

University of New Mexico

UNM Digital Repository

Nuclear Engineering ETDs

Engineering ETDs

Fall 2020

A Two-Dimensional Spherical-Polar Multigroup Discrete Ordinates Code for Atmospheric Particle Transport

Daniel T. Wakeford

University of New Mexico

Follow this and additional works at: https://digitalrepository.unm.edu/ne_etds



Part of the [Nuclear Engineering Commons](#)

Recommended Citation

Wakeford, Daniel T.. "A Two-Dimensional Spherical-Polar Multigroup Discrete Ordinates Code for Atmospheric Particle Transport." (2020). https://digitalrepository.unm.edu/ne_etds/97

This Dissertation is brought to you for free and open access by the Engineering ETDs at UNM Digital Repository. It has been accepted for inclusion in Nuclear Engineering ETDs by an authorized administrator of UNM Digital Repository. For more information, please contact disc@unm.edu.

Daniel T. Wakeford

Candidate

Nuclear Engineering

Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

Prof. Anil Prinja

, Chairperson

Dr. James Warsa

Prof. Cassiano Oliveira

Prof. Dmitriy Anistratov

**A Two-Dimensional Spherical-Polar
Multigroup Discrete Ordinates Code
for Atmospheric Particle Transport**

by

Daniel T. Wakeford

B.S., Physics, University of Guelph, 2002
M.S., Physics, University of Alberta, 2005

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy

Engineering

The University of New Mexico
Albuquerque, New Mexico

December, 2020

Acknowledgments

I thank Prof. Anil Prinja for his help and advice during the course of this dissertation work and, through his instruction, for sparking my interest in deterministic transport methods.

Thanks to Dr. Jim Warsa for teaching me methods of computational transport, for his interest in guiding this research, and for his willingness and patience to meet many times to discuss at length the challenges we faced in successfully deriving and implementing this new method.

Thanks to Prof. Cassiano Oliveira and to Prof. Dmitriy Anistratov for serving on my dissertation committee and for their insightful reviews of this work.

Thanks to my parents, Bryan and Ruth Wakeford, for deeply implanting in me the value of education.

Thanks to my wife Binh for her patience and encouragement, and to my son Corbin for giving me the motivation to follow through.

A Two-Dimensional Spherical-Polar Multigroup Discrete Ordinates Code for Atmospheric Particle Transport

by

Daniel T. Wakeford

B.S., Physics, University of Guelph, 2002

M.S., Physics, University of Alberta, 2005

Ph.D., Engineering, University of New Mexico, 2020

Abstract

A natural coordinate system for calculating neutral particle propagation in the atmosphere due to point sources is two-dimensional spherical-polar (r, θ) coordinates. Deterministic methods for solving the transport equation in this coordinate system require some interesting implementation details stemming from the two angular derivatives present in this form of the equation. In this dissertation, we present a bilinear discontinuous spatial discretization of the transport equation in this coordinate system. The weighted diamond difference discretization is applied to the angular derivative term in two angular dimensions, using zero-weighted starting directions to start the differencing. To our knowledge, the only previous deterministic code that solved the transport equation in the spherical-polar coordinate system was written more than 50 years ago; the development of the deterministic methods we present here and the application to atmospheric transport is new. Numerical experiments are shown to verify that method has better than second-order accuracy and correctly solves the spherical-polar form of the transport equation.

Contents

List of Figures	viii
List of Tables	xii
1 Introduction	1
1.1 Research Goals	1
1.2 Problem Definition	3
1.3 Current state of the problem	7
1.3.1 Solution by Monte Carlo methods	8
1.3.2 Solution by deterministic methods	10
1.4 Chapter outline	11
2 The Discrete Ordinates Method	12
2.1 The transport equation	14
2.2 Coordinate system	18
2.3 Angular discretization	19
2.4 Energy discretization	22
2.5 Spatial discretization	26
2.6 Discontinuous methods	34
2.7 TWOTRAN SPHERE	36
2.8 Solution methods	42
3 Bilinear Discontinuous Discretization in Two-Dimensional Spherical- Polar (r,θ) Coordinates	47
3.1 Form of the transport equation	47
3.2 Basis function expansion	50

3.3	The streaming term	53
3.4	The gradient term	61
3.5	The angular redistribution term	63
3.6	The removal term	68
3.7	The source term	69
3.8	Starting directions	71
3.8.1	Starting directions in μ	74
3.8.2	Starting directions in η	76
3.9	The fully discretized transport equations in the two-dimensional spherical-polar (r,θ) coordinate system	79
4	ATMOTRAN Implementation	83
4.1	Approach and main functional blocks	83
4.2	Details of <code>xsstore.cc</code>	86
4.3	Details of <code>geometry.cc</code>	88
4.4	Details of <code>quadrature.cc</code>	92
4.5	Details of <code>solver.cc</code>	93
4.6	Details of <code>sweep.cc</code>	95
4.7	Details of <code>utils.cc</code>	98
5	ATMOTRAN Verification	100
5.1	Thick-diffusion limit	100
5.2	Manufactured solution verification	102
5.2.1	Biquadratic isotropic manufactured solution	104
5.2.2	Bilinear isotropic manufactured solution	106
5.2.3	Biquadratic anisotropic manufactured source	109
5.3	Code-to-code comparison	112

5.4	Inspection of cases of interest	112
5.5	Boundary surface sources	114
5.6	Solid spheres	115
5.7	Point sources	116
5.8	Verification against Monte Carlo	121
5.9	Summary of verification results	124
6	Summary	127
7	Appendices	131
7.1	Appendix A - Matrix Elements	132
7.1.1	Tabulation of M_x Integrals	132
7.2	Appendix B - Atmospheric Calculation Results	136
7.2.1	Energy Spectrum Comparison	137
7.3	Fluence vs Angle Comparison	153
7.4	Appendix C - NJOY Input Files	169
	References	173

List of Figures

1	Problem geometry (not to scale).	4
2	Total atmospheric density (g/cm^3) as a function of altitude (km). . .	5
3	Total neutron cross section (b) as a function of energy (eV) for ^{14}N in the resonance region.	7
4	Neutron cross section as a function of scattering angle for ^{14}N , for neutron energies ranging from 1 MeV to 10 MeV.	8
5	Three-dimensional spherical coordinates extended view of vector directions	19
6	Two-dimensional spherical coordinates	20
7	Spatial and Angular cell indexing convention.	20
8	Variability of angle (indexed θ and χ) in the coordinate frame during straight line trajectory through a sphere.	21
9	Diagram of one-dimensional slab geometry.	33
10	Flux notation in the linear discontinuous spatial differencing method.	34
11	(Left) Sweeping (r, μ). (Right) Sweeping (θ, η).	39
12	TWOTRAN SPHERE Sweeping the angular mesh for small polar angle	40
13	TWOTRAN SPHERE Sweeping the angular mesh for large polar angle	41
14	Schematic of a computational cell.	53
15	Dimensions of the physical unit cell in spherical polar coordinates. . .	55
16	Ordering of directions in one-dimensional spherical geometry for S_6 approximation. The starting direction is denoted by s	72
17	Positioning of directions in two-dimensional spherical-polar geometry for S_6 approximation. The starting directions are shown as triangles.	73
18	ATMOTRAN functional block diagram.	84

19	ATMOTRAN: sweeping the angular mesh for small polar angle. . . .	97
20	ATMOTRAN: sweeping the angular mesh for large polar angle. . . .	98
21	Numerical result for 10x10 problem approaching the thick diffusion limit.	102
22	Numerical result for 32 x 32 biquadratic manufactured solution. . . .	106
23	Numerical result for bilinear manufactured solution, represented with 64 x 64 bins.	108
24	ATMOTRAN result for spherical shell with inner boundary source and vacuum outer boundary.	114
25	ATMOTRAN result for spherical shell with inner boundary source and reflecting outer boundary.	115
26	ATMOTRAN result for a pure scattering solid sphere with an internal source.	116
27	Comparison of weighted and standard diamond differencing for the angular discretization, using bilinear discontinuous spatial differencing on the solid sphere problem.	117
28	ATMOTRAN result for a point source in a solid sphere.	118
29	ATMOTRAN result for a point source in a solid sphere with bilinear discontinuous spatial discretization.	119
30	ATMOTRAN result for a point source in a solid sphere with diamond spatial discretization.	120
31	Atmospheric neutron energy spectrum at 1000 km $\theta = 0$, for 10 MeV source at 30 km	123
32	Atmospheric neutron fluence vs polar angle at 1000 km, for 10 MeV source at 30 km	124

33	Neutron energy spectrum at 1000 km $\theta = 0$, for 10 MeV source at 300 km	137
34	Energy spectrum at 1000 km $\theta = 0$, for 10 MeV source at 100 km . . .	138
35	Energy spectrum at 1000 km $\theta = 0$, for 10 MeV source at 30 km . . .	139
36	Energy spectrum at 1000 km $\theta = 0$, for 10 MeV source at 20 km . . .	140
37	Energy spectrum at 1000 km $\theta = 0$, for 5 MeV source at 300 km . . .	141
38	Energy spectrum at 1000 km $\theta = 0$, for 5 MeV source at 100 km . . .	142
39	Energy spectrum at 1000 km $\theta = 0$, for 5 MeV source at 30 km . . .	143
40	Energy spectrum at 1000 km $\theta = 0$, for 5 MeV source at 20 km . . .	144
41	Energy spectrum at 1000 km $\theta = 0$, for 2 MeV source at 300 km . . .	145
42	Energy spectrum at 1000 km $\theta = 0$, for 2 MeV source at 100 km . . .	146
43	Energy spectrum at 1000 km $\theta = 0$, for 2 MeV source at 30 km . . .	147
44	Energy spectrum at 1000 km $\theta = 0$, for 2 MeV source at 20 km . . .	148
45	Energy spectrum at 1000 km $\theta = 0$, for 1 MeV source at 300 km . . .	149
46	Energy spectrum at 1000 km $\theta = 0$, for 1 MeV source at 100 km . . .	150
47	Energy spectrum at 1000 km $\theta = 0$, for 1 MeV source at 30 km . . .	151
48	Energy spectrum at 1000 km $\theta = 0$, for 1 MeV source at 20 km . . .	152
49	Fluence vs polar angle at 1000 km, for 10 MeV source at 300 km . . .	153
50	Fluence vs polar angle at 1000 km, for 10 MeV source at 100 km . . .	154
51	Fluence vs polar angle at 1000 km, for 10 MeV source at 30 km . . .	155
52	Fluence vs polar angle at 1000 km, for 10 MeV source at 20 km . . .	156
53	Fluence vs polar angle at 1000 km, for 5 MeV source at 300 km . . .	157
54	Fluence vs polar angle at 1000 km, for 5 MeV source at 100 km . . .	158
55	Fluence vs polar angle at 1000 km, for 5 MeV source at 30 km	159
56	Fluence vs polar angle at 1000 km, for 5 MeV source at 20 km	160
57	Fluence vs polar angle at 1000 km, for 2 MeV source at 300 km . . .	161

58	Fluence vs polar angle at 1000 km, for 2 MeV source at 100 km . . .	162
59	Fluence vs polar angle at 1000 km, for 2 MeV source at 30 km . . .	163
60	Fluence vs polar angle at 1000 km, for 2 MeV source at 20 km . . .	164
61	Fluence vs polar angle at 1000 km, for 1 MeV source at 300 km . . .	165
62	Fluence vs polar angle at 1000 km, for 1 MeV source at 100 km . . .	166
63	Fluence vs polar angle at 1000 km, for 1 MeV source at 30 km . . .	167
64	Fluence vs polar angle at 1000 km, for 1 MeV source at 20 km . . .	168

List of Tables

1	Sweep-direction-dependent flux values in the streaming term	60
2	Spherical harmonic terms for the scattering operator	71
3	Gauss-Chebyshev-Legendre square S_6 η angles along the μ_1 level . . .	77
4	Simulation parameters with types and example values	85
5	Multigroup nuclear data for ^{14}N	87
6	Group-to-group scattering cross sections for ^{14}N	88
7	Composite atmosphere scattering fraction as a function of bin for the ATMOTRAN energy range	90
8	Number of mean free paths in the radial dimension for a 30 cell dis- cretization for several energies	91
9	S_4 Two-dimensional Gauss-Chebyshev-Legendre quadrature roots and weights	92
10	S_4 Angular redistribution coefficients in the μ direction	93
11	S_4 Angular redistribution coefficients in the η direction	93
12	S_4 Weighted-diamond angular differencing coefficients τ_m	93
13	S_4 Weighted-diamond angular differencing coefficients λ_ℓ	93
14	S_4 Starting directions	94
15	Spatial extents of the radial sweep	97
16	Spatial extents of the polar sweep	98
17	Method and description table for <code>solver_utils.cc</code>	99
18	Manufactured solution verification for isotropic biquadratic flux . . .	105
19	Manufactured solution verification for bilinear flux	108
20	Verification for $1 + 3\mu^2$ anisotropic biquadratic flux	110
21	Verification for $1 + 3\eta^2$ anisotropic biquadratic flux	111

22	ATMOTRAN and Capsaicin code-to-code comparison (Problem 1) . .	113
23	ATMOTRAN and Capsaicin code-to-code comparison (Problem 2) . .	113
24	Energy scale used in ATMOTRAN	172

1 Introduction

In this chapter, the research goals of the dissertation are discussed. Following this, we define the physical problem of outward transport of atmospheric neutrons. We continue by reviewing literature discussing past application of Monte Carlo and deterministic transport methods to solve variants of the problem.

1.1 Research Goals

Earth's atmosphere has enabled the development and sustainment of terrestrial life. Outside the thin atmospheric layer surrounding our planet there exists a universe dominated by the presence of intense radiation. Space radiation impacting the Earth includes the particles emanating from the Sun during solar flare events, the galactic cosmic rays originating from outside the solar system, and local magnetically-trapped radiation belts consisting of charged particles. Understanding how these extraterrestrial radiation sources interact with the Earth's magnetic field and atmosphere through mathematical modeling has been important to predicting terrestrial effects from solar storms, flares, and coronal mass ejections.

Radiation transport modeling has also enabled significant engineering advances in our abilities to harness the energy available in nuclear radiation. These include the design of nuclear power reactors, the development of safer techniques in medical imaging, defense and industrial applications, and space travel. These applications rely

on a capability to predict system performance as it relates to radiation physics. A complementary problem to understanding how radiation propagates inward through the atmosphere to the Earth, is how human-generated radiation might disperse as it propagates outward from the surface. Radiation transport in the atmosphere is an important numerical modeling calculation in many fields including astronomy, environmental science, civil engineering and defense.

Applications of these calculations may include, for example, the determination of a radiation dose field from a nuclear accident, and may inform the design of shielding suitable for a radiographic analysis system. At a larger geographic scale it may be useful to determine the radiation fields escaping the lower, thicker layers of the atmosphere and impinging on objects of interest in the upper atmosphere, for example high altitude aircraft and satellites.

This dissertation considers the application of neutral-particle deterministic transport methods to the assessment of radiation fields at high altitude from point sources either above or below, motivating the development of a new transport code, ATMOTRAN, that is designed to provide accurate, full phase-space results of radiation fields at a low computational expense relative to other methods of calculation. ATMOTRAN is intended to enable calculation of the propagation of a source of neutrons through an atmosphere with varying material properties. The code will calculate the expected neutron fluence at all altitudes and polar angles, as a function of energy, resulting from this source. This required us to derive an approximation to the transport equation solved in spherical coordinates and requires two spatial dimensions, one in a curvilinear direction. Because Monte Carlo codes are not computationally efficient in the target application, we chose to use deterministic methods. While there has been one deterministic code written for two-dimensional spherical coordinates [1], it did not utilize a discretization scheme that both preserves the diffusion limit and does

not put extremely challenging requirements on the cell size of any finite volumes used in the calculation. There are three-dimensional spherical codes, for example [2], but these use diamond differencing in space and angle, which is not practical for systems as large as the Earth. Our goal then, is to derive a new bilinear discontinuous discretization of the two-dimensional spherical-polar transport equation that generates results with high accuracy and, in conjunction with standard multigroup techniques, will provide fast results of neutron fluence at all points in the physical phase space of the atmospheric problem.

1.2 Problem Definition

Challenges of radiation transport modeling in the atmosphere include vast distances and widely varying material properties. While some applications of interest may be approximated, for example, by reducing the physical problem to a one-dimensional atmospheric slab [3], others require the full three-dimensional geometry of the atmosphere to be considered. In favor of expediency, others may neglect azimuthally-varying physical properties to reduce the physical atmosphere to a two-dimensional spherical-polar (r, θ) coordinate system. The propagation of a source of neutral particles through great distances in the atmosphere is such a physical system. Although the atmosphere varies with altitude, both greatly in density and only slightly in stoichiometry, only slight variations appear in polar and azimuthal angles (latitude and longitude). Considering the flow of an approximate point (or localized) source of particles, it is of course necessary to consider the distance traveled in both the radial and polar angle dimensions.

Figure 1 outlines the geometric problem. The smaller circle represents the Earth, modeled as a sphere with an approximate average radius R_E of 6378 km. The atmosphere A extends from the surface up to some boundary altitude. The results

in this dissertation use $A = 1000$ km, owing to available atmospheric models. The source of radiation is point-like at radius R_S from the center of the Earth, at angle ζ from the Earth's axis. When considering a two-dimensional spherical-polar system, we always set this system at $\zeta = 0$. The source axis defines a polar angle θ_0 (for our system $\theta_0 = 0$) such that a polar coordinate system extends from this base angle. Some point for which we want to calculate the particle fluence sits at angle θ and at radius R_D . The azimuthal dimension is represented by ϕ , and when $\zeta = 0$, the source is azimuthally isotropic, and the atmospheric material properties do not vary appreciable with azimuthal angle, the problem becomes azimuthally symmetric. This is the approximation we consider here.

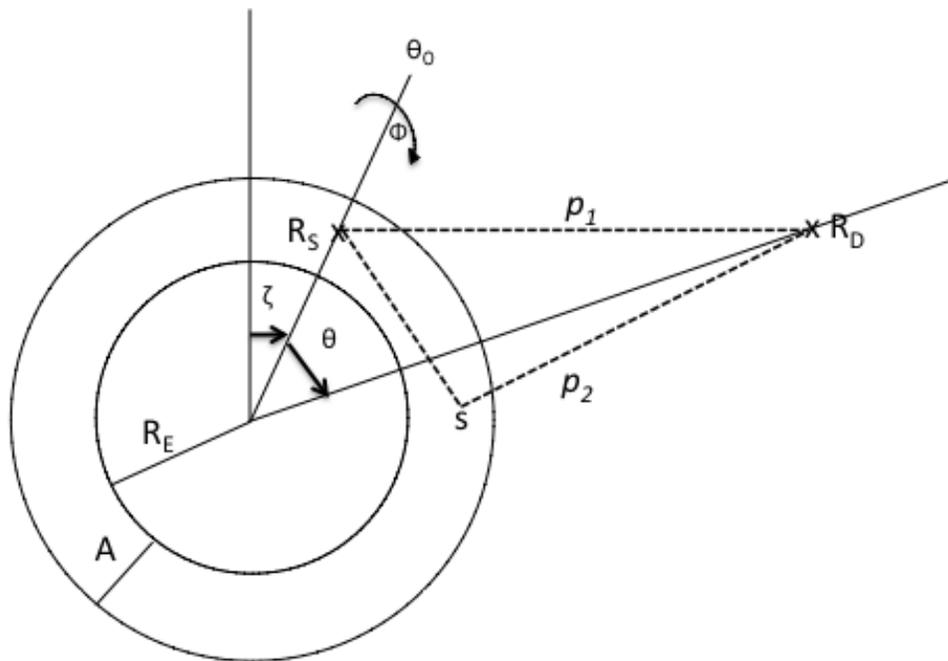


Figure 1: Problem geometry (not to scale).

The atmosphere is not a microscopically dense medium compared to solid material, but is optically dense to neutrons and gamma rays, considering the many tens or hundreds of kilometers that must be traversed to escape it. The total mean free path for fast neutrons at sea level is on the order of ten meters. The atmosphere is a highly scattering medium, although there is some absorption (capture and inelastic scattering) that results in secondary gamma-ray production. Along with this there are anisotropic scattering effects varying with particle energy. The probability of atmospheric neutron capture and scattering depends primarily on density. The density profile as a function of altitude is shown in Figure 2.

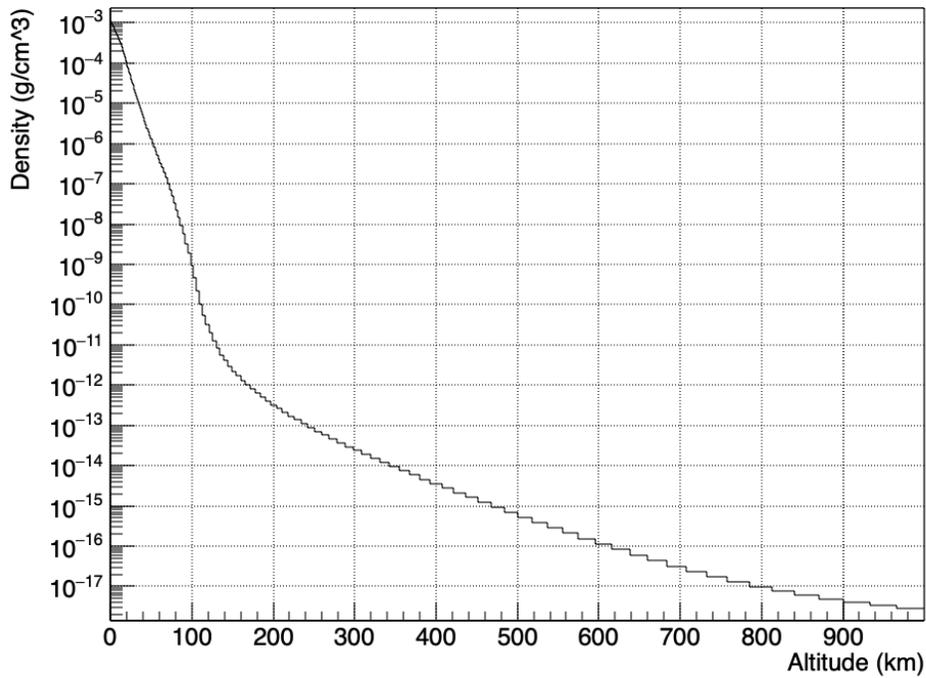


Figure 2: Total atmospheric density (g/cm^3) as a function of altitude (km).

As shown in the figure, the density of the atmosphere varies by some 15 orders of magnitude from Earth's surface up to 1000 km altitude. A secondary effect is variation in the atmospheric elemental constituents. Nearly all the atmospheric mass

is contained below 30 km altitude, and in this range the ratios of the gas elements are constant due to turbulent mixing [4]. The effect of the sharp density decline around 30 km impacts neutron transmission differently depending on the altitude of emission. For neutrons below 30 km, the atmosphere acts as an attenuator, while for neutrons above 30 km the atmosphere acts as a reflector, and for much higher altitude emission the atmosphere can act somewhat as a lens, with the net effect of bending neutron radiation, incident at the atmospheric edge, around the Earth to points where there is no direct line of sight to the source.

The United States Naval Research Laboratory maintains a freely available model of the atmosphere called Mass Spectrometer and Incoherent Scatter Radar Exosphere (MSISE) [5]. This model can be linked as a library into C++ simulation codes. In the model, the atmospheric density and elemental composition is provided as a function of altitude, latitude, longitude, year, day of year, etc., for altitudes ranging from the Earth's surface up to 1000 km. The dominant variation is in altitude, such that the implementation described in this work assumes a purely radial dependence of the atmospheric density and elemental composition. As mentioned earlier, this allows the problem to be reduced to two spatial dimensions, so that for a given source, the neutron flux can be spatially described by the radial distance from the center of the Earth, and the polar angle from the source (θ in Figure 1).

The cross sections of the dominant atmospheric constituents, ^{14}N and ^{16}O are 1-3 barns for keV-MeV neutrons. However, as shown in Figure 3, within the resonance region the total cross section can exceed 10 barns for narrow energy widths. At lower energies in this range, neutron capture is a significant component of the physics and neutron absorption is important. This property can hamper codes reliant on Monte Carlo methods that must employ variance reduction techniques to improve statistical accuracy in highly absorbing media. There is also angular anisotropy in the scattering

cross section as a function of incident energy as shown in Figure 4. In general, for neutrons around 1-2 MeV elastic scattering from nitrogen and oxygen the scattering is isotropic within a factor of two, and as neutron energy approaches 10 MeV and higher the scattering is forward-peaked.

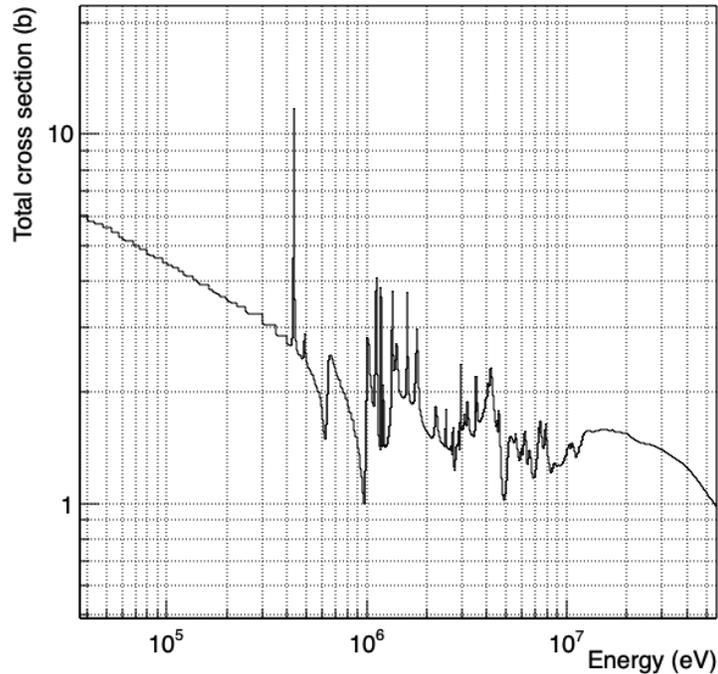


Figure 3: Total neutron cross section (b) as a function of energy (eV) for ^{14}N in the resonance region.

1.3 Current state of the problem

In this section we review some past methods employed to solve the problem of neutral particle radiation transport in the atmosphere, and curvilinear transport methods in general. The methods of primary interest that will be discussed here include Monte Carlo and deterministic methods. This literature review will motivate the development of ATMOTRAN.

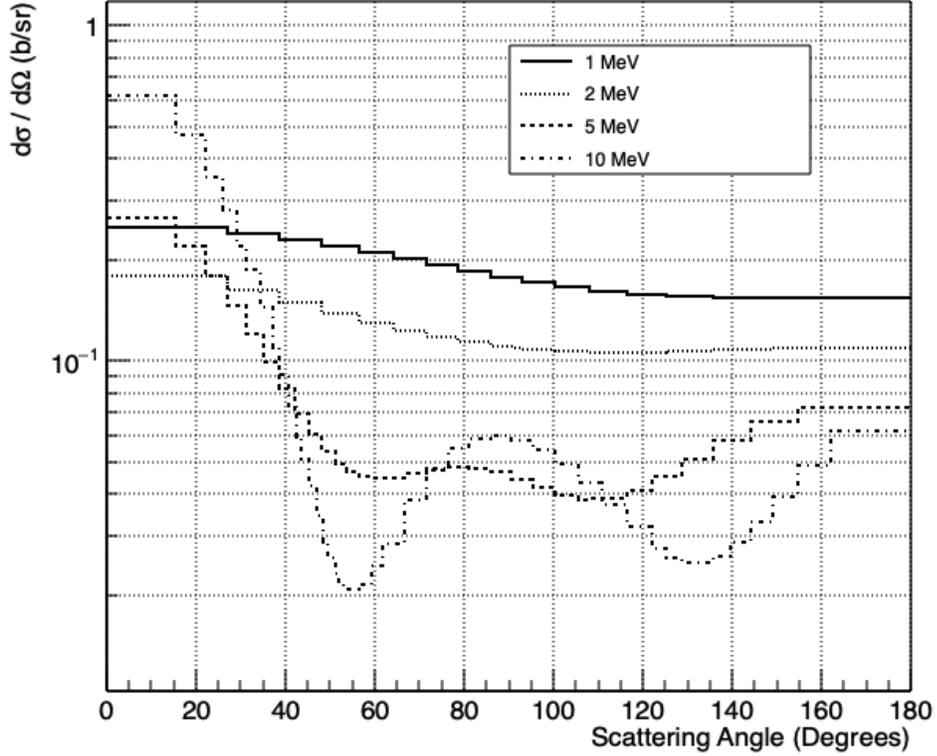


Figure 4: Neutron cross section as a function of scattering angle for ^{14}N , for neutron energies ranging from 1 MeV to 10 MeV.

1.3.1 Solution by Monte Carlo methods

Much of the existing peer-reviewed work on atmospheric neutron transport concerns particles impinging on the earth from solar or cosmic-ray origins [6–8], with additional specific reporting on transport of lightning-induced neutrons [9–11]. Point-source neutron propagation in the atmosphere has been attempted by setting up a Monte Carlo simulation using MCNP [12–15]. In [13], Culp performed MCNP simulations for a point source of neutrons at an altitude of 40 km and compared the Monte Carlo results with the SMAUG-II code, a deterministic code in one-dimensional spherical coordinates. He concluded that two-dimensional effects are important to model point source propagation. In [14] and [15], Byrd used MCNP to build an atmospheric model to examine various phase space points for this problem, calculating neutron

fluences as a function of angle and altitude and found interesting effects related to secondary particle production. While it is possible to perform such simulations in an analog Monte Carlo run, these runs are too time consuming to achieve suitable statistics at all points in the problem. Monte Carlo can be accelerated by using variance reduction techniques [16], although the opacity of the atmosphere will still cause significant issues due to the sheer size of the problem. For example while it is possible to use importance sampling (splitting) and weight windowing for deep shielding calculations these techniques have immense calculational burden when considering the atmospheric problem. A standard deep-shielding Monte Carlo simulation may calculate neutron shielding properties of several feet (tens of mean free paths) of concrete, requiring tens of artificially spatially-defined regions with boundaries at which to perform the splitting, one region at roughly every mean free path. The atmosphere at sea level is hundreds of thousands of mean free paths thick for neutrons traveling in the direction of the horizon. Additionally, the variance of the material properties requires varying cell sizes with altitude, and the number of variance reduction cells must number in the millions to achieve gains from splitting. Implicit absorption techniques are also possible, but again there are practical limitations to the achievable gains in such a large system. Thus, Monte Carlo is only practically used to calculate the neutron flux for some specific case of neutron emission, and that results take a very long time to achieve reliable statistical accuracy. For these reasons, we do not consider Monte Carlo to be a viable alternative to deterministic techniques to achieve a general purpose fast-running code, although in this dissertation we do present numerical comparison of Monte Carlo with ATMOTRAN results, to perform a verification spot-check.

1.3.2 Solution by deterministic methods

Work done at LANL in the late 2000's produced results from a diffusion code targeting the atmospheric problem [17]. The code blended Monte Carlo and diffusion, utilizing MCNP at higher emission altitudes and from a custom diffusion code at lower altitudes. The diffusion code used included a number of approximations, including a two-dimensional Cartesian model of the atmosphere over a flat-Earth and a purely exponential atmospheric density, which resulted in limited accuracy above 100 km, and angular inaccuracies due to the flat-Earth geometry. The technical limitations of this code brought on by these approximations preclude its use in the present problem.

We mentioned that ATMOTRAN is a two-dimensional, spherical-polar (r, θ) transport code. Although this coordinate system is quite useful for atmospheric problems it has only been considered once before in the deterministic transport code development community. The application of transport methods to curvilinear coordinate systems is commonly presented for one-dimensional spherical coordinates and one- or two-dimensional cylindrical coordinates $(r - z)$, and in all cases there is only one angular derivative to consider, simplifying both the method derivation and the code implementation. The literature suggests that solution of the transport equation in the spherical (r, θ) coordinate system has only been implemented once before in a code called TWOTRAN SPHERE [1, 18, 19], sometime around 1970. That work focused on peculiarities of the transport equation that must be considered including approximating the two angular derivatives in the transport equation, and study of the characteristic equations of motion through the spatio-angular mesh, which impact the required sweep progression in different halves of the problem. TWOTRAN SPHERE was diamond-differenced in angle and space, and used conventions of the time including the step approximation to begin the angular recursion, rather than zero-weighted starting directions, and no weighted-diamond-difference (WDD) for-

mulation in angle [20]. There is no evidence that the code was used for solving problems at the atmospheric scale, for which diamond-differencing in space would necessitate an immense number of spatial cells to avoid negative fluxes. We discuss this further in the next chapter. The present work proposes a new method of solution to the two-dimensional transport equation in (r, θ) spherical coordinates. Starting from some of the results of TWOTRAN SPHERE, most importantly the angular sweeps, we will develop a bilinear-discontinuous finite element method (DFEM) spatial discretization [21–23], with a WDD approximation to discretize the angular derivatives, using zero-weighted starting directions in the two angular dimensions. This dissertation describes details of the derivation and implementation, including a discussion on how to upwind the DFEM and sweep in this coordinate system.

1.4 Chapter outline

Following this introduction to the problem, Chapter 2 reviews the transport equation, discretization schemes, and the TWOTRAN SPHERE code. Chapter 3 provides details of the derivation of the bilinear discontinuous discretization of the two-dimensional spherical-polar form of the transport equation. Chapter 4 continues with details of the implementation of the discretization into a working code. Chapter 5 gives evidence that the discretization is valid as presented and that the code is implemented correctly, through verification by comparison with manufactured solutions and code-to-code comparisons. Chapter 6 summarizes the work and discusses anticipated future work.

2 The Discrete Ordinates Method

In this chapter we introduce the transport equation and discuss methods of solution. We provide background information on the methods of discretization of physical variables. We review the spherical (r, θ) code TWOTRAN SPHERE and motivate a new spatial discretization.

Particle transport calculations are foundational to the development of nuclear physics instrumentation. The Boltzmann equation is an integro-differential volumetric balance formulation describing the angular, spatial and energy variations of a particle distribution. Analytical solutions of the equation are not possible for all but the simplest physical problems, thus a computer program employing a discretized numerical method is often used [24]. The method of discrete ordinates is commonly used to discretize the angular variable; it is described in the original form in [25] and refined in [27]. The discrete ordinates (also called S_N) method simplifies the Boltzmann equation by evaluating the solution for certain values of the angular variable. Instead of integrating over the continuous range of scattering angles, the integral is converted to a weighted sum (quadrature). Angular discretization relies on particles undergoing enough scattering interactions in enough cells to average out the effects of discretization. This is truly the case in a dense, highly scattering material. If the source is distributed through the scattering medium the approximation works even better. For point-like sources there is a problem in particular if the medium is absorb-

ing (weakly scattering) or low density. In this case, source particles are emitted into a limited number of directions and can only scatter into a limited number of directions in adjacent cells. These "ray effects" [26] can limit the effectiveness of discrete ordinates for certain problems and a number of remedies are available although the fundamental issue remains. This will be true in the atmospheric problem particularly at high altitudes although an uncollided source method can help. Another issue in discrete ordinates is the iterative nature of the method that is typically employed to solve the discretized S_N equations. In the first iteration (Richardson method), source particles can scatter once into a new phase space region (in energy, angle). In the next iteration, those scattered particles can scatter once more into yet another phase space region. Highly scattering media will generate flux estimates with small differences between each source iteration. This slows the convergence of the method to such a degree that some problems require a high number of iterations. This has led to the development of acceleration techniques that are applied within an iteration and are designed to achieve convergence more quickly. The state of the art technique is to apply an acceleration method to the transport iteration based on a lower order approximation to the transport equation. A Krylov method, for example the Generalized Minimum Residual (GMRES), with such an acceleration method recast as a preconditioner, can be used to speed convergence [28, 29]. The inner acceleration operator must be consistently discretized with the outer transport operator to ensure the acceleration and preconditioning is effective and robust. A discretization method that can support this requirement is the linear discontinuous (LD) method [30, 31], or in two dimensions, the bilinear discontinuous method (BLD) [32]. The BLD method also preserves the diffusion limit, which is an important condition to ensure that solutions are accurate in optically thick problems without having to employ a large number of mesh cells.

2.1 The transport equation

The balance equation that describes neutral particle (neutron and photon) transport is known as the Boltzmann Equation. This equation is derived by balancing the effects related to particle balance for a general point in phase space (position, direction, energy and time). This phase space is generally seven-dimensional (seven independent variables: three in position, two in direction, one in each of energy and time). The complexity of the Boltzmann equation renders it impossible to solve analytically except in limited cases, for example in one-dimensional slab geometry. Approximations to the solution are obtained by numerical techniques involving the discretization of usually all of the continuous variables present in the equation. The full form of the integro-differential time-independent linear Boltzmann equation describing neutron transport and including prompt fission effects is given here:

$$\begin{aligned}
 & \boldsymbol{\Omega} \cdot \nabla \psi(\mathbf{r}, E, \boldsymbol{\Omega}) + \Sigma_t(\mathbf{r}, E) \psi(\mathbf{r}, E, \boldsymbol{\Omega}) \\
 &= \int_0^{E_{max}} dE' \int_{4\pi} d\boldsymbol{\Omega}' \Sigma_s(\mathbf{r}, E' \rightarrow E, \boldsymbol{\Omega}' \cdot \boldsymbol{\Omega}) \psi(\mathbf{r}, E', \boldsymbol{\Omega}') \\
 &+ \frac{\chi(E)}{4\pi} \int_0^{E_{max}} dE' \int_{4\pi} d\boldsymbol{\Omega}' \nu(\mathbf{r}, E') \Sigma_f(\mathbf{r}, E') \psi(\mathbf{r}, E', \boldsymbol{\Omega}') \\
 &+ Q(\mathbf{r}, E, \boldsymbol{\Omega}) \quad ,
 \end{aligned} \tag{1}$$

using standard notation as follows:

- \mathbf{r} , the spatial position vector, which is typically expressed in three-dimensional form in Cartesian (x, y, z) , cylindrical polar (r, θ, z) , or spherical polar (r, θ, ϕ) coordinates;
- $\boldsymbol{\Omega}$, the direction of neutron motion;
- E , the particle energy;

- $\Sigma_t(\mathbf{r}, E)$, the total cross section of the material at position \mathbf{r} for neutrons of energy E . The total cross section is the per unit path length probability of neutron collision in the material. This value is known as the macroscopic cross section, defined as $\Sigma = N\sigma$, where σ is the microscopic cross section in units of cm^2 and N is the neutron density in units of inverse volume;
- $\Sigma_s(\mathbf{r}, E' \rightarrow E, \boldsymbol{\Omega}' \cdot \boldsymbol{\Omega})$, the differential macroscopic scattering cross section of the material at position \mathbf{r} . This value is the probability per unit path length of a neutron at energy E' and direction $\boldsymbol{\Omega}'$ scattering into an energy window dE about energy E and direction window $d\Omega$ about $\boldsymbol{\Omega}$;
- $\chi(E)$, the energy spectrum of neutrons produced in fission;
- $\nu(\mathbf{r}, E)$, the multiplicity of neutrons produced in the material at position \mathbf{r} and energy E ;
- $\Sigma_f(\mathbf{r}, E)$, the macroscopic fission cross section of the material at position \mathbf{r} for neutrons of energy E . The total cross section is the per unit path length probability of a collision causing fission in a multiplying material;
- $Q(\mathbf{r}, E, \boldsymbol{\Omega})$, the source of neutrons at position \mathbf{r} at energy E and direction $\boldsymbol{\Omega}$. For the time-independent transport equation this source is always present at the same amplitude, and for time-dependent problems this source can be transient;
- $\psi(\mathbf{r}, E, \boldsymbol{\Omega})$ is the angular flux, equivalent to $vN(\mathbf{r}, E, \boldsymbol{\Omega})$, where $v = \sqrt{2E/m}$ is the neutron speed and N is the angular neutron density, and E and m are the neutron energy and mass.

The linear Boltzmann equation expresses a neutron balance equation for particles in a particular phase space. The phase space balance comprises independent variables

in position, energy and angle. A phase space volume can be expressed as the product of several differential values $dV dE d\Omega$. The derivation of this balance equation examines an arbitrary phase space volume, placing neutron gain terms on the right and neutron loss terms on the left.

