcal{R}_1(x, y, z, z) &= \frac{\mathcal{M}_0(x, x, y, z, z)}{\mathcal{M}_0(x, y, z)} = \frac{1}{xz} \\
\mathcal{R}_2(x, y, z) \mathcal{R}_2(x, y, y, z) &= \frac{\mathcal{M}_0(x, y, y, y, z)}{\mathcal{M}_0(x, y, z)} = \frac{1}{y^2} \\
\mathcal{R}_3(x, y, z) \mathcal{R}_2(x, y, z, z) &= \frac{\mathcal{M}_0(x, y, y, z, z)}{\mathcal{M}_0(x, y, z)} = \frac{1}{yz} \\
\mathcal{R}_3(x, y, z) \mathcal{R}_3(x, y, z, z) &= \frac{\mathcal{M}_0(x, y, z, z, z)}{\mathcal{M}_0(x, y, z)} = \frac{1}{z^2}
\end{aligned}$$

Else, if  $\max[\text{abs}(x1), \text{abs}(x3)] < 1.0$  then apply Identities 2 and 4 (Appendix A) to the numerator to find the Jacobian elements.

$$\mathcal{R}_1(x, y, z) \mathcal{R}_1(x, x, y, z) = \frac{\mathcal{M}_0(x, x, x, y, z)}{\mathcal{M}_0(x, y, z)} = \exp(-x) \frac{\mathcal{M}_1(-x, 0, y-x, z-x)}{\mathcal{M}_0(x, y, z)}$$

$$\mathcal{R}_2(x, y, z) \mathcal{R}_1(x, y, y, z) = \frac{\mathcal{M}_0(x, x, y, y, z)}{\mathcal{M}_0(x, y, z)} = \exp(-x) \frac{\mathcal{M}_1(-x, y-x, y-x, z-x)}{\mathcal{M}_0(x, y, z)}$$

$$\mathcal{R}_3(x, y, z) \mathcal{R}_1(x, y, z, z) = \frac{\mathcal{M}_0(x, x, y, z, z)}{\mathcal{M}_0(x, y, z)} = \exp(-x) \frac{\mathcal{M}_1(-x, y-x, z-x, z-x)}{\mathcal{M}_0(x, y, z)}$$

$$\mathcal{R}_2(x, y, z) \mathcal{R}_2(x, y, y, z) = \frac{\mathcal{M}_0(x, y, y, y, z)}{\mathcal{M}_0(x, y, z)} = \exp(-y) \frac{\mathcal{M}_1(x-y, -y, 0, z-y)}{\mathcal{M}_0(x, y, z)}$$

$$\mathcal{R}_3(x, y, z) \mathcal{R}_2(x, y, z, z) = \frac{\mathcal{M}_0(x, y, y, z, z)}{\mathcal{M}_0(x, y, z)} = \exp(-y) \frac{\mathcal{M}_1(x-y, -y, z-y, z-y)}{\mathcal{M}_0(x, y, z)}$$

$$\mathcal{R}_3(x, y, z) \mathcal{R}_3(x, y, z, z) = \frac{\mathcal{M}_0(x, y, z, z, z)}{\mathcal{M}_0(x, y, z)} = \exp(-z) \frac{\mathcal{M}_1(x-z, y-z, -z, 0)}{\mathcal{M}_0(x, y, z)}$$

Else,  $x1 < 38.0$  but the arguments are not small. In this case, use an algorithm similar to the source contribution algorithm in which the sort list is important. Fortunately, there are only six cases: xyz, xzy, yxz, yzx, zxy, and zyx. We present the case where the sort list is xyz. The other cases are permutations on this case. Note that

$\mathcal{M}_1 = \mathcal{M}_0 \rho$ . The previous case was needed because the following algorithm is unstable for all coefficients small and close together (Identity 1).

Applying Identities 2 and 4 (Appendix A):

$$\mathcal{R}_1(x, y, z) \mathcal{R}_1(x, x, y, z) = \frac{\mathcal{M}_0(x, x, x, y, z)}{\mathcal{M}_0(x, y, z)} = \frac{\mathcal{M}_1(-x, 0, y-x, z-x)}{\mathcal{M}_0(-x, y-x, z-x)}$$

$$\mathcal{R}_2(x, y, z) \mathcal{R}_1(x, y, y, z) = \frac{\mathcal{M}_0(x, x, y, y, z)}{\mathcal{M}_0(x, y, z)} = \frac{\mathcal{M}_1(-x, y-x, y-x, z-x)}{\mathcal{M}_0(-x, y-x, z-x)}$$

$$\mathcal{R}_3(x, y, z) \mathcal{R}_1(x, y, z, z) = \frac{\mathcal{M}_0(x, x, y, z, z)}{\mathcal{M}_0(x, y, z)} = \frac{\mathcal{M}_1(-x, y-x, z-x, z-x)}{\mathcal{M}_0(-x, y-x, z-x)}$$

Applying Identities 1 and 2 (Appendix A):

$$\begin{aligned}\mathcal{R}_2(x, y, z)\mathcal{R}_2(x, y, y, z) &= \frac{\mathcal{M}_0(x, y, y, y, z)}{\mathcal{M}_0(x, y, z)} \\ &= \frac{\mathcal{M}_0(-x, y-x, y-x, y-x) - \mathcal{M}_0(y-x, y-x, y-x, z-x)}{\mathcal{M}_0(-x, y-x) - \mathcal{M}_0(y-x, z-x)}\end{aligned}$$

$$\begin{aligned}\mathcal{R}_3(x, y, z)\mathcal{R}_2(x, y, z, z) &= \frac{\mathcal{M}_0(x, y, y, z, z)}{\mathcal{M}_0(x, y, z)} \\ &= \frac{\mathcal{M}_0(-x, y-x, y-x, z-x) - \mathcal{M}_0(y-x, y-x, z-x, z-x)}{\mathcal{M}_0(-x, y-x) - \mathcal{M}_0(y-x, z-x)}\end{aligned}$$

$$\begin{aligned}\mathcal{R}_3(x, y, z)\mathcal{R}_3(x, y, z, z) &= \frac{\mathcal{M}_0(x, y, z, z, z)}{\mathcal{M}_0(x, y, z)} \\ &= \frac{\mathcal{M}_0(-x, y-x, z-x, z-x) - \mathcal{M}_0(y-x, z-x, z-x, z-x)}{\mathcal{M}_0(-x, y-x) - \mathcal{M}_0(y-x, z-x)}\end{aligned}$$

The above algorithm would be more efficient if a general five-argument moment function routine existed. However, the algorithm is stable, robust and accurate without such a routine.

### Inflow Face Flux System and Jacobian Elements

Similar to the source system, the inflow flux system is composed of  $\mathcal{R}_j(x, y)$  functions, which are evaluated using the same approach used in *Inflow Flux Contribution to the Outflow Face Flux Moments*. Thus, we will not discuss this algorithm further.

As with the source system Jacobian, the inflow face flux Jacobian elements are comprised of products  $\mathcal{R}_j(x, y)$  and products of  $\mathcal{R}_j$  functions.

Briefly, these Jacobian elements are:

$$\begin{aligned}
J_{11}^f &= \mathcal{R}_1(x, y) [\mathcal{R}_1(x, y) - 2\mathcal{R}_1(x, x, y)], \\
J_{12}^f &= \mathcal{R}_2(x, y) [\mathcal{R}_1(x, y) - \mathcal{R}_1(x, y, y)], \\
J_{21}^f &= J_{12}^f, \\
J_{22}^f &= \mathcal{R}_2(x, y) [\mathcal{R}_2(x, y) - 2\mathcal{R}_2(x, y, y)].
\end{aligned}$$

The algorithm for evaluating  $\mathcal{R}_j(x, y)$  has been presented. The algorithm used to evaluate the other  $\mathcal{R}_j$  products is analogous to that used for the source Jacobian and is shown below.

Sort arguments  $x$  and  $y$  into ascending order: the new list is  $x_1$  and  $x_2$  where  $x_1 \leq x_2$ . Note that the original argument list is retained.

If  $x_1 \geq 38.0$  then use the asymptotic solutions for the needed ratios presented below.

$$\begin{aligned}
\mathcal{R}_1(x, y)\mathcal{R}_1(x, x, y) &= \frac{\mathcal{M}_0(x, x, x, y)}{\mathcal{M}_0(x, y)} = \frac{1}{x^2} \\
\mathcal{R}_2(x, y)\mathcal{R}_1(x, y, y) &= \frac{\mathcal{M}_0(x, x, y, y)}{\mathcal{M}_0(x, y)} = \frac{1}{xy} \\
\mathcal{R}_2(x, y)\mathcal{R}_2(x, y, y) &= \frac{\mathcal{M}_0(x, y, y, y)}{\mathcal{M}_0(x, y)} = \frac{1}{y^2}
\end{aligned}$$

Else, if  $x_1 < 38.0$  use Identities 2 and 4 along with stable routines to calculate two-, three-, and four-argument moment functions. We present the case of  $x_1 = x$ . The other case mirrors this case. Note that

$$\mathcal{M}_1 = \mathcal{M}_0 \rho.$$

Applying Identities 2 and 4 (Appendix A):

$$\begin{aligned}
\mathcal{R}_1(x, y)\mathcal{R}_1(x, x, y) &= \frac{\mathcal{M}_0(x, x, x, y)}{\mathcal{M}_0(x, y)} = \frac{\mathcal{M}_1(-x, 0, y-x)}{\mathcal{M}_0(-x, y-x)} \\
\mathcal{R}_2(x, y)\mathcal{R}_1(x, y, y) &= \frac{\mathcal{M}_0(x, x, y, y)}{\mathcal{M}_0(x, y)} = \frac{\mathcal{M}_1(-x, y-x, y-x)}{\mathcal{M}_0(-x, y-x)} \\
\mathcal{R}_2(x, y)\mathcal{R}_2(x, y, y) &= \frac{\mathcal{M}_0(x, y, y, y)}{\mathcal{M}_0(x, y)} = \frac{\mathcal{M}_0(-x, y-x, y-x, y-x)}{\mathcal{M}_0(-x, y-x)}
\end{aligned}$$

## Appendix D: Derivation of the Surface Cell Algorithm

### Motivation

We began the development the surface cell spatial quadrature to address a mesh problem that is quite difficult for all current unstructured mesh generators. Imagine a simple cube surrounded by a thin wall  $1/100^{\text{th}}$  the thickness of the cube (or the thin skin of an aircraft or satellite or 55 gallon drum of biological or chemical agent). When we mesh this problem using tetrahedra, the mesh generator will do one of three things. It might produce a coarse mesh with several poor aspect ratio tetrahedra in the thin region. Or, it might attempt to fill the thin region with many (thousands) of small, well-shaped tetrahedra. Or, finally, it might crash because of the memory requirements for making a proper mesh for this type of problem. The impact of the first case is that the poorly shaped tetrahedra are not numerically well conditioned and will introduce spatial and numerical errors into the transport problem. The second case is difficult in that the small mesh dimension will propagate to areas of the mesh where we can tolerate a coarse mesh, significantly driving up the computational cost of the problem. The final case prevents the solution of the problem.

These mesh problems stem from the requirement (for transport) that each tetrahedron in a mesh share the faces of its neighbors. We require that the faces be shared because the flux moments from an upstream cell are passed to downstream cells via affine transformations of the moments on the

shared faces. Presumably, finite element algorithms for stress and thermal require connected meshes for similar reasons. The requirement for connected cells is a difficult constraint to satisfy with a minimal number of well shaped cells when the interfacing regions of a mesh differ substantially in geometric extent, i.e. thin regions next to thick regions.

### **Tetrahedron Aspect Ratio**

The aspect ratio for a tetrahedron cell is defined as  $\gamma = \frac{E}{h}$ , where  $E$  is the length of the longest edge and  $h$  is the shortest height (distance from a node to opposite face). This ratio is shown in Figure 40.

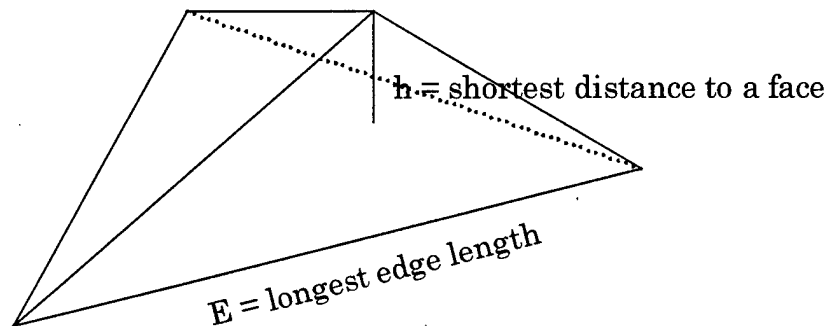


Figure 40. Definition of Tetrahedron Cell Aspect Ratio.

We can see from Figure 40 that if the cell is smashed flat or stretched long, the aspect ratio will be large. We seek to eliminate these types of cells by

using surface cells. The minimum possible aspect ratio, 1.2247, is achieved by using an equilateral (regular) tetrahedron.

Returning to our cube example, Figure 41 shows the first case (poor aspect ratio tetrahedra) for the simple cube surrounded by a thin layer of material. The mesh on the left contains 42 tetrahedra and the mesh on the right contains 4902 tetrahedra.

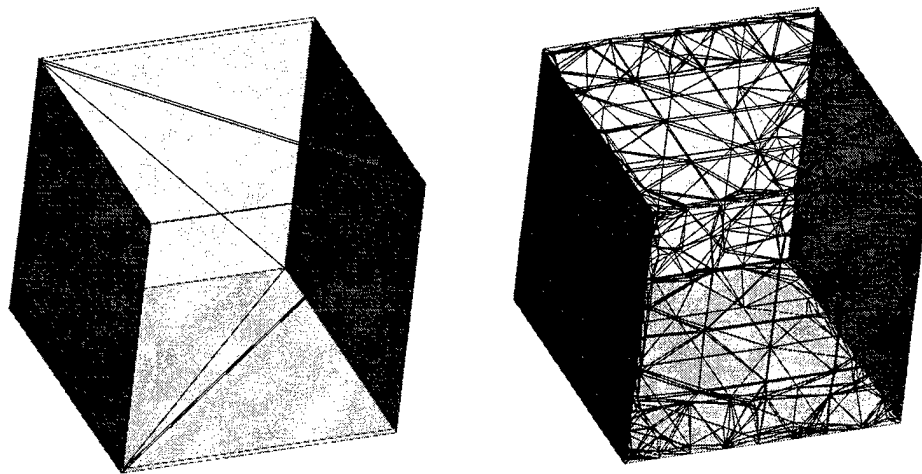


Figure 41. Examples of Unstructured Tetrahedra Meshes with Thin Regions.