Neutron gain terms include:

- $\int_0^{E_{max}} dE' \int_{4\pi} d\Omega' \Sigma_s(\mathbf{r}, E' \rightarrow E, \boldsymbol{\Omega}' \cdot \boldsymbol{\Omega}) \psi(\mathbf{r}, E', \boldsymbol{\Omega}') dV dE d\Omega$ is the neutron gain at $dV dE d\Omega$ at \mathbf{r} due to in-scattering to energy dE about E and direction $d\Omega$ about $\boldsymbol{\Omega}$ from all other energies E' and directions $\boldsymbol{\Omega}'$;
- $\frac{\chi(E)}{4\pi} \int_0^{E_{max}} dE' \int_{4\pi} d\Omega' \nu(\mathbf{r}, E) \Sigma_f(\mathbf{r}, E) \psi(\mathbf{r}, E', \boldsymbol{\Omega}') dV dE d\Omega$ is the neutron gain at $dV dE d\Omega$ at \mathbf{r} at energy E due to fission at all energies;
- $Q(\mathbf{r}, E, \boldsymbol{\Omega}) dV dE d\Omega$ is the neutron gain due to a direct source of neutrons.

Neutron loss terms include:

- $\Sigma_t(\mathbf{r}, E) \psi(\mathbf{r}, E, \boldsymbol{\Omega}) dV dE d\Omega$ is the neutron loss due to collisions with the medium within the phase space. This includes neutron scattering into, effectively, different volumes of phase space, and neutron absorption;
- $\boldsymbol{\Omega} \cdot \nabla \psi(\mathbf{r}, E, \boldsymbol{\Omega})$ is the neutron loss due to neutrons streaming out of the phase space volume. This term can be negative for some angles $\boldsymbol{\Omega}$, in which case those terms become those of neutron gain.

In the discrete ordinates method the differential cross section is usually defined as an expansion in Legendre polynomials

$$\Sigma_{s,g' \rightarrow g}(\mathbf{r}, E' \rightarrow E, \boldsymbol{\Omega}' \cdot \boldsymbol{\Omega}) = \sum_{\ell=0}^L \frac{2\ell + 1}{4\pi} \Sigma_{s\ell,g' \rightarrow g}(\mathbf{r}, E' \rightarrow E) P_\ell(\boldsymbol{\Omega}' \cdot \boldsymbol{\Omega}) \quad , \quad (2)$$

where

$$P_\ell(\boldsymbol{\Omega}' \cdot \boldsymbol{\Omega}) = \frac{2}{2\ell + 1} \sum_{m=-\ell}^{\ell} Y_{\ell m}(\boldsymbol{\Omega}') Y_{\ell m}^*(\boldsymbol{\Omega}) \quad , \quad (3)$$

where $Y_{\ell m}(\boldsymbol{\Omega})$ are the spherical harmonic functions following these relations

$$Y_{\ell m}(\boldsymbol{\Omega}) = Y_{\ell m}(\mu, \omega) = C_{\ell m} P_\ell^{|m|}(\mu) e^{im\omega} \quad , \quad (4)$$

where

$$C_{\ell m} = (-1)^{(m+|m|)/2} \left[\frac{2\ell + 1}{2} \frac{(\ell - |m|)!}{(\ell + |m|)!} \right]^{1/2} \quad , \quad (5)$$

and

$$P_\ell^m(\mu) = (-1)^m (1 - \mu^2)^{m/2} \frac{d^m}{d\mu^m} P_\ell \quad , \quad (6)$$

defining a relation

$$(-1)^m Y_{\ell, m}^*(\boldsymbol{\Omega}) = Y_{\ell, -m}(\boldsymbol{\Omega}) \quad . \quad (7)$$

In the case of isotropic scattering, the polynomial expansion is truncated at $\ell = 0$ and the scattering cross section reduces to

$$\Sigma_{s\ell}(\mathbf{r}, E' \rightarrow E, \boldsymbol{\Omega}' \cdot \boldsymbol{\Omega}) = \Sigma_{s0}(\mathbf{r}, E' \rightarrow E) \quad , \quad (8)$$

indicating that the relative change in the cosine of the scattering angle, $\boldsymbol{\Omega}' \cdot \boldsymbol{\Omega}$, is independent of orientation. This is a convenient simplification but is generally not the case in practical problems, especially at higher neutron energies where scattering becomes typically more forward-peaked in angle.

The Boltzmann equation must be solved subject boundary conditions, for example the following condition:

$$\psi(\mathbf{r}, E, \boldsymbol{\Omega}) = \psi^b(\mathbf{r}, E, \boldsymbol{\Omega}), \quad \mathbf{r} \in \partial V, \quad \boldsymbol{\Omega} \cdot \hat{\mathbf{n}} < 0, \quad 0 < E < \infty \quad . \quad (9)$$

This condition constrains the angular flux solution boundary of the problem domain for inward neutron directions. Here, \hat{n} is the outward pointing surface normal for the boundary surface and $\mathbf{\Omega}$ is the particle direction. For example, many problems specify $\psi^b = 0$, called the vacuum boundary condition, indicating that there is no incident source of particles either originating external to the problem geometry, or reflecting from its extents. For ATMOTRAN, the maximum radial boundary condition (the upper atmosphere) is a vacuum boundary condition, since there is no way for particles exiting the atmosphere to scatter back into it. Another common boundary condition, also used in ATMOTRAN, is the reflective boundary condition, which ensures symmetry of the solution when solving full circle spherical problems on a hemisphere:

$$\psi(\mathbf{r}, E, \mathbf{\Omega}) = \psi(\mathbf{r}, E, -\mathbf{\Omega}) \quad . \quad (10)$$

The discrete ordinates method discretizes the angular flux term in the differential form of the transport equation. Instead of solving for all possible angles of travel, only a subset of these angles are chosen, and the transport equation solved. The angular subset is chosen very specifically, following quadrature rules, in order to accurately match the integral value of the non-discretized equation.

2.2 Coordinate system

The three dimensional spherical coordinate system is shown in Figure 5. At a point specified by position (r, θ, ϕ) we show a direction vector $\mathbf{\Omega}$. The independent angular directions are μ and ω . The integration over the spatial azimuth ϕ reduces the spatial problem to two dimensions, shown in Figure 6, and we show a direction vector for the case $\xi = 0$, although this is not generally the case. The angular component η is

defined

$$\eta = (1 - \mu^2)^{1/2} \cos(\omega) \quad . \quad (11)$$

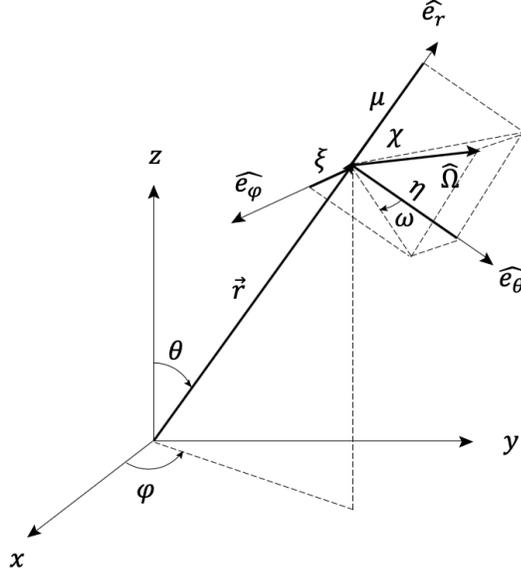


Figure 5: Three-dimensional spherical coordinates extended view of vector directions

The phase space is divided into cells in space and angle using the indexing conventions shown in Figure 7.

2.3 Angular discretization

The linear Boltzmann equation is continuous in angle. In the discrete ordinates method, the integrals over angle are converted to weighted sums using the quadrature method

$$\int_{4\pi} d\Omega \psi(\Omega) \approx \sum_{k=1}^K w_k \psi_k(\Omega_k) \quad , \quad (12)$$

where K is the total number of angles used in the sum, and Ω_k is a particular angle, and k indicates some point (m, ℓ) , referring to Figure 7. The angle values

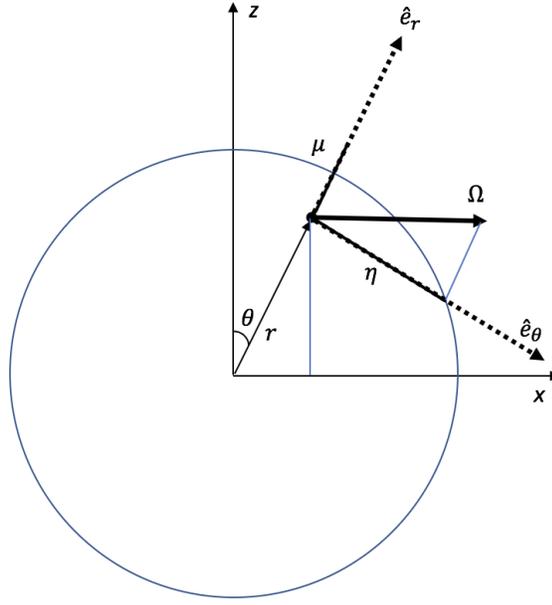


Figure 6: Two-dimensional spherical coordinates

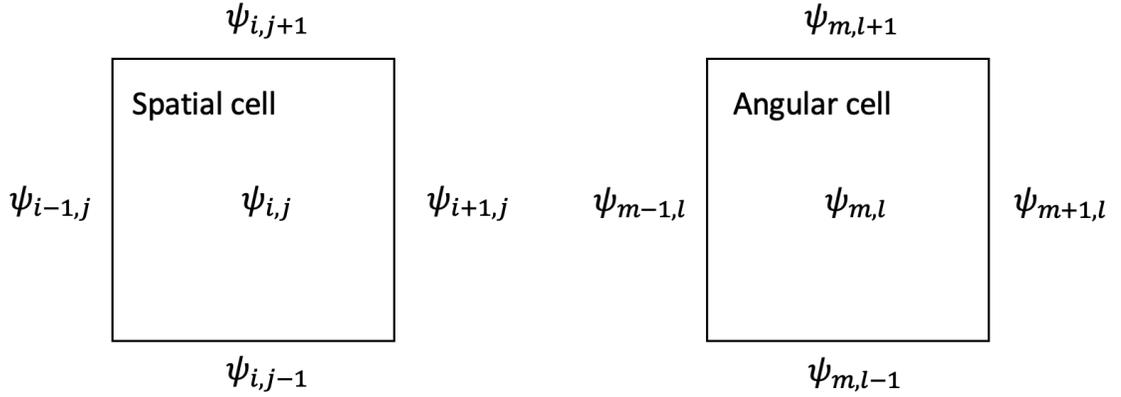


Figure 7: Spatial and Angular cell indexing convention.

and their weight w_k are chosen following quadrature rules, which ensure the integral values of the quantities under integration are conserved. For discrete ordinates codes, the selection of suitable quadrature sets is an important consideration to ensure accurate solutions. There are various methods of generating quadrature sets depending on the dimensionality of the problem or certain physical requirements

such as highly forward-peaked scattering [27, 33]. For multi-dimensional problems, a Gauss-Chebyshev-Legendre quadrature set is often used for its favorable integration properties [34]. Discretization rules must also be applied to angular derivatives. These are complicated for curvilinear problems since straight-line trajectories through the curvilinear coordinate system result in changing values of the angular components. This process is called “angular redistribution”. Figure 8 shows how the straight line trajectory forms a variable angle with the cell surface direction vectors depending on geometric position. For a standard diamond-differencing, an angular mesh cell

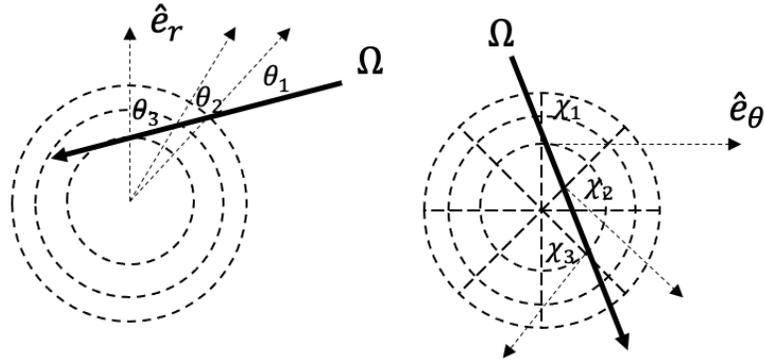


Figure 8: Variability of angle (indexed θ and χ) in the coordinate frame during straight line trajectory through a sphere.

at some intermediate angle index k , where the range of indices is $k \in [1, K]$, we approximate angular cell edge values in the following way

$$\psi_k = \frac{1}{2}(\psi_{k+1/2} + \psi_{k-1/2}) \quad , \quad (13)$$

where $\psi_k = \psi_i$, the spatial cell average flux. These relations are used when sweeping the angular mesh in a particular direction. If sweeping from low to high index value, at index k , $\psi_{k-1/2}$ is known from the previous calculation, and we solve for $\psi_{k+1/2}$ upon obtaining ψ_k . A special case is for index $k = 1$ for which we do not have a prior

calculation but must have some value $\psi_{1/2}$ with which to begin the recursion. This value is called the starting direction. For systems with transport equations having two angular derivatives, starting directions are required in two dimensions, and this is explored in the present document. Weighted diamond differencing was introduced as a method to improve the performance of diamond angular differencing by ensuring that the angular cell centers are collocated with the quadrature root, which in general they are not for typical quadrature sets [20]. This weighted diamond method is extended to two dimensions in this dissertation with demonstrated success. Weighted diamond is still a very common angular discretization although discontinuous methods have also been introduced for this purpose [35].

2.4 Energy discretization

The linear Boltzmann equation is continuous in energy. Deterministic methods solve the equation numerically by discretizing the independent phase space variables, including energy. This is unlike Monte Carlo, which treats energy as a continuous variable. Discretizing the energy domain amounts to picking a number of groups in energy within which all particles behave alike. Since particle behavior is most affected by material cross sections it is reasonable to want to pick enough groups to adequately reproduce the cross section data. However, the larger the number of groups the greater the computational memory required to store the solution. The number of energy bins required to reproduce the complex energy-dependent shapes of cross section data, particularly in the resonance region ($\sim 0.1-10$ MeV), is prohibitive for many materials. The problem then becomes how to get an approximate cross section value for an energy group when the cross section varies significantly. These problems are tackled in the multigroup form of the transport equation. To demonstrate the form of the multigroup approximation we return to the steady state, three-dimensional transport

equation. We assume non-multiplying media so the fission term is not present in the following equation:

$$\begin{aligned}
& \boldsymbol{\Omega} \cdot \nabla \psi(\mathbf{r}, E, \boldsymbol{\Omega}) + \Sigma_t(\mathbf{r}, E) \psi(\mathbf{r}, E, \boldsymbol{\Omega}) \\
&= \int_0^{E_{max}} dE' \int_{4\pi} d\boldsymbol{\Omega}' \Sigma_s(\mathbf{r}, E' \rightarrow E, \boldsymbol{\Omega}' \cdot \boldsymbol{\Omega}) \psi(\mathbf{r}, E', \boldsymbol{\Omega}') \\
&+ \frac{Q}{4\pi}(\mathbf{r}, E, \boldsymbol{\Omega}) \quad .
\end{aligned} \tag{14}$$

A finite number, G , of energy groups is chosen, typically ordered with subscripts such that the highest energy group is denoted E_0 . The relation between the groups, or energy bin centers is:

$$E_{G-1} < \dots < E_g < E_{g-1} < \dots < E_1 < E_0 \tag{15}$$

To give the best possible accuracy, the maximum extent of the energy range covered by the groups should ensure that no neutrons are present in the physical system at energies less than the lower bound of the E_G bin and conversely at E_0 . Generally, particles lose energy as they scatter, so they move upwards in group index number. The typical exception for neutrons is that neutrons at thermal energies, which equate to speeds comparable to the thermal motion of particles in the host material, can kinematically gain energy in collisions, although they do not gain appreciable energy, such that these effects can be contained within the lowest group or groups, depending on the required fidelity of the problem. Now, if we define the group angular flux and the group internal source in the following way,

$$\psi_g(\mathbf{r}, \boldsymbol{\Omega}) = \int_{E_g}^{E_{g-1}} dE \psi(\mathbf{r}, E, \boldsymbol{\Omega}) \quad , \tag{16}$$

and

$$Q_g(\mathbf{r}, \boldsymbol{\Omega}) = \int_{E_g}^{E_{g-1}} dE Q(\mathbf{r}, E, \boldsymbol{\Omega}) \quad , \quad (17)$$

and integrating the equation over an energy group

$$\begin{aligned} & \boldsymbol{\Omega} \cdot \nabla \int_{E_g}^{E_{g-1}} dE \psi(\mathbf{r}, E, \boldsymbol{\Omega}) + \int_{E_g}^{E_{g-1}} dE \Sigma_t(\mathbf{r}, E) \psi(\mathbf{r}, E, \boldsymbol{\Omega}) \\ &= \sum_{g'=1}^G \int_{E_g}^{E_{g-1}} dE \int_{E_{g'}}^{E_{g'-1}} dE' \int_{4\pi} d\boldsymbol{\Omega}' \Sigma_s(\mathbf{r}, E' \rightarrow E, \boldsymbol{\Omega}' \cdot \boldsymbol{\Omega}) \psi(\mathbf{r}, E', \boldsymbol{\Omega}') \\ &+ \int_{E_g}^{E_{g-1}} dE \frac{Q}{4\pi}(\mathbf{r}, E, \boldsymbol{\Omega}) \quad , \end{aligned} \quad (18)$$

and making the necessary substitutions we have

$$\begin{aligned} & \boldsymbol{\Omega} \cdot \nabla \psi_g(\mathbf{r}, \boldsymbol{\Omega}) + \int_{E_g}^{E_{g-1}} dE \Sigma_t(\mathbf{r}, E) \psi(\mathbf{r}, E, \boldsymbol{\Omega}) \\ &= \sum_{g'=1}^G \int_{E_g}^{E_{g-1}} dE \int_{E_{g'}}^{E_{g'-1}} dE' \int_{4\pi} d\boldsymbol{\Omega}' \Sigma_s(\mathbf{r}, E' \rightarrow E, \boldsymbol{\Omega}' \cdot \boldsymbol{\Omega}) \psi(\mathbf{r}, E', \boldsymbol{\Omega}') \\ &+ \frac{Q_g}{4\pi}(\mathbf{r}, \boldsymbol{\Omega}) \quad . \end{aligned} \quad (19)$$

The remaining integrals are over a convolution of cross section and flux functions. To get these in terms of the group angular flux, we multiply these terms by unity, in the form of the group angular flux definition:

$$\begin{aligned}
& \boldsymbol{\Omega} \cdot \nabla \psi_g(\mathbf{r}, \boldsymbol{\Omega}) + \left[\frac{\int_{E_g}^{E_{g-1}} dE \Sigma_t(\mathbf{r}, E) \psi(\mathbf{r}, E, \boldsymbol{\Omega})}{\int_{E_g}^{E_{g-1}} dE \psi(\mathbf{r}, E, \boldsymbol{\Omega})} \right] \psi_g(\mathbf{r}, \boldsymbol{\Omega}) \\
&= \sum_{g'=1}^G \int_{4\pi} d\Omega' \left[\frac{\int_{E_g}^{E_{g-1}} dE \int_{E_{g'}}^{E_{g'-1}} dE' \Sigma_s(\mathbf{r}, E' \rightarrow E, \boldsymbol{\Omega}' \cdot \boldsymbol{\Omega}) \psi(\mathbf{r}, E', \boldsymbol{\Omega}')}{\int_{E_{g'}}^{E_{g'-1}} dE' \psi(\mathbf{r}, E', \boldsymbol{\Omega}')} \right] \psi_{g'}(\mathbf{r}, \boldsymbol{\Omega}') \\
&+ \frac{Q_g}{4\pi}(\mathbf{r}, \boldsymbol{\Omega}) \quad .
\end{aligned} \tag{20}$$

More compactly, the multigroup equations can be written

$$\begin{aligned}
\boldsymbol{\Omega} \cdot \nabla \psi_g(\mathbf{r}, \boldsymbol{\Omega}) + \hat{\Sigma}_{t,g}(\mathbf{r}, \boldsymbol{\Omega}) \psi_g(\mathbf{r}, \boldsymbol{\Omega}) &= \sum_{g'=1}^G \int_{4\pi} d\Omega' \hat{\Sigma}_{s,g' \rightarrow g}(\mathbf{r}, \boldsymbol{\Omega}' \cdot \boldsymbol{\Omega}) \psi_{g'}(\mathbf{r}, \boldsymbol{\Omega}') \\
&+ \frac{Q_g}{4\pi}(\mathbf{r}, \boldsymbol{\Omega}) \quad ,
\end{aligned} \tag{21}$$

with the definitions for the multigroup constants

$$\hat{\Sigma}_{t,g}(\mathbf{r}, \boldsymbol{\Omega}) = \left[\frac{\int_{E_g}^{E_{g-1}} dE \Sigma_t(\mathbf{r}, E) \psi(\mathbf{r}, E, \boldsymbol{\Omega})}{\int_{E_g}^{E_{g-1}} dE \psi(\mathbf{r}, E, \boldsymbol{\Omega})} \right] \quad , \tag{22}$$

and

$$\hat{\Sigma}_{s,g' \rightarrow g}(\mathbf{r}, \boldsymbol{\Omega}' \cdot \boldsymbol{\Omega}) = \left[\frac{\int_{E_g}^{E_{g-1}} dE \int_{E_{g'}}^{E_{g'-1}} dE' \Sigma_s(\mathbf{r}, E' \rightarrow E, \boldsymbol{\Omega}' \cdot \boldsymbol{\Omega}) \psi(\mathbf{r}, E', \boldsymbol{\Omega}')}{\int_{E_{g'}}^{E_{g'-1}} dE' \psi(\mathbf{r}, E', \boldsymbol{\Omega}')} \right] \quad . \tag{23}$$

In this form the multigroup equations are an exact representation of the continuous energy transport equation, if the multigroup constants are known. However as evidenced by the above definitions, these quantities depend on the angular flux, which is the quantity we are aiming to solve for in the transport equation. The approximation that arises in the energy discretization is made at this point, where instead of ψ , a weighting function is introduced that is chosen for some application. This weighting function may take the form of a Watt fission spectrum, or a E^{-1} slowing-down spectrum, or some other application-specific choice is made. If the weighting can be assumed to be separable in energy and angle, the angular dependence can be taken out of the integral and cancels out, however this is not generally the case. The accuracy of the multigroup approximation is thus geometry and physics dependent and the complexity of the derivation of the flux guess and the number of bins hinges on the desired accuracy in the solution. For the atmospheric application described here an appropriate spectrum weighing is E^{-1} .

2.5 Spatial discretization

Spatial discretization using the diamond difference relations have been popular since the early days of discrete ordinates codes. For orthogonal grids the method is simple to code and to understand, and is second-order accurate. It suffers from shortcomings, however, in particular it can produce negative extrapolated fluxes that can lead to unphysical results [36, 37]. Although diamond performs quite well in some circumstances, concerns over the prevalence of negativity in physical quantities led to investigations of other methods, such as the step approximation, that guaranteed positivity at the expense of accuracy [38]. Following this, linear-discontinuous spatial discretization schemes were introduced [39], establishing a third-order accurate technique that was less susceptible to negative fluxes. Additionally this scheme does

not preserve the diffusion limit, which is necessary for accurate solutions in optically thick problems [40].

In this section, we examine the limitations and accuracy of the diamond spatial discretization as compared to a linear discontinuous discretization. It is particularly relevant to ATMOTRAN that we discuss spatial discretizations because this forms part of the motivation for the development of a new code. Production codes exist such as PARTISN which are fully three dimensional spherical (r, θ, ϕ) , however this code uses spatial diamond differencing, that performs poorly for optically thick and diffusive mesh cells such as we find in a practical atmospheric problem. To illustrate some properties of the two dimensional spherical transport equation, we can derive the diamond difference discretization to the discrete ordinates approximation. The transport equation has the form

$$\frac{\mu}{r^2} \frac{\partial}{\partial r} (r^2 \psi) + \frac{\eta}{r \sin(\theta)} \frac{\partial}{\partial \theta} (\sin(\theta) \psi) + \frac{1}{r} \frac{\partial}{\partial \mu} [(1 - \mu^2) \psi] - \frac{\cot(\theta)}{r} \frac{\partial}{\partial \omega} (\xi \psi) + \Sigma \psi = S \quad . \quad (24)$$

In the following derivation, the discretized angular flux $\psi_{i,j,m,\ell}$ is simplified as ψ , removing subscripts. Integrating this over the phase space volume $\int d\Omega dV$ we obtain the finite-difference representation of the transport equation. The source and removal terms are easy to evaluate since $\int d\Omega = w$, the quadrature weight, and $\int dV = V$, the cell volume. The divergence operator in two-dimensional spherical (r, θ) coordinates has four terms, derivatives in space and angle $\frac{\partial}{\partial r}$, $\frac{\partial}{\partial \theta}$, $\frac{\partial}{\partial \mu}$, $\frac{\partial}{\partial \omega}$, and each term can be expanded using the divergence theorem. Explicitly, the spherical volume element, and surface elements are

$$dV = r^2 \sin(\theta) dr d\theta d\phi \quad , \quad (25)$$

$$dS_r = r^2 \sin(\theta) d\theta d\phi \quad , \quad (26)$$

and

$$dS_\theta = r \sin(\theta) dr d\phi \quad . \quad (27)$$

Along the r-direction the finite-difference term becomes

$$\begin{aligned} \int d\Omega \oint_S \psi \Omega \cdot \hat{e}_r dS_r &= 2\pi w \mu \int \psi r^2 \sin(\theta) d\theta \\ &= 2\pi w \mu [r_{i+\frac{1}{2}}^2 \psi_{i+\frac{1}{2}} - r_{i-\frac{1}{2}}^2 \psi_{i-\frac{1}{2}}] (\cos(2\pi\theta_{j-\frac{1}{2}}) - \cos(2\pi\theta_{j+\frac{1}{2}})) \quad , \end{aligned} \quad (28)$$

and for the θ -direction

$$\begin{aligned} \int d\Omega \oint_S \psi \Omega \cdot \hat{e}_\theta dS_\theta &= 2\pi w \eta \int \psi r \sin(\theta) dr d\phi \\ &= 2\pi w \eta (\psi_{j+\frac{1}{2}} \sin(2\pi\theta_{j+\frac{1}{2}}) - \psi_{j-\frac{1}{2}} \sin(2\pi\theta_{j-\frac{1}{2}})) \left[\frac{r_{i+\frac{1}{2}}^2}{2} - \frac{r_{i-\frac{1}{2}}^2}{2} \right] \\ &= \pi w \eta (\psi_{j+\frac{1}{2}} \sin(2\pi\theta_{j+\frac{1}{2}}) - \psi_{j-\frac{1}{2}} \sin(2\pi\theta_{j-\frac{1}{2}})) \left[r_{i+\frac{1}{2}}^2 - r_{i-\frac{1}{2}}^2 \right] \quad . \end{aligned} \quad (29)$$

The angular derivatives are handled differently. An angular finite-difference method is assumed, analogous to the spatial finite-difference method, but with undetermined coefficients α and β , which are then solved for by satisfying particle balance in the divergenceless case. The coefficients can be expressed by a recursive formula using the quadrature data. For the μ derivative, the finite-difference term becomes

$$\begin{aligned}
& \int dS_r \int d\Omega \psi \Omega \cdot \hat{n}_\mu \\
&= 2\pi(\cos(2\pi\theta_{j-\frac{1}{2}}) - \cos(2\pi\theta_{j+\frac{1}{2}}))[r_{i+\frac{1}{2}}^2 - r_{i-\frac{1}{2}}^2] \int d\Omega \psi \Omega \cdot \hat{n}_\mu \\
&= 2\pi(\cos(2\pi\theta_{j-\frac{1}{2}}) - \cos(2\pi\theta_{j+\frac{1}{2}}))[r_{i+\frac{1}{2}}^2 - r_{i-\frac{1}{2}}^2](\alpha_{m+\frac{1}{2}}\psi_{m+\frac{1}{2}} - \alpha_{m-\frac{1}{2}}\psi_{m-\frac{1}{2}}) \quad .
\end{aligned} \tag{30}$$

Likewise for the ω derivative, the finite-difference term becomes

$$\begin{aligned}
& \int dS_\theta \int d\Omega \psi \Omega \cdot \hat{n}_\omega \\
&= \pi(\sin(2\pi\theta_{j+\frac{1}{2}}) - \sin(2\pi\theta_{j-\frac{1}{2}}))[r_{i+\frac{1}{2}}^2 - r_{i-\frac{1}{2}}^2] \int d\Omega \psi \Omega \cdot \hat{n}_\omega \\
&= \pi(\sin(2\pi\theta_{j+\frac{1}{2}}) - \sin(2\pi\theta_{j-\frac{1}{2}}))[r_{i+\frac{1}{2}}^2 - r_{i-\frac{1}{2}}^2](\beta_{l+\frac{1}{2}}\psi_{l+\frac{1}{2}} - \beta_{l-\frac{1}{2}}\psi_{l-\frac{1}{2}})
\end{aligned} \tag{31}$$

The other two terms in the transport equation are evaluated in the following way, for a unit cell at (i, j)

$$\int d\Omega \int_V \Sigma_t \psi dV = w \Sigma_t \psi 2\pi(\cos(2\pi\theta_{j-\frac{1}{2}}) - \cos(2\pi\theta_{j+\frac{1}{2}})) \left[\frac{r_{i+\frac{1}{2}}^3}{3} - \frac{r_{i-\frac{1}{2}}^3}{3} \right] \quad , \tag{32}$$

and

$$\int d\Omega \int_V S dV = w S 2\pi(\cos(2\pi\theta_{j-\frac{1}{2}}) - \cos(2\pi\theta_{j+\frac{1}{2}})) \left[\frac{r_{i+\frac{1}{2}}^3}{3} - \frac{r_{i-\frac{1}{2}}^3}{3} \right] \quad . \tag{33}$$

Bringing all the terms together, and dividing through by the angular quadrature

weight $w = w_{m,\ell}$ the finite-difference form of the transport equation is

$$\begin{aligned}
& 2\pi\mu[r_{i+\frac{1}{2}}^2\psi_{i+\frac{1}{2}} - r_{i-\frac{1}{2}}^2\psi_{i-\frac{1}{2}}](\cos(2\pi\theta_{j-\frac{1}{2}}) - \cos(2\pi\theta_{j+\frac{1}{2}})) \\
& + \pi\eta(\psi_{j+\frac{1}{2}}\sin(2\pi\theta_{j+\frac{1}{2}}) - \psi_{j-\frac{1}{2}}\sin(2\pi\theta_{j-\frac{1}{2}}))[r_{i+\frac{1}{2}}^2 - r_{i-\frac{1}{2}}^2] \\
& + 2\pi(\cos(2\pi\theta_{j-\frac{1}{2}}) - \cos(2\pi\theta_{j+\frac{1}{2}}))[r_{i+\frac{1}{2}}^2 - r_{i-\frac{1}{2}}^2]\frac{(\alpha_{m+\frac{1}{2}}\psi_{m+\frac{1}{2}} - \alpha_{m-\frac{1}{2}}\psi_{m-\frac{1}{2}})}{w} \\
& + \pi(\sin(2\pi\theta_{j+\frac{1}{2}}) - \sin(2\pi\theta_{j-\frac{1}{2}}))[r_{i+\frac{1}{2}}^2 - r_{i-\frac{1}{2}}^2]\frac{(\beta_{l+\frac{1}{2}}\psi_{l+\frac{1}{2}} - \beta_{l-\frac{1}{2}}\psi_{l-\frac{1}{2}})}{w} \\
& + \Sigma_t\psi 2\pi(\cos(2\pi\theta_{j-\frac{1}{2}}) - \cos(2\pi\theta_{j+\frac{1}{2}}))\left[\frac{r_{i+\frac{1}{2}}^3}{3} - \frac{r_{i-\frac{1}{2}}^3}{3}\right] \\
& = 2\pi S(\cos(2\pi\theta_{j-\frac{1}{2}}) - \cos(2\pi\theta_{j+\frac{1}{2}}))\left[\frac{r_{i+\frac{1}{2}}^3}{3} - \frac{r_{i-\frac{1}{2}}^3}{3}\right] .
\end{aligned} \tag{34}$$

There are now eight spatio-angular mesh cell edge fluxes in this equation. By sweeping the mesh, four of these will be known from prior calculation (or starting directions or other approximations such as the step approximation). The other four are cast as expressions involving the cell-averaged angular flux $\psi_{i,j}$ and the known quantities in the following way:

$$\begin{aligned}
2\psi_{i,j} &= \psi_{i+\frac{1}{2}} + \psi_{i-\frac{1}{2}} \\
2\psi_{i,j} &= \psi_{j+\frac{1}{2}} + \psi_{j-\frac{1}{2}} \\
2\psi_{i,j} &= \psi_{m+\frac{1}{2}} + \psi_{m-\frac{1}{2}} \\
2\psi_{i,j} &= \psi_{\ell+\frac{1}{2}} + \psi_{\ell-\frac{1}{2}} .
\end{aligned} \tag{35}$$

To illustrate the order of accuracy and limitations on the positivity of the diamond difference method, we can examine a simplified slab equation, with no scattering source and no internal source

$$\mu_n \frac{\partial}{\partial x} \psi_n(x) + \Sigma \psi_n(x) = 0 . \tag{36}$$

Rearranging and adding an integrating factor $e^{\frac{x\Sigma}{\mu}}$, we have

$$\frac{\partial}{\partial x}\psi_n(x) + \frac{\Sigma}{\mu_n}\psi_n(x) = \frac{\partial}{\partial x}\left[e^{\frac{x\Sigma}{\mu}}\psi_n(x)\right] = 0 \quad . \quad (37)$$

The solutions to this equation, at points x and x' are

$$\psi_n(x) = Ce^{-\frac{x\Sigma}{\mu}} \quad (38)$$

and

$$\psi_n(x') = Ce^{-\frac{x'\Sigma}{\mu}} \quad , \quad (39)$$

where C is a constant of integration which can be eliminated from the two equations, leaving

$$\psi_n(x') = \psi_n(x)e^{-\frac{(x'-x)\Sigma}{\mu}} \quad (40)$$

for $x' > x$. In the discretized mesh, if the points x' and x are taken to be the spatial mesh cell edges, then $\Delta x = x' - x$ and we have

$$\psi_{i+1/2,n} = \psi_{i-1/2,n}e^{-\frac{\Delta x\Sigma}{\mu}} \quad . \quad (41)$$

Expanding the exact solution as a Taylor series about zero, we have

$$\psi_{i+1/2,n} = e^{-2\gamma}\psi_{i-1/2} = \left[1 - 2\gamma + 2\gamma^2 - \frac{4}{3}\gamma^3 + \frac{4}{3}\gamma^4 + O(\gamma^5)\right]\psi_{i-1/2,n} \quad , \quad (42)$$

where $\gamma = \frac{\Delta x\Sigma}{2|\mu|}$. For the diamond approximation we can integrate the slab equation over the spatial variable of a single cell

$$\int_{x_{i-1/2}}^{x_{i+1/2}} dx \left[\mu_n \frac{\partial}{\partial x} \psi_n(x) + \Sigma(x)\psi_n(x) = 0 \right] \quad , \quad (43)$$

which in compact notation can be written

$$\mu_n(\psi_{i+1/2,n} - \psi_{i-1/2,n}) + \Delta x_i \Sigma_i \psi_{i,n} = 0 \quad . \quad (44)$$

Now we can connect the three ψ values by a relation in order to solve this equation.

With the diamond difference relation

$$\psi_{i,n} = \frac{1}{2}(\psi_{i+1/2,n} + \psi_{i-1/2,n}) \quad , \quad (45)$$

we can combine the above two equations to arrive at

$$\psi_{i+1/2,n} = \frac{1 - \gamma}{1 + \gamma} \psi_{i-1/2,n} \quad , \quad (46)$$

and so we find that if $\gamma > 1$ there will be negative flux values from this extrapolation.

Rearranging, the condition on γ , to avoid negative fluxes, becomes

$$\Delta x < 2\mu_{min}\lambda \quad , \quad (47)$$

where $\lambda = \Sigma^{-1}$ physically represents the mean free path. The μ_{min} here is the smallest (closest to zero) value in the quadrature root set (represented in Figure 9). This becomes a difficult condition to meet in many problems. For example, an optically thick problem where λ is small will require small mesh cell sizes, as will any problems that require high order S_N solutions.

Examining the result of the diamond difference method we can observe that it preserves the first three terms of the exact solution (to second order, $O(\gamma^2)$). The Taylor expansion of the diamond difference approximation is

$$\psi_{i+1/2,n} = \frac{1 - \gamma}{1 + \gamma} \psi_{i-1/2,n} = \left[1 - 2\gamma + 2\gamma^2 - 2\gamma^3 + O(\gamma^4) \right] \psi_{i-1/2,n} \quad (48)$$

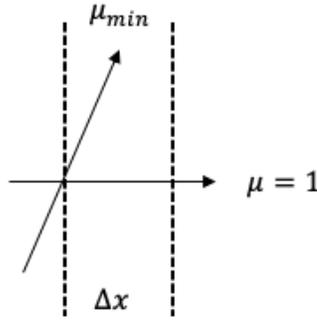


Figure 9: Diagram of one-dimensional slab geometry.

While the diamond difference approximation preserves the first two terms $O(\gamma^2)$ of the exact solution, the approximation is prone to negative fluxes for larger cells. The appearance of negative fluxes can cause unphysical oscillations in the solution, but the simplicity of the diamond differencing is appealing and many fix-up routines have been developed over the years to permit the use of diamond differencing when modified by these fix-ups. This process can degrade the order of accuracy. One of these methods is to use a first-order accurate $O(\gamma)$ approximation called the step approximation in cases that produce a negative flux. This method, referencing the geometry of a 1D slab, uses the known (incident) cell edge flux to calculate a cell-centered flux, and then sets the unknown extrapolated edge equal to the cell-centered value.

As a comparison, the step approximation only preserves the first order of the true solution $O(\gamma)$ so it is expected to be less accurate, but with the benefit of guaranteeing positive flux results. The Taylor expansion of the step approximation is

$$\psi_{i+1/2,n} = \frac{1}{1+2\gamma}\psi_{i-1/2,n} = \left[1 - 2\gamma + 4\gamma^2 - 8\gamma^3 + O(\gamma^4)\right]\psi_{i-1/2,n} \quad . \quad (49)$$

2.6 Discontinuous methods

While the diamond difference scheme requires continuity on the mesh boundaries, discontinuous spatial differencing methods relax that requirement by allowing discontinuities at the spatial mesh boundary by, in the case of linear discontinuous (LD) methods, applying a linear functional representation within the cell [41]. In LD, the flux in a cell i for angle index n is represented by

$$\psi = \frac{1}{\Delta x} \left[\psi_L(x_R - x) + \psi_R(x - x_L) \right] . \quad (50)$$

A diagram depicting this situation is shown in Figure 10.

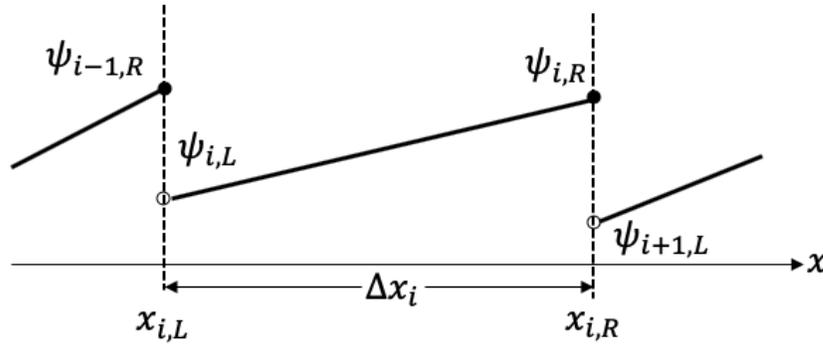


Figure 10: Flux notation in the linear discontinuous spatial differencing method.