In the coarse mesh case (the left cube), 36 out of the 42 tetrahedra (all in the shield) have aspect ratios of 144! Figure 42 shows a histogram of the distribution of aspect ratios for the finer mesh on the right. Note also that the right mesh is quite irregular around the corners of the cube, presumably attempting to detail possible stress risers for a structural finite element model.



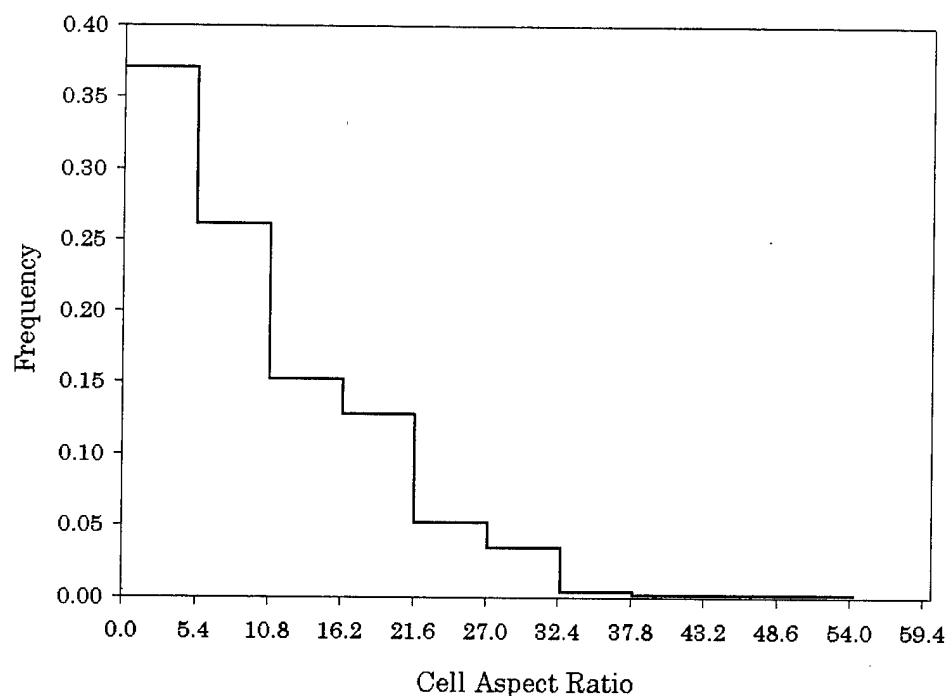


Figure 42. Distribution of Cell Aspect Ratios for Fine Meshed Example Problem.

Obviously, the distribution of aspect ratios is better than in the coarse mesh case for Figure 42. However, there are still an overwhelming fraction of cells with large aspect ratios in this mesh. Indeed, the only way to dramatically reduce the average aspect ratio, other than using surface cells, is to mesh the problem with cells whose maximum dimension is on the order of the shield thickness. This mesh is not shown because in every attempt to produce such a mesh I locked up the mesh generator, Parametric Technology Corporation's Pro/Mesh™, Release 18.0 (Parametric Technology Corporation, 97). The

hardware used in these attempts was the ASC MSRC's SGI Power Onyx, which has 1 Gigabyte of memory (ASC, 1998)!

### The Surface Cell Algorithm

Clearly, there is need for better ways to approach thin volume regions in an unstructured mesh. Our approach is the surface cell algorithm.

Consider Figure 43 below.

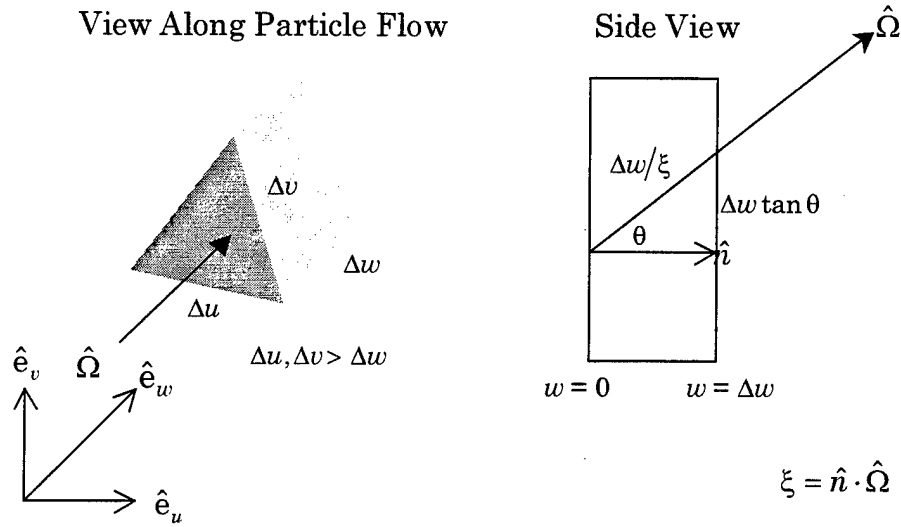


Figure 43. Cell and Side View of a Surface Cell.

Figure 43 shows an extruded triangular prism along the local  $\hat{e}_w$  axis of the  $(\hat{e}_u, \hat{e}_v, \hat{e}_w)$  coordinate system. This cell has an intrinsic thickness,  $\Delta w$ , and an optical path length of  $\frac{\Delta w}{\xi}$  with respect to the streaming direction,  $\hat{\Omega}$ .

Particles enter the front face at  $w = 0$  and exit the back face at  $w = \Delta w$ . Actually, a few particles enter or exit through the sides but we will ignore these since we assume that  $\Delta w$  is small compared to the lateral extents of

the cell. We erect the  $(\hat{e}_u, \hat{e}_v, \hat{e}_w)$  coordinate system to span the surface cell as though it were a triangular prism. Indeed, the surface cell is represented as having no thickness in the mesh (one would have to take measures to preserve the surface cells material in implementing the quadrature).

The key approximation for the surface cell algorithm is that we can neglect the lateral displacement of the flux exiting the cell and treat the transport in the cell in a slab-like manner (Figure 44). Clearly, as  $\Delta w \rightarrow 0$ , this approximation becomes increasingly accurate.

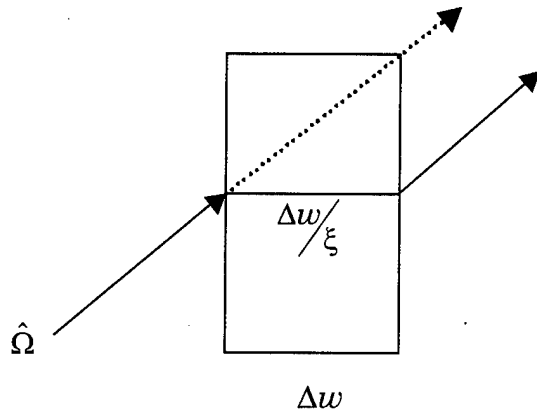


Figure 44. Surface Cell Approximation.

Note that in Figure 44 we have increased the path length across the cell to match the path length of the correctly propagated ray (dashed arrow).

With the geometric foundation set and approximations presented, we are now ready to derive the surface cell spatial quadrature for the linear and exponential characteristic methods.

## Derivation of Surface Cell Spatial Quadratures

As with characteristic methods based on a tetrahedron cell, the surface cell method is also based on the characteristic equation for the cell flux,

$$\psi(u, v, w) = f(u, v) \exp\left(\frac{-\varepsilon w}{\Delta w}\right) + \int_0^w \frac{dw'}{\xi} S(u, v, w') \exp\left(\frac{-\varepsilon(w-w')}{\Delta w}\right), \quad (128)$$

where  $f(u, v)$  is a distribution for the inflow face flux and  $\xi$  is the extended path length across the surface cell. As will be shown,  $f(u, v)$  need not actually be specified, nor any form assumed. Only its moments will be needed. The outflow flux is the flux at  $w = \Delta w$ ,

$$\psi^{\text{out}}(u, v) = \psi(u, v, \Delta w) = f(u, v) \exp(-\varepsilon) + \exp(-\varepsilon) \int_0^{\Delta w} \frac{dw'}{\xi} S(u, v, w') \exp\left(\frac{\varepsilon w'}{\Delta w}\right). \quad (129)$$

Lastly, we define a cell volume moment operator for the surface cell,

$$M[g] = 2 \int_0^1 du \int_0^u dv \int_0^{\Delta w} \frac{dw}{\Delta w} g(u, v, w), \quad (130)$$

where this is the volume operator for a regular triangular prism. If we separate the lateral and axial integration operators, we can define  $M$  as

$$\begin{aligned} M[g(u, v, w)] &= L[g(u, v)] A[g(w)], \\ A[g(w)] &= \int_0^{\Delta w} \frac{dw}{\Delta w} g(w), \\ L[g(u, v)] &= 2 \int_0^1 du \int_0^u dv g(u, v). \end{aligned} \quad (131)$$

Indeed, the key approximation for the surface cell algorithm is the separation of variables as implied by the operators in (131).

We now derive the linear and exponential characteristic versions of the surface cell algorithm.

### Linear Characteristic Surface Cell Approximation

We begin the linear characteristic surface cell derivation by assuming that the normalized source is distributed linearly in the  $w$ -dimension across the cell,

$$S(u, v, w) = [P_0(w) + \theta P_1(w)] g(u, v), \quad (132)$$

where  $P_0(w) = 1$ ,  $P_1(w) = 2 \frac{w}{\Delta w} - 1$ ,  $\theta = \frac{S_w}{S_A}$ , and  $g(u, v)$  is a source distribution

whose average is

$$\begin{aligned} M[S(u, v, w)] &= 2 \int_0^1 du \int_0^u dv g(u, v) \int_0^{\Delta w} \frac{dw'}{\Delta w} [P_0(w') + \theta P_1(w')] \\ &= 2 \int_0^1 du \int_0^u dv g(u, v) = S_A. \end{aligned} \quad (133)$$

Thus, the functional form of the cell flux is

$$\psi(u, v, w) = f(u, v) \exp\left(-\varepsilon \frac{w}{\Delta w}\right) + \frac{g(u, v)}{\xi} \int_0^w dw' [1 + \theta P_1(w')] \exp\left[\frac{-\varepsilon(w - w')}{\Delta w}\right]. \quad (134)$$

For the outflow flux, we start with (129) and substitute the source

distribution, noting that  $2\frac{w'}{\Delta w} - 1 = 1 - 2(1 - \frac{w'}{\Delta w})$ . Changing variables such that

$t = \frac{w'}{\Delta w}$ , the outflow flux becomes

$$\begin{aligned} \psi^{\text{out}}(u, v) = & f(u, v) \exp(-\varepsilon) + g(u, v) \exp(-\varepsilon) \frac{\Delta w}{\xi} K_0(-\varepsilon) + \\ & \theta g(u, v) \exp(-\varepsilon) \frac{\Delta w}{\xi} [2K_1(-\varepsilon) - K_0(-\varepsilon)], \end{aligned} \quad (135)$$

where the  $K$ -functions are functions originally presented in the linear characteristic paper (Mathews, 1998) and are defined as

$$K_{i_1, i_2, \dots, i_m}(\varepsilon) = \int_0^1 dt_1 \int_0^{t_1} dt_2 \dots \int_0^{t_{m-1}} dt_m t_1^{i_1} t_2^{i_2} \dots t_m^{i_m} e^{-\varepsilon t_m}. \quad (136)$$

Operating on (134) with  $A$ , we calculate the zeroth moment of the flux with respect to  $w$ ,

$$\begin{aligned} \psi_0(u, v) = & A[\psi(u, v, w')] \\ = & \int_0^{\Delta w} \frac{dw'}{\Delta w} f(u, v) \exp\left(\frac{-\varepsilon w'}{\Delta w}\right) + \\ & \frac{g(u, v)}{\xi} \int_0^{\Delta w} \frac{dw'}{\Delta w} \int_0^{w'} dw'' [1 + \theta P_1(w'')] \exp\left(\frac{-\varepsilon(w' - w'')}{\Delta w}\right). \end{aligned} \quad (137)$$

Changing variables such that  $t' = \frac{w'}{\Delta w}$  and  $t'' = \frac{w''}{\Delta w}$ , we transform (137) into

$$\begin{aligned}\psi_0(u,v) = & \int_0^1 dt' f(u,v) \exp(-\varepsilon t') + \\ & \frac{g(u,v) \Delta w}{\xi} \int_0^1 dt' \int_0^{t'} dt'' [1 + \theta(2t'' - 1)] \exp(-\varepsilon(t' - t'')).\end{aligned}\quad (138)$$

Changing variables again such that  $t = t' - t''$  and performing the integrations, we find that

$$\begin{aligned}\psi_0(u,v) = & f(u,v) K_0(\varepsilon) + g(u,v) \frac{\Delta w}{\xi} K_{0,0}(\varepsilon) + \\ & \theta g(u,v) \frac{\Delta w}{\xi} [2K_{1,0}(\varepsilon) - 2K_{0,1}(\varepsilon) - K_{0,0}(\varepsilon)].\end{aligned}\quad (139)$$

With  $\psi_0(u,v)$  in hand, we now derive the first moment of the flux with respect to  $w$ ,  $\psi_1(u,v)$ . As before, we are only evaluating the inner integral of M using A. In the first moment case,

$$\psi_1(u,v) = A[3P_1(w')\psi(u,v,w')].\quad (140)$$

Using the same changes of variables as for  $\psi_0(u,v)$ , we find that

$$\begin{aligned}\psi_1(u,v) = & 3f(u,v)[2K_1(\varepsilon) - K_0(\varepsilon)] + 3g(u,v) \frac{\Delta w}{\xi} [2K_{1,0}(\varepsilon) - K_{0,0}(\varepsilon)] + \\ & 3\theta g(u,v) \frac{\Delta w}{\xi} [4K_{2,0}(\varepsilon) - 4K_{1,1}(\varepsilon) - 4K_{1,0}(\varepsilon) + 2K_{0,1}(\varepsilon) + K_{0,0}(\varepsilon)].\end{aligned}\quad (141)$$