To evaluate the flux for example in the $\mu > 0$ direction of travel, we plug the linear representation into the sourceless slab transport equation from earlier

$$\mu_n \frac{\partial}{\partial x} \psi_n(x) + \Sigma \psi_n(x) = 0 \quad , \quad (51)$$

and integrate over the mesh cell, using weighting functions 1 and $x - x_L$. This results

in two equations describing the two unknowns ψ_L and ψ_R

$$\left(\mu + \frac{\Sigma\Delta}{x}\right)\psi_{i,R} + \frac{\Sigma\Delta x}{2}\psi_{i,L} = \mu\psi_{i-1,R} \quad , \quad (52)$$

and

$$(-3\mu + \Sigma\Delta x)\psi_{i,L} + (3\mu + 2\Sigma\Delta x)\psi_{i,R} = 0 \quad . \quad (53)$$

Solving the second of these equations for $\psi_{i,L}$, inserting this into the first equation, and rearranging, we get

$$\psi_{i,R} = \left[\frac{3 - 2\gamma}{2\gamma^2 + 4\gamma + 3}\right]\psi_{i,L} \quad . \quad (54)$$

Again expanding this as a Taylor series, we have

$$\psi_{i,R} = \left[\frac{3 - 2\gamma}{2\gamma^2 + 4\gamma + 3}\right]\psi_{i,L} = \left[1 - 2\gamma + 2\gamma^2 - \frac{4}{3}\gamma^3 + \frac{4}{9}\gamma^4 + O(\gamma^5)\right]\psi_{i,L} \quad , \quad (55)$$

which demonstrates third-order ($O(\gamma^3)$) accuracy of the LD method, with respect to the exact solution of our previous slab problem, at the rightmost cell boundary when the direction of neutron travel is to the right. The cell-averaged scalar flux also exhibits third-order accuracy in LD schemes. In summary, LD improves the accuracy of the solution over the diamond methods and is less prone, though not immune, to negative fluxes at a higher computational cost, as the number of unknowns per cell is doubled. The LD method has been successfully extended to some curvilinear systems and multidimensional systems [23].

2.7 TWOTRAN SPHERE

TWOTRAN [18] was one of the first general purpose two-dimensional discrete ordinates codes. Although some two-dimensional geometries could be modeled with TWOTRAN, the two-dimensional spherical-polar system, and specifically the treatment of two angular derivatives, required complexities of implementation that were not anticipated in the TWOTRAN development. TWOTRAN SPHERE [1] was a separate code designed to solve the transport equation in this system. To our knowledge, no other code has been since developed to accomplish this. Limitations of TWOTRAN codes include the exclusive use of diamond spatial differencing, since discontinuous methods had not been developed yet, and the use of initialization angular fluxes to start the angular recursion derived from the step approximation. These limitations are eliminated with the development of the ATMOTRAN discretizations. TWOTRAN SPHERE was an important influence to ATMOTRAN since it carefully examined the requirements of directionality in sweeping the spatio-angular mesh. We provide the salient details in this section.

The authors of TWOTRAN SPHERE noticed difficulties in producing solutions that were homogeneous in the polar coordinate and traced the effect back to a requirement imposed by the characteristic equations of particle motion in this geometry. Sweeping the mesh in the direction of particle motion requires consideration of how straight-line particle motion varies in terms of the angular coordinates. Returning to the conservation form of the transport equation we may examine these characteristic equations in detail. Evaluating the derivative terms we arrive at

$$\nabla \cdot \Omega \psi = \mu \frac{\partial \psi}{\partial r} + \frac{\eta}{r} \frac{\partial \psi}{\partial \theta} + \frac{1 - \mu^2}{r} \frac{\partial \psi}{\partial \mu} - \frac{\xi \cot(\theta)}{r} \frac{\partial \psi}{\partial \omega} \quad . \quad (56)$$

Making a change of variables for the second angular derivative $\eta = (1 - \mu^2)^{1/2} \kappa$,

where $\kappa = \cos(\omega)$, we have

$$\nabla \cdot \Omega \psi = \mu \frac{\partial \psi}{\partial r} + \frac{\eta}{r} \frac{\partial \psi}{\partial \theta} + \frac{1 - \mu^2}{r} \frac{\partial \psi}{\partial \mu} + \frac{(1 - \mu^2)^{1/2} \cot(\theta)(1 - \kappa^2)}{r} \frac{\partial \psi}{\partial \kappa} . \quad (57)$$

This equation is now in the form of a total derivative with respect to the distance s in the direction Ω

$$\nabla \cdot \Omega \psi = \frac{\partial r}{\partial s} \frac{\partial \psi}{\partial r} + \frac{\partial \theta}{\partial s} \frac{\partial \psi}{\partial \theta} + \frac{\partial \mu}{\partial s} \frac{\partial \psi}{\partial \mu} + \frac{\partial \kappa}{\partial s} \frac{\partial \psi}{\partial \kappa} , \quad (58)$$

such that

$$\frac{\partial r}{\partial s} = \mu, \quad \frac{\partial \mu}{\partial s} = \frac{1 - \mu^2}{r}, \quad \frac{\partial \theta}{\partial s} = \frac{\eta}{r}, \quad \frac{\partial \kappa}{\partial s} = \frac{(1 - \mu^2)^{1/2} \cot(\theta)(1 - \kappa^2)}{r} . \quad (59)$$

Focusing on (r, μ) -space, we combine the two relations, eliminating ∂s we get

$$\frac{\partial r}{\partial \mu} = \frac{\mu r}{(1 - \mu^2)} , \quad (60)$$

which can be rearranged

$$\frac{\partial r}{r} = \frac{\mu d\mu}{(1 - \mu^2)} , \quad (61)$$

and integrated to give

$$\ln r + c' = -\frac{1}{2} \ln(1 - \mu^2) + c'' , \quad (62)$$

where c terms are constants of integration and can be combined into a single constant.

Taking the exponential of each side, we have

$$r = c_1(1 - \mu^2)^{-1/2} \quad , \quad (63)$$

and rearranging,

$$r(1 - \mu^2)^{1/2} = c_1 \quad , \quad (64)$$

as the characteristic equation of motion in this plane. Similarly, equating the (θ, κ) terms, we find

$$\frac{\partial\theta}{\partial\kappa} = \frac{\kappa}{\cot(\theta)(1 - \kappa^2)} \quad , \quad (65)$$

which can be rearranged

$$\cot(\theta)d\theta = \frac{\kappa d\kappa}{(1 - \kappa^2)} \quad , \quad (66)$$

and integrated to give

$$\ln \sin(\theta) + c' = -\frac{1}{2} \ln(1 - \kappa^2) + c'' \quad . \quad (67)$$

Taking the exponential of each side, we have

$$\sin(\theta) = c_2(1 - \mu^2)^{-1/2} \quad , \quad (68)$$

and rearranging, we have

$$\sin(\theta)(1 - \kappa^2)^{1/2} = c_2 \quad , \quad (69)$$

as the characteristic equation of motion in this plane. For the integrated equations, c_1 and c_2 are indefinite constants of integration. Figure 11 shows projections of these

characteristic equations in the relevant coordinates, where here we have switched back to η for consistency with the code derivation presented in the following chapter.

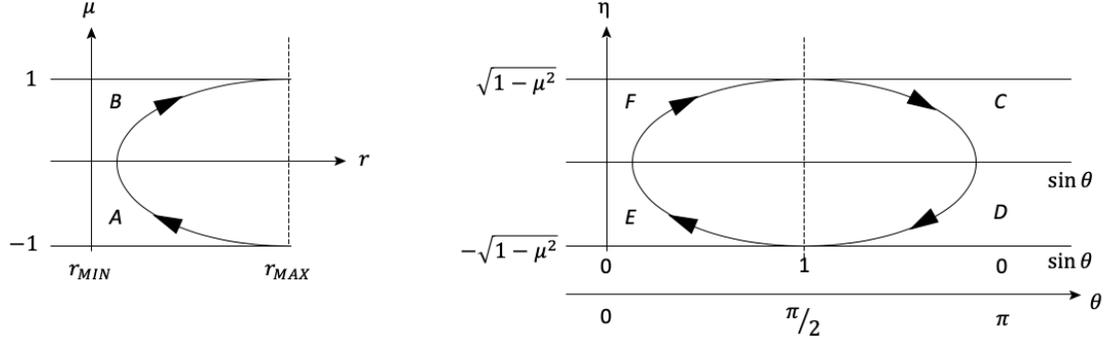


Figure 11: (Left) Sweeping (r, μ) . (Right) Sweeping (θ, η) .

For one-dimensional codes, typically the spatial sweep starts at the outer shell of the sphere r_{MAX} , and with the most negative angle (usually a starting direction of $\mu = -1$), and then sweeps the spatial mesh inward for each angle and the angular mesh with increasing μ values. This follows paths A-B in Figure 11. The progression in (η, θ) is more complicated than the progression in (μ, r) , however, and for TWOTRAN SPHERE the following angular sweep method is necessary to produce the correct results. For full spheres, they start at a spatial cell with $\theta_{j+1/2} = \pi$ for negative η directions, moving downward in η until $\theta_{j-1/2} = 0$. These are paths D-E in Figure 11. Once all negative η values are computed to $\theta_{j-1/2} = 0$, the sweep proceeds for all paths F and then C. With this procedure of following the characteristic paths, TWOTRAN SPHERE can obtain solutions homogeneous in polar angle. The TWOTRAN SPHERE angular sweep followed the following ordering in Figures 12 and 13, following the characteristic plots in Figure 11, and is shown here for an S_6 angular discretization.

These sweeps, both in space and angle, are slightly modified for ATMOTRAN given that zero-weighted starting directions supplant the step-initialization directions,

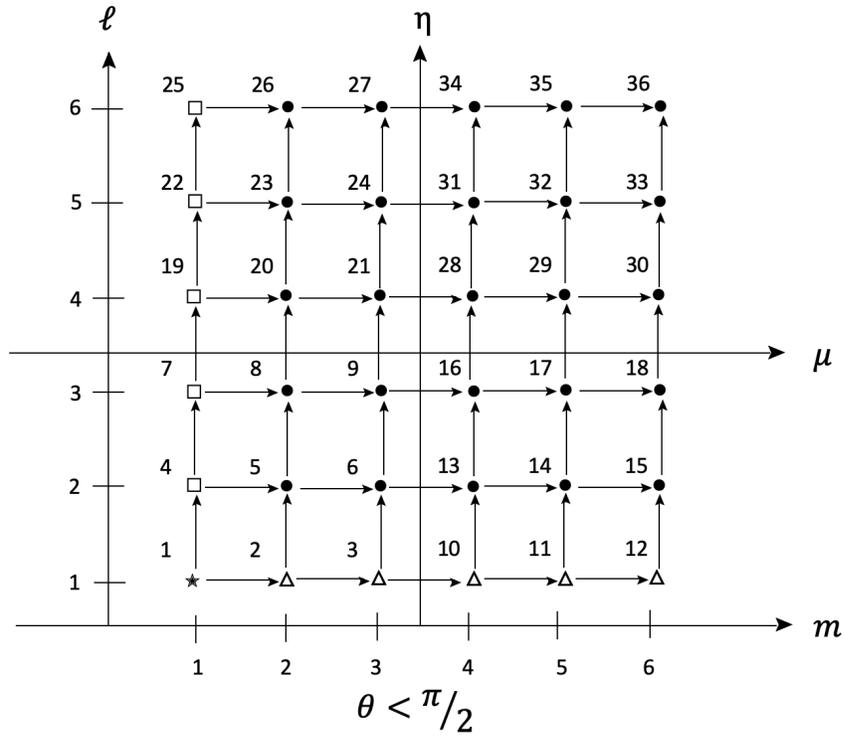


Figure 12: TWOTRAN SPHERE Sweeping the angular mesh for small polar angle

and for the bilinear method we find the spatial sweep should start around $\pi/2$ rather than at π . This requires an iteratively-lagged value to be stored with which to begin the computation. Details of these sweeps are provided in Chapter 4.

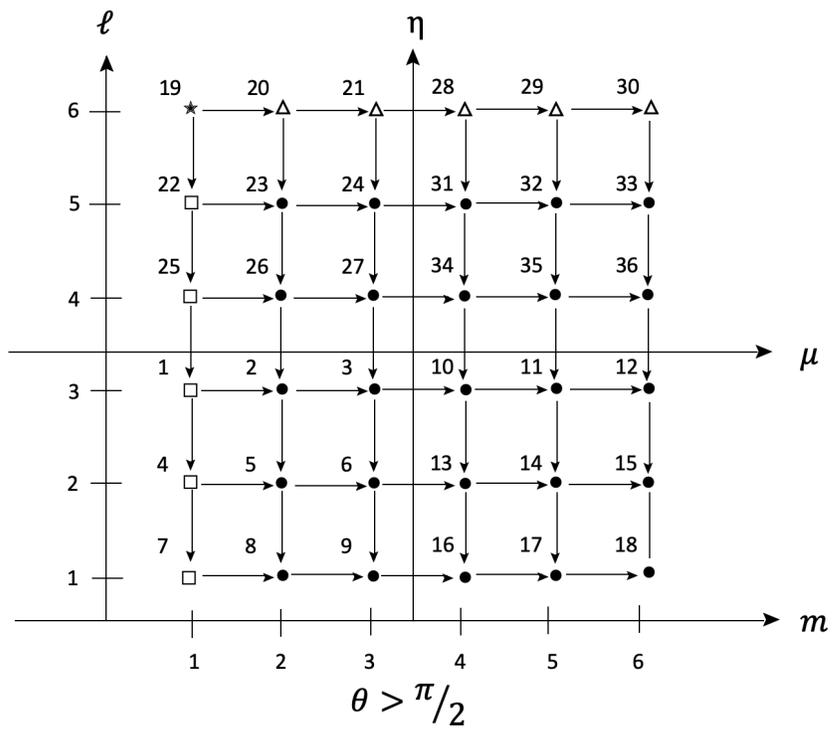


Figure 13: TWOTRAN SPHERE Sweeping the angular mesh for large polar angle

2.8 Solution methods

Neutron transport problems are solved either by deterministic or stochastic methods. Stochastic methods include the Monte Carlo method, which tracks individual simulated particles and computes probabilities of scattering and absorption events, creating a statistically average solution. Deterministic methods algebraically solve the transport equation either analytically, in the case that a solution exists, or through discretization and direct inversion or iterative methods. Analytical solutions do not exist for all but the simplest physical problems, and direct inversion is typically not attempted for many practical problems either because it is not possible to create the system or because the full system requires too much computational storage. We are then left with iterative methods, which form the most commonly encountered class of deterministic methods of solution. To motivate iterative methods of solution we begin by considering the steady-state, monoenergetic, discrete ordinates neutron transport equation in operator notation

$$\mathbf{L}\psi = \mathbf{MSD}\psi + q \quad . \quad (70)$$

In this form, \mathbf{L} represents the discretized streaming and removal operator for all angles in the approximation. The operator \mathbf{M} maps scalar flux moments onto angular flux moments, where for isotropic problems this operator is $(4\pi)^{-1}$ and anisotropy is handled by expansion in the spherical harmonics. The operator \mathbf{S} applies scattering cross sections, and the operator \mathbf{D} maps the angular flux moments onto scalar flux moments, $\phi = \mathbf{D}\psi$, which amounts to summing up the angular fluxes weighting by quadrature, and applying spherical harmonic expansions for anisotropic problems.

Combining these relations we can write

$$(\mathbf{L} - \mathbf{MSD})\psi = q \quad , \quad (71)$$

which is a problem of the general form

$$\mathbf{A}x = b \quad , \quad (72)$$

where in relation to the transport equation, the operator \mathbf{A} represents all particle loss terms and all sources of particles from scattering and other physical processes, while b is the external source. The operator \mathbf{A} can be written generally as a combination of two generalized operators, $\mathbf{A} = \mathbf{M} - \mathbf{S}$ such that

$$(\mathbf{M} - \mathbf{S})x = b \quad , \quad (73)$$

and expanding this

$$\mathbf{M}x = b + \mathbf{S}x \quad , \quad (74)$$

and adding $\mathbf{M}x$ terms summing to unity, we have

$$\begin{aligned} \mathbf{M}x &= b + \mathbf{S}x + \mathbf{M}x - \mathbf{M}x \\ &= b - (\mathbf{M} - \mathbf{S})x + \mathbf{M}x \\ &= b - \mathbf{A}x + \mathbf{M}x \end{aligned} \quad (75)$$

Inverting to solve for x we find

$$x = \mathbf{M}^{-1}(b - \mathbf{A}x) + x \quad , \quad (76)$$

and we can cast this in iterative form if we make an initial guess for x_0

$$x_{n+1} = \mathbf{M}^{-1}(b - \mathbf{A}x_n) + x_n \quad . \quad (77)$$

Defining a residual $r_n = b - \mathbf{A}x_n$, which would be zero if our solution is exactly correct $x_n = x$,

$$x_{n+1} = \mathbf{M}^{-1}r_n + x_n \quad . \quad (78)$$

We began by setting $\mathbf{A} = \mathbf{M} - \mathbf{S}$. The exact form this splitting takes depends on the iterative method to be used. There is not necessarily a best method, and the efficacy of the iterative method depends on the parameters of the problem, however in general we seek to reduce the residual to near zero as quickly as possible. This implies that we want the fewest iterations and the smallest computational time per iteration. There are many iterative methods ranging from simple intuitive methods that should have very low computational time per iteration but take more iterations to complete, and more complex methods that converge in fewer iterations but require additional calculations (in particular costly matrix inversions) at each iteration. There are two main types of iterative methods, stationary and Krylov. Stationary methods include Jacobi and Gauss-Seidel, both of which decompose \mathbf{A} into diagonal components for easier inversion. Krylov subspace methods include the conjugate gradient and generalized minimal residual methods, which seek to minimize a residual by spanning an expanding basis of products of the initial residual and successive inverse matrix powers.

The simplest method, Richardson iteration (also called source iteration) has $\mathbf{M} = \mathbf{I}$ and $\mathbf{S} = \mathbf{I} - \mathbf{A}$, where \mathbf{I} is the identity operator. With these substitutions, source iteration is represented by

$$x_{n+1} = r_n + x_n \quad . \quad (79)$$

If we define the difference between the solution at an iteration x_n and the true solution x as the error $\epsilon_n = x - x_n$, then with source iteration we have

$$\epsilon_{n+1} = (\mathbf{I} - \mathbf{A})\epsilon_n \quad , \quad (80)$$

such that for an initial guess x_0 the error at iteration $n + 1$ is

$$\epsilon_{n+1} = (\mathbf{I} - \mathbf{A})^{n+1}\epsilon_0 \quad , \quad (81)$$

and so long as the norm $\|\mathbf{I} - \mathbf{A}\| < 1$, then the error will approach zero and the method will converge, meaning that it approaches the true solution. The source iteration method will converge quickly if $\|\mathbf{I} - \mathbf{A}\| \ll 1$ and slowly if $\|\mathbf{I} - \mathbf{A}\| \rightarrow 1$. The latter case is true for physical systems that are dominated by scattering $\Sigma_s \approx \Sigma_t$ in some or all regions of the problem. A need to solve such problems has spurred development of methods to improve the source iteration convergence rate by acceleration methods, reviewed in detail elsewhere [42].

To relate the above derivation back to the physical quantities in operator notation, we remark that source iteration can be expressed in the following series of calculations. First, we start with an initial guess of the scalar flux ϕ_0 (which, for example, can be uniformly zero or one for all mesh cells), and solve the linear expression

$$\mathbf{L}\psi_1 = \mathbf{M}\mathbf{S}\phi_0 + q \quad (82)$$

for ψ_1 . We then calculate the first iteration scalar flux $\phi_1 = \mathbf{D}\psi_1$, calculate the residual $\epsilon_1 = \|\phi_1 - \phi_0\|$, check this against the convergence tolerance, and repeat until the tolerance is satisfied. The general form of the source iteration solution at

iteration n , becomes

$$\begin{aligned}\mathbf{L}\psi_{n+1} &= \mathbf{MS}\phi_n + q \\ \phi_{n+1} &= \mathbf{D}\psi_{n+1} \\ \epsilon_{n+1} &= \|\phi_{n+1} - \phi_n\| \quad .\end{aligned}\tag{83}$$

3 Bilinear Discontinuous Discretization in Two-Dimensional Spherical-Polar (r, θ) Coordinates

In this chapter we begin with the steady-state two-dimensional spherical-polar (r, θ) transport equation and proceed to derive a discontinuous-Galerkin finite element method (DFEM) for its solution. The bilinear basis approximation and terminology is discussed. Each term in the transport equation is considered in detail, proceeding through the DFEM to the formulation of the equations that arise from this method. The method of angular redistribution and the associated weighted-diamond differencing (WDD) for two angular dimensions is derived, along with the equations for the zero-weighted starting directions. This chapter is confined to the general method of solution, while specific details of the ATMOTRAN implementation of this method are left to Chapter 4.

3.1 Form of the transport equation

The transport equation can be written in two-dimensional spherical coordinates, in conservation form, as

$$\begin{aligned} \frac{\mu}{r^2} \frac{\partial}{\partial r}(r^2 \psi) + \frac{\eta}{r \sin(\theta)} \frac{\partial}{\partial \theta}(\sin(\theta) \psi) \\ + \frac{1}{r} \frac{\partial}{\partial \mu}[(1 - \mu^2) \psi] - \frac{\cot(\theta)}{r} \frac{\partial}{\partial \omega}(\xi \psi) + \Sigma \psi = S \quad . \end{aligned} \tag{84}$$

For brevity we are eliminating the energy dependence from this derivation since we are concerned with the details of the spatial and angular differencing. The energy dependence can be reintroduced following the methods in the previous chapter. Aligning the arguments of the spatial differential terms to prepare for integration and use of the divergence theorem, we can write

$$\begin{aligned} \frac{\mu}{r^2 \sin(\theta)} \frac{\partial}{\partial r} (r^2 \sin(\theta) \psi) + \frac{\eta}{r^3 \sin(\theta)} \frac{\partial}{\partial \theta} (r^2 \sin(\theta) \psi) \\ + \frac{1}{r} \frac{\partial}{\partial \mu} [(1 - \mu^2) \psi] - \frac{\cot(\theta)}{r} \frac{\partial}{\partial \omega} (\xi \psi) + \Sigma \psi = S \quad . \end{aligned} \quad (85)$$

The components of the directional unit vector are μ , η , and ξ , and are related in the conventional definition by

$$\begin{aligned} \eta &= \sqrt{1 - \mu^2} \cos(\omega) \\ \xi &= \sqrt{1 - \mu^2} \sin(\omega) \\ 1 &= \mu^2 + \eta^2 + \xi^2 \quad . \end{aligned} \quad (86)$$

To get the angular derivatives into a finite-difference form we follow the method of Carlson, casting these terms as functions of quadrature weights and directions with undetermined coefficients and then finding the coefficients by satisfying particle conservation. For the two angular derivatives these approximations are

$$\begin{aligned} \frac{\partial}{\partial \mu} [(1 - \mu^2) \psi] &\approx \frac{\alpha_{m+1/2}}{w_m} \psi_{m+1/2, \ell} - \frac{\alpha_{m-1/2}}{w_m} \psi_{m-1/2, \ell} \\ \frac{\partial}{\partial \omega} (\xi \psi) &\approx \frac{\beta_{m, \ell+1/2}}{w_{m, \ell}} \psi_{m, \ell+1/2} - \frac{\beta_{m, \ell-1/2}}{w_{m, \ell}} \psi_{m, \ell-1/2} \quad . \end{aligned} \quad (87)$$

Also, the angular cell flux terms are put into a finite-difference scheme. This creates an angular mesh grid on which to solve. For example, the flux at spatial cell (i, j) also

sits at angular cell indices (m, ℓ) as shown in Figure 7. We recognize that angular mesh cell edges exist akin to spatial mesh edges. For example an upper edge of angular cell m has edge flux $\psi_{m+1/2}$. We cast the equation into the following form

$$\begin{aligned}
& \frac{\mu}{r^2 \sin(\theta)} \frac{\partial}{\partial r} (r^2 \sin(\theta) \psi) + \frac{\eta}{r^3 \sin(\theta)} \frac{\partial}{\partial \theta} (r^2 \sin(\theta) \psi) \\
& + \frac{1}{r} \left[\frac{\alpha_{m+1/2}}{w_m} \psi_{m+1/2, \ell} - \frac{\alpha_{m-1/2}}{w_m} \psi_{m-1/2, \ell} \right] \\
& - \frac{\cot(\theta)}{r} \left[\frac{\beta_{m, \ell+1/2}}{w_{m\ell}} \psi_{m, \ell+1/2} - \frac{\beta_{m, \ell-1/2}}{w_{m\ell}} \psi_{m, \ell-1/2} \right] \\
& + \Sigma \psi = S \quad .
\end{aligned} \tag{88}$$

This form of the equation still represents particle balance. The third and fourth terms are the angular redistribution terms, which represent gain and loss of particles into angular mesh cells during straight line trajectories through the curvilinear spatial mesh. It is important that the angular redistribution terms conserve neutrons since they account for a purely geometrical effect. Satisfying particle conservation amounts to ensuring that these terms with undetermined coefficients neither create nor destroy particles. As is done typically, we apply a test case of divergenceless particle flow satisfying $\nabla \cdot (\mathbf{\Omega} \psi) = 0$ (here we chose $\psi = S/\Sigma$) and equate the remaining terms (μ with α , η with β) to find two recurrence relations

$$\begin{aligned}
\alpha_{m+1/2} &= \alpha_{m-1/2} - 2\mu_m w_m \\
\beta_{m, \ell+1/2} &= \beta_{m, \ell-1/2} + \eta_{m, \ell} w_{m\ell} \quad .
\end{aligned} \tag{89}$$

The boundary conditions are specific to the problem. Radially there are two boundaries in our intended application. The lower radial bound is along the surface of the Earth. For neutrons this could be treated as a reflective surface or an absorbing surface. The upper radial bound is actually in a vacuum, and a vacuum boundary condition is appropriate. No exiting neutrons can scatter back in, and there is

no source of neutrons from outside the geometry. The boundary conditions can be described in the following way.

$$\begin{aligned}\psi^b(R, \theta, \mu, \eta, E) &= 0, & \mu < 0 \\ \psi^b(0, \theta, \mu, \eta, E) &= 0, & \mu > 0 \quad .\end{aligned}\tag{90}$$

In the theta direction, we model only a hemisphere, from 0 to π radians. The boundaries are reflective, and are calculated using an initial guess, when starting on the reflective boundary, and updating this guess with every iteration. The reflective boundary flux lags the solution by one iteration.

$$\begin{aligned}\psi^b(r, \theta = 0, \mu, \eta, E) &= \psi(r, \theta = 0, \mu, -\eta, E), & \eta < 0 \\ \psi^b(r, \theta = \pi, \mu, \eta, E) &= \psi(r, \theta = 0, \mu, -\eta, E), & \eta > 0 \quad .\end{aligned}\tag{91}$$

3.2 Basis function expansion

With the transport equation in this form we can now work toward a DFEM spatial representation. This includes defining vector basis functions to relate quantities within a spatial cell, allowing expansion of quantities in terms of these basis functions, and integrating the terms in the transport equation over phase space volume for individual cells. First we define these basis functions, and then we proceed through the six terms in the above transport equation to obtain the DFEM representation. Weighted residual methods such as this are often employed to solve differential equations for which no practical analytical solution exists, to include the transport equation in many real world cases. A brief background of the method follows, limited to what is essential to understand the derivation presented in this chapter. To illustrate the method of weighted residuals, a linear differential equation may be written

$$L(u(x)) = f(x) \quad , \quad (92)$$

where L is a linear differential operator. An approximate solution $\tilde{u}(x)$ satisfies

$$\tilde{u}(x) = \sum_{i=1}^N u_i \phi_i(x) \quad . \quad (93)$$

The approximate solution generates a residual

$$R(x) = L\tilde{u}(x) - f(x) = L\left(\sum_{i=1}^N u_i \phi_i(x)\right) - f(x) \neq 0 \quad . \quad (94)$$

Weighted residual methods are based on minimizing the value of this residual by utilizing a set of N weighting functions and integrating the product over the problem domain. This amounts to solving $i = 1, N$ equations of the form

$$\int_a^b dx w_i(x) R(x) = 0 \quad . \quad (95)$$

In the Galerkin method (this method is also called Bubnov-Galerkin to distinguish it from other variants), the weighting function equals the guess solution ($w_i = \phi_i$). This being the case, if the true solution can be described by a linear combination of the basis functions then the approximate solution will be exact. Both continuous (cG) and discontinuous Galerkin (dG) methods may be used in the solution of the transport equation. Both methods approximate the solution in a finite physical cell by a linear combination of basis functions, which are piecewise polynomials. cG requires the polynomials to be continuous at the mesh cell boundaries, while dG allows for discontinuities at these boundaries. dG methods are generally better at resolving flux gradients that are steep, and so are considered more robust to a general set of

physical problems, however dG methods have a greater number of unknowns requiring more computation. cG methods have been applied to two-dimensional diffusion problems [43], and successfully to transport problems, sometimes employing Petrov-Galerkin methods [44]. cG is the diamond difference spatial discretization employed in TWOTRAN-SPHERE, for example. dG was implemented for neutron transport and presented with some advantages as an alternative to cG [39] and has since been applied by many other researchers.

The starting point of the dG method is the choice of polynomial. In this derivation, and for the results in this dissertation, we employ a linear representation in two dimensions. Higher order polynomials are of course possible (e.g. quadratic) and can allow for more accurate solutions at the cost of greater complexity. The present bilinear discontinuous (BLD) method uses two-dimensional cardinal basis functions that can be defined in the following way for spherical coordinates. The spatial cell computational grid is defined in Figure 14. The four corners of the spatial mesh cell are denoted by indices 1 to 4. Surfaces in the polar angle direction are named T(op) and B(ottom) with the unit vector in the theta direction pointing from B to T, In the radial direction the surfaces are named L(eft) and R(ight) with the unit vector pointing from L to R.

The bilinear basis functions are indexed at each of the four corners:

$$\begin{aligned}
 B_1(r, \theta) &= \left(\frac{r_R - r}{\Delta r} \right) \left(\frac{\theta_T - \theta}{\Delta \theta} \right) \\
 B_2(r, \theta) &= \left(\frac{r - r_L}{\Delta r} \right) \left(\frac{\theta_T - \theta}{\Delta \theta} \right) \\
 B_3(r, \theta) &= \left(\frac{r - r_L}{\Delta r} \right) \left(\frac{\theta - \theta_B}{\Delta \theta} \right) \\
 B_4(r, \theta) &= \left(\frac{r_R - r}{\Delta r} \right) \left(\frac{\theta - \theta_B}{\Delta \theta} \right) .
 \end{aligned} \tag{96}$$

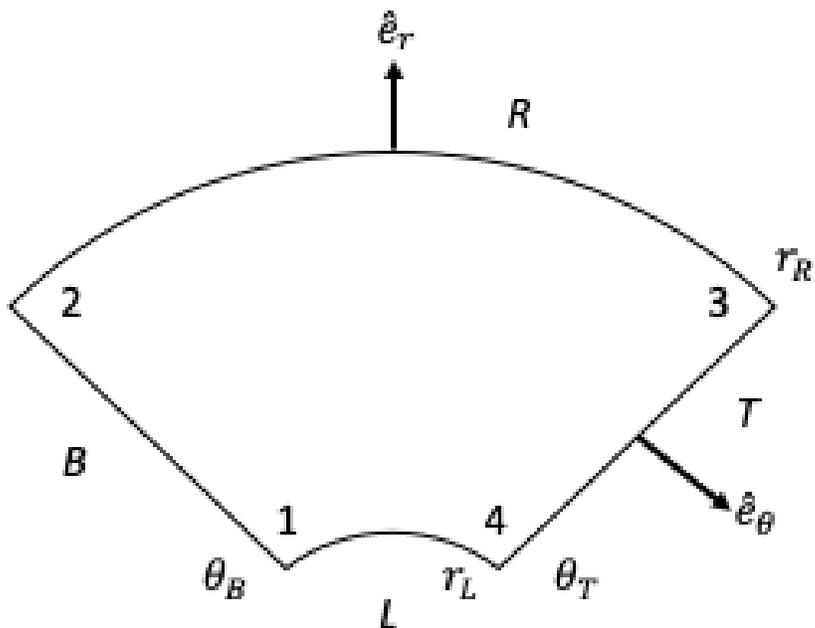


Figure 14: Schematic of a computational cell.

The angular flux can be expanded in terms of these basis equations. For example the weighted residual approximation to the angular flux in a cell (i, j) with particle direction (m, ℓ) is

$$\tilde{\psi}_{ijm\ell}(r, \theta) = \sum_{k=1}^4 \psi_{kijm\ell} B_{kij}(r, \theta) \quad . \quad (97)$$

3.3 The streaming term

In two-dimensional spherical polar (r, θ) coordinates, the gradient is

$$\nabla = \hat{r} \frac{\partial}{\partial r} + \frac{\hat{\theta}}{r} \frac{\partial}{\partial \theta} \quad . \quad (98)$$

The first two terms in the conservation form of the transport equation (Equation 85) can be cast in a form reflecting the above gradient expression. Using $\mu = \mathbf{\Omega} \cdot \hat{e}_r$ and

$\eta = \mathbf{\Omega} \cdot \hat{e}_\theta$, we have

$$\begin{aligned} \frac{\mu}{r^2 \sin(\theta)} \frac{\partial}{\partial r} (r^2 \sin(\theta) \psi) + \frac{\eta}{r^3 \sin(\theta)} \frac{\partial}{\partial \theta} (r^2 \sin(\theta) \psi) + \dots \\ = \frac{1}{r^2 \sin(\theta)} \left[\mathbf{\Omega} \cdot \nabla (r^2 \sin \theta \psi) \right] + \dots \end{aligned} \quad (99)$$

This rearrangement allows for expanding this expression with the divergence theorem, which allows us to separate the contributions from what we will call streaming and gradient effects. Multiplying the transport equation by the basis and integrating over the cell volume, this term can be written

$$\int \frac{1}{r^2 \sin(\theta)} \mathbf{\Omega} \cdot \nabla (r^2 \sin \theta \psi) B dV = \mathbf{\Omega} \cdot \int_{\partial E} \hat{n} \tilde{\psi} B dA - \mathbf{\Omega} \cdot \int_E \tilde{\psi} \nabla B dV \quad . \quad (100)$$

The second term on the right-hand side of this equation represents the volumetric term of the spatial gradient, and we refer to this the gradient term. The first term is the surface integral term of the spatial gradient and we refer to this as the streaming term. We will deal with expanding this streaming term presently. For reference we also have the physical volume and surface area elements

$$\begin{aligned} dV &= r^2 \sin(\theta) dr d\theta \\ dA_\mu &= r^2 \sin(\theta) d\theta \\ dA_\eta &= r \sin(\theta) dr \quad , \end{aligned} \quad (101)$$

which follow from the geometrical definitions in Figure 15. We find that a physical cell in the spherical-polar geometry, when normalized by 2π for the azimuthal dimension, has a surface area in the outward μ direction of $r^2 \sin(\theta) d\theta$ and in the η direction of $r \sin(\theta) dr$.

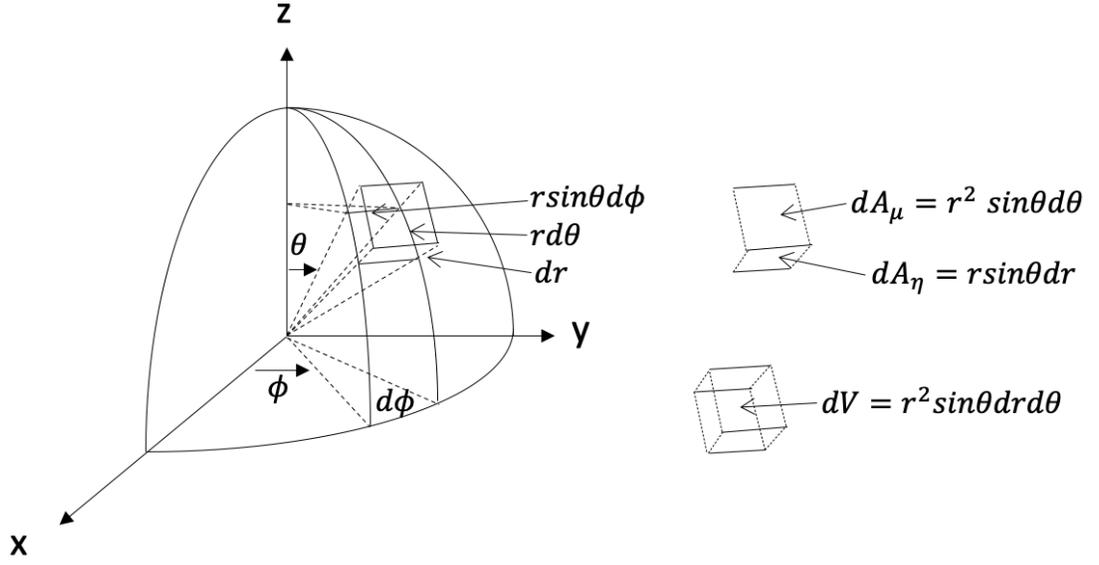


Figure 15: Dimensions of the physical unit cell in spherical polar coordinates.