With the above moments, we are ready to define the surface cell approximation. We define the results of the following moment operator operations:

$$\begin{aligned}
L[f(u,v)] &= \psi_A^{\text{in}}, \\
L[u f(u,v)] &= \psi_u^{\text{in}}, \\
L[v f(u,v)] &= \psi_v^{\text{in}}, \\
L[g(u,v)] &= S_A, \\
L[u g(u,v)] &= S_u, \\
L[v g(u,v)] &= S_v,
\end{aligned} \tag{142}$$

recalling that we defined the source normalization to ensure that

$L[g(u,v)] = S_A$ . The cell flux moments based on the source normalization are:

$$\begin{aligned}
\psi_A^{\text{cell}} = L[\psi_0(u,v)] &= \psi_A^{\text{in}} K_0(\epsilon) + S_A \frac{\Delta w}{\xi} K_{0,0}(\epsilon) + \\
&\quad \theta S_A \frac{\Delta w}{\xi} [2K_{1,0}(\epsilon) - 2K_{0,1}(\epsilon) - K_{0,0}(\epsilon)],
\end{aligned} \tag{143}$$

$$\begin{aligned}
\psi_u^{\text{cell}} = L[u \psi_0(u,v)] &= \psi_u^{\text{in}} K_0(\epsilon) + S_u \frac{\Delta w}{\xi} K_{0,0}(\epsilon) + \\
&\quad \theta S_u \frac{\Delta w}{\xi} [2K_{1,0}(\epsilon) - 2K_{0,1}(\epsilon) - K_{0,0}(\epsilon)],
\end{aligned} \tag{144}$$

$$\begin{aligned}
\psi_v^{\text{cell}} = L[v \psi_0(u,v)] &= \psi_v^{\text{in}} K_0(\epsilon) + S_v \frac{\Delta w}{\xi} K_{0,0}(\epsilon) + \\
&\quad \theta S_v \frac{\Delta w}{\xi} [2K_{1,0}(\epsilon) - 2K_{0,1}(\epsilon) - K_{0,0}(\epsilon)],
\end{aligned} \tag{145}$$

and

$$\begin{aligned}
\psi_w^{\text{cell}} = L[\psi_1(u,v)] &= 3\psi_A^{\text{in}} [2K_1(\epsilon) - K_0(\epsilon)] + 3S_A \frac{\Delta w}{\xi} [2K_{1,0}(\epsilon) - K_{0,0}(\epsilon)] + \\
&\quad 3\theta S_A \frac{\Delta w}{\xi} [4K_{2,0}(\epsilon) - 4K_{1,1}(\epsilon) - 4K_{1,0}(\epsilon) + 2K_{0,1}(\epsilon) + K_{0,0}(\epsilon)].
\end{aligned} \tag{146}$$



Operating on  $\psi^{\text{out}}(u,v)$  with L gives us the outflow flux moments:

$$\begin{aligned}\psi_A^{\text{out}} = L[\psi^{\text{out}}(u,v)] &= \psi_A^{\text{in}} \exp(-\varepsilon) + S_A \frac{\Delta w}{\xi} K_0(\varepsilon) + \\ &\quad \theta S_A \frac{\Delta w}{\xi} (K_0(\varepsilon) - 2K_1(\varepsilon)),\end{aligned}\tag{147}$$

$$\begin{aligned}\psi_u^{\text{out}} = L[u \psi^{\text{out}}(u,v)] &= \psi_u^{\text{in}} \exp(-\varepsilon) + S_u \frac{\Delta w}{\xi} K_0(\varepsilon) + \\ &\quad \theta S_u \frac{\Delta w}{\xi} (K_0(\varepsilon) - 2K_1(\varepsilon)),\end{aligned}\tag{148}$$

$$\begin{aligned}\psi_v^{\text{out}} = L[v \psi^{\text{out}}(u,v)] &= \psi_v^{\text{in}} \exp(-\varepsilon) + S_v \frac{\Delta w}{\xi} K_0(\varepsilon) + \\ &\quad \theta S_v \frac{\Delta w}{\xi} (K_0(\varepsilon) - 2K_1(\varepsilon)).\end{aligned}\tag{149}$$

Finally, we derive the conservation relationships by applying M to the BTE, just as we did in the tetrahedron cell case. The conservation equations are:

$$\text{Average:} \quad \psi_A^{\text{out}} - \psi_A^{\text{in}} + \varepsilon \psi_A^{\text{cell}} = S_A \frac{\Delta w}{\xi},\tag{150}$$

$$u\text{-moment:} \quad \psi_u^{\text{out}} - \psi_u^{\text{in}} + \varepsilon \psi_u^{\text{cell}} = S_u \frac{\Delta w}{\xi},\tag{151}$$

$$v\text{-moment:} \quad \psi_v^{\text{out}} - \psi_v^{\text{in}} + \varepsilon \psi_v^{\text{cell}} = S_v \frac{\Delta w}{\xi},\tag{152}$$

$$\text{and } w\text{-moment:} \quad 3\psi_A^{\text{out}} + 3\psi_A^{\text{in}} - 6\psi_A^{\text{cell}} + \varepsilon \psi_w^{\text{cell}} = S_w \frac{\Delta w}{\xi}.\tag{153}$$

The step characteristic quadrature is obtained by using equations (143), (147), and (150) for the average flux in the cell. With the linear characteristic surface cell equations developed, we now extend the algorithm to the exponential characteristic method.

### Exponential Characteristic Surface Cell Approximation

The exponential characteristic surface cell assumes an exponentially distributed source in the  $w$  direction,

$$S(u, v, w) = \frac{\exp(-\beta \frac{w}{\Delta w})}{\mathcal{M}_0(\beta)} g(u, v), \quad (154)$$

where again we have introduced a normalization  $(\frac{1}{\mathcal{M}_0(\beta)})$  such that

$A[S(w)] = 1$  and  $L[g(u, v)] = S_A$ . However, in this case we must determine the value of the source parameter,  $\beta$ , via root solving as is done in the exponential characteristic method for slab geometry (Mathews, 1993). Indeed, the source parameter,  $\beta$ , is determined by the root-solving algorithm used for the EC method on unstructured grids of triangles (Mathews, 1997). In this instance, we take the ratio of  $\theta = \frac{S_w}{S_A}$  and solve for  $\rho(\beta)$ , giving us

$$\rho(\beta) = \frac{3 - \theta}{6}, \quad (155)$$

which is solved for  $\beta$  using the method referenced above. Note that  $-3 < \theta < 3$  so that  $0 < \rho(\beta) < 1$ .

We begin the derivation of the cell flux with (128) and substitute for  $S(u, v, w)$ ,

$$\psi(u, v, w) = f(u, v) \exp\left(\frac{-\varepsilon w}{\Delta w}\right) + \frac{g(u, v)}{\xi \mathcal{M}_0(\beta)} \int_0^w dw' \exp\left(-\beta \frac{w}{\Delta w}\right) \exp\left(\frac{-\varepsilon(w-w')}{\Delta w}\right). \quad (156)$$

For the outflow flux, we substitute (154) into (129):

$$\begin{aligned} \psi^{\text{out}}(u, v) = \psi(u, v, \Delta w) = & f(u, v) \exp(-\varepsilon) + \\ & \frac{g(u, v)}{\xi \mathcal{M}_0(\beta)} \int_0^{\Delta w} dw' \exp\left(-\beta \frac{w'}{\Delta w}\right) \exp\left(\frac{\varepsilon(w-w')}{\Delta w}\right). \end{aligned} \quad (157)$$

As in the LC case, if we change variables such that  $t = \frac{w'}{\Delta w}$  and use the definition of the moment functions, we find that the outflow flux is

$$\psi^{\text{out}}(u, v) = f(u, v) \exp(-\varepsilon) + \frac{g(u, v)}{\mathcal{M}_0(\beta)} \frac{\Delta w}{\xi} \exp(-\varepsilon) \mathcal{M}_0(\beta - \varepsilon). \quad (158)$$