Each integration over the element E (or surface ∂E) amounts to an integral in the radial direction added to an integral in the polar direction. Considering $B_1(r, \theta)$, for example, the streaming term becomes

$$\begin{aligned}
& \Omega \cdot \int_{\partial E} \hat{n} \tilde{\psi} B_1 dA \\
&= \Omega \cdot \int_{r_L}^{r_R} \hat{n}_B \tilde{\psi}_B(\theta_B) B_1(\theta_B) r \sin(\theta_B) dr \\
&+ \Omega \cdot \int_{\theta_B}^{\theta_T} \hat{n}_L \tilde{\psi}_L(r_L) B_1(r_L) r_L^2 \sin(\theta) d\theta \quad .
\end{aligned} \tag{102}$$

$B_1(r, \theta)$ is unity at point 1 on the mesh cell and is zero at the other three points. On the bottom and left surfaces, $B_1(r, \theta)$ will have linear dependence. The angular fluxes along these surfaces can be written as

$$\tilde{\psi}_B = \tilde{\psi}_{1B} B_1(\theta_B) + \tilde{\psi}_{2B} B_2(\theta_B) \tag{103}$$

and

$$\tilde{\psi}_L = \tilde{\psi}_{1L}B_1(r_L) + \tilde{\psi}_{4L}B_4(r_L) \quad , \quad (104)$$

where

$$B_1(r, \theta_B) = \left(\frac{r_R - r}{\Delta r} \right) \quad , \quad (105)$$

$$B_2(r, \theta_B) = \left(\frac{r - r_L}{\Delta r} \right) \quad , \quad (106)$$

and

$$B_1(r_L, \theta) = \left(\frac{\theta_T - \theta}{\Delta \theta} \right) \quad , \quad (107)$$

$$B_4(r_L, \theta) = \left(\frac{\theta - \theta_B}{\Delta \theta} \right) \quad . \quad (108)$$

As defined in the computational cell, the following relations hold

$$\begin{aligned} \boldsymbol{\Omega} \cdot \hat{n}_B &= \boldsymbol{\Omega} \cdot -\hat{e}_\theta = -\eta \\ \boldsymbol{\Omega} \cdot \hat{n}_T &= \boldsymbol{\Omega} \cdot \hat{e}_\theta = \eta \\ \boldsymbol{\Omega} \cdot \hat{n}_L &= \boldsymbol{\Omega} \cdot -\hat{e}_r = -\mu \\ \boldsymbol{\Omega} \cdot \hat{n}_R &= \boldsymbol{\Omega} \cdot \hat{e}_r = \mu \quad . \end{aligned} \quad (109)$$

The resulting integration for the $B_1(r, \theta)$ term can now be evaluated. Note that in the final step the integrals have been assigned the designation M_x . The solutions of the individual integrals are tabulated in Appendix A. For $B_1(r, \theta)$ we have

$$\begin{aligned}
& \Omega \cdot \int_{r_L}^{r_R} \hat{n}_B r \sin(\theta_B) \tilde{\psi}_B(\theta_B) B_1(\theta_B) dr + \Omega \cdot \int_{\theta_B}^{\theta_T} \hat{n}_L r_L^2 \sin(\theta) \tilde{\psi}_L(r_L) B_1(r_L) d\theta \\
&= -\eta \sin(\theta_B) \frac{1}{(\Delta r)^2} \left[\left(\int_{r_L}^{r_R} dr r (r_R - r) (r_R - r) \tilde{\psi}_{1B} \right) \right. \\
&+ \left. \left(\int_{r_L}^{r_R} dr r (r - r_L) (r_R - r) \tilde{\psi}_{2B} \right) \right] \\
&- \mu r_L^2 \frac{1}{(\Delta \theta)^2} \left[\left(\int_{\theta_B}^{\theta_T} d\theta \sin(\theta) (\theta_T - \theta) (\theta_T - \theta) \tilde{\psi}_{1L} \right) \right. \\
&+ \left. \left(\int_{\theta_B}^{\theta_T} d\theta \sin(\theta) (\theta - \theta_B) (\theta_T - \theta) \tilde{\psi}_{4L} \right) \right] \\
&= -\frac{\eta \sin(\theta_B)}{(\Delta r)^2} (M_9 \tilde{\psi}_{1B} + M_{10} \tilde{\psi}_{2B}) - \frac{\mu r_L^2}{(\Delta \theta)^2} (M_4 \tilde{\psi}_{1L} + M_5 \tilde{\psi}_{4L}) \quad ,
\end{aligned} \tag{110}$$

for $B_2(r, \theta)$

$$\begin{aligned}
& \Omega \cdot \int_{r_L}^{r_R} \hat{n}_B r \sin(\theta_B) \tilde{\psi}_B(\theta_B) B_2(\theta_B) dr + \Omega \cdot \int_{\theta_B}^{\theta_T} \hat{n}_R r_R^2 \sin(\theta) \tilde{\psi}_R(r_R) B_2(r_R) d\theta \\
&= -\eta \sin(\theta_B) \frac{1}{(\Delta r)^2} \left[\left(\int_{r_L}^{r_R} dr r (r_R - r) (r - r_L) \tilde{\psi}_{1B} \right) \right. \\
&+ \left. \left(\int_{r_L}^{r_R} dr r (r - r_L) (r - r_L) \tilde{\psi}_{2B} \right) \right] \\
&+ \mu r_R^2 \frac{1}{(\Delta \theta)^2} \left[\left(\int_{\theta_B}^{\theta_T} d\theta \sin(\theta) (\theta_T - \theta) (\theta_T - \theta) \tilde{\psi}_{2R} \right) \right. \\
&+ \left. \left(\int_{\theta_B}^{\theta_T} d\theta \sin(\theta) (\theta - \theta_B) (\theta_T - \theta) \tilde{\psi}_{3R} \right) \right] \\
&= -\frac{\eta \sin(\theta_B)}{(\Delta r)^2} (M_{10} \tilde{\psi}_{1B} + M_{11} \tilde{\psi}_{2B}) + \frac{\mu r_R^2}{(\Delta \theta)^2} (M_4 \tilde{\psi}_{2R} + M_5 \tilde{\psi}_{3R}) \quad ,
\end{aligned} \tag{111}$$

for $B_3(r, \theta)$

$$\begin{aligned}
& \Omega \cdot \int_{r_L}^{r_R} \hat{n}_T r \sin(\theta_T) \tilde{\psi}_T(\theta_T) B_3(\theta_T) dr + \Omega \cdot \int_{\theta_B}^{\theta_T} \hat{n}_R r_R^2 \sin(\theta) \tilde{\psi}_R(r_R) B_3(r_R) d\theta \\
&= \eta \sin(\theta_T) \frac{1}{(\Delta r)^2} \left[\left(\int_{r_L}^{r_R} dr r (r - r_L) (r - r_L) \tilde{\psi}_{3T} \right) \right. \\
&+ \left. \left(\int_{r_L}^{r_R} dr r (r - r_L) (r_R - r) \tilde{\psi}_{4T} \right) \right] \\
&+ \mu r_R^2 \frac{1}{(\Delta \theta)^2} \left[\left(\int_{\theta_B}^{\theta_T} d\theta \sin(\theta) (\theta_T - \theta) (\theta - \theta_B) \tilde{\psi}_{2R} \right) \right. \\
&+ \left. \left(\int_{\theta_B}^{\theta_T} d\theta \sin(\theta) (\theta - \theta_B) (\theta - \theta_B) \tilde{\psi}_{3R} \right) \right] \\
&= \frac{\eta \sin(\theta_T)}{(\Delta r)^2} (M_{11} \tilde{\psi}_{3T} + M_{10} \tilde{\psi}_{4T}) + \frac{\mu r_R^2}{(\Delta \theta)^2} (M_5 \tilde{\psi}_{2R} + M_6 \tilde{\psi}_{3R}) \quad ,
\end{aligned} \tag{112}$$

and for $B_4(r, \theta)$

$$\begin{aligned}
& \Omega \cdot \int_{r_L}^{r_R} \hat{n}_T r \sin(\theta_T) \tilde{\psi}_T(\theta_T) B_4(\theta_T) dr + \Omega \cdot \int_{\theta_B}^{\theta_T} \hat{n}_L r_L^2 \sin(\theta) \tilde{\psi}_L(r_L) B_4(r_L) d\theta \\
&= \eta \sin(\theta_T) \frac{1}{(\Delta r)^2} \left[\left(\int_{r_L}^{r_R} dr r (r_R - r) (r - r_L) \tilde{\psi}_{3T} \right) \right. \\
&+ \left. \left(\int_{r_L}^{r_R} dr r (r_R - r) (r_R - r) \tilde{\psi}_{4T} \right) \right] \\
&- \mu r_L^2 \frac{1}{(\Delta \theta)^2} \left[\left(\int_{\theta_B}^{\theta_T} d\theta \sin(\theta) (\theta_T - \theta) (\theta - \theta_B) \tilde{\psi}_{1L} \right) \right. \\
&+ \left. \left(\int_{\theta_B}^{\theta_T} d\theta \sin(\theta) (\theta - \theta_B) (\theta - \theta_B) \tilde{\psi}_{4L} \right) \right] \\
&= \frac{\eta \sin(\theta_T)}{(\Delta r)^2} (M_{10} \tilde{\psi}_{3T} + M_9 \tilde{\psi}_{4T}) - \frac{\mu r_L^2}{(\Delta \theta)^2} (M_5 \tilde{\psi}_{1L} + M_6 \tilde{\psi}_{4L}) \quad .
\end{aligned} \tag{113}$$

Combining the terms we arrive at

$$\begin{aligned}
& \frac{\eta}{(\Delta r)^2} \begin{bmatrix} -\sin(\theta_B)M_9 & -\sin(\theta_B)M_{10} & 0 & 0 \\ -\sin(\theta_B)M_{10} & -\sin(\theta_B)M_{11} & 0 & 0 \\ 0 & 0 & \sin(\theta_T)M_{11} & \sin(\theta_T)M_{10} \\ 0 & 0 & \sin(\theta_T)M_{10} & \sin(\theta_T)M_9 \end{bmatrix} \begin{bmatrix} \tilde{\psi}_{1B} \\ \tilde{\psi}_{2B} \\ \tilde{\psi}_{3T} \\ \tilde{\psi}_{4T} \end{bmatrix} \\
& + \frac{\mu}{(\Delta\theta)^2} \begin{bmatrix} -r_L^2 M_4 & 0 & 0 & -r_L^2 M_5 \\ 0 & r_R^2 M_4 & r_R^2 M_5 & 0 \\ 0 & r_R^2 M_5 & r_R^2 M_6 & 0 \\ -r_L^2 M_5 & 0 & 0 & -r_L^2 M_6 \end{bmatrix} \begin{bmatrix} \tilde{\psi}_{1L} \\ \tilde{\psi}_{2R} \\ \tilde{\psi}_{3R} \\ \tilde{\psi}_{4L} \end{bmatrix}, \tag{114}
\end{aligned}$$

which can be written in matrix form as

$$\eta \mathbf{V}_\eta \begin{bmatrix} \tilde{\psi}_{1B} \\ \tilde{\psi}_{2B} \\ \tilde{\psi}_{3T} \\ \tilde{\psi}_{4T} \end{bmatrix} + \mu \mathbf{V}_\mu \begin{bmatrix} \tilde{\psi}_{1L} \\ \tilde{\psi}_{2R} \\ \tilde{\psi}_{3R} \\ \tilde{\psi}_{4L} \end{bmatrix}, \tag{115}$$

where

$$\mathbf{V}_\eta = \frac{1}{(\Delta r)^2} \begin{bmatrix} -\sin(\theta_B)M_9 & -\sin(\theta_B)M_{10} & 0 & 0 \\ -\sin(\theta_B)M_{10} & -\sin(\theta_B)M_{11} & 0 & 0 \\ 0 & 0 & \sin(\theta_T)M_{11} & \sin(\theta_T)M_{10} \\ 0 & 0 & \sin(\theta_T)M_{10} & \sin(\theta_T)M_9 \end{bmatrix}, \tag{116}$$

and

$$\mathbf{V}_\mu = \frac{1}{(\Delta\theta)^2} \begin{bmatrix} -r_L^2 M_4 & 0 & 0 & -r_L^2 M_5 \\ 0 & r_R^2 M_4 & r_R^2 M_5 & 0 \\ 0 & r_R^2 M_5 & r_R^2 M_6 & 0 \\ -r_L^2 M_5 & 0 & 0 & -r_L^2 M_6 \end{bmatrix}. \quad (117)$$

Depending on the sweep directions, two of the angular fluxes denoted by $\tilde{\psi}$, also known as numerical fluxes, are assigned upwind values from neighboring spatial mesh cells. These upwind values, when multiplied by corresponding terms in the matrices $\mu\mathbf{V}_\mu$ and $\eta\mathbf{V}_\eta$, will be negative. These negative terms are moved to the right hand side of the transport equation, essentially becoming boundary sources on the entering surface of the spatial mesh cell. The sweep-direction-dependent flux values at each corner are shown in Table 1 for combinations of angular polarity. For boundary cells, the upwind fluxes become boundary fluxes.

Table 1: Sweep-direction-dependent flux values in the streaming term

	$\eta > 0$	$\eta < 0$	$\mu > 0$	$\mu < 0$
$\tilde{\psi}_{1B}$	$\psi_{4,i,j-1}$	$\psi_{1,i,j}$	-	-
$\tilde{\psi}_{2B}$	$\psi_{3,i,j-1}$	$\psi_{2,i,j}$	-	-
$\tilde{\psi}_{3T}$	$\psi_{3,i,j}$	$\psi_{2,i,j+1}$	-	-
$\tilde{\psi}_{4T}$	$\psi_{4,i,j}$	$\psi_{1,i,j+1}$	-	-
$\tilde{\psi}_{1L}$	-	-	$\psi_{2,i-1,j}$	$\psi_{1,i,j}$
$\tilde{\psi}_{4L}$	-	-	$\psi_{3,i-1,j}$	$\psi_{4,i,j}$
$\tilde{\psi}_{2R}$	-	-	$\psi_{2,i,j}$	$\psi_{1,i+1,j}$
$\tilde{\psi}_{3R}$	-	-	$\psi_{3,i,j}$	$\psi_{4,i+1,j}$

3.4 The gradient term

The gradient term is as follows, retaining the negative sign from the earlier equation so that the result can be simply added into the resultant transport equation

$$-\mathbf{\Omega} \cdot \int_E \tilde{\psi} \nabla B dV \quad , \quad (118)$$

which expands to

$$-\mathbf{\Omega} \cdot \int_E \tilde{\psi} \left[\hat{r} \frac{\partial B}{\partial r} + \frac{\hat{\theta}}{r} \frac{\partial B}{\partial \theta} \right] r^2 \sin(\theta) dr d\theta \quad . \quad (119)$$

Recalling the approximation to the angular flux, $\tilde{\psi} = \sum_{i=1}^4 \psi_i B_i$, and taking the case of $B_1(r, \theta)$ through $B_4(r, \theta)$ as the weighting functions, four equations result

$$-\mathbf{\Omega} \cdot \int_{r_L}^{r_R} dr \int_{\theta_B}^{\theta_T} d\theta \sin(\theta) \sum_{i=1}^4 \psi_i B_i \left[-\frac{r^2 \hat{r}}{\Delta r \Delta \theta} (\theta_T - \theta) - \frac{r \hat{\theta}}{\Delta r \Delta \theta} (r_R - r) \right] \quad (120)$$

$$-\mathbf{\Omega} \cdot \int_{r_L}^{r_R} dr \int_{\theta_B}^{\theta_T} d\theta \sin(\theta) \sum_{i=1}^4 \psi_i B_i \left[\frac{r^2 \hat{r}}{\Delta r \Delta \theta} (\theta_T - \theta) - \frac{r \hat{\theta}}{\Delta r \Delta \theta} (r_R - r) \right] \quad (121)$$

$$-\mathbf{\Omega} \cdot \int_{r_L}^{r_R} dr \int_{\theta_B}^{\theta_T} d\theta \sin(\theta) \sum_{i=1}^4 \psi_i B_i \left[\frac{r^2 \hat{r}}{\Delta r \Delta \theta} (\theta_T - \theta) + \frac{r \hat{\theta}}{\Delta r \Delta \theta} (r_R - r) \right] \quad (122)$$

$$-\mathbf{\Omega} \cdot \int_{r_L}^{r_R} dr \int_{\theta_B}^{\theta_T} d\theta \sin(\theta) \sum_{i=1}^4 \psi_i B_i \left[-\frac{r^2 \hat{r}}{\Delta r \Delta \theta} (\theta_T - \theta) + \frac{r \hat{\theta}}{\Delta r \Delta \theta} (r_R - r) \right] \quad . \quad (123)$$

Evaluating the vector dot products and inserting the angular flux approximation and the basis, the B_1 term becomes

$$\begin{aligned}
&= \mu \frac{1}{\Delta r \Delta \theta} \left[\left(\int_{r_L}^{r_R} dr r^2 \left(\frac{r_R - r}{\Delta r} \right) \int_{\theta_B}^{\theta_T} d\theta \sin(\theta) \left(\frac{\theta_T - \theta}{\Delta \theta} \right) (\theta_T - \theta) \right) \psi_1 \right. \\
&\quad + \left(\int_{r_L}^{r_R} dr r^2 \left(\frac{r - r_L}{\Delta r} \right) \int_{\theta_B}^{\theta_T} d\theta \sin(\theta) \left(\frac{\theta_T - \theta}{\Delta \theta} \right) (\theta_T - \theta) \right) \psi_2 \\
&\quad + \left(\int_{r_L}^{r_R} dr r^2 \left(\frac{r - r_L}{\Delta r} \right) \int_{\theta_B}^{\theta_T} d\theta \sin(\theta) \left(\frac{\theta - \theta_B}{\Delta \theta} \right) (\theta_T - \theta) \right) \psi_2 \\
&\quad \left. + \left(\int_{r_L}^{r_R} dr r^2 \left(\frac{r_R - r}{\Delta r} \right) \int_{\theta_B}^{\theta_T} d\theta \sin(\theta) \left(\frac{\theta - \theta_B}{\Delta \theta} \right) (\theta_T - \theta) \right) \psi_4 \right] \\
&+ \eta \frac{1}{\Delta r \Delta \theta} \left[\left(\int_{r_L}^{r_R} dr r (r_R - r) \left(\frac{r_R - r}{\Delta r} \right) \int_{\theta_B}^{\theta_T} d\theta \sin(\theta) \left(\frac{\theta_T - \theta}{\Delta \theta} \right) \right) \psi_1 \right. \\
&\quad + \left(\int_{r_L}^{r_R} dr r (r_R - r) \left(\frac{r - r_L}{\Delta r} \right) \int_{\theta_B}^{\theta_T} d\theta \sin(\theta) \left(\frac{\theta_T - \theta}{\Delta \theta} \right) \right) \psi_2 \\
&\quad + \left(\int_{r_L}^{r_R} dr r (r_R - r) \left(\frac{r - r_L}{\Delta r} \right) \int_{\theta_B}^{\theta_T} d\theta \sin(\theta) \left(\frac{\theta - \theta_B}{\Delta \theta} \right) \right) \psi_3 \\
&\quad \left. + \left(\int_{r_L}^{r_R} dr r (r_R - r) \left(\frac{r_R - r}{\Delta r} \right) \int_{\theta_B}^{\theta_T} d\theta \sin(\theta) \left(\frac{\theta - \theta_B}{\Delta \theta} \right) \right) \psi_4 \right] . \tag{124}
\end{aligned}$$

The other three terms are similar, swapping out the relevant basis functions. All together, the gradient matrices can be expressed as the sum of two matrix terms

$$\mathbf{L}_\mu = \frac{1}{(\Delta r)^2 (\Delta \theta)^2} \begin{bmatrix} M_7 M_4 & M_8 M_4 & M_8 M_5 & M_7 M_5 \\ -M_7 M_4 & -M_8 M_4 & -M_8 M_5 & -M_7 M_5 \\ -M_7 M_5 & -M_8 M_5 & -M_8 M_6 & -M_7 M_6 \\ M_7 M_5 & M_8 M_5 & M_8 M_6 & M_7 M_6 \end{bmatrix} \tag{125}$$

and

$$\mathbf{L}_\eta = \frac{1}{(\Delta r)^2(\Delta\theta)^2} \begin{bmatrix} M_9M_{12} & M_{10}M_{12} & M_{10}M_{13} & M_9M_{13} \\ M_{10}M_{12} & M_{11}M_{12} & M_{11}M_{13} & M_{10}M_{13} \\ -M_{10}M_{12} & -M_{11}M_{12} & -M_{11}M_{13} & -M_{10}M_{13} \\ -M_9M_{12} & -M_{10}M_{12} & -M_{10}M_{13} & -M_9M_{13} \end{bmatrix} . \quad (126)$$

The gradient term of the transport equation can now be expressed in a compact form

$$\mu\mathbf{L}_\mu\psi + \eta\mathbf{L}_\eta\psi \quad . \quad (127)$$

As a final comment, unlike the streaming term, the gradient term does not change with angular sweep direction.

3.5 The angular redistribution term

From Equation 88, we begin with the angular redistribution terms, where the flux is expanded in terms of the basis

$$\frac{1}{r} \left[\frac{\alpha_{m+1/2}}{w_m} \tilde{\psi}_{m+1/2} - \frac{\alpha_{m-1/2}}{w_m} \tilde{\psi}_{m-1/2} \right] - \frac{\cot(\theta)}{r} \left[\frac{\beta_{\ell+1/2}}{w_{m\ell}} \tilde{\psi}_{\ell+1/2} - \frac{\beta_{\ell-1/2}}{w_{m\ell}} \tilde{\psi}_{\ell-1/2} \right] \quad . \quad (128)$$

For the angular mesh cell average flux $\tilde{\psi}_{m,\ell}$ defined as in Equation 97, we extend to two angular dimensions the weighted diamond difference approximation for the angular flux

$$\begin{aligned} \tilde{\psi}_{m,\ell}(r, \theta) &= \tau_m \tilde{\psi}_{m+1/2,\ell} + (1 - \tau_m) \tilde{\psi}_{m-1/2,\ell} \\ \tilde{\psi}_{m,\ell}(r, \theta) &= \lambda_{m,\ell} \tilde{\psi}_{m,\ell+1/2} + (1 - \lambda_{m,\ell}) \tilde{\psi}_{m,\ell-1/2} \quad , \end{aligned} \quad (129)$$

where the coefficients are defined as

$$\begin{aligned}\tau_m &= \frac{\mu_m - \mu_{m-1/2}}{\mu_{m+1/2} - \mu_{m-1/2}} \\ \lambda_{m,\ell} &= \frac{\eta_{m,\ell} - \eta_{m,\ell-1/2}}{\eta_{m,\ell+1/2} - \eta_{m,\ell-1/2}} .\end{aligned}\tag{130}$$

We can now apply the DFEM method to these terms. We will begin with the term pertaining to $\frac{\partial}{\partial \mu}$. Since our sweep convention will be that μ is increasing, then $\tilde{\psi}_{m-1/2,\ell}$ is known and the diamond relation can be used to express the unknown $\tilde{\psi}_{m+1/2,\ell}$ as

$$\tilde{\psi}_{m+1/2,\ell}(r, \theta) = \frac{1}{\tau_m} \tilde{\psi}_{m,\ell}(r, \theta) - \frac{1 - \tau_m}{\tau_m} \tilde{\psi}_{m-1/2,\ell}(r, \theta) \quad ,\tag{131}$$

such that in the above expression the first term can be written

$$\frac{1}{r} \left[\frac{\alpha_{m+1/2}}{\tau_m w} \tilde{\psi}_{m,\ell} - \frac{1}{w} \left(\alpha_{m+1/2} \left(\frac{1 - \tau_m}{\tau_m} \right) + \alpha_{m-1/2} \right) \tilde{\psi}_{m-1/2,\ell} \right] .\tag{132}$$

After integration, the second term in Equation 132 is moved to the right side of the transport equation to become part of the source term. The integration over the computational cell volume dV and basis functions proceeds in the following way, for example over basis function B_1

$$\begin{aligned}
& \int_{r_L}^{r_R} dr \int_{\theta_B}^{\theta_T} d\theta B_1 r \sin(\theta) \\
& \left[\frac{\alpha_{m+1/2}}{\tau_m w_{m,\ell}} \tilde{\psi}_{m,\ell} - \frac{1}{w_{m,\ell}} \left(\alpha_{m+1/2} \left(\frac{1-\tau_m}{\tau_m} \right) + \alpha_{m-1/2} \right) \tilde{\psi}_{m-1/2} \right] \\
& = \frac{\alpha_{m+1/2}}{\tau_m w_{m,\ell} (\Delta r)^2 (\Delta \theta)^2} \\
& \left[\left(\int_{r_L}^{r_R} dr r (r_R - r) (r_R - r) \int_{\theta_B}^{\theta_T} d\theta \sin(\theta) (\theta_T - \theta) (\theta_T - \theta) \right) \psi_{1,m,\ell} \right. \\
& + \left(\int_{r_L}^{r_R} dr r (r_R - r) (r - r_L) \int_{\theta_B}^{\theta_T} d\theta \sin(\theta) (\theta_T - \theta) (\theta_T - \theta) \right) \psi_{2,m,\ell} \\
& + \left(\int_{r_L}^{r_R} dr r (r_R - r) (r - r_L) \int_{\theta_B}^{\theta_T} d\theta \sin(\theta) (\theta_T - \theta) (\theta - \theta_B) \right) \psi_{3,m,\ell} \\
& + \left. \left(\int_{r_L}^{r_R} dr r (r_R - r) (r_R - r) \int_{\theta_B}^{\theta_T} d\theta \sin(\theta) (\theta_T - \theta) (\theta - \theta_B) \right) \psi_{4,m,\ell} \right] \tag{133} \\
& - \frac{1}{w_{m,\ell} (\Delta r)^2 (\Delta \theta)^2} \left(\alpha_{m+1/2} \left(\frac{1-\tau_m}{\tau_m} \right) + \alpha_{m-1/2} \right) \\
& \left[\left(\int_{r_L}^{r_R} dr r (r_R - r) (r_R - r) \int_{\theta_B}^{\theta_T} d\theta \sin(\theta) (\theta_T - \theta) (\theta_T - \theta) \right) \psi_{1,m-1/2,\ell} \right. \\
& + \left(\int_{r_L}^{r_R} dr r (r_R - r) (r - r_L) \int_{\theta_B}^{\theta_T} d\theta \sin(\theta) (\theta_T - \theta) (\theta_T - \theta) \right) \psi_{2,m-1/2,\ell} \\
& + \left(\int_{r_L}^{r_R} dr r (r_R - r) (r - r_L) \int_{\theta_B}^{\theta_T} d\theta \sin(\theta) (\theta_T - \theta) (\theta - \theta_B) \right) \psi_{3,m-1/2,\ell} \\
& + \left. \left(\int_{r_L}^{r_R} dr r (r_R - r) (r_R - r) \int_{\theta_B}^{\theta_T} d\theta \sin(\theta) (\theta_T - \theta) (\theta - \theta_B) \right) \psi_{4,m-1/2,\ell} \right] .
\end{aligned}$$

This can be expressed in matrix form

$$\frac{\alpha_{m+1/2}}{\tau_m w_{m,\ell}} \mathbf{P} \begin{bmatrix} \psi_{1,m,\ell} \\ \psi_{2,m,\ell} \\ \psi_{3,m,\ell} \\ \psi_{4,m,\ell} \end{bmatrix} - \frac{1}{w_{m,\ell}} \left(\alpha_{m+1/2} \left(\frac{1-\tau_m}{\tau_m} \right) + \alpha_{m-1/2} \right) \mathbf{P} \begin{bmatrix} \psi_{1,m-1/2,\ell} \\ \psi_{2,m-1/2,\ell} \\ \psi_{3,m-1/2,\ell} \\ \psi_{4,m-1/2,\ell} \end{bmatrix} , \tag{134}$$

with the definition

$$\mathbf{P} = \frac{1}{(\Delta r)^2(\Delta\theta)^2} \begin{bmatrix} M_9M_4 & M_{10}M_4 & M_{10}M_5 & M_9M_5 \\ M_{10}M_4 & M_{11}M_4 & M_{11}M_5 & M_{10}M_5 \\ M_{10}M_5 & M_{11}M_5 & M_{11}M_6 & M_{10}M_6 \\ M_9M_5 & M_{10}M_5 & M_{10}M_6 & M_9M_6 \end{bmatrix}. \quad (135)$$

Likewise, the second angular redistribution term, pertaining to $\frac{\partial}{\partial\omega}$, differs from the above only in the differencing variable, now ω , and with a cosine coefficient instead of sine. However, we have two options in our sweep direction over the η directions, which adds complexity. When η is increasing, from most negative to most positive values, $\tilde{\psi}_{\ell-1/2}$ is known and

$$\tilde{\psi}_{m,\ell+1/2} = \frac{1}{\lambda_{m,\ell}}\tilde{\psi}_{m,\ell} - \frac{1-\lambda_{m,\ell}}{\lambda_{m,\ell}}\tilde{\psi}_{m,\ell-1/2} \quad (136)$$

and the term becomes

$$-\frac{\cot(\theta)}{r} \left[\frac{\beta_{m,\ell+1/2}}{\lambda_{m,\ell}w_{m,\ell}}\tilde{\psi}_{m,\ell} - \frac{1}{w_{m,\ell}} \left(\beta_{m,\ell+1/2} \left(\frac{1-\lambda_{m,\ell}}{\lambda_{m,\ell}} \right) + \beta_{m,\ell-1/2} \right) \tilde{\psi}_{m,\ell-1/2} \right]. \quad (137)$$

When η is decreasing, from most positive to most negative values, $\tilde{\psi}_{m,\ell+1/2}$ is known and

$$\tilde{\psi}_{m,\ell-1/2} = \frac{1}{1-\lambda_{m,\ell}}\tilde{\psi}_{m,\ell} - \frac{\lambda_{m,\ell}}{1-\lambda_{m,\ell}}\tilde{\psi}_{m,\ell+1/2}, \quad (138)$$

and the term becomes

$$-\frac{\cot(\theta)}{r} \left[-\frac{\beta_{m,\ell-1/2}}{(1-\lambda_{m,\ell})w_{m,\ell}} \tilde{\psi}_{m,\ell} + \frac{1}{w_{m,\ell}} \left(\beta_{m,\ell+1/2} + \left(\frac{\lambda_{m,\ell}}{1-\lambda_{m,\ell}} \right) \beta_{m,\ell-1/2} \right) \tilde{\psi}_{m,\ell+1/2} \right] . \quad (139)$$

The integration over volume with B_1 is then, for η increasing,

$$\begin{aligned} & - \int_{r_L}^{r_R} dr \int_{\theta_B}^{\theta_T} d\theta B_1 r \cos(\theta) \\ & \quad \left[\frac{\beta_{m,\ell+1/2}}{\lambda_{m,\ell} w_{m,\ell}} \tilde{\psi}_{m,\ell} - \frac{1}{w_{m,\ell}} \left(\beta_{m,\ell+1/2} \left(\frac{1-\lambda_{m,\ell}}{\lambda_{m,\ell}} \right) + \beta_{m,\ell-1/2} \right) \tilde{\psi}_{\ell-1/2} \right] \\ & = \frac{\beta_{m,\ell+1/2}}{\lambda_{m,\ell} w_{m,\ell} (\Delta r)^2 (\Delta \theta)^2} \\ & \quad \left[\left(\int_{r_L}^{r_R} dr r (r_R - r) (r_R - r) \int_{\theta_B}^{\theta_T} d\theta \cos(\theta) (\theta_T - \theta) (\theta_T - \theta) \right) \psi_{1,m,\ell} \right. \\ & \quad + \left(\int_{r_L}^{r_R} dr r (r_R - r) (r - r_L) \int_{\theta_B}^{\theta_T} d\theta \cos(\theta) (\theta_T - \theta) (\theta_T - \theta) \right) \psi_{2,m,\ell} \\ & \quad + \left(\int_{r_L}^{r_R} dr r (r_R - r) (r - r_L) \int_{\theta_B}^{\theta_T} d\theta \cos(\theta) (\theta_T - \theta) (\theta - \theta_B) \right) \psi_{3,m,\ell} \\ & \quad + \left. \left(\int_{r_L}^{r_R} dr r (r_R - r) (r_R - r) \int_{\theta_B}^{\theta_T} d\theta \cos(\theta) (\theta_T - \theta) (\theta - \theta_B) \right) \psi_{4,m,\ell} \right] \\ & - \frac{1}{w_{m,\ell} (\Delta r)^2 (\Delta \theta)^2} \left(\beta_{m,\ell+1/2} \left(\frac{1-\lambda_{m,\ell}}{\lambda_{m,\ell}} \right) + \beta_{m,\ell-1/2} \right) \\ & \quad \left[\left(\int_{r_L}^{r_R} dr r (r_R - r) (r_R - r) \int_{\theta_B}^{\theta_T} d\theta \cos(\theta) (\theta_T - \theta) (\theta_T - \theta) \right) \psi_{1,m,\ell-1/2} \right. \\ & \quad + \left(\int_{r_L}^{r_R} dr r (r_R - r) (r - r_L) \int_{\theta_B}^{\theta_T} d\theta \cos(\theta) (\theta_T - \theta) (\theta_T - \theta) \right) \psi_{2,m-1/2,\ell-1/2} \\ & \quad + \left(\int_{r_L}^{r_R} dr r (r_R - r) (r - r_L) \int_{\theta_B}^{\theta_T} d\theta \cos(\theta) (\theta_T - \theta) (\theta - \theta_B) \right) \psi_{3,m-1/2,\ell-1/2} \\ & \quad + \left. \left(\int_{r_L}^{r_R} dr r (r_R - r) (r_R - r) \int_{\theta_B}^{\theta_T} d\theta \cos(\theta) (\theta_T - \theta) (\theta - \theta_B) \right) \psi_{4,m-1/2,\ell-1/2} \right] . \quad (140) \end{aligned}$$

This can be expressed in matrix form

$$\frac{\beta_{m,\ell+1/2}}{\lambda_{m,\ell}w_{m,\ell}}\mathbf{Q}\begin{bmatrix} \psi_{1,m,\ell} \\ \psi_{2,m,\ell} \\ \psi_{3,m,\ell} \\ \psi_{4,m,\ell} \end{bmatrix} - \frac{1}{w_{m,\ell}}\left(\beta_{m,\ell+1/2}\left(\frac{1-\lambda_{m,\ell}}{\lambda_{m,\ell}}\right) + \beta_{m,\ell-1/2}\right)\mathbf{Q}\begin{bmatrix} \psi_{1,m,\ell-1/2} \\ \psi_{2,m,\ell-1/2} \\ \psi_{3,m,\ell-1/2} \\ \psi_{4,m,\ell-1/2} \end{bmatrix}, \quad (141)$$

with the definition

$$\mathbf{Q} = \frac{1}{(\Delta r)^2(\Delta\theta)^2}\begin{bmatrix} M_9M_{14} & M_{10}M_{14} & M_{10}M_{15} & M_9M_{15} \\ M_{10}M_{14} & M_{11}M_{14} & M_{11}M_{15} & M_{10}M_{15} \\ M_{10}M_{15} & M_{11}M_{15} & M_{11}M_{16} & M_{10}M_{16} \\ M_9M_{15} & M_{10}M_{15} & M_{10}M_{16} & M_9M_{16} \end{bmatrix}. \quad (142)$$

For η decreasing from most positive to most negative, the term changes only by coefficients to become

$$-\frac{\beta_{m,\ell-1/2}}{(1-\lambda_{m,\ell})w_{m,\ell}}\mathbf{Q}\begin{bmatrix} \psi_{1,m,\ell} \\ \psi_{2,m,\ell} \\ \psi_{3,m,\ell} \\ \psi_{4,m,\ell} \end{bmatrix} + \frac{1}{w_{m,\ell}}\left(\beta_{m,\ell+1/2} + \beta_{m,\ell-1/2}\left(\frac{\lambda_{m,\ell}}{1-\lambda_{m,\ell}}\right)\right)\mathbf{Q}\begin{bmatrix} \psi_{1,m,\ell+1/2} \\ \psi_{2,m,\ell+1/2} \\ \psi_{3,m,\ell+1/2} \\ \psi_{4,m,\ell+1/2} \end{bmatrix}. \quad (143)$$

3.6 The removal term

The removal term and source term have the same geometric coefficients and are operated on by the collision matrix, denoted here as \mathbf{T} . The integration against basis function $B_1(r, \theta)$ and dV , for example, is

$$\begin{aligned}
& \int_{r_L}^{r_R} dr \int_{\theta_B}^{\theta_T} d\theta r^2 \sin(\theta) \Sigma_t \tilde{\psi}_{m,\ell} B_1 \\
&= \frac{\Sigma_t}{(\Delta r)^2 (\Delta \theta)^2} \left[\left(\int_{r_L}^{r_R} dr r^2 (r_R - r)(r - r_L) \int_{\theta_B}^{\theta_T} d\theta \sin(\theta) (\theta_T - \theta)(\theta - \theta_B) \right) \psi_{1,m,\ell} \right. \\
&+ \left(\int_{r_L}^{r_R} dr r^2 (r_R - r)(r - r_L) \int_{\theta_B}^{\theta_T} d\theta \sin(\theta) (\theta_T - \theta)(\theta_T - \theta) \right) \psi_{2,m,\ell} \\
&+ \left(\int_{r_L}^{r_R} dr r^2 (r_R - r)(r - r_L) \int_{\theta_B}^{\theta_T} d\theta \sin(\theta) (\theta_T - \theta)(\theta - \theta_B) \right) \psi_{3,m,\ell} \\
&\left. + \left(\int_{r_L}^{r_R} dr r^2 (r_R - r)(r - r_L) \int_{\theta_B}^{\theta_T} d\theta \sin(\theta) (\theta_T - \theta)(\theta - \theta_B) \right) \psi_{4,m,\ell} \right] . \tag{144}
\end{aligned}$$

Integration over the remaining basis vectors follow similarly, and we can define the collision matrix

$$\mathbf{T} = \frac{1}{(\Delta r)^2 (\Delta \theta)^2} \begin{bmatrix} M_1 M_4 & M_2 M_4 & M_2 M_5 & M_1 M_5 \\ M_2 M_4 & M_3 M_4 & M_3 M_5 & M_2 M_5 \\ M_2 M_5 & M_3 M_5 & M_3 M_6 & M_2 M_6 \\ M_1 M_5 & M_2 M_5 & M_2 M_6 & M_1 M_6 \end{bmatrix} , \tag{145}$$

such that in compact form the removal term can be expressed as $\Sigma_t \mathbf{T} \psi$.

3.7 The source term

The source term, S , has been expanded in the same nodal basis as ψ , for a given cell as

$$S = \sum_{i=1}^4 S_i B_i . \tag{146}$$

If the source term is constant in the spatial coordinates, then it can be expressed as $\mathbf{T}S$, where S is a vector $S_i = \Sigma_{sg'} \phi_i + q_i$. Generally, for the definitions that follow, we will leave it as S . The scattering source consists of self-scattering (within an energy

group), and in-scattering from other energy groups. The scattering source can be expressed as (omitting energy indices for brevity)

$$q_{scat}(r, \theta, \mu, \omega) = \sum_{\ell=0}^L \sum_{m=0}^{\ell} (2 - \delta_{m0}) Y_{\ell m}^e(\mu, \omega) \Sigma_{\ell} \phi_{\ell}^m(r, \theta, \mu, \omega) \quad , \quad (147)$$

where the scalar flux moments are (translating ω back to the η unit used in the definitions)

$$\phi_{\ell}^m(r, \theta, \mu, \omega) = \sum_{\mu, \eta} w_{\mu, \eta} \psi_{r, \theta, \mu, \eta} Y_{\ell m}^e(\mu, \omega) \quad , \quad (148)$$

and the azimuthally-even two-dimensional forms of the spherical harmonics are given by the following relations

$$Y_{\ell m}^e(\mu, \omega) = \sqrt{C_{\ell m}} P_{\ell}^m(\mu) \cos(m\omega) \quad , \quad (149)$$

where

$$C_{\ell m} = \frac{(2\ell + 1)(\ell - m)!}{(\ell + m)!} \quad , \quad (150)$$

and

$$P_{\ell}^m(\mu) = \frac{(-1)^m}{2^{\ell} \ell!} (1 - \mu^2)^{m/2} \frac{d^{\ell+m}}{d\mu^{\ell+m}} (\mu^2 - 1)^{\ell} \quad . \quad (151)$$

The values of $P_{\ell}^m(\mu)$, up to $L=3$ are given in Table 2.

In the case of isotropic scattering, the polynomial expansion is truncated at $\ell = 0$ and the scattering cross section reduces to

$$q_{scat}(r, \theta, \mu, \omega) = \Sigma_0 \phi_0(r, \theta, \mu, \omega) \quad . \quad (152)$$

Table 2: Spherical harmonic terms for the scattering operator

ℓ	m	$\sqrt{C_{\ell m}}$	$P_{\ell}^m(\mu)$
0	0	1	1
1	0	$\sqrt{3}$	μ
1	1	$\sqrt{\frac{3}{2}}$	$-(1 - \mu^2)^{1/2}$
2	0	$\sqrt{5}$	$\frac{1}{2}(3\mu^2 - 1)$
2	1	$\sqrt{\frac{5}{6}}$	$-3\mu(1 - \mu^2)^{1/2}$
2	2	$\sqrt{\frac{5}{24}}$	$3(1 - \mu^2)$
3	0	$\sqrt{7}$	$\frac{1}{2}(5\mu^3 - 3\mu)$
3	1	$\sqrt{\frac{7}{12}}$	$\frac{1}{2}(1 - \mu^2)^{1/2}(15\mu^2 - 3)$
3	2	$\sqrt{\frac{7}{120}}$	$15\mu(1 - \mu^2)$
3	3	$\sqrt{\frac{7}{720}}$	$-15\mu(1 - \mu^2)^{3/2}$

3.8 Starting directions

Starting directions are used to initialize the angular mesh recursions. These starting directions are treated slightly differently than the directions that arise from the quadrature approximation to the continuous direction operator, that we call the weighted directions. One difference is that the starting directions have zero weight, so they do not contribute to the scalar flux calculation. The lack of a suitable weight means that the terms involving the quadrature weight w are treated differently, as we describe in this section. The WDD coefficients τ and λ are also adjusted for these directions and in some cases disappear altogether. For some of the starting directions there is no angular redistribution at all, and for some there is angular redistribution along only one angular component. We begin by examining the one-dimensional case and then we build up the starting direction transport equations we need for the present discretization.

In one-dimensional spherical coordinates the starting direction is typically $\mu = -1$. Figure 16 shows the starting direction placement for an S_6 discretization in one-

dimensional spherical coordinates. This represents a particle direction radially inward, and is a direction that does not change with respect to the coordinate axes as the radial position changes. This implies the angular redistribution terms are zero for this starting direction, since there is no angular redistribution along the particle path owing purely to geometrical considerations.

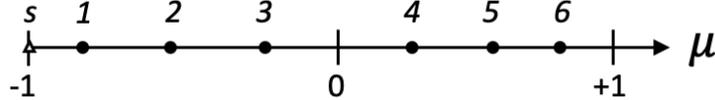


Figure 16: Ordering of directions in one-dimensional spherical geometry for S_6 approximation. The starting direction is denoted by s .