As before, we will first apply the  $A$  operator on equations (156) and (158) to obtain the  $w$  contribution to the flux moment. For the sake of brevity and since the operations and changes of variables are identical to the LC case (we just have moment functions instead of  $K$  functions), I will just present the results of the operations:

$$\psi_0(u,v) = A[\psi(u,v,w')] = f(u,v) \mathcal{M}_0(\varepsilon) + \frac{g(u,v)}{\mathcal{M}_0(\beta)} \frac{\Delta w}{\xi} \mathcal{M}_0(\varepsilon, \beta), \quad (159)$$

$$\begin{aligned} \psi_1(u,v) &= A[3P_1(w')\psi(u,v,w')] \\ &= 3f(u,v) \mathcal{M}_0(\varepsilon)[1-2\rho(\varepsilon)] + 3\frac{g(u,v)}{\mathcal{M}_0(\beta)} \frac{\Delta w}{\xi} \mathcal{M}_0(\varepsilon, \beta)[1-2\rho(\varepsilon, \beta)]. \end{aligned} \quad (160)$$

With  $\psi_0(u,v)$ ,  $\psi_1(u,v)$ , and  $\psi^{\text{out}}(u,v)$  defined, we now operate on them with  $L$  and use the moment definitions in (142), producing the flux moments for the cell:

$$\begin{aligned} \psi_A^{\text{cell}} &= L[\psi_0(u,v)] \\ &= \psi_A^{\text{in}} \mathcal{M}_0(\varepsilon) + S_A \frac{\Delta w}{\xi} \frac{\mathcal{M}_0(\varepsilon, \beta)}{\mathcal{M}_0(\beta)}, \end{aligned} \quad (161)$$

$$\begin{aligned} \psi_u^{\text{cell}} &= L[u \psi_0(u,v)] \\ &= \psi_u^{\text{in}} \mathcal{M}_0(\varepsilon) + S_u \frac{\Delta w}{\xi} \frac{\mathcal{M}_0(\varepsilon, \beta)}{\mathcal{M}_0(\beta)}, \end{aligned} \quad (162)$$

$$\begin{aligned} \psi_v^{\text{cell}} &= L[v \psi_0(u,v)] \\ &= \psi_v^{\text{in}} \mathcal{M}_0(\varepsilon) + S_v \frac{\Delta w}{\xi} \frac{\mathcal{M}_0(\varepsilon, \beta)}{\mathcal{M}_0(\beta)}, \end{aligned} \quad (163)$$

and

$$\begin{aligned} \psi_w^{\text{cell}} &= L[\psi_1(u,v)] \\ &= 3\psi_A^{\text{in}} \mathcal{M}_0(\varepsilon)[1-2\rho(\varepsilon)] + 3S_A \frac{\Delta w}{\xi} \frac{\mathcal{M}_0(\varepsilon, \beta)}{\mathcal{M}_0(\beta)}[1-2\rho(\varepsilon, \beta)]. \end{aligned} \quad (164)$$

Similarly, the outflow face flux moments are:

$$\begin{aligned}
\psi_A^{\text{out}} &= \mathcal{L}[\psi^{\text{out}}(u, v)] \\
&= \psi_A^{\text{in}} \exp(-\varepsilon) + S_A \exp(-\varepsilon) \frac{\Delta w}{\xi} \frac{\mathcal{M}_0(\beta - \varepsilon)}{\mathcal{M}_0(\beta)},
\end{aligned} \tag{165}$$

$$\begin{aligned}
\psi_u^{\text{out}} &= \mathcal{L}[u \psi^{\text{out}}(u, v)] \\
&= \psi_u^{\text{in}} \exp(-\varepsilon) + S_u \exp(-\varepsilon) \frac{\Delta w}{\xi} \frac{\mathcal{M}_0(\beta - \varepsilon)}{\mathcal{M}_0(\beta)},
\end{aligned} \tag{166}$$

and

$$\begin{aligned}
\psi_v^{\text{out}} &= \mathcal{L}[v \psi^{\text{out}}(u, v)] \\
&= \psi_v^{\text{in}} \exp(-\varepsilon) + S_v \exp(-\varepsilon) \frac{\Delta w}{\xi} \frac{\mathcal{M}_0(\beta - \varepsilon)}{\mathcal{M}_0(\beta)}.
\end{aligned} \tag{167}$$

The conservation equations for the average-,  $u$ -, and  $v$ -moments are exactly the same as those for the LC case; equations (150), (151), and (152).

The equation for  $w$ -moment conservation is

$$3\psi_A^{\text{out}} + 3\psi_A^{\text{in}} - 6\psi_A^{\text{cell}} + \varepsilon\psi_w^{\text{cell}} = 3S_A \frac{\Delta w}{\xi} [1 - 2\rho(\beta)]. \tag{168}$$

Clearly, we can see that if we can trade a surface cell calculation for a tetrahedron calculation, we gain great efficiencies. The surface cell algorithm requires no cell splitting and in the case of EC, only a single one-dimensional root solve [which was optimized by Mathews (Mathews, 1997)]. Both EC and LC require rotation of  $\psi_u^{\text{in}}$  and  $\psi_v^{\text{in}}$  into the surface cell coordinate system, and determination of the orientation of the streaming direction with respect to the cell zero coordinate to correctly account for the  $w$  moments as is

normally done for slab codes (by sign reversal). Contrast this with a tetrahedron cell, which requires:

- cell splitting,
- translation/rotation of inflow and source moments,
- multi-dimensional non-linear root solves for the source and inflow flux,
- many calls to exponential functions for the spatial quadrature,
- and, re-assembling of the sub-cell moments.

Further, we avoid poorly shaped cells, which propagate throughout the mesh.

### **Implementation of the Surface Cell Quadrature**

Unfortunately, the above spatial quadrature was not implemented in this research. Before implementing the surface cell algorithm, we need a mesh generator that adequately addresses the special needs of a mixed mesh transport code: connected meshes that conserve volumes and materials. Pro/Mesh was inadequate to this task. Thus, we derived the quadrature for future use and proceeded to develop the parallel version of TETRAN.

## Bibliography

- Advanced Visual Systems, Inc., AVS 5 User's Manual, Appendix E (1996).
- Aeronautical Systems Center High Performance Computing Center, "HPC Consolidated Hardware Listing," <http://www.asc.hpc.mil/cgi-bin/hwbymfg.html>, 25 Jul 98.
- Alcouffe, Raymond E., "THREEDANT: A Code to Perform Three-Dimensional, Neutral Particle Transport Calculations", Proc. Topical Mtg. International Conference on Mathematics and Reactor Physics and Environmental Analyses, Portland OR, April 30- May 4, American Nuclear Society, 1, p. 461-469 (1995).
- Alcouffe, R. E., R. S. Baker, F. W. Brinkley, D. R. Marr, R. D. O'Dell, and W. F. Walters, "DANTSYS: A Diffusion Accelerated Neutral Particle Transport Code System," LA-12969-M, Los Alamos National Laboratory, Los Alamos NM, June 1995.
- Anistratov, Dmitriy Yu., M. L. Adams, and E. W. Larsen, "Acceleration Method for the Nonlinear Corner Balance Method," Transactions of the American Nuclear Society, 75, p. 156-157 (1996).
- Brennan, Charles R., Captain, USAF, Characteristic Spatial Quadrature for Discrete Ordinates Neutral Particle Transport on Arbitrary Tetrahedral Meshes, Ph.D. Dissertation, AFIT/DS/ENP/96-01. . School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, June 1996.
- Briesmeister, Judith F., MCNP™ - A General Monte Carlo Code for Neutron and Photon Transport, Los Alamos National Laboratory, Los Alamos NM (1991).
- Burden, R. L., and J. D. Faires, Numerical Analysis, 5<sup>th</sup> Edition, International Thomson Publishing, p. 560-566 (1993).
- Carlson, B. G. and G. I. Bell, "Solution of the Transport Equation by the SN Method", Proc. U. N. Intl. Conference on the Peaceful Uses of Atomic Energy, Geneva, 2, p. 2386 (1958).
- Castrianni, C.L. and M.L. Adams, "A Nonlinear Corner-Balance Spatial Discretization for Transport on Arbitrary Grids," Nucl. Sci. Eng., 128, p. 278-296 (1998).

- Chandrasekhar, S., Radiative Transfer, Dover Publications, NY, Chapter 2 (1960).
- Chin, Steven, "P@SC Overview: Architecture, Application Tuning and Optimization," International Business Machines, RS/6000 Division, Poughkeepsie, NY, October 16, 1996.
- DeHart, M. D., R. E. Pevey, and T. A. Parish, "An Extended Step Characteristic Method for Solving the Transport Equation in General Geometries," Nucl. Sci. Eng., 118, p. 79-90 (1994).
- DelGrande, J. M., Major, "Neutron Transport with Highly Anisotropic Scattering," Research Prospectus, Air Force Institute of Technology (1998).
- Dongarra, Jack, "The Performance Database: Linpack Benchmark," <http://performance.netlib.org/performance/html/linpack>, 26 Jul 98.
- Dowd, Kevin, and Charles Severance, High Performance Computing, 2nd Edition, O'Reilly (1998).
- Foster, Ian, Designing and Building Parallel Programs, Chapter 1, Addison-Wesley (1995).
- "MCNPXS: Standard Neutron, Photon, and Electron Data Libraries for MCNP4B," DLC-189, Radiation Safety Information Computational Center, January 1997.
- Geist, A., A. Beguelin, J. Dongarra, W. Jiang, B. Manchek, and V. Sunderman, PVM: Parallel Virtual Machine – A User's Guide and Tutorial for Network Parallel Computing, The MIT Press (1994).
- Glasstone, S., and A. Sesonke, Nuclear Reactor Engineering, 3ed, Krieger, Malabar, FL, Chapter 10 (1981).
- Hendricks, J. S., S. C. Frankle, and J. D. Court, "ENDF/B-VI Data for MCNP™," LA-1289, Los Alamos National Laboratory, Los Alamos NM, December 1994.
- Intel Corporation, "Intel Microprocessor Quick Reference Guide," <http://www.intel.com/pressroom/kits/processors/quickref.html>, 9 Jul 98.
- International Business Machines, Optimization and Tuning Guide for Fortran, C, and C++, SC09-1705-00, December 1993.



- Jaynes, E. T., "Information Theory and Statistical Mechanics," Phys. Rev., 106, p. 620 (1957).
- Koelbel, C. H., D. B. Loveman, R. S. Schreiber, G. L. Steele, Jr., and M. E. Zosel, The High Performance Fortran Handbook, The MIT Press (1994).
- Kornreich, D.E., and B. D. Ganapol, "The Suite of Analytical Benchmarks for Neutral Particle Transport in Infinite Isotropically Scattering Media," Nucl. Sci. Eng., 125, 1, p. 24-50 (1997).
- Lathrop, K. D., "Spatial Differencing of the Transport Equation: Positivity vs. Accuracy," J. Comput. Phys., 4, p.475-498 (1969).
- Lathrop, K.D., "Remedies for Ray Effects," Nucl. Sci. Eng. 45, 255 (1971).
- Lathrop, K.D., "THREETRAN - A Program to Solve the Multigroup Discrete Ordinates Transport Equation in ( x, y , z) Geometry," LA-4848-MS, Los Alamos Scientific Laboratory, Los Alamos NM (1976).
- Lewis, E. E., and W. F. Miller, Jr., Computational Methods of Neutron Transport, LaGrange Park, Illinois: American Nuclear Society (1993).
- Little, Jeffery K., Major, Private conversation regarding performance of HPF code different memory architectures (August 1998).
- McGhee, J.M., J. E. Morel, and E.W. Larsen, "A 3-D Time-Dependent Unstructured Tetrahedral-Mesh SP<sub>N</sub> Method," Proceedings of the International Conference, Mathematics and Computations, Reactor Physics, and Environmental Analyses, Portland, OR, Volume 2, p. 887-896 (1995).
- MacFarlane, R. E., "TRANSX 2: A Code for Interfacing MATXS Cross-Section Libraries to Nuclear Transport Codes," LA-12312-MS, Los Alamos National Laboratory, Los Alamos NM, July 1992.
- Mathews, Kirk A., LCDR, USN, "Adaptive Characteristic Spatial Quadratures for Discrete Ordinates Neutral Particle Transport - The Slab Geometry Case", Transport Theory and Statistical Physics, 19, p. 419-458 (1990).
- Mathews, K. A. and B. M. Minor, "Step Adaptive Characteristic Spatial Quadrature for Discrete Ordinates Neutral Particle Transport in Two-Dimensional Cartesian Coordinates", Intl. Topical Mtg. Advances in Mathematics. Computations and Reactor Physics, Pittsburgh, PA., April 28 - May 2, 1991, American Nuclear Society.

- Mathews, K. A. and B. M. Minor, "Adaptive Characteristic Spatial Quadratures for Discrete Ordinates Neutral Particle Transport - The Rectangular Cell Case", *Transport Theory and Statistical Physics*, 22, p. 655-686 (1993).
- Mathews, K. A., G. Sjoden, and B. Minor, "Exponential Characteristic Spatial Quadratures for Discrete Ordinates Radiation Transport in Slab Geometry", *Nuclear Science and Engineering*, 118, p. 24-37 (1994).
- Mathews, Kirk A., and C.R. Brennan, "Exponential Characteristic Nonlinear Radiation Transport Method for Unstructured Grids of Triangular Cells," *Nucl. Sci. Eng.*, 126, p. 264-281 (1997).
- Mathews, K. A., "On the Propagation of Rays in Discrete Ordinates," submitted to *Nuclear Science and Engineering* (1998).
- Mathews, K. A., R. L. Miller, and C. R. Brennan, "Split-Cell Characteristic Transport Methods on Unstructured Tetrahedral Meshes Part I: Linear Methods," to be submitted to *Nuclear Science and Engineering* (1998).
- Mathews, K.A., Private conversation regarding MCSN development (November 1998).
- Metcalf, Michael, and John Reid, FORTTRAN 90/95 Explained, Oxford Science Publications (1996).
- Miller, D. J., K. A. Mathews and C. R. Brennan, "Split Cell Discrete Ordinates Transport on and Unstructured Grid of Triangular Cells", *Transport Theory and Statistical Physics*, 25(7), p. 833-867 (1996).
- Miller, Dennis J., Linear Characteristic Spatial Quadrature for Discrete Ordinates Neutral Particle Transport on Arbitrary Triangles, Ph.D. Dissertation AFIT/DS/ENP93-02. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, June 1993.
- Minor, Bryan, and K. A. Mathews, "Exponential Characteristic Spatial Quadrature for Discrete Ordinates Radiation Transport with Rectangular Cells," *Nucl. Sci. Eng.*, 120, p. 165-186 (1995).
- Minor, Bryan M. , Exponential Characteristic Spatial Quadrature for Discrete Ordinates Neutral Particle Transport in Two-Dimensional Cartesian Coordinates, Ph.D. Dissertation AFIT/DS/ENP93-7. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, September 1993.

Ott, K.O., and W. A. Bezella, Introductory Nuclear Reactor Statics, American Nuclear Society, La Grange Park, IL, p. 304-348 (1983).