In analogy to the one-dimensional spherical system, we find that our equations for two-dimensional spheres demand that we define starting directions in both angular dimensions to begin our angular recursions. In Figure 17 we show the two-dimensional starting direction locations on the angular coordinate system as triangular points, and the weighted S_6 quadrature directions as filled dots. Note that the information flow between starting and weighted directions for the ATMOTRAN implementation will be detailed in Chapter 4. The point arrangement is Gauss-Legendre in μ and Gauss-Chebyshev in η . The third angle axis ξ points toward the reader. The starting directions all have the feature of $\xi = 0$ (these are located in the $\mu-\eta$ plane). Returning to the original definition of the transport equation we can proceed to derive discretized equations for the starting directions. Like the one-dimensional system, the angular recursion in the μ direction is started by calculation in the $\mu = -1$ direction. For this direction we can confirm there is no η component by the geometrical relation

$$\eta_{m\ell} = (1 - \mu_m^2)^{1/2} \cos(\omega_{m\ell}) = 0 \quad . \quad (153)$$

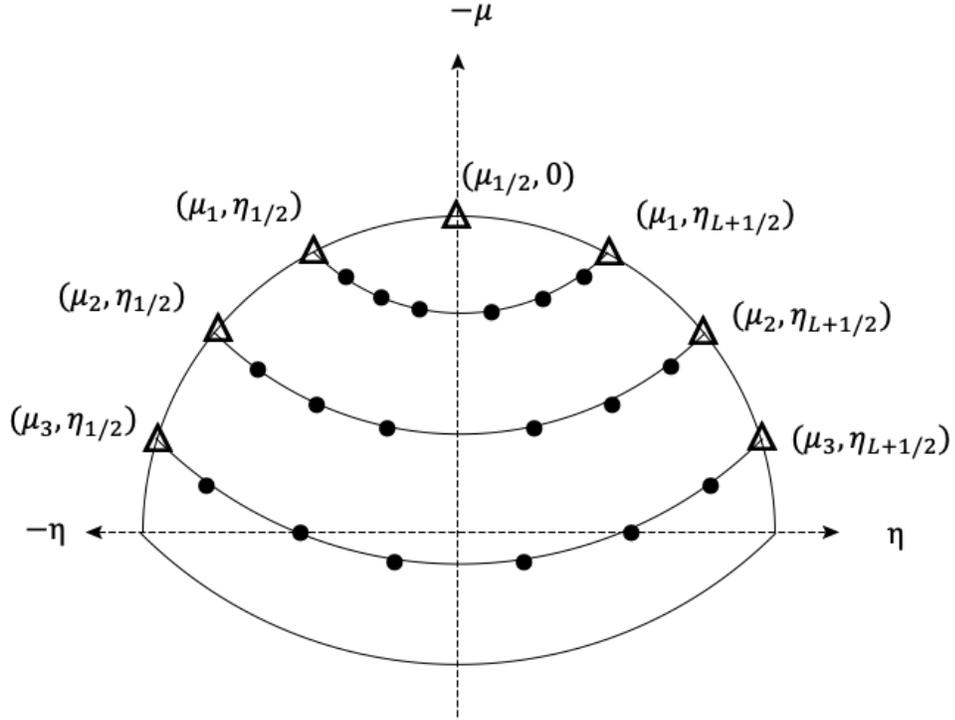


Figure 17: Positioning of directions in two-dimensional spherical-polar geometry for S_6 approximation. The starting directions are shown as triangles.

In one-dimensional spherical geometry the weighted directions are a set $\{\mu_1 \dots \mu_M\}$. For reflected geometries it is useful to calculate $\mu_{M+1/2} = +1$. Both these directions do not require angular redistribution, as evidenced by the disappearance of the angular derivative in the transport equation for $(\mu = \pm 1, \eta = 0)$. However, the second angular derivative adds a starting direction (and a reflected complement) for each μ level m where, for an angular index of the format $\eta_{m,\ell}$, these starting directions are $\eta_{m,1/2}$ and $\eta_{m,L+1/2}$, where there are L weighted directions on a μ level (e.g. for S_6 there are six directions on a level in Gauss-Chebyshev-Legendre quadrature). Note that for the implementation in ATMOTRAN, the use of a square Gauss-Chebyshev-Legendre quadrature means $M = L$. Examining the transport equations, these η starting directions do have angular redistribution to consider in the μ component of

the equation. In Chapter 2 we introduced the reasoning for bidirectional angular mesh sweeps due to the characteristic particle paths in this coordinate system. Information flow in the angular mesh must be reversed depending on the cell location being in either the upper or lower half of the spatial θ coordinate. In the upper half we begin with the most positive η directions and work downward, while in the lower half we begin with the most negative η and work upward. At the boundary $\theta = \pi/2$ we need to allow a spatial upwind flux value for the starting directions (using the same value of η) so we must have all η values calculated for both upper and lower halves of the physical sphere. Of course, there is also mathematical need for these angles in the case of reflection at the $\theta = 0$ and $\theta = \pi$, but these values are trivial to the solution of the transport equation for full spheres since at these points the physical surface area is zero and there cannot be particle flow. However, solutions on spherical segments would require these directions. So, while the μ complement ($\mu = +1$) is required for iterative solutions of problems with reflection conditions in the radial dimension (a reflected sphere, for example), more η starting directions are required to permit upwinding of angular fluxes across the $\theta = \pi/2$ boundary. That is, they are both required, but for different reasons in the implementation. We now proceed to derive discretized transport equations for these starting directions.

3.8.1 Starting directions in μ

Returning to the two-dimensional spherical-polar conservation form of the transport equation (Equation 84) and evaluating the derivatives, where $\mu = -1$ and $\eta, \xi = 0$, the transport equation reduces to

$$-\frac{\partial}{\partial r}\psi + \Sigma\psi = S \quad . \quad (154)$$

This is the one-dimensional slab transport equation in the direction $\mu = -1$. This equation has no angular redistribution term, and so we must ensure this is also the case when we evaluate the discretized form of the starting direction equation. This starting direction is sub-scripted $\mu_{1/2}$ or $\mu_{M+1/2}$, given M weighted directions. Recall the angular redistribution term is

$$\frac{1}{r} \left[\frac{\alpha_{m+1/2}}{w_m} \psi_{m+1/2} - \frac{\alpha_{m-1/2}}{w_m} \psi_{m-1/2} \right] . \quad (155)$$

The discretized form of this term, for $\mu_{1/2}$ is

$$\frac{1}{r} \left[\frac{\alpha_1}{w_{1/2}} \psi_1 - \frac{\alpha_0}{w_{1/2}} \psi_0 \right] . \quad (156)$$

At least within the recursion relation definition, α is only defined at edge points, which are half-integer values, and there is no physical meaning to weights $w_{1/2}$ for a zero-weighted starting direction. These undefined terms can be replaced with tractable values in the following manner. For the case of the $\mu = 1/2$ starting direction, the recursion relation for α is

$$\alpha_1 - \alpha_0 = -2w_{1/2}\mu_{1/2} \quad , \quad (157)$$

and this term is undefined outside the recursion relation, so $\alpha_0 \equiv 0$. So we have, when calculating the $m=1/2$ starting direction,

$$\frac{\alpha_1}{w_{1/2}} = -2\mu_{1/2} = 2 \quad . \quad (158)$$

Likewise, for $m = M + 1/2$, we have

$$\alpha_{M+1} - \alpha_M = -2w_{M+1/2}\mu_{M+1/2} \quad , \quad (159)$$

and this term is undefined outside the recursion relation, so $\alpha_{M+1} \equiv 0$. So we have,

$$\frac{\alpha_M}{w_{M+1/2}} = 2\mu_{M+1/2} = 2 \quad . \quad (160)$$

Since there is no angular redistribution, there is no WDD coefficient (τ_m) for these starting angles.

3.8.2 Starting directions in η

The starting directions in η exist on the weighted μ -levels and are slightly more complicated. Examining the derivative in the azimuthal angle, an extreme value can be chosen, analogous to the choice of $\mu = -1$. When sweeping from most negative to most positive η , we choose $\omega_S = \pi$, for which $\cos(\omega_S) = -1$ and then η can be calculated, given μ . When sweeping from most positive to most negative values of η , we choose $\omega_S = 0$ as the initial azimuthal direction. These choices are consistent with the geometrical requirement that $\xi = 0$ for all starting directions (recall the definition $\xi = \sqrt{1 - \mu^2} \sin(\omega_S) = 0$). Calculation of the starting η value is by the following relation

$$\eta_{m\ell} = (1 - \mu_m^2)^{1/2}(-1) \quad , \quad (161)$$

which provides the unweighted (μ, η) pairs needed to initialize the angular mesh recursion in the η direction. Example values for S_6 are shown in Table 3, for only the most negative μ level.

With this process to define the η values for the starting directions, we can now evaluate the recursion relation coefficients α and β . Returning to the the transport equation in conservation form (Equation 84)

Table 3: Gauss-Chebyshev-Legendre square S_6 η angles along the μ_1 level

index ℓ	μ	η	w
START	-9.3246951420E-01	-3.6124871151E-01	0
1	-9.3246951420E-01	-3.4893942475E-01	8.9705294440E-02
2	-9.3246951420E-01	-2.5544138768E-01	8.9705294440E-02
3	-9.3246951420E-01	-9.3498037070E-02	8.9705294440E-02
4	-9.3246951420E-01	9.3498037070E-02	8.9705294440E-02
5	-9.3246951420E-01	2.5544138768E-01	8.9705294440E-02
6	-9.3246951420E-01	3.4893942475E-01	8.9705294440E-02

$$\frac{\mu}{r^2} \frac{\partial}{\partial r} (r^2 \psi) + \frac{\eta}{r \sin(\theta)} \frac{\partial}{\partial \theta} (\sin(\theta) \psi) + \frac{1}{r} \frac{\partial}{\partial \mu} [(1 - \mu^2) \psi] - \frac{\cot(\theta)}{r} \frac{\partial}{\partial \omega} (\xi \psi) + \Sigma \psi = S \quad . \quad (162)$$

We note there are two terms that involve η and β ,

$$\frac{\eta}{r \sin(\theta)} \frac{\partial}{\partial \theta} (\sin(\theta) \psi) - \frac{\cot(\theta)}{r} \frac{\partial}{\partial \omega} (\xi \psi) \quad , \quad (163)$$

or equivalently

$$\frac{\eta}{r} \frac{\partial \psi}{\partial \theta} - \xi \frac{\cot \theta}{r} \frac{\partial \psi}{\partial \omega} \quad . \quad (164)$$

We showed earlier that all starting directions have $\xi = 0$, so the second of the above terms is zero for all starting directions in both dimensions. The first term is zero for the μ_s starting directions since $\eta = 0$. For the starting directions where $\eta \neq 0$, this term is merely the slab equation (in the polar direction), for which there is no angular redistribution. Thus we expect the angular redistribution in η to always reduce to zero for the η starting directions. This is consistent with the one-dimensional concept that there is no angular redistribution along the starting dimension axis for a starting direction. Explicitly, this does not prohibit angular redistribution in the μ direction

for the η starting directions, and we find there are indeed valid α coefficients to describe this redistribution. Since none of the full set of starting directions make use of weighted η levels at all, the recursion terms in β do not exist. We recall the discretization of the angular variable ω resulted in this term

$$-\frac{\cot\theta}{r} \left[\frac{\beta_{m,\ell+1/2}}{w_{m,\ell}} \psi_{m,\ell+1/2} - \frac{\beta_{m,\ell-1/2}}{w_{m,\ell}} \psi_{m,\ell-1/2} \right] \quad (165)$$

There are two η starting directions, the maximum positive and maximum negative values, corresponding to $\omega = 0, \pi$ respectively. As for the μ starting directions we need to swap out the recursion and weight terms for an equivalent value that is defined. For the case of the η starting directions, $\ell = 1/2$ or $\ell = L + 1/2$, the recursion relation becomes, for $\ell = 1/2$,

$$\beta_{m,1} - \beta_{m,0} = w_{m,1/2} \eta_{m,1/2} \quad . \quad (166)$$

Since this term is undefined outside the recursion relation, $\beta_{m,0} \equiv 0$. So when calculating the $\ell = 1/2$ starting direction

$$\frac{\beta_{m,1}}{w_{m,1/2}} = \eta_{m,1/2} \quad . \quad (167)$$

Likewise, for $\ell = L + 1/2$

$$\beta_{m,L+1} - \beta_{m,L} = w_{m,L+1/2} \eta_{m,L+1/2} \quad , \quad (168)$$

and this term is undefined outside the recursion relation, so $\beta_{m,L+1} \equiv 0$, and

$$-\frac{\beta_{m,L}}{w_{m,L+1/2}} = \eta_{m,L+1/2} \quad . \quad (169)$$

3.9 The fully discretized transport equations in the two-dimensional spherical-polar (r, θ) coordinate system

Collecting the terms from the previous sections we can write for a given cell at energy group $g \in [1, G]$, for weighted directions, for η increasing

$$\begin{aligned}
& \eta \mathbf{V}_\eta \begin{bmatrix} \tilde{\psi}_{1B,g} \\ \tilde{\psi}_{2B,g} \\ \tilde{\psi}_{3T,g} \\ \tilde{\psi}_{4T,g} \end{bmatrix} + \mu \mathbf{V}_\mu \begin{bmatrix} \tilde{\psi}_{1L,g} \\ \tilde{\psi}_{2R,g} \\ \tilde{\psi}_{3R,g} \\ \tilde{\psi}_{4L,g} \end{bmatrix} + \mu \mathbf{L}_\mu \Psi_{m,\ell,g} + \eta \mathbf{L}_\eta \Psi_{m,\ell,g} \\
& + \left[\frac{\alpha_{m+1/2}}{w\tau_m} \mathbf{P} - \frac{\beta_{m,\ell+1/2}}{w\lambda_{m,\ell}} \mathbf{Q} \right] \Psi_{m,\ell,g} + \Sigma_{t,g} \mathbf{T} \Psi_{m,\ell,g} \\
& = \frac{1}{w} \left(\alpha_{m+1/2} \left(\frac{1-\tau_m}{\tau_m} \right) + \alpha_{m-1/2} \right) \mathbf{P} \Psi_{m-1/2,\ell,g} \\
& - \frac{1}{w} \left(\beta_{m,\ell+1/2} \left(\frac{1-\lambda_{m,\ell}}{\lambda_\ell} \right) + \beta_{m,\ell-1/2} \right) \mathbf{Q} \Psi_{m,\ell-1/2,g} \\
& + \sum_{g'=1}^G \Sigma_{s,g' \rightarrow g} \phi_{g'} + q_{m,\ell,g} \quad ,
\end{aligned} \tag{170}$$

and for weighted directions, for η decreasing

$$\begin{aligned}
& \eta \mathbf{V}_\eta \begin{bmatrix} \tilde{\psi}_{1B,g} \\ \tilde{\psi}_{2B,g} \\ \tilde{\psi}_{3T,g} \\ \tilde{\psi}_{4T,g} \end{bmatrix} + \mu \mathbf{V}_\mu \begin{bmatrix} \tilde{\psi}_{1L,g} \\ \tilde{\psi}_{2R,g} \\ \tilde{\psi}_{3R,g} \\ \tilde{\psi}_{4L,g} \end{bmatrix} + \mu \mathbf{L}_\mu \Psi_{m,\ell,g} + \eta \mathbf{L}_\eta \Psi_{m,\ell,g} \\
& + \left[\frac{\alpha_{m+1/2}}{w\tau_m} \mathbf{P} + \frac{\beta_{m,\ell-1/2}}{w(1-\lambda_{m,\ell})} \mathbf{Q} \right] \Psi_{m,\ell,g} + \Sigma_{t,g} \mathbf{T} \Psi_{m,\ell,g} \\
& = \frac{1}{w} \left(\alpha_{m+1/2} \left(\frac{1-\tau_m}{\tau_m} \right) + \alpha_{m-1/2} \right) \mathbf{P} \Psi_{m-1/2,\ell,g} \\
& + \frac{1}{w} \left(\beta_{m,\ell+1/2} + \beta_{m,\ell-1/2} \left(\frac{\lambda_{m,\ell}}{1-\lambda_{m,\ell}} \right) \right) \mathbf{Q} \Psi_{m,\ell+1/2,g} \\
& + \sum_{g'=1}^G \Sigma_{s,g' \rightarrow g} \phi_{g'} + q_{m,\ell,g} \quad .
\end{aligned} \tag{171}$$

The starting direction equation for μ_s , where $s = 1/2, M + 1/2$ starting directions is

$$\begin{aligned}
\mu \mathbf{V}_{\mu_s} \begin{bmatrix} \tilde{\psi}_{s,1L,g} \\ \tilde{\psi}_{s,2R,g} \\ \tilde{\psi}_{s,3R,g} \\ \tilde{\psi}_{s,4L,g} \end{bmatrix} + \mu \mathbf{L}_\mu \Psi_{s,g} - 2\mu_s \mathbf{P} \Psi_{s,g} + \Sigma_{t,g} \mathbf{T} \Psi_{s,g} = \sum_{g'=1}^G \Sigma_{s,g' \rightarrow g} \phi_{g'} + q_{m,\ell,g} \quad ,
\end{aligned} \tag{172}$$

and the transport equation for the $\eta_{m,\ell=s}$ starting directions in either sweep direction is

$$\begin{aligned}
& \eta_{m,s} \mathbf{V}_\eta \begin{bmatrix} \tilde{\psi}_{m,s,1B,g} \\ \tilde{\psi}_{m,s,2B,g} \\ \tilde{\psi}_{m,s,3T,g} \\ \tilde{\psi}_{m,s,4T,g} \end{bmatrix} + \mu_m \mathbf{V}_\mu \begin{bmatrix} \tilde{\psi}_{m,s,1L,g} \\ \tilde{\psi}_{m,s,2R,g} \\ \tilde{\psi}_{m,s,3R,g} \\ \tilde{\psi}_{m,s,4L,g} \end{bmatrix} + \mu_m \mathbf{L}_\mu \Psi_{m,s,g} + \eta_{m,s} \mathbf{L}_\eta \Psi_{m,s,g} \\
& + \left[\frac{\alpha_{m+1/2}}{w_m \tau_m} \mathbf{P} \right] \Psi_{m,s,g} - \eta_{m,s} \mathbf{Q} \Psi_{m,s,g} + \Sigma_{t,g} \mathbf{T} \Psi_{m,s,g} \\
& = \frac{1}{w_m} \left(\alpha_{m+1/2} \left(\frac{1 - \tau_m}{\tau_m} \right) + \alpha_{m-1/2} \right) \mathbf{P} \Psi_{m-1/2,s,g} \\
& + \sum_{g'=1}^G \Sigma_{s,g' \rightarrow g} \phi_{g'} + q_{m,\ell,g} \quad .
\end{aligned} \tag{173}$$

These equations use the following definitions:

μ is the direction vector, $\mu = \cos(\theta)$,

η is the direction vector, $\eta = \sqrt{1 - \mu^2} \cos(\omega)$,

w is quadrature weight for angle index (m, ℓ) ,

$\alpha_{m+1/2}$ is the angular redistribution coefficient for the upper edge of cell m ,

$\alpha_{m-1/2}$ is the angular redistribution coefficient for the lower edge of cell m ,

$\beta_{m,\ell+1/2}$ is the angular redistribution coefficient for the upper edge of cell (m, ℓ) ,

$\beta_{m,\ell-1/2}$ is the angular redistribution coefficient for the lower edge of cell (m, ℓ) ,

τ_m is the weighted diamond difference mixing parameter for cell m ,

$\lambda_{m,\ell}$ is the weighted diamond difference mixing parameter for cell (m, ℓ) ,

$\Sigma_{t,g}$ is the total interaction cross-section,

$\Sigma_{s,g' \rightarrow g}$ is the scattering interaction cross-section from energy group g' to energy group g ,

$\phi_{g'}$ is the scalar flux at energy group g' ,

$q_{m,\ell,g}$ is the external source at angle indices m, ℓ at energy group g ,

\mathbf{V}_μ is a matrix of surface integrals over the cell, representing surface gradients in the μ direction,

\mathbf{V}_η is a matrix of surface integrals over the cell, representing surface gradients in the η direction,

\mathbf{L}_μ is a matrix of volume integrals over the cell, representing volume gradients in the μ direction,

\mathbf{L}_η is a matrix of volume integrals over the cell, representing volume gradients in the η direction,

\mathbf{P} is a matrix of volume integrals over the cell, representing angular redistribution amongst the μ directions,

\mathbf{Q} is a matrix of volume integrals over the cell, representing angular redistribution amongst the η directions,

\mathbf{T} is the mass matrix,

$\tilde{\psi}_{ik,g}$ is the numerical flux at cell corner i at surface k , energy group g , and

$\Psi_{m,\ell,g}$ is a vector of angular fluxes at each cell corner for angle index (m, ℓ) , energy group g .

4 ATMOTRAN Implementation

This chapter includes functional block diagrams of ATMOTRAN, brief code module descriptions, and some practical computational details of the code. Details focus on those elements that are unique to ATMOTRAN, or are required in order to understand the verification results that follow in Chapter 5.

4.1 Approach and main functional blocks

ATMOTRAN is a C++ program comprising around 7000 lines of code, split into six main functional blocks:

`geometry.cc`: Builds data structures related to problem physical geometry.

`xsstore.cc`: Loads multigroup data for application to geometry.

`quadrature.cc`: Builds quadrature sets, angular redistribution and angular weighted diamond difference coefficients.

`solver.cc`: Sets up data structures including all angular and scalar fluxes and sources. This function includes the main outer iteration (in energy), inner iteration angle selection, and calculation of convergence.

`sweep.cc`: This code performs the spatial portion of the inner iteration for a specific angle. The sweep direction and extents are determined by the value of

the angle.

`utils.cc`: This includes functions that are called from the sweep and solver routines to access data structures and perform repeated calculations.

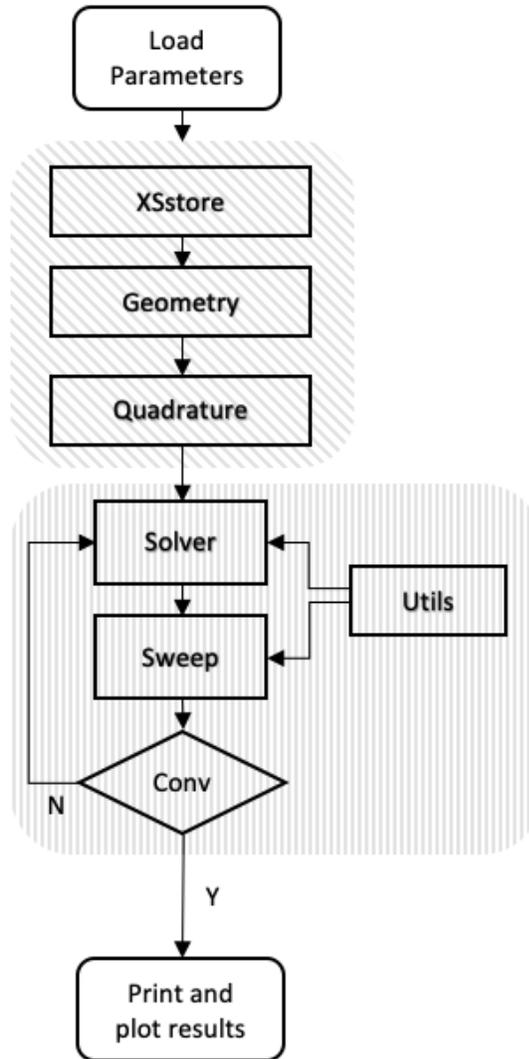


Figure 18: ATMOTRAN functional block diagram.

The current code depends on Eigen [45], an open-source header-only set of template files for linear algebra, matrix and vector operations, and numerical algorithms.

Eigen was used primarily to provide a set of methods for computing matrix inverses and for matrix and vector classes since they were available, however standard template library containers could also be used. Data analysis for the results show in Chapter 5 used scripts written in ROOT [46], with some plotting in Python matplotlib [47]. The build system is GNU autotools [48]. Simulation parameters are set in a text file and read in at runtime. This approach allows for running successive simulations without recompiling the code and also allows for running simultaneous simulations with different parameters, for example on multiple CPUs. The generic parameters file for a non-atmospheric-scale geometry with a single material includes the following:

Table 4: Simulation parameters with types and example values

Par.	Type	Example	Description
quad	int	6	quadrature order
nx	int	10	number of cells in radial dimension
ny	int	10	number of cells in polar dimension
xl	float	10.0	minimum radial edge in cm
xr	float	20.0	maximum radial edge in cm
yl	float	0.0	minimum polar edge in units of π
yr	float	1.0	maximum polar edge in units of π
xt	float	1.0	total cross section $N\sigma$, units cm^{-1}
c	float	0.9	ratio of scattering to total cross section
tol	float	1E-9	convergence tolerance
reflectx	boolean	0	reflection in the radial dimension
reflecty	boolean	1	reflection in the polar dimension
q	vector float	1.0,0,0...	internal source as function of energy, cm^{-3}
qb	vector float	0,0,0...	boundary source as function of energy, cm^{-3}

The parameters shown in Table 4 relate to a general problem geometry. For atmospheric geometries the physical extents and cross-sectional data shown in this parameter file are overwritten in the `geometry` function with altitude-dependent values. Some additional detail on each main functional block follows.

4.2 Details of `xsstore.cc`

This function reads nuclear cross section data text files processed and formatted by the NJOY2016 code [49] and makes it available to be combined with the physical geometric definitions to create cell-averaged cross section quantities. The NJOY Nuclear Data Processing System is a code designed to read evaluated data in ENDF [50] format, to transform it as requested by a user, and to export it in another format suitable for transport code applications. For ATMOTRAN, NJOY was used to create representative multigroup cross section data. NJOY output files are contained within ATMOTRAN and read in at runtime. Appendix C contains more details related to how ATMOTRAN interfaces with the nuclear data. Some example data is shown in Table 5, for the isotope ^{14}N , which is the principal component of the atmosphere. Group-to-group isotropic scattering fractions are shown in Table 6.

Table 5: Multigroup nuclear data for ^{14}N

Bin	Energy low (eV)	Energy high (eV)	σ_t	σ_{s0}	c
1	1.39E-04	1.52E-01	17.04390	9.91140	0.58152
2	1.52E-01	4.14E-01	10.52340	9.91132	0.94184
3	4.14E-01	1.13E+00	10.28160	9.91118	0.96397
4	1.13E+00	3.06E+00	10.13540	9.91075	0.97784
5	3.06E+00	8.32E+00	10.04630	9.90991	0.98642
6	8.32E+00	2.26E+01	9.99016	9.90747	0.99172
7	2.26E+01	6.14E+01	9.95045	9.90028	0.99496
8	6.14E+01	1.67E+02	9.91491	9.88452	0.99693
9	1.67E+02	4.54E+02	9.85023	9.83187	0.99814
10	4.54E+02	1.24E+03	9.73478	9.72369	0.99886
11	1.24E+03	3.35E+03	9.44609	9.43944	0.99930
12	3.35E+03	9.12E+03	8.77021	8.76628	0.99955
13	9.12E+03	2.48E+04	7.53824	7.53590	0.99969
14	2.48E+04	6.76E+04	5.86866	5.86711	0.99974
15	6.76E+04	1.84E+05	4.29462	4.29325	0.99968
16	1.84E+05	3.03E+05	3.35065	3.34873	0.99943
17	3.03E+05	5.00E+05	2.90588	2.88641	0.99330
18	5.00E+05	8.23E+05	2.07450	2.01835	0.97293
19	8.23E+05	1.35E+06	1.76938	1.74411	0.98572
20	1.35E+06	1.74E+06	2.16885	2.07264	0.95564
21	1.74E+06	2.23E+06	1.77285	1.66523	0.93930
22	2.23E+06	2.87E+06	1.44567	1.29339	0.89466
23	2.87E+06	3.68E+06	1.65835	1.32017	0.79607
24	3.68E+06	6.07E+06	1.58309	1.22565	0.77421
25	6.07E+06	7.79E+06	1.38198	0.99244	0.71813
26	7.79E+06	1.00E+07	1.30806	0.87574	0.66950
27	1.00E+07	1.20E+07	1.40930	0.85001	0.60314
28	1.20E+07	1.35E+07	1.56053	0.97698	0.62605
29	1.35E+07	1.50E+07	1.57297	0.96980	0.61654
30	1.50E+07	1.70E+07	1.58238	0.97801	0.61806

Table 6: Group-to-group scattering cross sections for ^{14}N

From	To	σ_{s0}	From	To	σ_{s0}	From	To	σ_{s0}
1	1	9.9114	13	12	1.1107	23	22	0.7060
2	1	1.3589	13	13	6.4252	23	23	0.5899
2	2	8.5524	14	13	0.8962	24	22	0.0104
3	2	1.3561	14	14	4.9709	24	23	0.2944
3	3	8.5551	15	14	0.6487	24	24	0.9209
4	3	1.3668	15	15	3.6446	25	24	0.4062
4	4	8.5440	16	15	0.9210	25	25	0.5862
5	4	1.3611	16	16	2.4278	26	24	0.0067
5	5	8.5488	17	16	0.7482	26	25	0.3115
6	5	1.3623	17	17	2.1382	26	26	0.5576
6	6	8.5452	18	17	0.5320	27	25	0.0096
7	6	1.3613	18	18	1.4864	27	26	0.3279
7	7	8.5390	19	18	0.4040	27	27	0.5125
8	7	1.3584	19	19	1.3401	28	26	0.0584
8	8	8.5261	20	19	1.0221	28	27	0.3255
9	8	1.3544	20	20	1.0506	28	28	0.5931
9	9	8.4775	21	19	0.0292	29	27	0.0932
10	9	1.3427	21	20	0.8764	29	28	0.2584
10	10	8.3810	21	21	0.7597	29	29	0.6182
11	10	1.3209	22	20	0.0106	30	27	0.0077
11	11	8.1186	22	21	0.7541	30	28	0.0608
12	11	1.2499	22	22	0.5287	30	29	0.2122
12	12	7.5164	23	21	0.0243	30	30	0.6973

4.3 Details of `geometry.cc`

This function creates indexed data structures related to the physical cells of the problem: cell areas and volumes, edge lengths, and cell centers. Here we also utilize the processed nuclear data from `xsstore` to create cell-indexed quantities for macroscopic cross sections, employing the relevant densities from the NRL-MSISE atmospheric model introduced in Chapter 1. For the results shown here, we compose the atmosphere only of 78% nitrogen and 22% oxygen (specifically ^{14}N and ^{16}O). These values can be changed in the code, and other relevant elements can be added

as required. It is also possible to change the mixing fraction of gases as a function of altitude. The scattering fraction c , the ratio of scattering to total cross section, as a function of energy is shown in Table 7. The typical geometry used in ATMOTRAN is an atmosphere that extends from a round Earth surface 6378 km in radius, up to 1000 km altitude. The Earth is typically set to be a perfect absorber as a simplification for problems of interest, but this is not a requirement. Note that in Chapter 5 we show verification results of correct code performance for reflecting surfaces and for spheres with no central hollow. For the present atmospheric demonstration we employ an orthogonal grid that is binned logarithmically in altitude, following the density profile of the atmosphere. In Table 8 we show the number of mean free paths in the radial direction using a 30 cell logarithmic spacing approach, as a function of altitude, for four energies. With this spatial cell size, we have from roughly 1-10 mean free paths per cell, with some energy dependence related to the higher cross section at lower energies. The high values in the first altitude bin are artificial and reflect the absorbing Earth layer. It is evident that above 30 km altitude, as the density of the atmosphere drops off, the probability of interaction is tiny. If we have an identical number of cells in the radial and the polar dimensions, the polar dimension is roughly 20 times larger than the radial dimension, in this case around 20,000 km. Generally, more bins should be used in the polar dimension to avoid negative fluxes, depending on the application of interest. In the polar dimension, we typically employ equidistant bins but this is not a requirement of ATMOTRAN, and for point sources a nonlinear binning that is finer near the source could be useful.

Table 7: Composite atmosphere scattering fraction as a function of bin for the AT-MOTRAN energy range

Bin	Energy low edge (eV)	Energy high edge (eV)	c
1	1.39E-04	1.52E-01	0.6038
2	1.52E-01	4.14E-01	0.9467
3	4.14E-01	1.13E+00	0.9670
4	1.13E+00	3.06E+00	0.9797
5	3.06E+00	8.32E+00	0.9876
6	8.32E+00	2.26E+01	0.9924
7	2.26E+01	6.14E+01	0.9954
8	6.14E+01	1.67E+02	0.9972
9	1.67E+02	4.54E+02	0.9983
10	4.54E+02	1.24E+03	0.9990
11	1.24E+03	3.35E+03	0.9994
12	3.35E+03	9.12E+03	0.9996
13	9.12E+03	2.48E+04	0.9997
14	2.48E+04	6.76E+04	0.9998
15	6.76E+04	1.84E+05	0.9997
16	1.84E+05	3.03E+05	0.9995
17	3.03E+05	5.00E+05	0.9957
18	5.00E+05	8.23E+05	0.9802
19	8.23E+05	1.35E+06	0.9910
20	1.35E+06	1.74E+06	0.9644
21	1.74E+06	2.23E+06	0.9512
22	2.23E+06	2.87E+06	0.9086
23	2.87E+06	3.68E+06	0.8485
24	3.68E+06	6.07E+06	0.8125
25	6.07E+06	7.79E+06	0.7239
26	7.79E+06	1.00E+07	0.6489
27	1.00E+07	1.20E+07	0.6029
28	1.20E+07	1.35E+07	0.6172
29	1.35E+07	1.50E+07	0.6148
30	1.50E+07	1.70E+07	0.6150

Table 8: Number of mean free paths in the radial dimension for a 30 cell discretization for several energies

Bin	Alt. (km)	N_λ 1 MeV	N_λ 2 MeV	N_λ 5 MeV	N_λ 10 MeV
0	1.025	1.0E+06	1.0E+06	1.0E+06	1.0E+06
1	2.335	4.84238	3.80322	3.44626	3.06808
2	2.97	5.57772	4.38076	3.9696	3.53398
3	3.745	6.25238	4.91064	4.44974	3.96144
4	4.695	6.99285	5.49221	4.97673	4.43059
5	5.87	7.65853	6.01503	5.45048	4.85236
6	7.315	8.0623	6.33216	5.73784	5.10818
7	9.09	8.28415	6.5064	5.89573	5.24875
8	11.275	8.10688	6.36717	5.76956	5.13643
9	13.965	7.22302	5.67298	5.14053	4.57642
10	17.275	5.42376	4.25984	3.86002	3.43644
11	21.345	3.39408	2.66572	2.41552	2.15045
12	26.35	1.87202	1.47029	1.33229	1.18609
13	32.51	0.925414	0.726823	0.658606	0.586332
14	40.09	0.399263	0.313582	0.28415	0.252969
15	49.41	0.14988	0.117716	0.106668	0.0949622
16	60.88	0.0483454	0.0379706	0.0344068	0.0306311
17	74.995	0.0113103	0.00888313	0.00804938	0.00716607
18	92.355	0.00135574	0.0010648	0.000964863	0.000858982
19	113.71	5.43299E-05	4.26709E-05	3.86659E-05	3.44228E-05
20	139.985	1.96399E-06	1.54252E-06	1.39774E-06	1.24436E-06
21	172.315	3.49795E-07	2.7473E-07	2.48944E-07	2.21626E-07
22	212.085	1.05798E-07	8.30942E-08	7.52952E-08	6.70326E-08
23	261.01	3.54621E-08	2.78521E-08	2.52379E-08	2.24684E-08
24	321.205	1.15991E-08	9.11001E-09	8.25497E-09	7.34909E-09
25	395.265	3.4758E-09	2.72991E-09	2.47368E-09	2.20223E-09
26	486.375	8.99903E-10	7.06786E-10	6.40449E-10	5.70168E-10
27	598.46	1.92573E-10	1.51247E-10	1.37051E-10	1.22012E-10
28	736.36	3.63868E-11	2.85783E-11	2.5896E-11	2.30543E-11
29	906.015	8.937E-12	7.01915E-12	6.36035E-12	5.66239E-12

4.4 Details of quadrature.cc

This function calculates the roots and weights for the Gauss-Chebyshev-Legendre quadrature using standard definitions [51]. Associated values are also calculated and stored into data structures for later retrieval, including the angular redistribution and weighted diamond difference coefficients. The following tables show relevant data for an S_4 angular approximation. Table 9 shows the quadrature roots and weights. The angular redistribution recursion coefficients α and β are tabulated in Tables 10 and 11, where for η we only need to show half the values, since those for $m=3,4$ are mirror images of $m=2,1$ respectively. The weighted diamond coefficients are shown in Tables 12 and 13, where there is a simplification since λ values, by definition, are the same for every μ level. The two-dimensional zero-weighted starting directions are calculated and shown in Table 14.

Table 9: S_4 Two-dimensional Gauss-Chebyshev-Legendre quadrature roots and weights

μ	η	\mathbf{w}
-0.861136312	-0.469676451	0.273204557
-0.861136312	-0.194546356	0.273204557
-0.339981044	-0.868846143	0.512193607
-0.339981044	-0.359887856	0.512193607
-0.861136312	0.469676451	0.273204557
-0.861136312	0.194546356	0.273204557
-0.339981044	0.868846143	0.512193607
-0.339981044	0.359887856	0.512193607
0.861136312	-0.469676451	0.273204557
0.861136312	-0.194546356	0.273204557
0.339981044	-0.868846143	0.512193607
0.339981044	-0.359887856	0.512193607
0.861136312	0.469676451	0.273204557
0.861136312	0.194546356	0.273204557
0.339981044	0.868846143	0.512193607
0.339981044	0.359887856	0.512193607

Table 10: S_4 Angular redistribution coefficients in the μ direction

m	μ	$\alpha_{m-1/2}$	$\alpha_{m+1/2}$
1	-8.6113631159E-01	0.0	4.7053272819E-01
2	-3.3998104358E-01	4.7053272819E-01	8.1880496217E-01
3	3.3998104358E-01	8.1880496217E-01	4.7053272819E-01
4	8.6113631159E-01	4.7053272819E-01	0.0

Table 11: S_4 Angular redistribution coefficients in the η direction

m	ℓ	η	$\beta_{m,\ell-1/2}$	$\beta_{m,\ell+1/2}$
1	1	-4.6967645066E-01	0.0	-1.2831774640E-01
1	2	-1.9454635579E-01	-1.2831774640E-01	-1.8146869725E-01
1	3	1.9454635579E-01	-1.8146869725E-01	-1.2831774640E-01
1	4	4.6967645066E-01	-1.2831774640E-01	0.0
2	1	-8.6884614343E-01	0.0	-4.4501744004E-01
2	2	-3.5988785622E-01	-4.4501744004E-01	-6.2934969920E-01
2	3	3.5988785622E-01	-6.2934969920E-01	-4.4501744004E-01
2	4	8.6884614343E-01	-4.4501744004E-01	0.0

Table 12: S_4 Weighted-diamond angular differencing coefficients τ_m

m	τ_m
1	3.9920009839E-01
2	4.7867274479E-01
3	5.2132725521E-01
4	6.0079990161E-01

Table 13: S_4 Weighted-diamond angular differencing coefficients λ_ℓ

ℓ	λ_ℓ
1	2.5989153247E-01
2	4.5880389985E-01
3	5.4119610015E-01
4	7.4010846753E-01

4.5 Details of solver.cc

This routine essentially sets up the problem including creating data structures for the calculation and aligning the iterative procedure. The matrices related to integration of the basis functions over the cell volumes are performed here for all cells and then

Table 14: S₄ Starting directions

μ	η
-1.0000000000e+00	0.0
-8.6113631159E-01	-5.0837412685E-01
-3.3998104358E-01	-9.4043228890E-01
3.3998104358E-01	-9.4043228890E-01
8.6113631159E-01	-5.0837412685E-01
-8.6113631159E-01	5.0837412685E-01
-3.3998104358E-01	9.4043228890E-01
3.3998104358E-01	9.4043228890E-01
8.6113631159E-01	5.0837412685E-01
1.0000000000e+00	0.0

stored in a structure to be accessed later within the spatial sweep. Depending on the angular order of the calculation, starting direction and weighted direction ordinates are set up and the ordering through the angular mesh is established and executed. The internal source is set up for all cells and angles. All these data structures are made available to the sweep routine specifying an angular direction set and a flag for whether the inner spatial sweep is to solve in the upper or lower physical hemisphere of the problem. As detailed earlier, this flag affects sweep direction. When the inner iteration returns from the sweep routine, the convergence criterion C is calculated, summing scalar flux values over all indices (v: cell corner, i: radial cell index, j: polar cell index, e: energy index, s: scattering order index) and evaluating the difference between successive iterations, scaled by the total number of cells N in the problem if the following way

$$C = \frac{1}{N} \sum_{v,i,j,e,s} \left(\phi_n - \phi_{n-1} \right) . \quad (174)$$

The iteration continues until C is smaller than some threshold value set as an input parameter to the calculation (e.g. $C_{thresh} = 1 \times 10^{-9}$). This routine also calculates quantities of interest such as particle balance and error calculations for manufactured solutions.

4.6 Details of sweep.cc

This routine, given an angle and a flag for upper/lower hemisphere, sets up the spatial cell range and direction for the problem. We introduced the spatio-angular mesh sweep for two-dimensional spheres in Chapter 2 and detailed the progression used in TWOTRAN SPHERE. The progression is slightly different with ATMOTRAN for two reasons: (1) ATMOTRAN uses zero-weighted starting directions (rather than the step-approximation-generated initialization directions in TWOTRAN SPHERE) so the information flow must include these, and (2) ATMOTRAN uses a discontinuous spatial discretization which benefits from non-zero upwind fluxes, and so we find starting the spatial sweep at the center of the sphere ($\theta = \pi/2$), where all cell surfaces have non-zero area, gives better results than starting at either extreme end of the θ range, as was done using spatial diamond differencing in TWOTRAN SPHERE.