Parametric Technology Corporation, Pro/Mesh and Pro/FEM POST User's Guide, Release 18.0 (1997).

Paternoster, R. R., and W. F. Walters, "A Linear Characteristic-Nodal Transport Method for Arbitrary Triangular Meshes in (x, y) Geometry," *Trans. Am. Nucl. Soc.*, 46, p.431-433 (1984).

Petrovic, Bojan G., and Alireza Haghighat, "Analysis of Inherent Oscillations in Multidimensional  $S_N$  Solutions of the Neutron Transport Equation," *Nucl. Sci. Eng.*, 124, p. 31-62 (1996).

PGHPF User's Guide, The Portland Group, Inc. (1996).

Radiation Safety Information Computational Center, "RSIC CODE PACKAGE CCC-650-DOORS3.2: One, Two- and Three-Dimensional Discrete Ordinates Neutron/Photon Transport Code System," <http://epicws.epm.ornl.gov/codes/ccc/ccc6/ccc-650.html>, 27 Jul 98.

Ramone, G. L., M. L. Adams, and P. F. Nowak, "A Transport Synthetic Acceleration Method for Transport Iterations," *Nucl. Sci. Eng.*, 125, p. 257-283 (1997).

Rhoades, Wayne A. and Yousry Y. Azmy, "Three Dimensional  $S_N$  Calculations with the Oak Ridge TORT Code", *Proc. Topical Mtg. International Conference on Mathematics and Reactor Physics and Environmental Analyses*, Portland, OR, April 30- May 4, American Nuclear Society, 1, p. 480-489 (1995).

Rose, R.F., and C. L. Dunford, "ENDF-102, Data Formats and Procedures for the Evaluated Nuclear Data File, ENDF-6," Brookhaven National Laboratory report BNL-NCS-44945, July 1990.

Sjoden, G. E., Exponential Characteristic Spatial Quadratures for Discrete Ordinates Neutral Particle Transport in Slab Geometry, M. S. Thesis, AFIT/GNE/ENP/92M-10. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH. March 1992.

Snir, , M., S. Otto, S. Huss-Lederman, D. Walker, and J. Dongarra, MPI: The Complete Reference, The MIT Press (1996).

United States Department of Energy Accelerated Strategic Computing Initiative (ASCI), "Accelerated Strategic Computing Initiative: Overview," <http://www.llnl.gov/asci/overview>, 26 Jul 98.

- Walters, W. F. and R. D. O'Dell, "Nodal Methods for Discrete Ordinates Transport Problems in (XY) Geometry", Proc. Topical Mtg. Advances in Mathematical Methods for the Solution of Engineering Problems, Munich, FRG, April 27-29, 1981, American Nuclear Society, 1, 115 (1981).
- Walters, W. F., and T. A. Wareing, "A Non-Linear Positive Method for Solving the Transport Equation on Coarse Meshes," Proceedings Eighth International Conference on Radiation Shielding, Volume 1, Arlington, Texas (1994).
- Walters, W. F., T. A. Wareing, and D. R. Marr, "The Non-Linear Characteristic Scheme for X-Y Geometry Transport Problems", Proceedings International Conference on Mathematics and Computations, Reactor Physics and Environmental Analyses, Portland, Oregon, April 30-May 4, 1995.
- Walters, Wallace F., and T. A. Wareing, "An Accurate, Strictly Positive, Nonlinear Characteristic Scheme for the Discrete-Ordinate Equations," Transport Theory and Statistical Physics, 25(2), p. 197-215 (1996).
- Wareing, Todd A., W. F. Walters, and J. E. Morel, "An  $S_2$  - like Acceleration Method for the Nonlinear Characteristic Transport Scheme," Transactions of the American Nuclear Society, 71, p. 267-268 (1994).
- Wareing, T. A., W. F. Walters and J. E. Morel, "A Diffusion Accelerated Solution Method for the Nonlinear Characteristic Scheme in Slab Geometry", LA-UR-95-235, Los Alamos National Laboratory, Los Alamos NM (1995).
- Wareing, T. A., and W. F. Walters, "An Exponential Discontinuous Scheme for X-Y Geometry Transport Problems," Transactions of the American Nuclear Society, 72 (1995).
- Wareing, Todd A., and R. E. Alcouffe, "An Exponential Discontinuous Scheme for X-Y-Z Geometry Transport Problems," Los Alamos National Laboratory, Unpublished (1996).
- Wareing, T. A., J. M. McGhee, and J. E. Morel, "ATTILA: A Three-Dimensional, Unstructured Tetrahedral Mesh Discrete Ordinates Transport Code," Transactions of the American Nuclear Society, 75, p. 146-147 (1996).
- Wareing, T. A., D. K. Parsons, and S. Pautz, "A Reactor Pressure Vessel Dosimetry Calculation Using ATTILA, An Unstructured Tetrahedral

Mesh Discrete-Ordinates Code," Proc. Joint International Conference on Mathematical Methods and Supercomputing for Nuclear Applications, Saratoga Springs, NY, October 5-9, American Nuclear Society, 2, p. 1024-1031 (1997).

Wareing, T. A., Private Communication, 3 Aug 98.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE November 1998		3. REPORT TYPE AND DATES COVERED Doctoral Dissertaion
4. TITLE AND SUBTITLE DEVELOPMENT OF A DISCRETE ORDINATES CODE SYSTEM FOR UNSTRUCTURED MESHES OF TETRAHEDRAL CELLS, WITH SERIAL AND PARALLEL IMPLEMENTATIONS			5. FUNDING NUMBERS	
6. AUTHOR(S)  Rodney L. Miller, Captain, USAF				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology/ENP 2950 P. St. Building 640 WPAFB, OH 45433-7765 Attn: Prof. Kirk A. Mathews			8. PERFORMING ORGANIZATION REPORT NUMBER  AFIT/DS/ENP/98-02	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) SA-ALC/NWIC 1651 First St. SE KAFB, NM 87117-5617 Attn: Mr. Harvey Dayhoff Ph: (505) 846-6176			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
Defense Threat Reduction Agency Attn: SWEW (Mr. Rob Kehlet) 45045 Aviation Dr. Dulles, VA 20166-7517 Ph: (703) 325-2046				
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION AVAILABILITY STATEMENT  Approved for Public Release; Distribution Unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) A numerically stable, accurate, and robust form of the exponential characteristic (EC) method, used to solve the time-independent linearized Boltzmann Transport Equation, is derived using direct affine coordinate transformations on unstructured meshes of tetrahedra. This quadrature, as well as the linear characteristic (LC) spatial quadrature, is implemented in our transport code, called TETRAN. This code solves multi-group neutral particle transport problems with anisotropic scattering and was parallelized using High Performance Fortran and angular domain decomposition. A new, parallel algorithm for updating the scattering source is introduced. The EC source and inflow flux coefficients are efficiently evaluated using Broyden's rootsolver, started with special approximations developed here. TETRAN showed robustness, stability and accuracy on a variety of challenging test problems. Parallel speed-up was observed as the number of processors was increased using an IBM SP computer system.				
14. SUBJECT TERMS Boltzmann Equation, Neutron Transport Theory, Numerical Methods, Unstructured Grids, Radiation Transport, Radiation Shielding, High Performance Fortran, Anisotropic Scattering, Multigroup Theory			15. NUMBER OF PAGES 196	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT  UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE  UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT  UNCLASSIFIED	20. LIMITATION OF ABSTRACT  UL	