For the (μ, r) sweep, we start with the most negative angle (a starting direction of $\mu = -1$) and then sweep the angular mesh for increasing μ values, following paths A-B in Figure 11. For the (η, θ) sweep we find the following progression is necessary to produce the correct results. For full spheres, we start at a spatial cell with $\theta_B = \pi/2$ and start at a maximum positive η value, sweeping in angle to decreasing yet positive values of η and up to $\theta = \pi$ (note, the μ sweep is always from most negative to most positive and the radial sweep from the outer radius inward as usual for one-dimensional codes). Then we return from $\theta = \pi$ to $\theta = \pi/2$ using negative values of η until we reach the most negative value. Then, we start with the cell that has $\theta_T = \pi/2$ sweep down to $\theta = 0$ using the negative values from most negative to closest to zero. Finally we sweep from $\theta = 0$ back to $\theta = \pi/2$ starting the smallest positive value of eta and working up to the largest positive value. This description follows successively paths C-D-E-F in Figure 11. With this procedure of following the characteristic paths we can obtain solutions homogeneous in polar angle when this is

expected.

The above procedure requires that we check what hemisphere we are in, and properly upwind the spatio-angular solutions across the $\theta = \pi/2$ boundary. This requires that we are able to sweep the angular mesh for η in either direction, accounting for the direction of flow through the angular mesh, and this is why we derived transport equations for both flows in Chapter 3. The nesting of sweeps in ATMOTRAN is the same as other multigroup transport codes, so that the outer iteration is over energy, from highest to lowest energy in the problem, then we proceed in the inner iteration through the angular mesh and then the spatial mesh. Diagrams of information flow in the S_6 angular mesh are shown for both hemispheres below, Figure 19 shows the case of $\theta < \pi/2$ and Figure 20 shows $\theta > \pi/2$. Note that this procedure implies that an even number of cells must be used in the θ direction. The figures are grid-plotted by (m, ℓ) index, not by (μ, η) value. Starting directions are shown as triangles and are indexed with letter from a to n and weighted directions are filled dots indexed with numbers from 1 to 36. Information flow is along the small arrows, meaning that the calculation of a particular direction requires information from all arrows flowing into it.

Some comments are appropriate for clarity of this rather complicated sweep. We can see that starting direction index a (representing $\mu = -1$) feeds into all fluxes on the next most negative μ -level, both starting and weighted. Starting directions in η , appearing along the horizontal, can feed into the weighted direction calculation (indices b through g), since there is angular redistribution to consider for the weighted direction, but the weighted direction fluxes never feed into starting direction fluxes. This is consistent with our derivation in Chapter 3 where we found that there is no angular redistribution in the η dimension for any of the starting fluxes (indices b through m). To be clear, these η starting direction fluxes (indices h through m) that

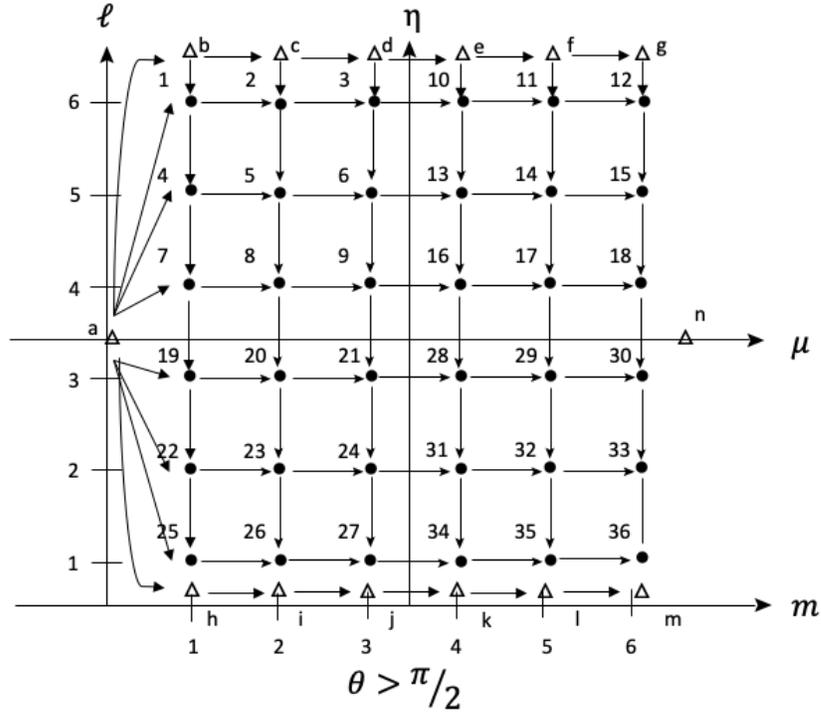


Figure 20: ATMOTRAN: sweeping the angular mesh for large polar angle.

Table 16: Spatial extents of the polar sweep

η	$(\theta > \pi/2)?1 : 0$	j_{start}	j_{end}	$j_{direction}$
< 0	1	J	$J/2$	-
< 0	0	$J/2 - 1$	$J/2$	-
≥ 0	1	$J/2$	J	+
≥ 0	0	0	$J/2 - 1$	+

Eigen package that performs LU decomposition with pivoting.

4.7 Details of `utils.cc`

This function defines a utility class with methods required for repeated calculations in the solver and sweep functions. The methods are described in Table 17.

Table 17: Method and description table for `solver_utils.cc`

Method	Description
GetIntegral	Calculate the individual terms that go into the BLD matrices for a specified computational cell
GetMatrices	Build the BLD matrix terms (V, L, M, T, P, Q) for a specified computational cell
GetPhiQloc	Retrieve the previous iteration scalar flux and current source for a specified computational cell
GetCellAvgFlux	Calculate the cell-averaged flux for a spherical segment in the physical geometry given the angular flux calculated with the computational cell
GetAngleMatch	Find the appropriate flux for unwinding the angular mesh
GetAlphaBeta	Retrieve the angular redistribution coefficients for this angle index, given the cell location and status of whether or not the angle is a zero-weighted starting direction
GetWDD	Retrieve the weighted diamond difference angular coefficients τ and λ

5 ATMOTRAN Verification

The required mathematical derivations and code implementation details of ATMOTRAN have been presented in Chapters 3 and 4. In the following chapter we discuss the correctness of the code implementation and the validity of the fundamental assumptions that went into the derivations. First we verify that the discretization scheme does preserve the thick-diffusion limit. Then we proceed with numerical experiments to verify code performance with results from several manufactured solutions, code-to-code comparisons with another transport code, comparison of multigroup performance with Monte Carlo results, and some individual examples of interesting cases.

5.1 Thick-diffusion limit

Since the discretization method is new for this coordinate system, and given that the intended application is an optically thick problem at least in some regions, it is useful to investigate performance in the thick-diffusion limit. It has been established that the diffusion equation is an asymptotic limit of the transport equation [52]. The diffusion limit is analytically established by scaling the physical properties of the transport equation such that it becomes increasingly optically thick and scattering dominated. This is achieved by introducing a parameter ϵ to scale the cross sections, source, and mesh size. The thick-diffusion limit is evaluated by keeping the mesh size

fixed while scaling the cross sections and source such that the mesh cells approach an infinite optical thickness. Schemes that preserve this limit are accurate so long as the cell size is small compared to the diffusion length, which is a measure of how far neutrons will diffuse before absorption and is quite large for highly diffusive problems. Schemes that do not preserve this limit are only accurate when the cell size is small compared to a mean free path, which is typically much smaller than the diffusion length. If a scheme does not preserve this limit, more cells will need to be used in order to get accurate solutions.

To demonstrate that the present scheme does preserve the thick-diffusion limit we show here numerical results evaluated when $\epsilon \rightarrow 0$. Selecting a problem with a constant isotropic source with vacuum boundary conditions, we plot a numerical S_8 solution for $r \in [1, 2]$ and a 10 x 10 spatial mesh, with a convergence tolerance of 10^{-8} . Starting with cross section values $\sigma_{t0} = 1.0$ and $\sigma_{a0} = 0.1$ we successively show, the numerical solution for values

$$\sigma_t = \frac{\sigma_{t0}}{\epsilon}, \quad \sigma_a = \epsilon\sigma_{a0}, \quad q(r, \theta) = \epsilon \quad . \quad (175)$$

where $\epsilon = 2^{-k}$ for $k = 0, 2, 4, 6, 8, 10$. As we asymptotically approach the thick-diffusion limit the solution converges on a non-zero solution, as shown in Figure 21, demonstrating that the discretization scheme preserves the thick-diffusion limit.

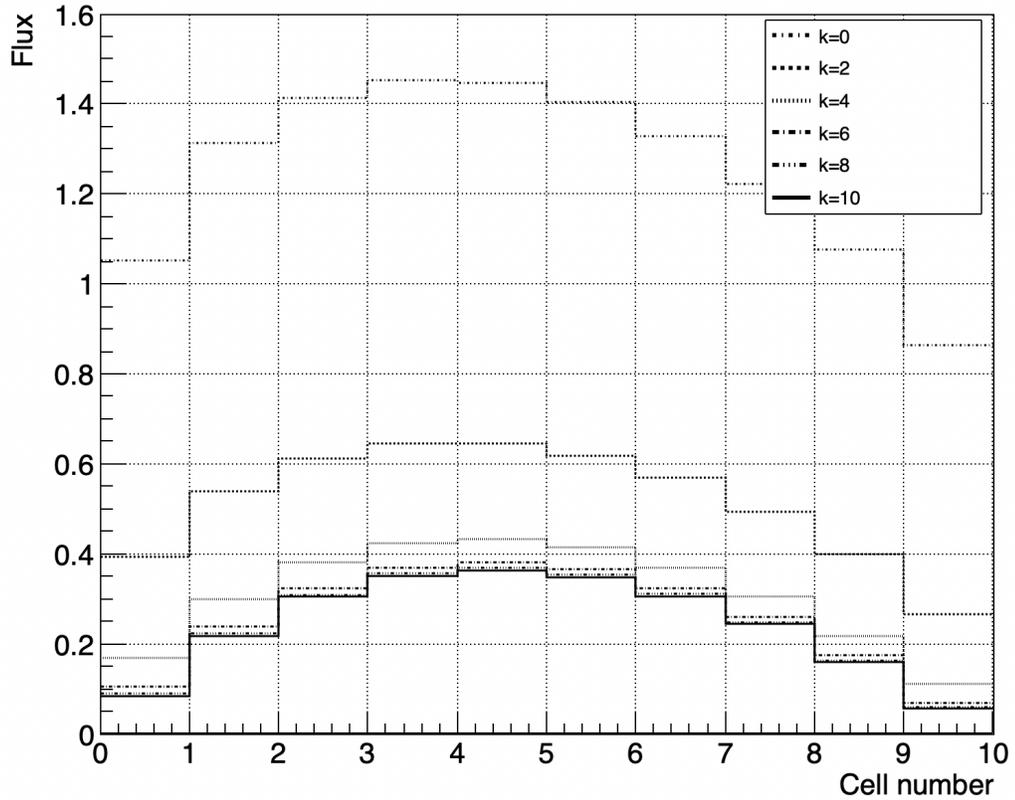


Figure 21: Numerical result for 10x10 problem approaching the thick diffusion limit.

5.2 Manufactured solution verification

Manufactured solutions may be used to test the properties of transport discretization schemes in the common event that analytic S_N solutions do not exist [53]. The general process is to specify some angular flux with properties that can test relevant aspects of the code. Since the present code allows for flux variation in two dimensions the specified flux should depend on the spatial variables in order to test the spatial discretization scheme. Using the transport equation, this flux is used to calculate a spatially discrete source term that corresponds to the flux, and numerical solutions computed with this flux are computed with increasingly refined meshes in order to

generate an angular flux for comparison with the manufactured solution. A correctly implemented discretization scheme will show that the error between the numerical and the manufactured solutions will decrease as the mesh cell size decreases, and the rate of decrease will be proportional to the reduction in cell size, in which case the numerical method is convergent.

We present verification of the methods described above with a manufactured solution. In this section we show results for four manufactured solutions $\Psi_M(r, \theta)$ on a annular shell test problem geometry. The hollow inner spherical void is a boundary source for these problems. The four manufactured solutions are

1. a biquadratic isotropic flux (quadratic in both radial and polar dimensions) of the form $\Psi_M(r, \theta) = \theta(\pi - \theta)r(2 - r)$,
2. a bilinear isotropic flux (linear in both the radial and polar dimensions) of the form $\Psi_M(r, \theta) = (\pi - \theta)(20 - r)$,
3. a biquadratic flux anisotropic in μ of the form $\Psi_M(r, \theta) = (1 + 3\mu^2)\theta(\pi - \theta)r(2 - r)$, and
4. a biquadratic flux anisotropic in η of the form $\Psi_M(r, \theta) = (1 + 3\eta^2)\theta(\pi - \theta)r(2 - r)$.

The numerically calculated scalar flux solution, ϕ , is compared to the scalar flux corresponding to the manufactured solution, $\Phi_M = \int d\Omega \Psi_M$, using a discrete L_2 -norm of the difference in the cell-average scalar fluxes, that is,

$$\delta = \left[\frac{1}{N_x N_y} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \left(\Phi_{M(i,j)} - \phi_{i,j} \right)^2 \right]^{1/2}, \quad (176)$$

where N_x and N_y are the number of cells in the radial and polar directions, respectively.

5.2.1 Biquadratic isotropic manufactured solution

Since our basis functions are linear in (r, θ) the code will not be able to exactly model higher order manufactured solutions, for example quadratic solutions. However, the bilinear code should converge to the quadratic solution with mesh refinement since geometrically shorter polynomial elements are more accurately represented by line elements. This convergence property is confirmed by this verification. To begin, we specify an angular flux, quadratic in both radius and polar angle, of the form

$$\Psi_M(r, \theta) = \theta(\pi - \theta)r(2 - r), \quad r \in [1, 2]\text{cm}, \theta \in [0, \pi] \quad , \quad (177)$$

which gives a scalar flux

$$\Phi_M(r, \theta) = \int d\Omega \Psi_M(r, \theta) = 4\pi\theta(\pi - \theta)r(2 - r) \quad . \quad (178)$$

Substituting these fluxes into the transport equation we can derive a continuous source

$$\begin{aligned} S(r, \theta, \mu, \eta) &= 2\eta\pi + r(-\pi\eta) + \theta(2\pi\mu - 4\eta) + r\theta(-2\pi\mu + 2\eta + 2\pi(\sigma_t - \sigma_s)) \\ &+ \theta^2(-2\mu) + r\theta^2(2\mu - 2(\sigma_t - \sigma_s)) + r^2\theta(-\pi(\sigma_t - \sigma_s)) \\ &+ r^2\theta^2(\sigma_t - \sigma_s) \quad . \end{aligned} \quad (179)$$

From this expression, the spatially discrete isotropic source q is evaluated at each of the four cell corners by the following integration for a single mesh cell, as

$$q_k = \sum_{i=1}^4 \int dV S B_i \quad . \quad (180)$$

Because this source is nonzero at the leftmost physical radial boundary, ($r_L = 1$ cm), we must evaluate the source term at the boundary conditions. The source needs to be integrated over the boundary surface and subtracted from q at that face. These

terms are, for the $r_L = 1$ cm face

$$q_{1L} = \Omega \cdot \hat{n}_L \int_{\theta_B}^{\theta_T} \Psi_{M,L}(r_L, \theta) B_1(r_L, \theta) r_L^2 \sin \theta d\theta \quad , \quad (181)$$

and

$$q_{4L} = \Omega \cdot \hat{n}_L \int_{\theta_B}^{\theta_T} \Psi_M(r_L, \theta) B_4(r_L, \theta) r_L^2 \sin \theta d\theta \quad , \quad (182)$$

where $\Psi_M(r_L, \theta) = \theta(\pi - \theta)$. Note that the various integrals used in these evaluations are included in Appendix A for reference. Proper implementation of the code will demonstrate that as the spatial mesh is refined, the solutions will converge at a predictable rate. We refine the spatial mesh by solving the problem successively for increasing values of k , where $N_x = N_y = 2^k$. The iterative convergence tolerance is 10^{-9} , the angular approximation is S_8 , $\Sigma_t = 2 \text{ cm}^{-1}$ and $c = 0.99$. Examining the ratio of successive errors, one can estimate the order of accuracy by calculating $\log_2(\delta_{k-1}/\delta_k)$, as seen in Table 18. These results indicate the method is better than second order order accurate.

Table 18: Manufactured solution verification for isotropic biquadratic flux

k	N_x	N_y	$N_x N_y$	δ_k	δ_{k-1}/δ_k
3	8	8	64	2.3868E-02	—
4	16	16	256	5.3373E-03	4.4719
5	32	32	1024	1.0467E-03	5.0994
6	64	64	4096	1.9093E-04	5.4818
7	128	128	16384	3.3646E-05	5.6747
8	256	256	65536	5.8513E-06	5.7503
9	512	512	262144	1.0146E-06	5.7672

A two-dimensional visualization of this solution is shown in Figure 22. The cell-averaged scalar flux (in units cm^{-2}) is shown for each cell. For this plot there are 32 radial cells extending from 1 cm to 2 cm and 32 polar cells from 0 radians to π radians.

The solution from π to 2π radians is a mirror image, no additional calculations are done on that range. Since this is a two-dimensional code the flux is integrated over the azimuthal direction, that extend out to the reader.

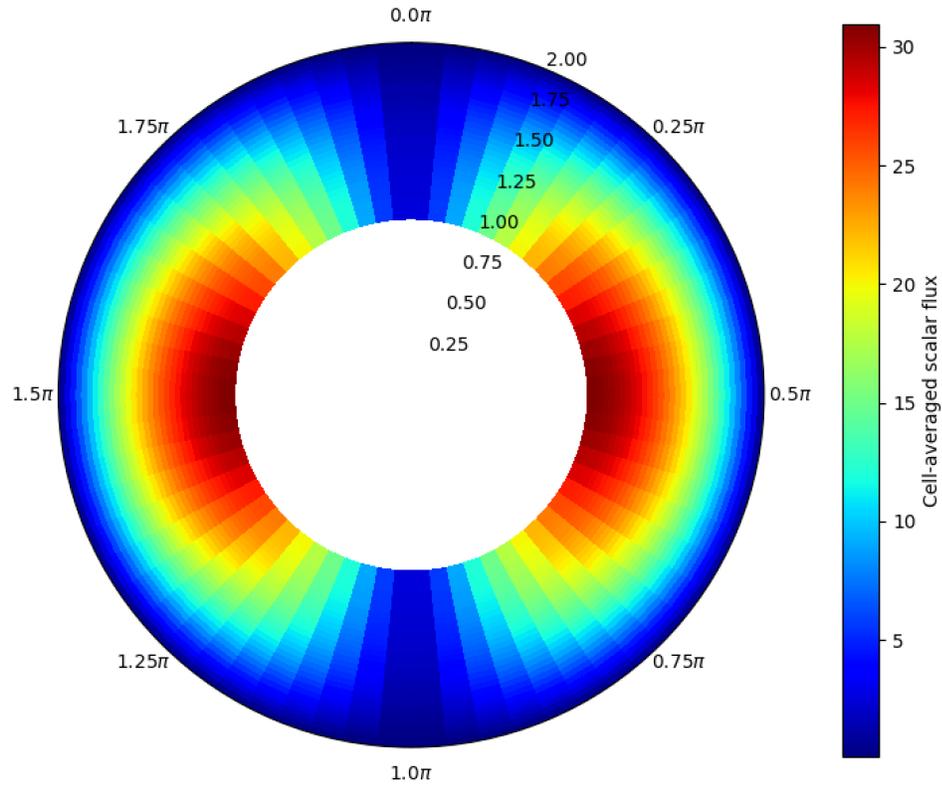


Figure 22: Numerical result for 32 x 32 biquadratic manufactured solution.

5.2.2 Bilinear isotropic manufactured solution

The manufactured solution is bilinear for this problem. Because the finite element basis is also bilinear, the code should reproduce the posited flux to the level of the iterative convergence tolerance of the numerical solution. We specify an angular flux, linear in radius and polar angle, of the form

$$\Psi_M(r, \theta) = (\pi - \theta)(20 - r) \quad r \in [10, 20] \quad \text{cm}, \quad \theta \in [0, \pi] \quad , \quad (183)$$

which gives a scalar flux

$$\Phi_M(r, \theta) = \int d\Omega \Psi_M(r, \theta) = 4\pi(\pi - \theta)(20 - r) \quad . \quad (184)$$

Again, from the transport equation we can derive the continuous source

$$\begin{aligned} S(r, \theta, \mu, \eta) = & (-\mu\pi + \eta + (\Sigma_t - \Sigma_s)) + \theta(\mu - 20(\Sigma_t - \Sigma_s)) \\ & - \frac{20\eta}{r} - r(\pi(\Sigma_t - \Sigma_s)) + r\theta(\Sigma_t - \Sigma_s) \quad . \end{aligned} \quad (185)$$

We find that just as in the biquadratic case, this source is nontrivial at the leftmost physical radial boundary, ($r_L = 10$ cm), and the bottommost polar boundary ($\theta_B = 0$) and we again must evaluate the source term at the boundary conditions. The source needs to be integrated over the boundary surface and subtracted from q at that face. These terms for the $r_L = 10$ cm face are the same form as for the biquadratic case (Equations 181 and 182) with $\Psi_M(r_L, \theta) = 10(\pi - \theta)$. For the bottom face, we have

$$q_{1B} = \Omega \cdot \hat{n}_B \int_{r_L}^{r_R} \Psi_M(r, \theta_B) B_1(r, \theta_B) r \sin \theta_B dr \quad , \quad (186)$$

and

$$q_{2B} = \Omega \cdot \hat{n}_B \int_{r_L}^{r_R} \Psi_M(r, \theta_B) B_2(r, \theta_B) r \sin \theta_B dr \quad . \quad (187)$$

Note that these terms are zero for full spheres since $\sin \theta_B = 0$ for this cell. The physical interpretation is that since there is no surface area at the extreme ends in the polar dimension as defined in our coordinate system, there can be no leakage across that face. Results were computed on the same spatial domain as before, with

S_8 , $\Sigma_t = 2.0$ and $c = 0.99$, and $N_x = N_y = 16$, and δ was calculated for a series of iterative convergence tolerance. The results in Table 19 confirm that the numerical method reproduced the bilinear solution as expected. A two-dimensional visualization of this solution is shown in Figure 23 with the same setup as the image shown for the previous biquadratic plot.

Table 19: Manufactured solution verification for bilinear flux

tol.	δ
1E-06	3.286E-06
1E-08	3.588E-08
1E-10	3.507E-10
1E-12	2.378E-11

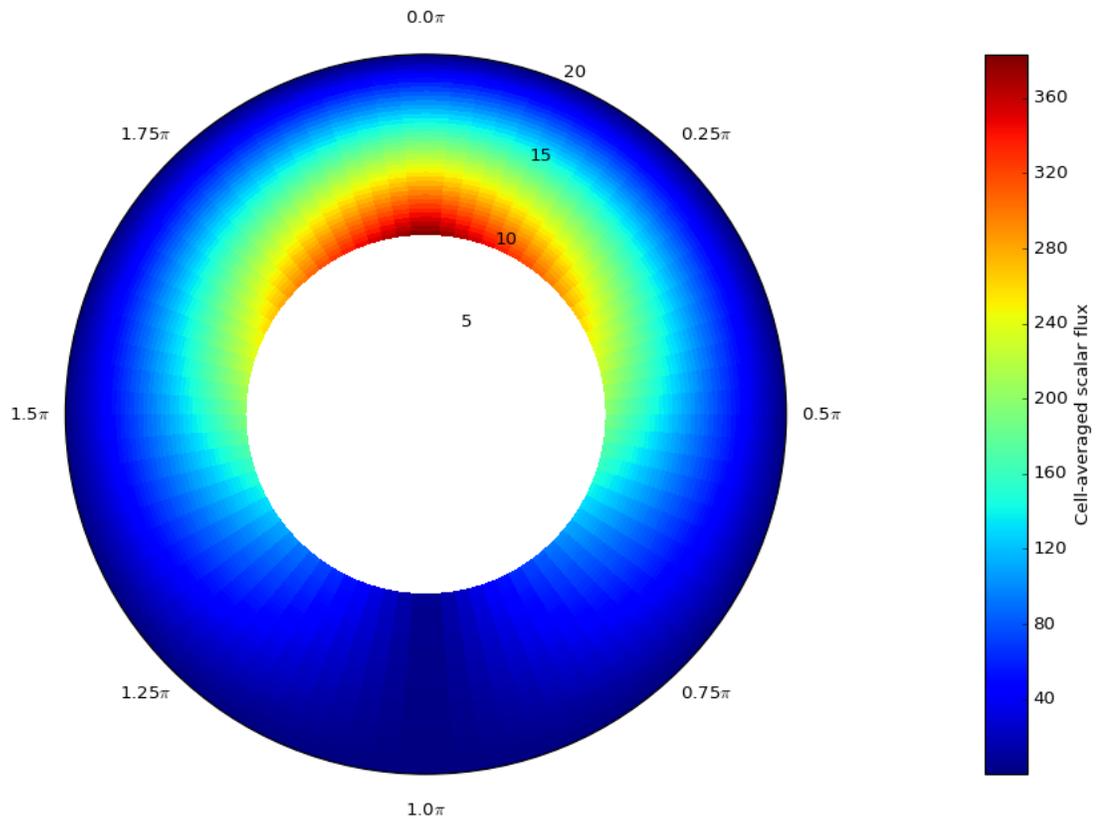


Figure 23: Numerical result for bilinear manufactured solution, represented with 64 x 64 bins.

5.2.3 Biquadratic anisotropic manufactured source

To investigate the accuracy of the weighted diamond difference angular approximation we can use a manufactured source corresponding to an anisotropic flux of the form

$$\Psi_M(r, \theta, \mu) = (1 + 3\mu^2)\theta(\pi - \theta)r(2 - r) \quad r \in [1, 2] \quad \text{cm}, \quad \theta \in [0, \pi] \quad . \quad (188)$$

This flux, similar to the isotropic biquadratic flux, has a boundary source on the inner radial edge, although for this test the source is anisotropic. This angular flux corresponds to a scalar flux

$$\Phi_M(r, \theta) = \int d\Omega \Psi_M(r, \theta, \mu) = 8\pi\theta(\pi - \theta)r(2 - r) \quad . \quad (189)$$

We can then derive the manufactured source

$$\begin{aligned} S(r, \theta, \mu, \eta) = & 2\pi(\eta + 3\mu^2\eta) + r(-\pi(\eta + 3\mu^2\eta)) + \theta(14\pi\mu - 6\pi\mu^3 - 4\eta - 12\mu^2\eta) \\ & + r\theta(-8\pi\mu + 2\eta + 6\mu^2\eta + 2\pi(\sigma_t(1 + 3\mu^2) - 2\sigma_s)) \\ & + \theta^2(6\mu^3 - 14\mu) + r\theta^2(8\mu - 2(\sigma_t(1 + 3\mu^2) - 2\sigma_s)) \\ & + r^2\theta(-\pi(\sigma_t(1 + 3\mu^2) - 2\sigma_s)) + r^2\theta^2(\sigma_t(1 + 3\mu^2) - 2\sigma_s) \quad . \end{aligned} \quad (190)$$

The anisotropic boundary source on the $r_L = 1.0$ edge is, at the corners,

$$q_{1L} = \Omega \cdot \hat{n}_L(1 + 3\mu^2) \int_{\theta_B}^{\theta_T} \Psi_{M,L}(r_L, \theta) B_1(r_L, \theta) r_L^2 \sin \theta d\theta \quad , \quad (191)$$

and

$$q_{4L} = \Omega \cdot \hat{n}_L(1 + 3\mu^2) \int_{\theta_B}^{\theta_T} \Psi_M(r_L, \theta) B_4(r_L, \theta) r_L^2 \sin \theta d\theta \quad . \quad (192)$$

To solve this numerically we use $\sigma_t = 1.0$, $c = 0.1$, and employ a mesh refinement procedure in both angle and space. With multiple levels of mesh refinement it is easy to see when the error in the solution is due to the angular derivative approximation rather than the spatial accuracy. As before, the spatial meshes employ an equal number of cells in the radial and polar dimensions. We show in Table 20 that the error induced by an anisotropic source hinders the order of accuracy of the solution at low angular quadrature. Once we reach S_{96} the errors are on the order of the spatial error, around a second order accuracy.

Table 20: Verification for $1 + 3\mu^2$ anisotropic biquadratic flux

	8x8	16x16	32x32	64x64
S_8	2.3537E-01	2.3440E-01	2.3440E-01	2.3450E-01
S_{16}	7.0436E-02	6.3358E-02	6.2527E-02	6.2415E-02
S_{32}	3.2970E-02	1.7797E-02	1.5978E-02	1.5802E-02
S_{64}	2.8030E-02	8.2490E-03	4.3881E-03	4.0242E-03
S_{96}	2.7558E-02	7.2269E-03	2.4317E-03	6.2308E-04

Similarly, we can investigate the accuracy of the angular discretization scheme in the η direction with an appropriate manufactured source. If we posit an angular flux with the form

$$\Psi_M(r, \theta, \eta) = (1 + 3\eta^2)\theta(\pi - \theta)r(2 - r) \quad r \in [1, 2] \quad \text{cm}, \quad \theta \in [0, \pi] \quad , \quad (193)$$

giving a scalar flux, identical to the previous case

$$\Phi_M(r, \theta) = \int d\Omega \Psi_M(r, \theta, \eta) = 8\pi\theta(\pi - \theta)r(2 - r) \quad . \quad (194)$$

We can then derive the manufactured source

$$\begin{aligned}
S(r, \theta, \mu, \eta) = & 2\pi(\eta + 3\eta^3) + r(-\pi(\eta + 3\eta^3)) \\
& + \theta(2\pi\mu - 6\pi\mu\eta^2 - 4\eta - 12\eta^3) \\
& + r\theta(-2\pi\mu + 2\eta + 6\eta^3 + 2\pi(\sigma_t(1 + 3\eta^2) - 2\sigma_s)) \\
& + \theta^2(-2\mu + 6\mu\eta^2) + r\theta^2(4\mu - 2(\sigma_t(1 + 3\eta^2) - 2\sigma_s)) \\
& + r^2\theta(-\pi(\sigma_t(1 + 3\eta^2) - 2\sigma_s)) + r^2\theta^2(\sigma_t(1 + 3\eta^2) - 2\sigma_s) \\
& + \theta \cot(\theta)(12\eta\xi^2\pi) + \theta^2 \cot(\theta)(-12\eta\xi^2) \\
& + r\theta \cot(\theta)(-6\pi\eta\xi^2) + r\theta^2 \cot(\theta)(6\eta\xi^2) \quad ,
\end{aligned} \tag{195}$$

and the anisotropic boundary source on the $r_L = 1.0$ edge is, at the corners,

$$q_{1L} = \Omega \cdot \hat{n}_L(1 + 3\eta^2) \int_{\theta_B}^{\theta_T} \Psi_{M,L}(r_L, \theta) B_1(r_L, \theta) r_L^2 \sin \theta d\theta \quad , \tag{196}$$

and

$$q_{4L} = \Omega \cdot \hat{n}_L(1 + 3\eta^2) \int_{\theta_B}^{\theta_T} \Psi_M(r_L, \theta) B_4(r_L, \theta) r_L^2 \sin \theta d\theta \quad . \tag{197}$$

We show in Table 21 the combined mesh and angular refinement results for this source.

Table 21: Verification for $1 + 3\eta^2$ anisotropic biquadratic flux

	8x8	16x16	32x32	64x64
S_8	2.0729E-01	1.9524E-01	1.8928E-01	1.8753E-01
S_{16}	7.3558E-02	5.6426E-02	5.1838E-02	5.0434E-02
S_{32}	4.4890E-02	1.9800E-02	1.4606E-02	1.3526E-02
S_{64}	3.8734E-02	1.1276E-02	4.7892E-03	3.6422E-03
S_{96}	3.7683E-02	9.873E-03	3.0598E-03	9.8071E-04

5.3 Code-to-code comparison

Here we present a comparison of results for two problems as calculated by ATMOTRAN and by Capsaicin, a production transport code that can perform transport calculations for one-dimensional spherical geometries using a linear-discontinuous spatial discretization [55]. Problem 1 is a spherical geometry problem with a minimum radius of 4 cm, inside of which is a pure absorber, and a maximum radius of 6.25 cm, outside of which is vacuum. The relevant cross sections are $\sigma_t = 1.0 \text{ cm}^{-1}$ and $c = 0.9$. The radial dimension is split into 18 cells, indexed 0 through 17, and there is no internal source except for cells 8 and 9 which have a source of 2 cm^{-3} . An S_{64} angular discretization is used. For ATMOTRAN we used 10 spatial cells in the polar dimension and the result is equal for all such spherical wedges, to the convergence tolerance of the simulation. For comparison here we select one of the wedges to compare with the one-dimensional Capsaicin result in Table 22.

A similar S_{64} comparison is made for Problem 2, again an annular sphere with a central absorber and vacuum outer boundary, this time with a minimum radius of 10 cm, a maximum radius of 20 cm, split into 10 radial cells indexed 0 through 9, with cross sections $\sigma_t = 1.0 \text{ cm}^{-1}$ and $c = 0.9$, this time with an internal source 1 cm^{-3} in all cells. The comparison is shown in Table 23.

5.4 Inspection of cases of interest

Here we show results for some cases of interest to demonstrate code performance. First we show purely scattering spherical shells with a boundary source on the inner radial boundary and either vacuum or reflector at the outer boundary. We then show a solid sphere and confirm that there is no unphysical dip in the scalar flux at the center of the sphere. Finally we show qualitative performance with a localized

Table 22: ATMOTRAN and Capsaicin code-to-code comparison (Problem 1)

Cell	ATMOTRAN	Capsaicin
0	0.3961226643	0.3961211683
1	0.4983106567	0.4983096455
2	0.5829085590	0.5829083320
3	0.6645138383	0.6645150605
4	0.7492715793	0.7492748502
5	0.8437496236	0.8437547241
6	0.9628213314	0.9628257701
7	1.1523253142	1.1523241506
8	1.3956210210	1.3956145831
9	1.3673339199	1.3673321226
10	1.0794017583	1.0794039559
11	0.8593869421	0.8593873823
12	0.7177477545	0.7177477930
13	0.6083192923	0.6083192277
14	0.5155590534	0.5155589948
15	0.4328732092	0.4328732803
16	0.3558618708	0.3558622104
17	0.2787093301	0.2787097084

Table 23: ATMOTRAN and Capsaicin code-to-code comparison (Problem 2)

Cell	ATMOTRAN	Capsaicin
0	5.0142310580	5.0142310598
1	7.3660355575	7.3660355605
2	8.4390145650	8.4390145687
3	8.9288563723	8.9288563762
4	9.0638032856	9.0638032895
5	8.9314569865	8.9314569903
6	8.5284452286	8.5284452316
7	7.7709942846	7.7709942870
8	6.4764570799	6.4764570817
9	4.1184026202	4.1184026213

point source in a solid sphere and in a spherical shell, and for the latter we also make a comparison of the ATMOTRAN solution to a spatial diamond-differenced discretization.

5.5 Boundary surface sources

The geometry is a spherical shell on $r \in [0.5, 2.0]$ cm, $\theta \in [0, \pi]$, with $c = 1.0$, $\sigma_t = 1.0$ cm⁻¹, and the code uses an S_6 angular approximation. Figure 24 shows the result for a source 1 cm⁻³ at the inner boundary, and with vacuum at the outer boundary. Figure 25 shows the result for the same problem but adding a specular reflection at the outer boundary, resulting as expected in a flat unity scalar flux.

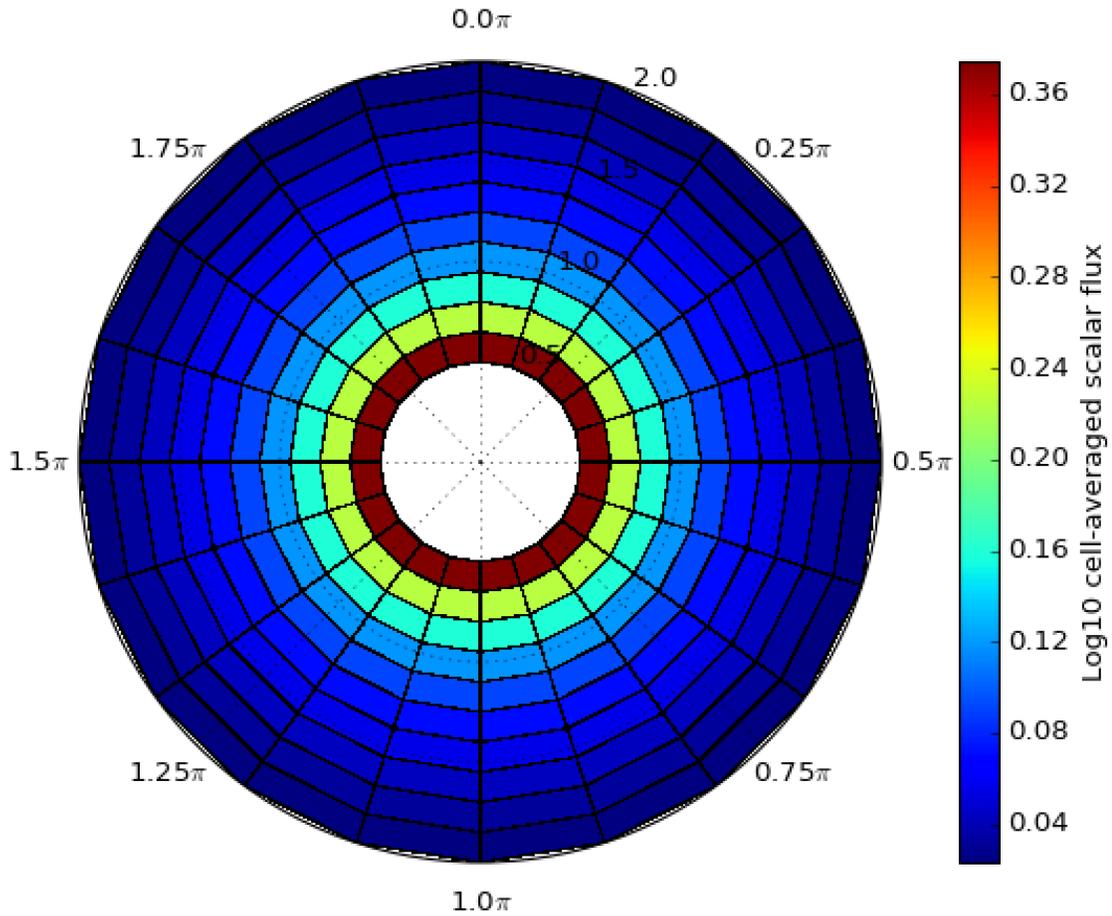


Figure 24: ATMOTRAN result for spherical shell with inner boundary source and vacuum outer boundary.

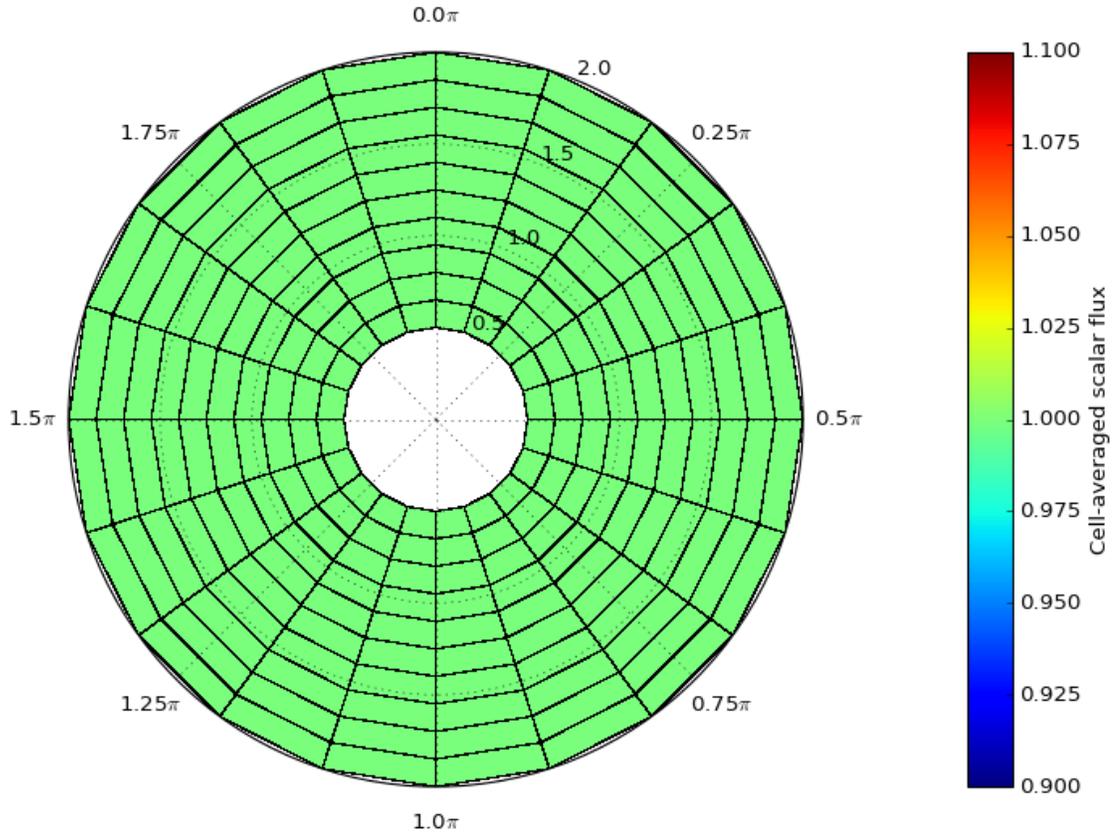


Figure 25: ATMOTRAN result for spherical shell with inner boundary source and reflecting outer boundary.

5.6 Solid spheres

Solid spheres are generally interesting although not needed for the atmospheric transport application. Here we show performance of ATMOTRAN using a solid sphere geometry. Figure 26 shows the solution of a problem with $r \in [0.0, 1.0]$ cm, $\theta \in [0, \pi]$, with $c = 1.0$, $\sigma_t = 1.0 \text{ cm}^{-1}$ and an internal source 1 cm^{-3} , with an S_2 angular approximation. This figure uses the ATMOTRAN weighted diamond difference angular discretization. In Figure 27 we show a one-dimensional representation of the same problem and compare the result with a diamond differenced angular discretization. Both cases used bilinear discontinuous spatial discretization. This plot motivates the

use of weighted diamond angular discretizations and confirms that the unphysical flux dip observed and mitigated in [20] for spatial diamond differencing is also observed for discontinuous spatial differencing.

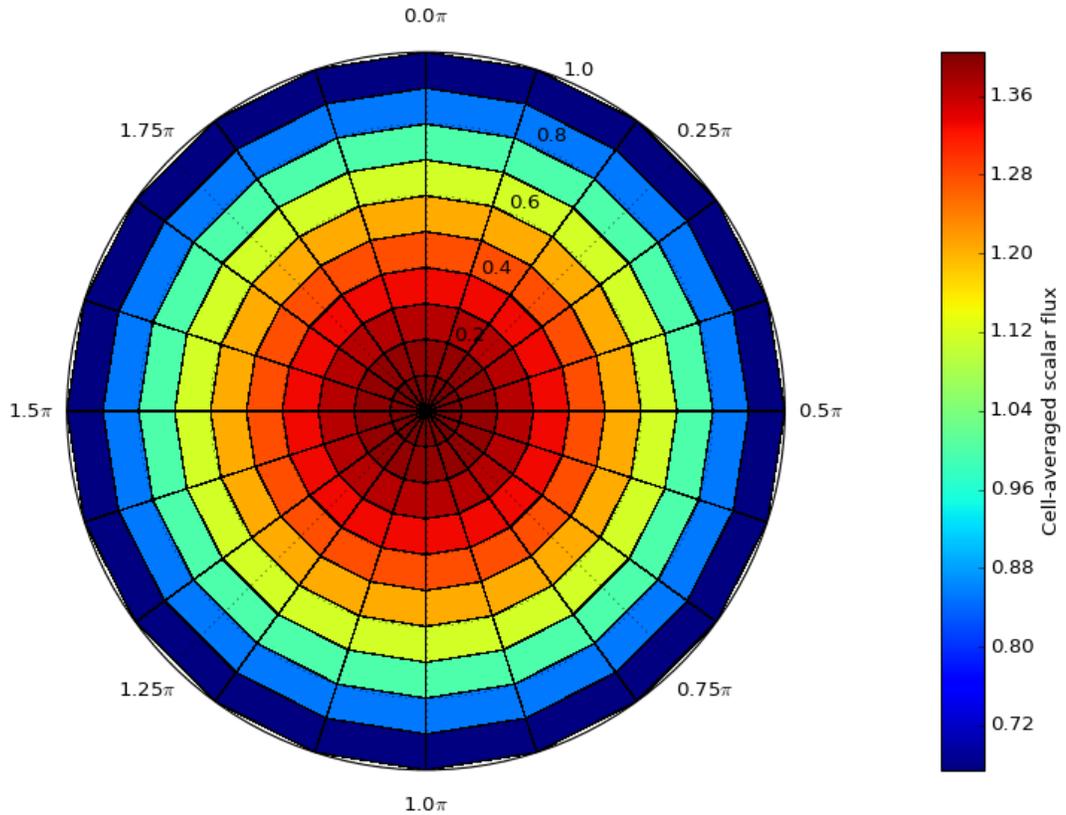


Figure 26: ATMOTRAN result for a pure scattering solid sphere with an internal source.

5.7 Point sources

The two-dimensional spherical-polar geometry allows for point sources within a spherical medium. This is a critical requirement for the atmospheric problem. The atmospheric verification is done by comparing one dimensional data, for example an energy spectrum or fluence at a point, since it is hard to compare the full two-dimensional data graphically. Some interesting qualitative results are shown here for small geome-

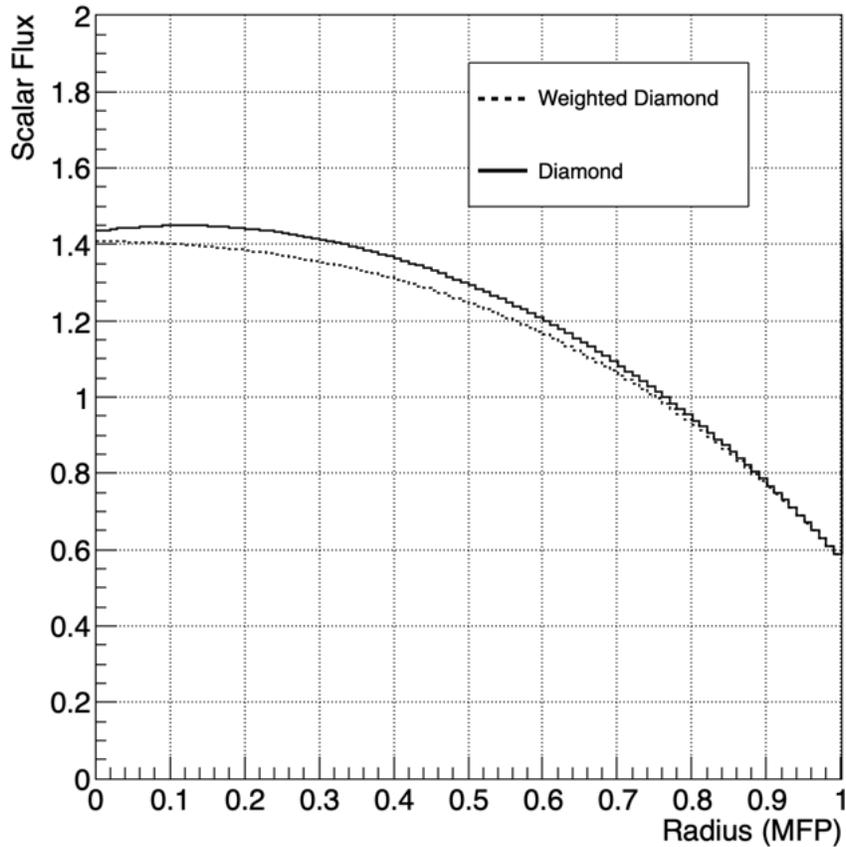


Figure 27: Comparison of weighted and standard diamond differencing for the angular discretization, using bilinear discontinuous spatial differencing on the solid sphere problem.

tries in two dimensions to illustrate the performance of the code for point sources. Figure 28 shows an example of a solid sphere with 40×40 cells, on $r \in [0.0, 20.0]$ cm, $\theta \in [0, \pi]$, with $c = 0.9$, $\sigma_t = 1.0 \text{ cm}^{-1}$ and an internal source 1 cm^{-3} at cell (20,40), with an S_4 angular approximation. From this result, we can see that the flux is isotropic from the source, and is represented properly on the curvilinear orthogonal grid. Figure 29 shows an example of a spherical shell with 10×40 cells, on $r \in [10.0, 20.0]$ cm, $\theta \in [0, \pi]$, with $c = 0.9$, $\sigma_t = 1.0 \text{ cm}^{-1}$ and an internal source 1 cm^{-3} at cell (0,10), with an S_4 angular approximation and BLD spatial discretiza-

tion. Figure 30 shows the performance of spatial diamond-difference code for the same problem. Cells in white are negative fluxes, illustrating this key deficiency of the diamond discretization, which is particularly obvious for two-dimensional point source problems.

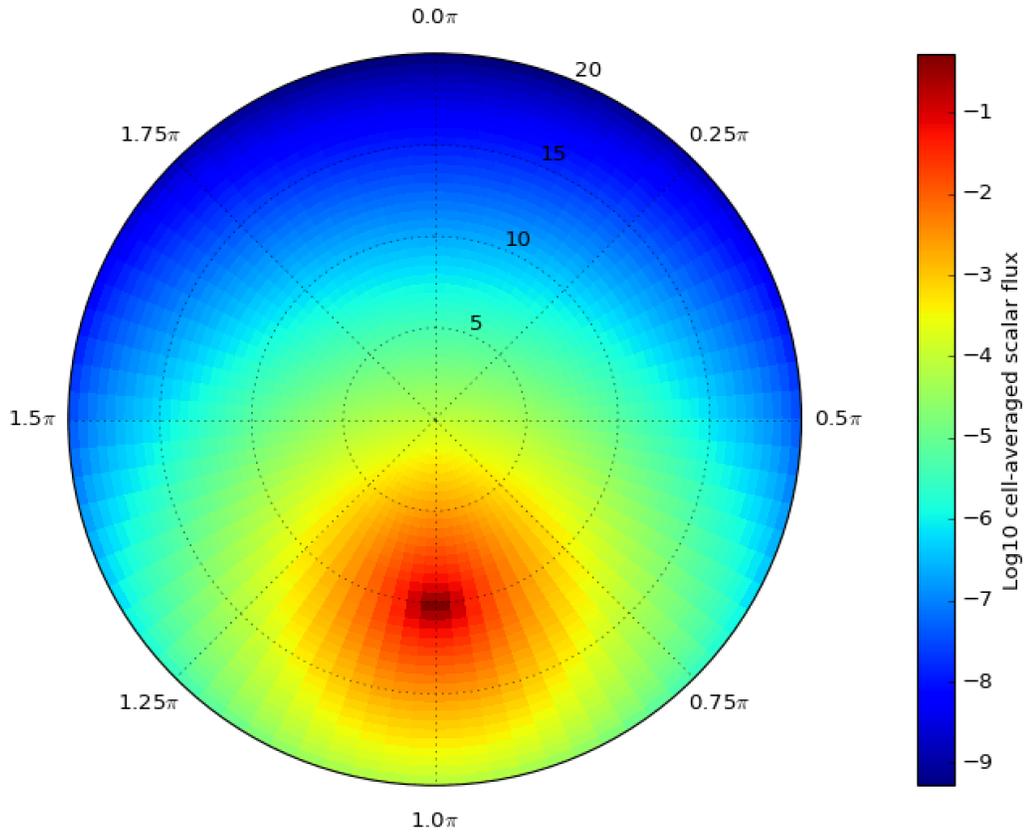


Figure 28: ATMOTRAN result for a point source in a solid sphere.

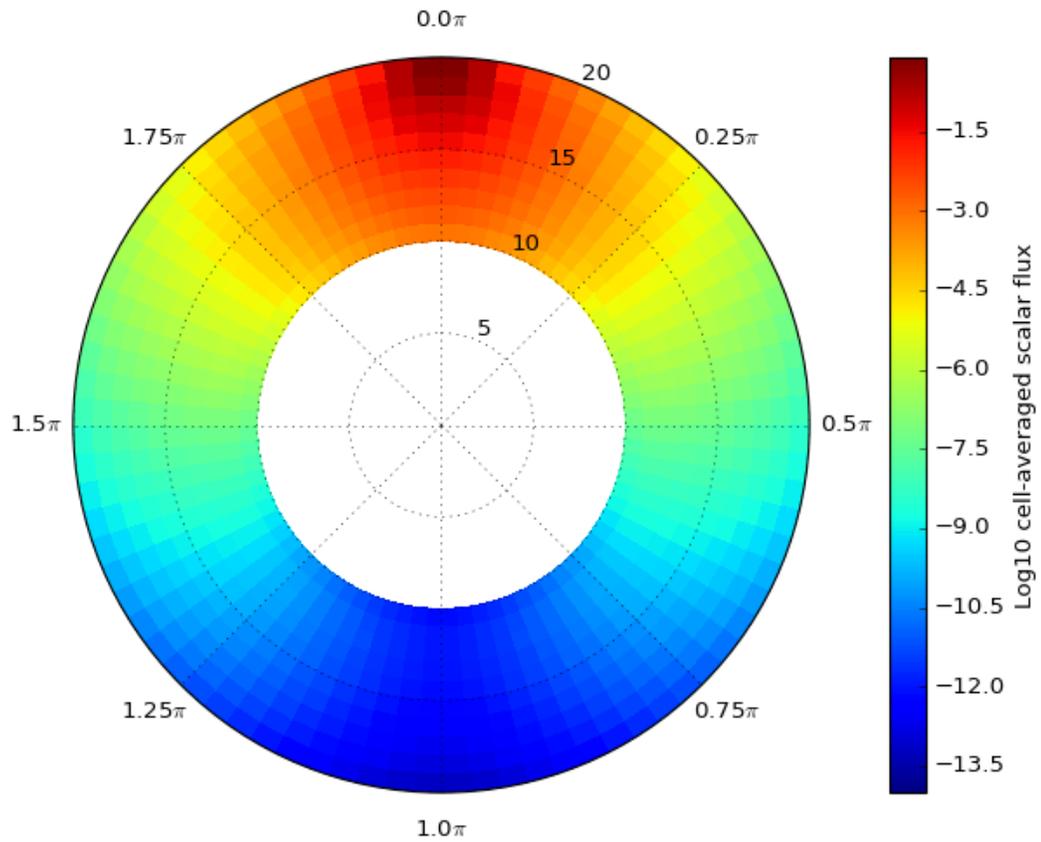


Figure 29: ATMOTRAN result for a point source in a solid sphere with bilinear discontinuous spatial discretization.

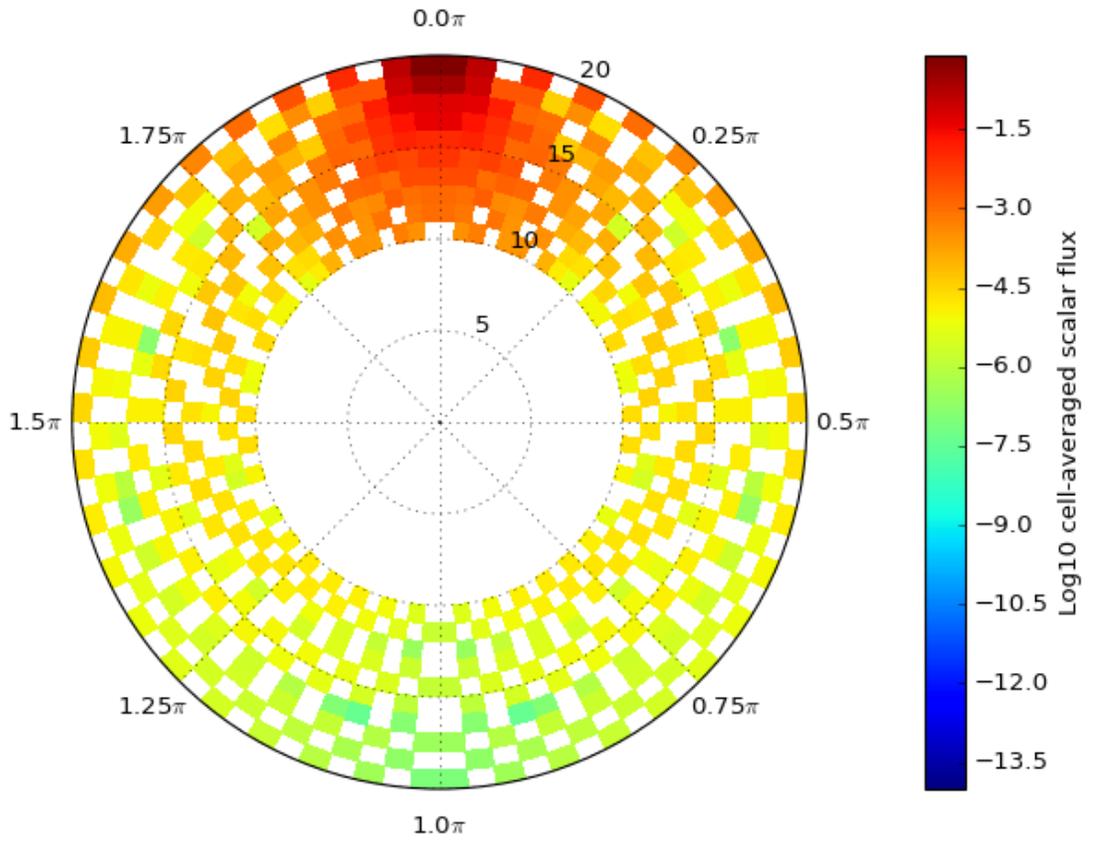


Figure 30: ATMOTRAN result for a point source in a solid sphere with diamond spatial discretization.

5.8 Verification against Monte Carlo

An important check is to verify how well the code can represent the atmospheric problem of interest. This section shows ATMOTRAN results compared to Geant4 [54] Monte Carlo results for the same physical problem. For the results presented here, the Geant4 package (version 4.10.01) was configured to utilize the QGSP-BERT-HP physics model, which performs high precision neutron transport at low energies (< 20 MeV). The Geant4 geometry included a spherical Earth with a radius 6378 km, and an atmosphere extending to 1000 km. The density and composition of the atmosphere was segmented into 200 layers, logarithmically spaced in radius. Each layer is assigned material properties consistent with the NRL MSISE-00 model introduced in Chapter 1. A point source of neutrons is placed at a particular altitude and the fluence of neutrons is scored leaving the atmosphere as a function of energy and angle. Neutron fluence is scored in 60 angular bins from 0 to π radians, and in 30 energy bins (see Table 24). Although the Monte Carlo data was scored into a finite number of bins, the simulation treats energy as a continuous variable, which will reveal some energy discretization error in ATMOTRAN. ATMOTRAN was configured similarly, with 30 energy bins and 60 angular bins, and used 60 radial bins, somewhat coarser than the Monte Carlo geometry. We compare two quantities, the energy spectrum of neutrons leaving that atmosphere directly overhead (Figure 31), and the energy-integrated fluence of neutrons leaving the atmosphere as a function of polar angle (Figure 32). The two plots contained this this section are examples of the full data set contained in Appendix B. Note that these plots compare Monte Carlo data with ATMOTRAN at increasing angular discretization order, and includes a comparison with a spatial diamond difference discretization. Weighted diamond angular differencing is always used. The lower panels of the plots show point-wise ratios of Monte Carlo to ATMOTRAN (S_{10}) fluence values. These comparisons confirm several effects. First, that

ATMOTRAN is properly down-scattering neutrons in energy. These examples are all for monoenergetic sources, and so the lower energy portion of the spectrum is due to the multigroup implementation in ATMOTRAN. We can see in Figure 31 that the ATMOTRAN energy spectrum compares well with the Monte Carlo results particularly at the lower energy ranges below the resonance region. Second, in Figure 32 we can see that the fluence as a function of angle agrees well between higher order ATMOTRAN and Monte Carlo, up to a point where horizon effects and geometry differences are present. We can also see that the diamond-differenced solution starts to oscillate and give unphysical results with higher angle as a result of negative fluxes.

10MeV_30km

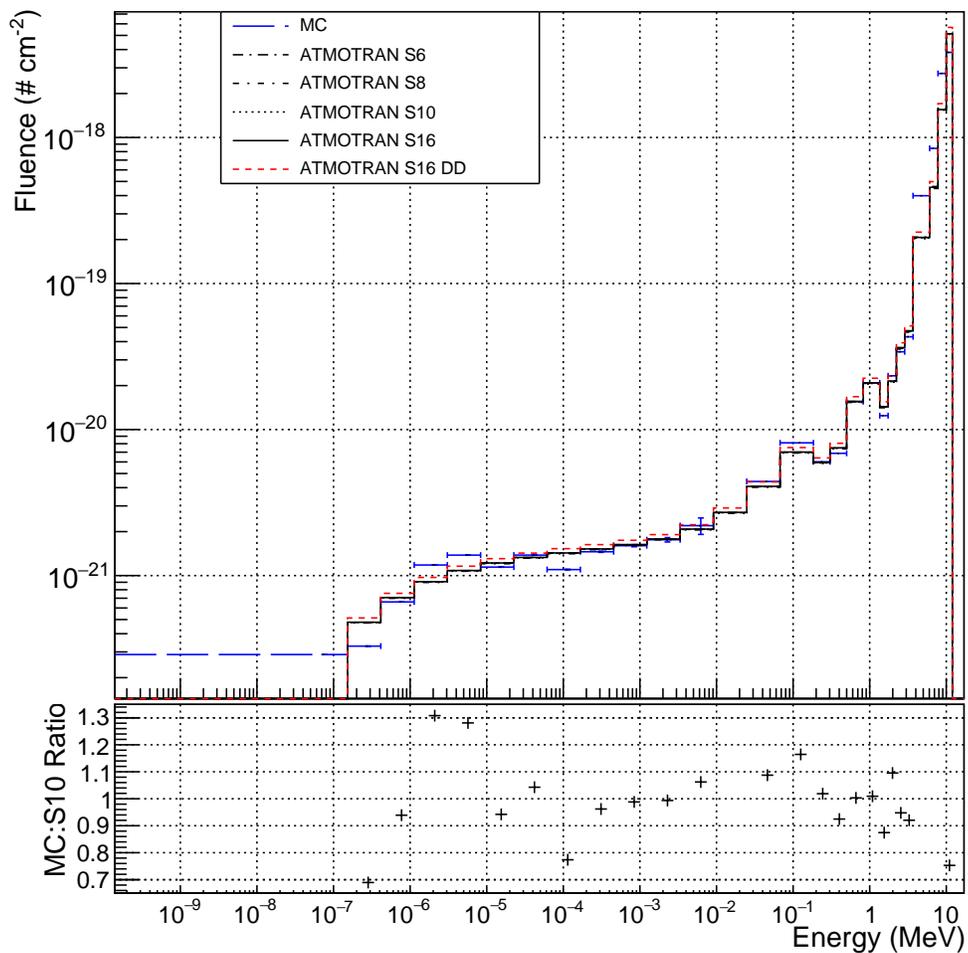


Figure 31: Atmospheric neutron energy spectrum at 1000 km $\theta = 0$, for 10 MeV source at 30 km

10MeV_30km

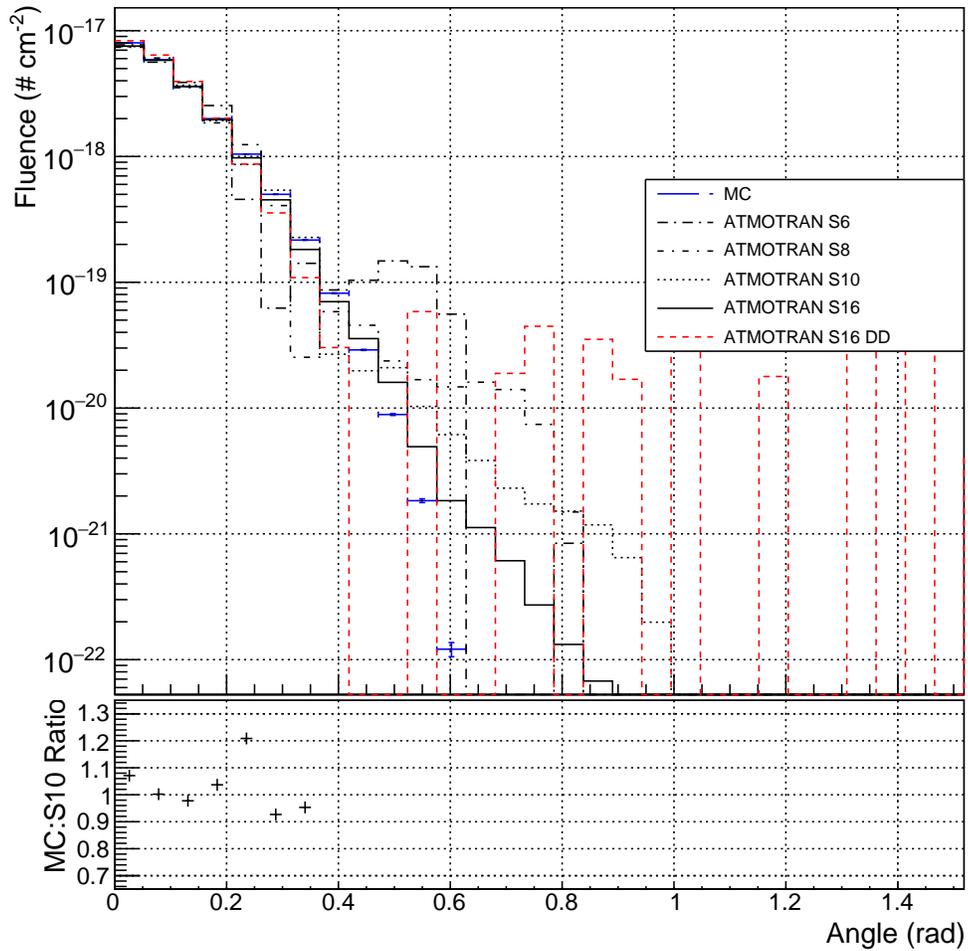


Figure 32: Atmospheric neutron fluence vs polar angle at 1000 km, for 10 MeV source at 30 km

5.9 Summary of verification results

In this chapter we showed evidence that the ATMOTRAN spatio-angular discretization scheme and sweep is correctly implemented without errors. We accomplished this through several tests. First we performed an asymptotic scaling test to verify that

the scheme does preserve the thick-diffusion limit. This is expected for the linear discontinuous class of schemes and it important to ensure that accurate solutions can be achieved for optically thick problems without resorting to very large numbers of mesh cells. We then checked the accuracy of the scheme through comparisons of numerical results with manufactured solutions. This confirms that the bilinear discontinuous discretization of the transport equation in two-dimensional spherical coordinates, the two dimensional weighted diamond angular differencing, and the two-dimensional starting directions, all presented here for the first time, are correctly derived and implemented. The biquadratic isotropic manufactured solution test demonstrated that this scheme has better than second-order accuracy, and in fact a fractional order of accuracy near 2.5. This is an interesting result and more study of the causes may prove fruitful. Two-dimensional Cartesian (X-Y) BLD schemes achieve third order convergence, but two dimensional cylindrical (R-Z) achieve second order, so the present scheme falls in between. The bilinear isotropic manufactured solution confirmed that the code exactly reproduces a solution of the same order as the basis. We showed two anisotropic biquadratic manufactured solutions and results from a multi-level mesh refinement in space and angle. These results demonstrated that with the WDD method many angular points are required to approach the level of the spatial discretization error. This was true for both μ and η anisotropic sources. Depending on the application it may be useful to investigate angular schemes other than WDD that can provide better accuracy with fewer points. Confirmatory code-to-code comparisons were provided by showing good agreement between ATMOTRAN and Capsaicin for two small problems. We showed several interesting cases including a boundary source, a point source within a sphere, and a solid sphere. Finally, to test the multigroup code we expanded the geometry to include a physical model the size of the Earth with an atmosphere extending to 1000 km, and seeded mono-energetic

point sources at several altitudes and energies. Results of these problems showed down-scattered neutron energy spectrum and escaping fluence as a function of polar angle agreed reasonably well between ATMOTRAN and the Monte Carlo simulation.

6 Summary

In this chapter we summarize the main findings of this dissertation and discuss future work that would benefit the application of ATMOTRAN to the atmospheric transport problem. Specifically, this dissertation derived, implemented and verified the methods needed to realize a bilinear discontinuous discretization of the two-dimensional transport equation in (r, θ) spherical coordinates. We will now summarize the main points of each chapter.

In Chapter 1 we introduced the overall goals of this research and defined the problem of interest. Although the work done for this dissertation in terms of methods development for deterministic transport is general, there is a specific target application which we reviewed. We reviewed past work involving deterministic solution of incoming plane waves of particles and contrasted this with the present method. We discussed the shortcomings involved in using Monte Carlo methods for this application, motivating the work presented in this dissertation.

In Chapter 2 we reviewed the discrete ordinates method and discussed techniques of discretizing the transport equation in angle, energy and space to permit accurate solution. Since it is particularly relevant to this dissertation we contrasted the diamond-differenced and linear discontinuous spatial discretization methods. Following this we touched on current techniques in the iterative solution of the transport equation that are used in ATMOTRAN. We concluded this chapter with a detailed

review of the most relevant prior work, that of the implementation of the diamond-differenced TWOTRAN SPHERE code from 1970. This key reference provided insight into the requirements of information flow in the spatio-angular mesh for discrete ordinates solutions in the relevant coordinate system. We discussed the derivation of the characteristic equations of motion and their implications, which are relevant to any transport solution in this system regardless of discretization technique.

In Chapter 3 we introduced the bilinear discontinuous discretization in (r, θ) spherical coordinates. We rearranged the form of the transport equation and worked through the details to derive a finite element spatial representation. This included defining vector basis functions to relate quantities within a spatial cell allowing expansion of quantities in terms of these basis functions, and integrating the terms in the transport equation over phase space volume for individual cells. Once this was completed, we presented the fully discretized transport equations in a form that is readily implemented for solution on a computer. We concluded by providing a derivation of the discretized transport equations for the zero-weighted starting directions in two dimensions.

In Chapter 4 we provided details of the implementation of the method of Chapter 3 into the functional ATMOTRAN code. We described the six main functional blocks of the code. We began by describing how the physical atmospheric geometry is defined in the code, including use of external library data for atmospheric density and composition, along with nuclear cross-sectional data. We then showed a sampling of some of the data including mean free paths as a function of energy and altitude, along with scattering fractions for the same ranges to provide a general feel for how this large physical problem might relate to other problems with a similar phase space. Following this, we presented data in the S_4 angular approximation for the Gauss-Chebyshev-Legendre quadrature, angular redistribution and weighted

diamond angular differencing coefficients, along with starting direction points. We presented the iterative method of solution and presented some sample iterative convergence measurements. We concluded with details of the spatio-angular sweep information flow noting that, although it follows the characteristic equations of particle motion it is somewhat different from, and more complicated than, the already complex sweep method presented in TWOTRAN SPHERE because of the presence of starting directions.

In Chapter 5 we considered the correctness of the code and presented verification data confirming the validity of the fundamental assumptions that went into the derivations and the ATMOTRAN implementation. We presented verification results from manufactured solutions, code-to-code comparisons with another transport code, comparison of multigroup performance with Monte Carlo results, and some individual examples of interesting cases. We showed that the scheme presented preserves the thick-diffusion limit, which is necessary to ensure accurate solutions in optically thick problems, such as neutron transport in the atmospheric transport application targeted in this work. Numerical verification demonstrated that it is feasible to use this discretization approach for such applications and that ATMOTRAN has successfully accomplished this goal.

To emphasize the novelty of this work, we state that the bilinear discontinuous spatial discretization of the transport equation in this coordinate system, the application of weighted diamond difference discretization to the angular derivative term in two angular dimensions, and the use of zero-weighted starting directions and complicated information flow presented are all new to the field and generally applicable to any physical system to be represented in these coordinates. The application of a deterministic, spherical (r, θ) code to the solution of a problem involving the transport of point sources of neutral particles through the atmosphere is also new.

Future work may include further investigation and analysis of the source of the fractional order of convergence shown for this discretization scheme in the isotropic biquadratic manufactured solution. Also, it may be useful to investigate other angular differencing schemes that may improve the convergence speed of the solution with fewer angles in the case of anisotropic source terms. Future work is anticipated to investigate other improvements including the use of acceleration techniques for faster convergence speed, and scaling down memory requirements of the code to improve usability.

7 Appendices

Appendix A - Matrix Elements

Appendix B - Atmospheric Calculation Results

Appendix C - NJOY Input Files

7.1 Appendix A - Matrix Elements

This section contains formulations for the integrals used in the matrix representations in the derivation and verification sections.

7.1.1 Tabulation of M_x Integrals

$$M_1 = \int_{r_L}^{r_R} dr r^2 (r_R - r)(r - r_L) = \frac{(\Delta r)^3}{30} (6r_L^2 + 3r_L r_R + r_R^2)$$

$$M_2 = \int_{r_L}^{r_R} dr r^2 (r_R - r)(r - r_L) = \frac{1}{60} (-3r_L^5 + 5r_L^4 r_R - 5r_L r_R^4 + 3r_R^5)$$

$$M_3 = \int_{r_L}^{r_R} dr r^2 (r - r_L)(r - r_L) = \frac{(\Delta r)^3}{30} (r_L^2 + 3r_L r_R + 6r_R^2)$$

$$M_4 = \int_{\theta_B}^{\theta_T} d\theta \sin(\theta)(\theta_T - \theta)(\theta_T - \theta)$$

$$= 2\Delta\theta \sin(\theta_B) + ((\Delta\theta)^2 - 2) \cos(\theta_B) + 2 \cos(\theta_T)$$

$$M_5 = \int_{\theta_B}^{\theta_T} d\theta \sin(\theta)(\theta_T - \theta)(\theta - \theta_B)$$

$$= -\Delta\theta (\sin(\theta_B) + \sin(\theta_T)) + 2 \cos(\theta_B) - 2 \cos(\theta_T)$$

$$M_6 = \int_{\theta_B}^{\theta_T} d\theta \sin(\theta)(\theta - \theta_B)(\theta - \theta_B)$$

$$= 2\Delta\theta \sin(\theta_T) - ((\Delta\theta)^2 - 2) \cos(\theta_T) - 2 \cos(\theta_B)$$

$$M_7 = \int_{r_L}^{r_R} dr r^2 (r_R - r) = \frac{1}{12} (3r_L^4 - 4r_L^3 r_R + r_R^4)$$

$$M_8 = \int_{r_L}^{r_R} dr r^2 (r - r_L) = \frac{1}{12} (r_L^4 - 4r_L r_R^3 + 3r_R^4)$$

$$M_9 = \int_{r_L}^{r_R} dr r (r_R - r)(r - r_L) = \frac{(\Delta r)^3}{12} (3r_L + r_R)$$

$$M_{10} = \int_{r_L}^{r_R} dr r (r_R - r)(r - r_L) = \frac{(\Delta r)^3}{12} (r_L + r_R)$$

$$M_{11} = \int_{r_L}^{r_R} dr r (r - r_L)(r - r_L) = \frac{(\Delta r)^3}{12} (r_L + 3r_R)$$

$$M_{12} = \int_{\theta_B}^{\theta_T} d\theta \sin(\theta)(\theta_T - \theta) = \Delta\theta \cos(\theta_B) + \sin(\theta_B) - \sin(\theta_T)$$

$$\begin{aligned}
M_{13} &= \int_{\theta_B}^{\theta_T} d\theta \sin(\theta)(\theta - \theta_B) = -\Delta\theta \cos(\theta_T) - \sin(\theta_B) + \sin(\theta_T) \\
M_{14} &= \int_{\theta_B}^{\theta_T} d\theta \cos(\theta)(\theta_T - \theta)(\theta_T - \theta) \\
&= -((\Delta\theta)^2 - 2) \sin(\theta_B) + 2\Delta\theta \cos(\theta_B) - 2 \sin(\theta_T) \\
M_{15} &= \int_{\theta_B}^{\theta_T} d\theta \cos(\theta)(\theta_T - \theta)(\theta - \theta_B) \\
&= -\Delta\theta \cos(\theta_B) - \Delta\theta \cos(\theta_T) - 2 \sin(\theta_B) + 2 \sin(\theta_T) \\
M_{16} &= \int_{\theta_B}^{\theta_T} d\theta \cos(\theta)(\theta - \theta_B)(\theta - \theta_B) \\
&= ((\Delta\theta)^2 - 2) \sin(\theta_T) + 2\Delta\theta \cos(\theta_T) + 2 \sin(\theta_B) \\
M_{17} &= \int_{r_L}^{r_R} dr r^3 (r_R - r) = \frac{1}{20} (4r_L^5 - 5r_L^4 r_R + r_R^5) \\
M_{18} &= \int_{r_L}^{r_R} dr r^3 (r - r_L) = \frac{1}{20} (r_L^5 - 5r_L r_R^4 + 4r_R^5) \\
M_{19} &= \int_{r_L}^{r_R} dr r^3 (r_R - r)(r_R - r) = \frac{1}{60} (-10r_L^6 + 24r_L^5 r_R - 15r_L^4 r_R^2 + r_R^6) \\
M_{20} &= \int_{r_L}^{r_R} dr r^3 (r_R - r)(r - r_L) = \frac{1}{60} (-2r_L^6 + 3r_L^5 r_R - 3r_L r_R^5 + 2r_R^6) \\
M_{21} &= \int_{r_L}^{r_R} dr r^3 (r - r_L)(r - r_L) = \frac{1}{60} (-r_L^6 + 15r_L^2 r_R^4 - 24r_L r_R^5 + 10r_R^6) \\
M_{22} &= \int_{\theta_B}^{\theta_T} d\theta \theta \sin \theta (\theta_T - \theta)(\theta_T - \theta) \\
&= -(3\theta_B^2 - 4\theta_B \theta_T + \theta_T^2 - 6) \sin \theta_B \\
&\quad + (\theta_B(\Delta\theta^2 - 6) + 4\theta_T) \cos \theta_B - 6 \sin \theta_T + 2\theta_T \cos \theta_T \\
M_{23} &= \int_{\theta_B}^{\theta_T} d\theta \theta \sin \theta (\theta_T - \theta)(\theta - \theta_B) \\
&= \theta_B^2 \sin \theta_B - \theta_B \theta_T \sin \theta_B + \theta_B \theta_T \sin \theta_T + (4\theta_B - 2\theta_T) \cos \theta_B \\
&\quad + 2(\theta_B - 2\theta_T) \cos \theta_T - 6 \sin \theta_B - \theta_T^2 \sin \theta_T + 6 \sin \theta_T
\end{aligned}$$

$$\begin{aligned}
M_{24} &= \int_{\theta_B}^{\theta_T} d\theta \sin \theta (\theta - \theta_B) (\theta - \theta_B) \\
&= -(\theta_B^2 \theta_T - 2\theta_B \theta_T^2 + 4\theta_B + \theta_T^3 - 6\theta_T) \cos \theta_T + \theta_B^2 \sin \theta_T - 4\theta_B \theta_T \sin \theta_T \\
&\quad + 6 \sin \theta_B - 2\theta_B \cos \theta_B + 3\theta_T^2 \sin \theta_T - 6 \sin \theta_T \\
M_{25} &= \int_{r_L}^{r_R} dr r^4 (r_R - r) (r_R - r) = \frac{1}{105} (-15r_L^7 + 35r_L^6 r_R - 21r_L^5 r_R^2 + r_R^7) \\
M_{26} &= \int_{r_L}^{r_R} dr r^4 (r_R - r) (r - r_L) = \frac{1}{210} (-5r_L^7 + 7r_L^6 r_R - 7r_L r_R^6 + 5r_R^7) \\
M_{27} &= \int_{r_L}^{r_R} dr r^4 (r - r_L) (r - r_L) = \frac{1}{105} (-r_L^7 + 21r_L^2 r_R^5 - 35r_L r_R^6 + 15r_R^7) \\
M_{28} &= \int_{\theta_B}^{\theta_T} d\theta \theta^2 \sin \theta (\theta_T - \theta) (\theta_T - \theta) \\
&= -2(2\theta_B - \theta_T) (\theta_B^2 - \theta_B \theta_T - 6) \sin \theta_B \\
&\quad + (\theta_B^4 - 2\theta_B^3 \theta_T + \theta_B^2 (\theta_T^2 - 12) + 12\theta_B \theta_T - 2\theta_T^2 + 24) \cos \theta_B \\
&\quad + 2(\theta_T^2 - 12) \cos \theta_T - 12\theta_T \sin \theta_T \\
M_{29} &= \int_{\theta_B}^{\theta_T} d\theta \theta^2 \sin \theta (\theta_T - \theta) (\theta - \theta_B) \\
&= \theta_B^3 \sin \theta_B - \theta_B^2 \theta_T \sin \theta_B + (6\theta_B^2 - 4\theta_B \theta_T - 24) \cos \theta_B + \theta_B \theta_T^2 \sin \theta_T \\
&\quad + (4\theta_B \theta_T - 6\theta_T^2 + 24) \cos \theta_T - 6\theta_B \sin \theta_T + 6\theta_T \sin \theta_B - 18\theta_B \sin \theta_B \\
&\quad - \theta_T^3 \sin \theta_T + 18\theta_T \sin \theta_T \\
M_{30} &= \int_{\theta_B}^{\theta_T} d\theta \theta^2 \sin \theta (\theta - \theta_B) (\theta - \theta_B) \\
&= -(\theta_B^2 (\theta_T^2 - 2) - 2\theta_B \theta_T (\theta_T^2 - 6) + \theta_T^4 - 12\theta_T^2 + 24) \cos \theta_T \\
&\quad - 2(\theta_B^2 - 12) \cos \theta_B + 2(\theta_B - 2\theta_T) (\theta_B \theta_T - \theta_T^2 + 6) \sin \theta_T + 12\theta_B \sin \theta_B
\end{aligned}$$

$$\begin{aligned}
M_{31} &= \int_{\theta_B}^{\theta_T} d\theta \cos \theta (\theta_T - \theta) (\theta_T - \theta) \\
&= -(3\theta_B^2 - 4\theta_B\theta_T + \theta_T^2 - 6) \cos(\theta_B) \\
&\quad - (\theta_B(\Delta\theta^2 - 6) + 4\theta_T) \sin(\theta_B) - 2\theta_T \sin(\theta_T) - 6 \cos(\theta_T) \\
M_{32} &= \int_{\theta_B}^{\theta_T} d\theta \cos \theta (\theta_T - \theta) (\theta - \theta_B) \\
&= (\theta_B^2 - \theta_B\theta_T - 6) \cos(\theta_B) + (\theta_B\theta_T - \theta_T^2 + 6) \cos(\theta_T) \\
&\quad + 2\theta_T \sin(\theta_B) - 2\theta_B \sin(\theta_T) - 4\theta_B \sin(\theta_B) + 4\theta_T \sin(\theta_T) \\
M_{33} &= \int_{\theta_B}^{\theta_T} d\theta \cos \theta (\theta - \theta_B) (\theta - \theta_B) \\
&= (\theta_B^2 - 4\theta_B\theta_T + 3\theta_T^2 - 6) \cos(\theta_T) + \theta_B^2\theta_T \sin(\theta_T) - 2\theta_B\theta_T^2 \sin(\theta_T) \\
&\quad + 4\theta_B \sin(\theta_T) + 2\theta_B \sin(\theta_B) + 6 \cos(\theta_B) + \theta_T^3 \sin(\theta_T) - 6\theta_T \sin(\theta_T) \\
M_{34} &= \int_{\theta_B}^{\theta_T} d\theta \theta^2 \cos \theta (\theta_T - \theta) (\theta_T - \theta) \\
&= -2(2\theta_B - \theta_T)(\theta_B^2 - \theta_B\theta_T - 6) \cos(\theta_B) - 2(\theta_T^2 - 12) \sin(\theta_T) - 12\theta_T \cos(\theta_T) \\
&\quad - (\theta_B^4 - 2\theta_B^3\theta_T + \theta_B^2(\theta_T^2 - 12) + 12\theta_B\theta_T - 2\theta_T^2 + 24) \sin(\theta_B) \\
M_{35} &= \int_{\theta_B}^{\theta_T} d\theta \theta^2 \cos \theta (\theta_T - \theta) (\theta - \theta_B) = -6\theta_B^2 \sin(\theta_B) \\
&\quad + (\theta_B^3 - \theta_B^2\theta_T - 18\theta_B + 6\theta_T) \cos(\theta_B) + (\theta_B\theta_T^2 - 6\theta_B - \theta_T^3 + 18\theta_T \cos(\theta_T)) \\
&\quad + 4\theta_B\theta_T(\sin(\theta_B) - \sin(\theta_T)) + 24 \sin(\theta_B) + 6\theta_T^2 \sin(\theta_T) - 24 \sin(\theta_T) \\
M_{36} &= \int_{\theta_B}^{\theta_T} d\theta \theta^2 \cos \theta (\theta - \theta_B) (\theta - \theta_B) = 2(\theta_B - 2\theta_T)(\theta_B\theta_T - \theta_T^2 + 6) \cos(\theta_T) \\
&\quad + (\theta_B^2(\theta_T^2 - 2) - 2\theta_B(\theta_T^2 - 6) + \theta_T^4 - 12\theta_T^2 + 24) \sin(\theta_T) \\
&\quad + 2(\theta_B^2 - 12) \sin(\theta_B) + 12\theta_B \cos(\theta_B)
\end{aligned}$$

7.2 Appendix B - Atmospheric Calculation Results

This Appendix shows ATMOTRAN results compared to Geant4 Monte Carlo results for the same physical problem. Details of the models are discussed in Chapter 5. In summary, a mono-energetic isotropic point source of neutrons is placed at a particular altitude and the fluence of neutrons is scored leaving the atmosphere as a function of energy and angle. Fluence is scored in 60 angular bins from 0 to π radians, and in 30 energy bins. ATMOTRAN results are shown for several angular approximations along with a sample spatial diamond-differenced discretization to illustrate the limitations of that method. The following plots are grouped into two sections that compare different quantities: Figures 33 to 48 compare the energy spectrum of neutrons leaving the atmosphere directly overhead, while Figures 49 to 64 compare the energy-integrated fluence of neutrons leaving the atmosphere as a function of polar angle.

7.2.1 Energy Spectrum Comparison

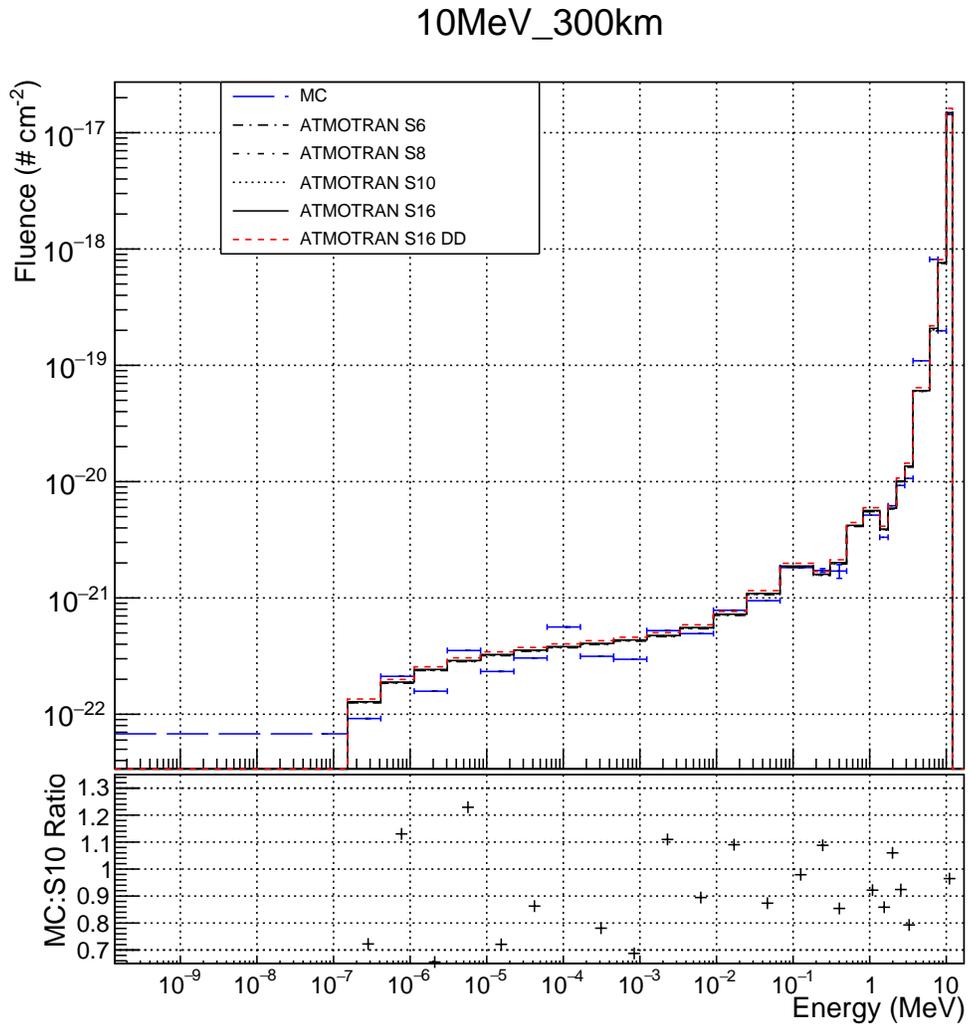


Figure 33: Neutron energy spectrum at 1000 km $\theta = 0$, for 10 MeV source at 300 km

10MeV_100km

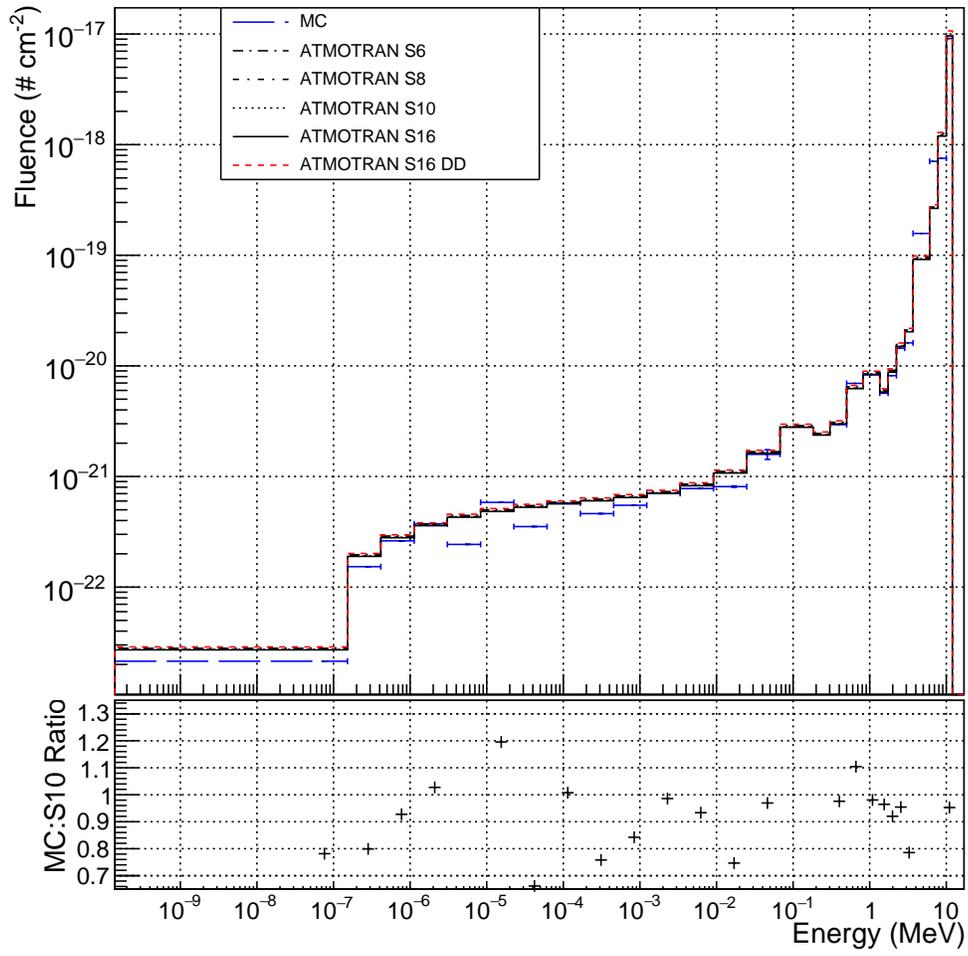


Figure 34: Energy spectrum at 1000 km $\theta = 0$, for 10 MeV source at 100 km

10MeV_30km

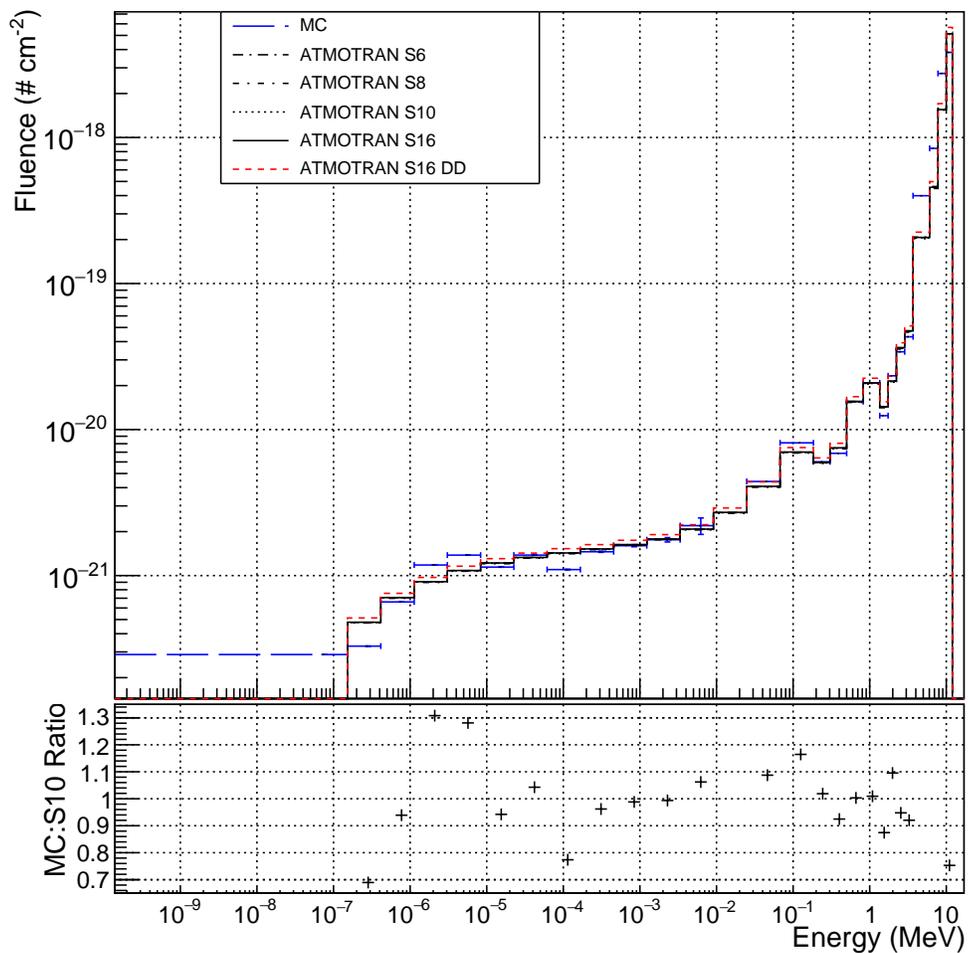


Figure 35: Energy spectrum at 1000 km $\theta = 0$, for 10 MeV source at 30 km

10MeV_20km

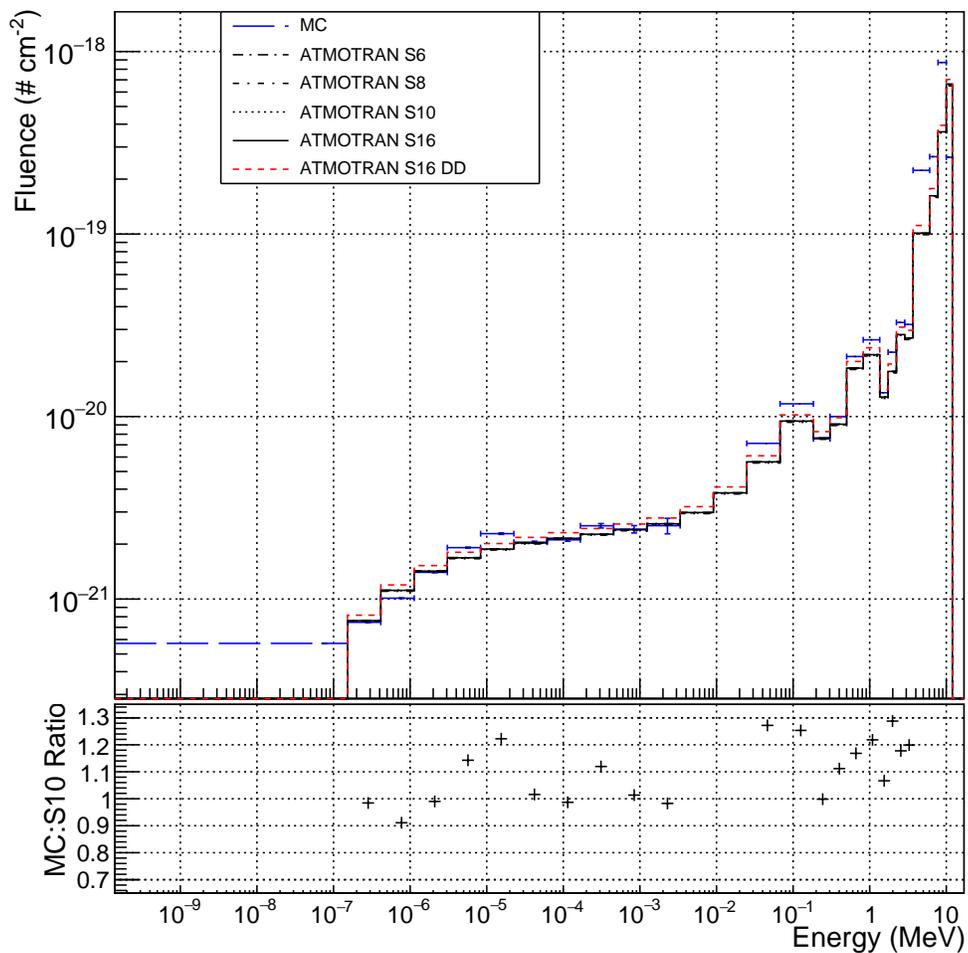


Figure 36: Energy spectrum at 1000 km $\theta = 0$, for 10 MeV source at 20 km

5MeV_300km

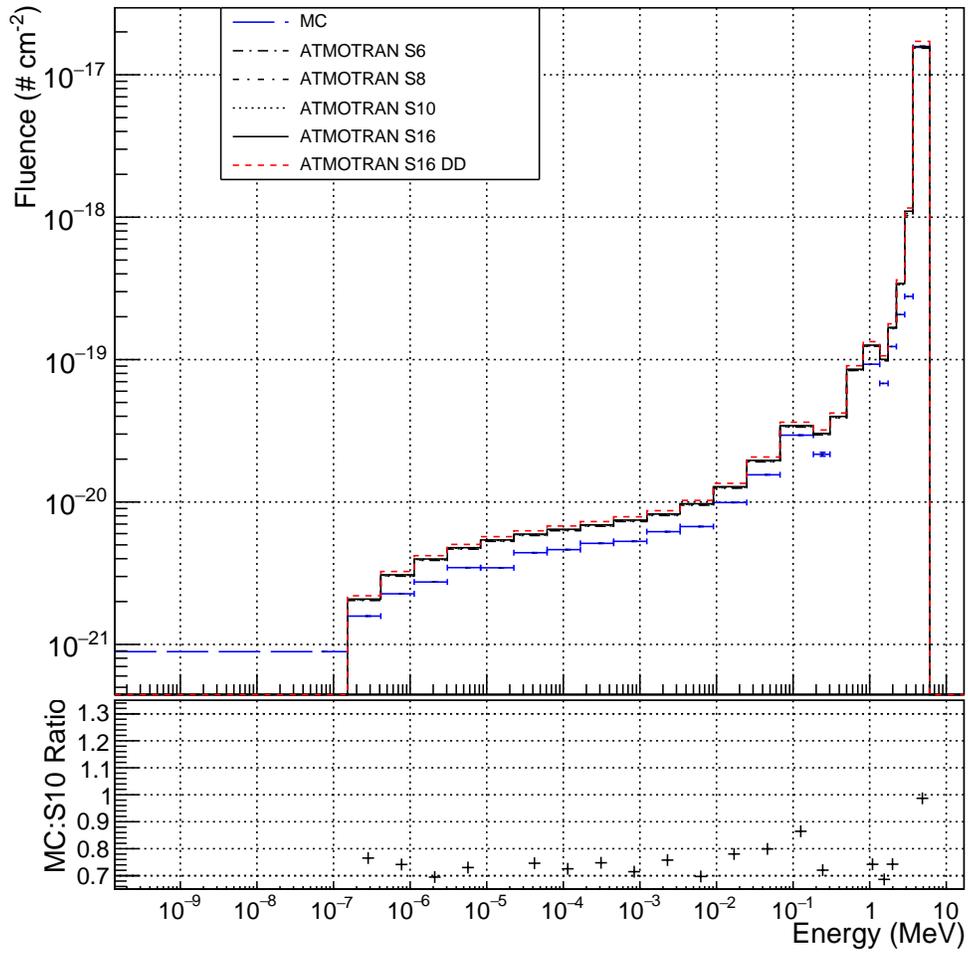


Figure 37: Energy spectrum at 1000 km $\theta = 0$, for 5 MeV source at 300 km

5MeV_100km

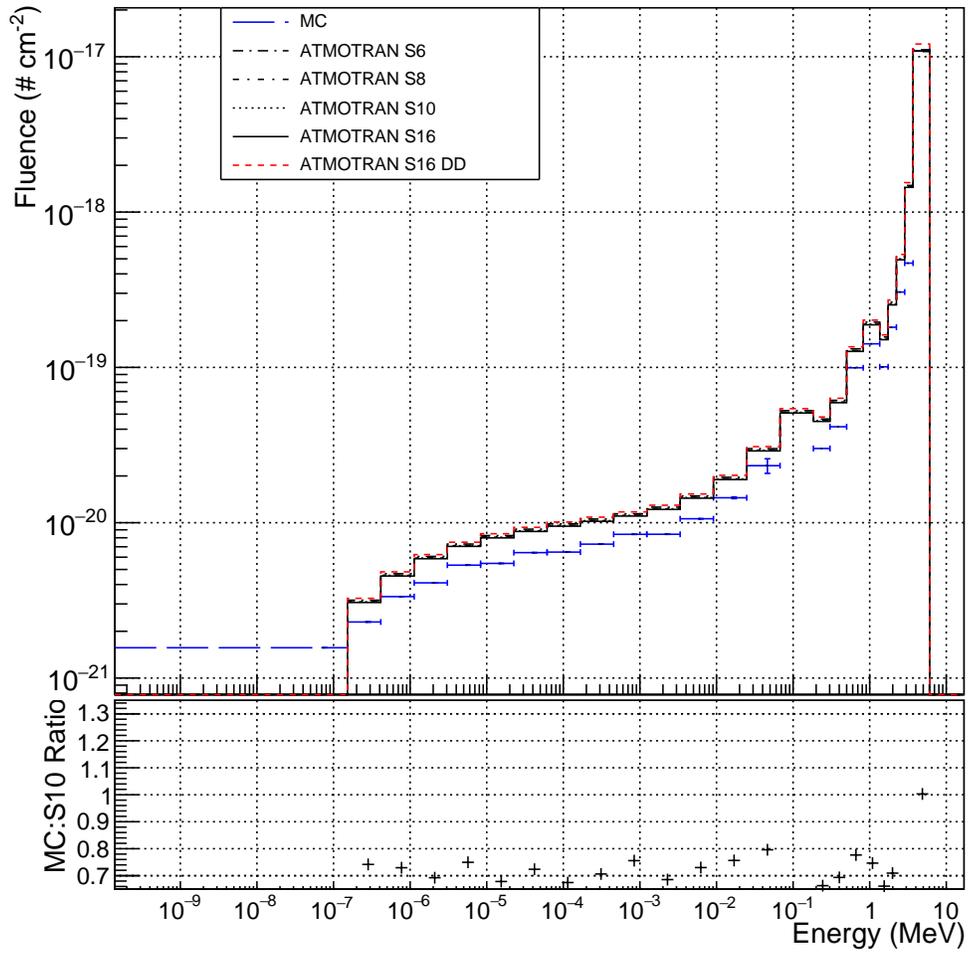


Figure 38: Energy spectrum at 1000 km $\theta = 0$, for 5 MeV source at 100 km

5MeV_30km

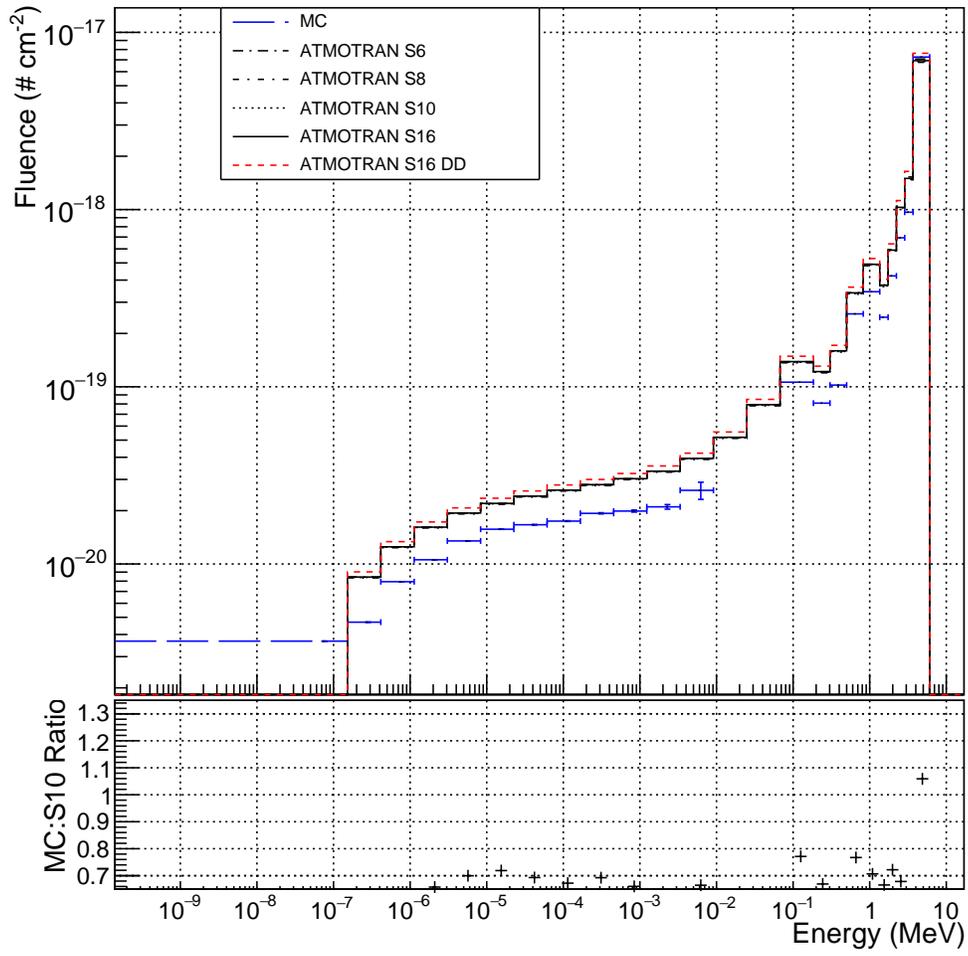


Figure 39: Energy spectrum at 1000 km $\theta = 0$, for 5 MeV source at 30 km

5MeV_20km

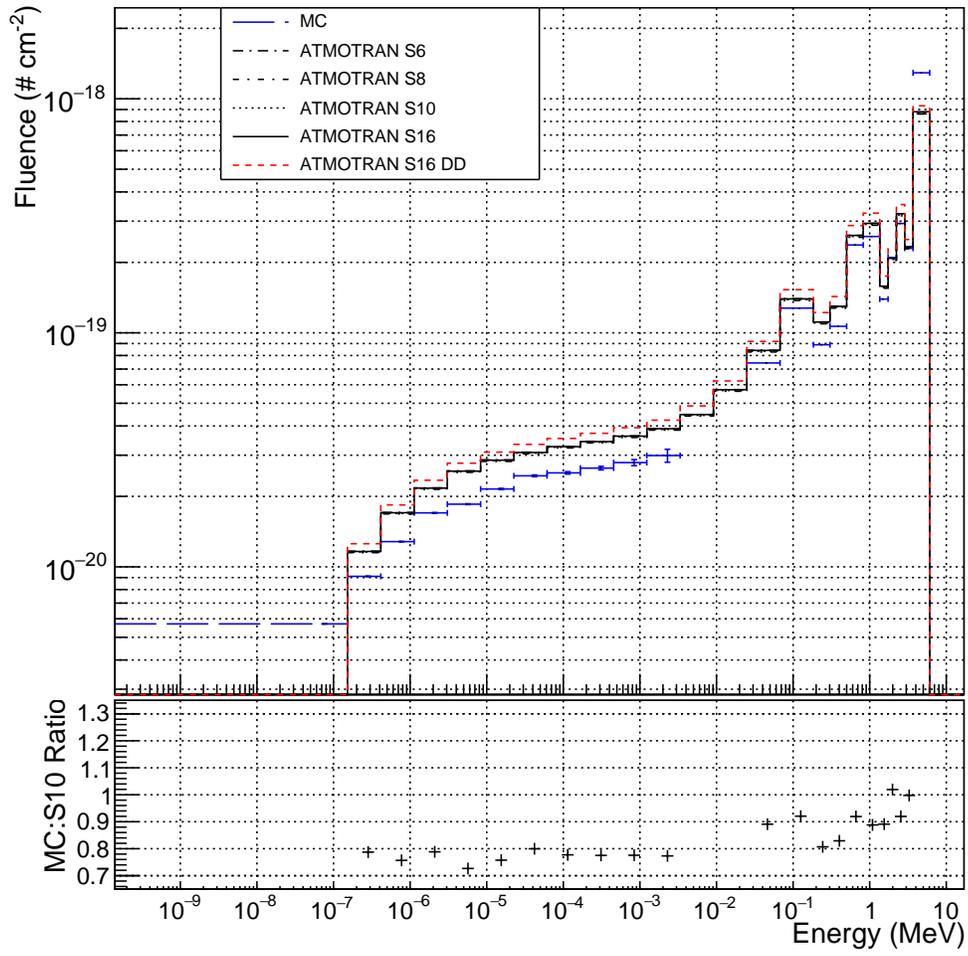


Figure 40: Energy spectrum at 1000 km $\theta = 0$, for 5 MeV source at 20 km

2MeV_300km

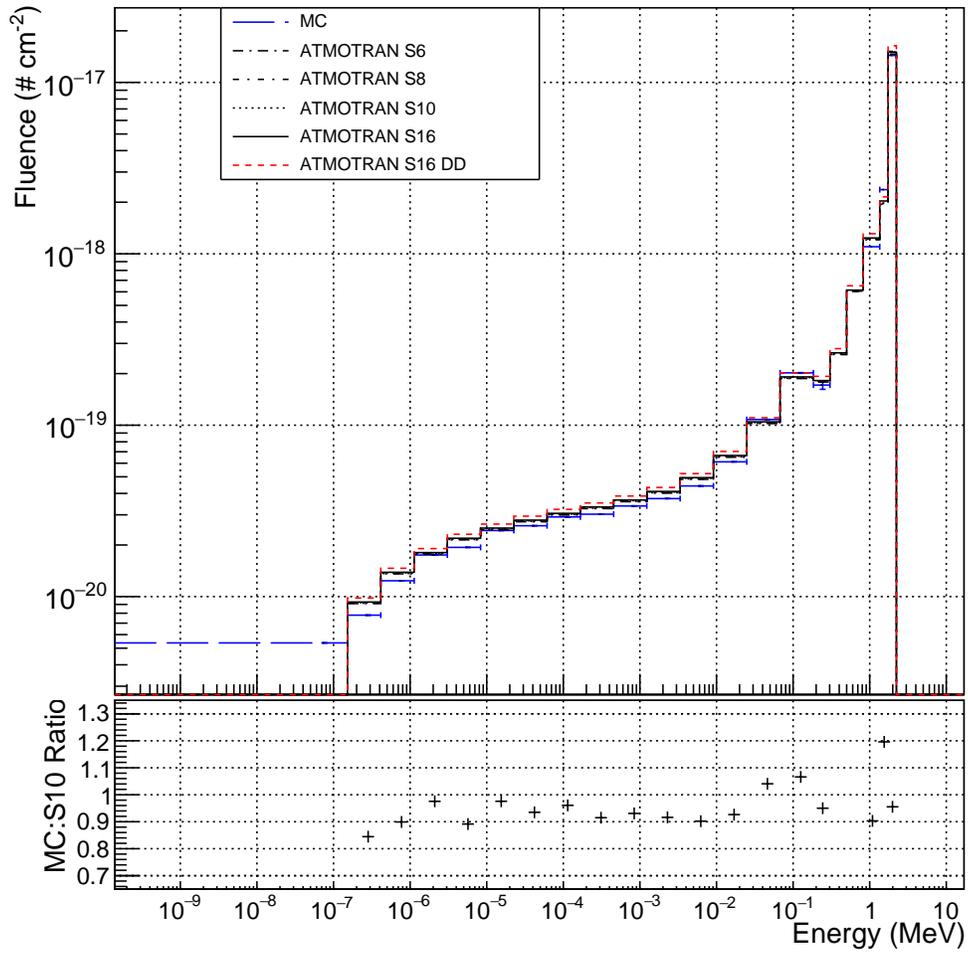


Figure 41: Energy spectrum at 1000 km $\theta = 0$, for 2 MeV source at 300 km

2MeV_100km

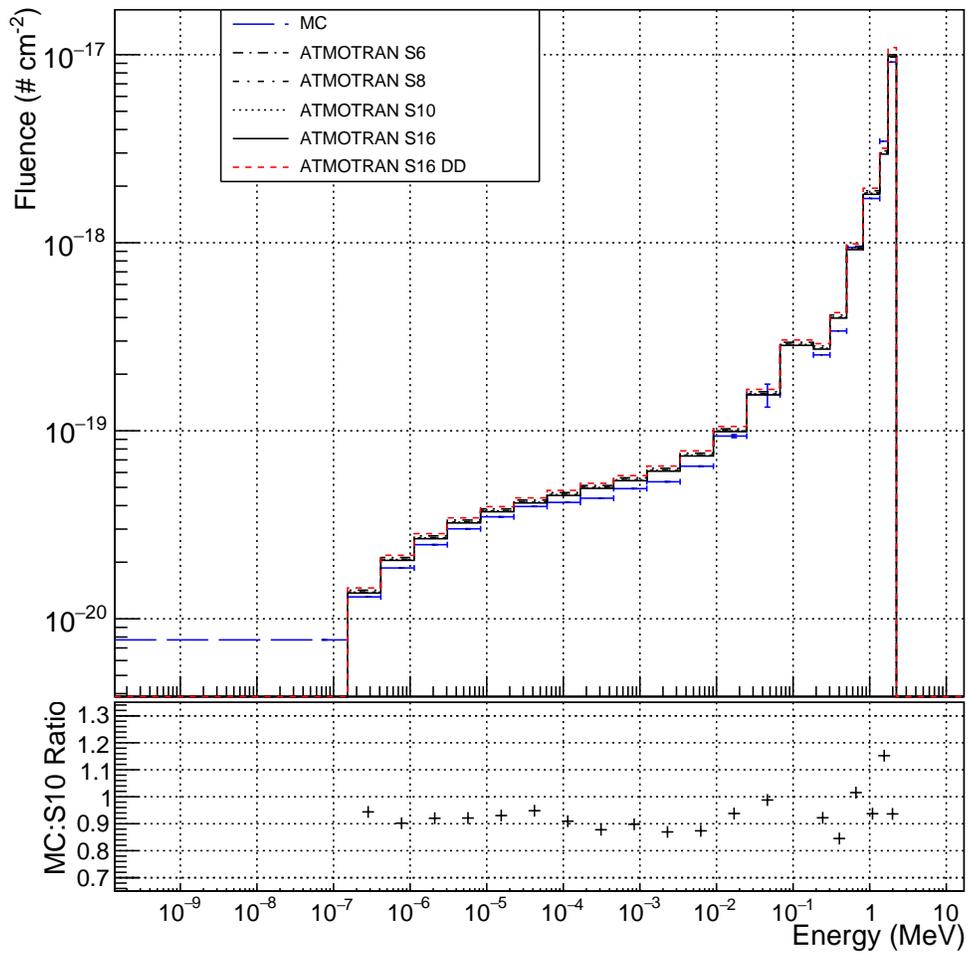


Figure 42: Energy spectrum at 1000 km $\theta = 0$, for 2 MeV source at 100 km

2MeV_30km

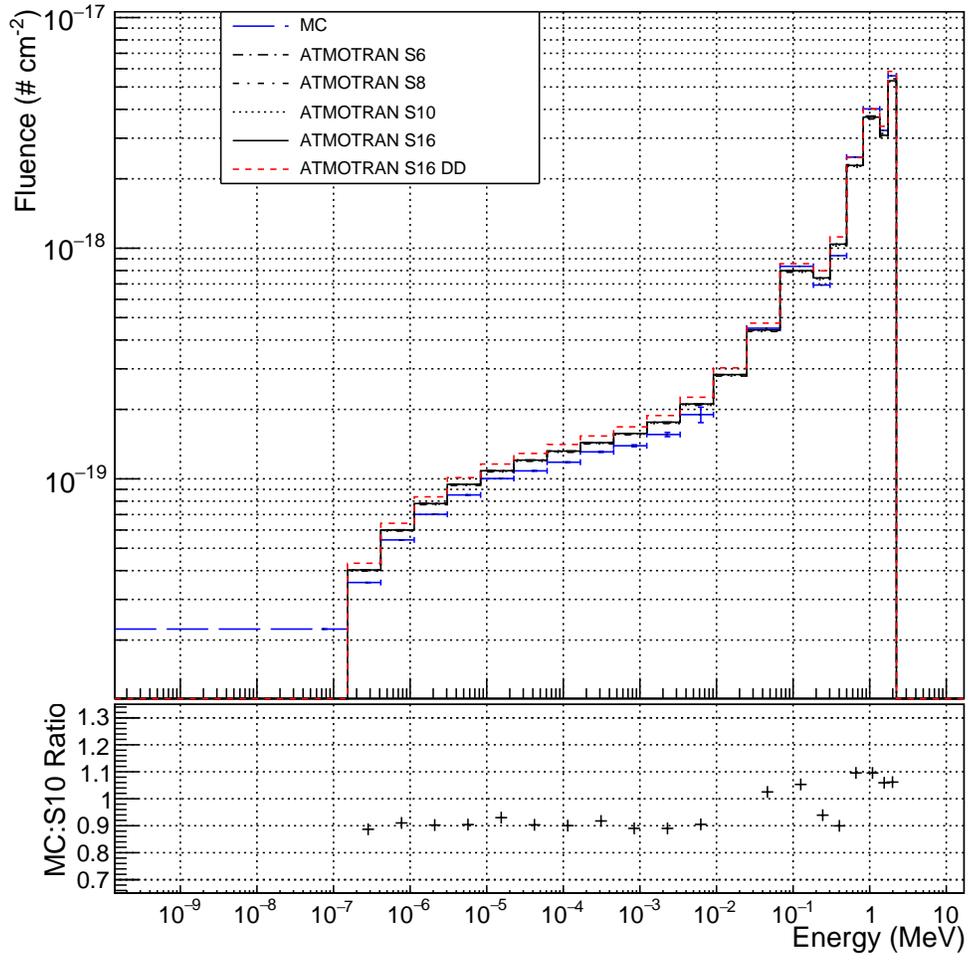


Figure 43: Energy spectrum at 1000 km $\theta = 0$, for 2 MeV source at 30 km

2MeV_20km

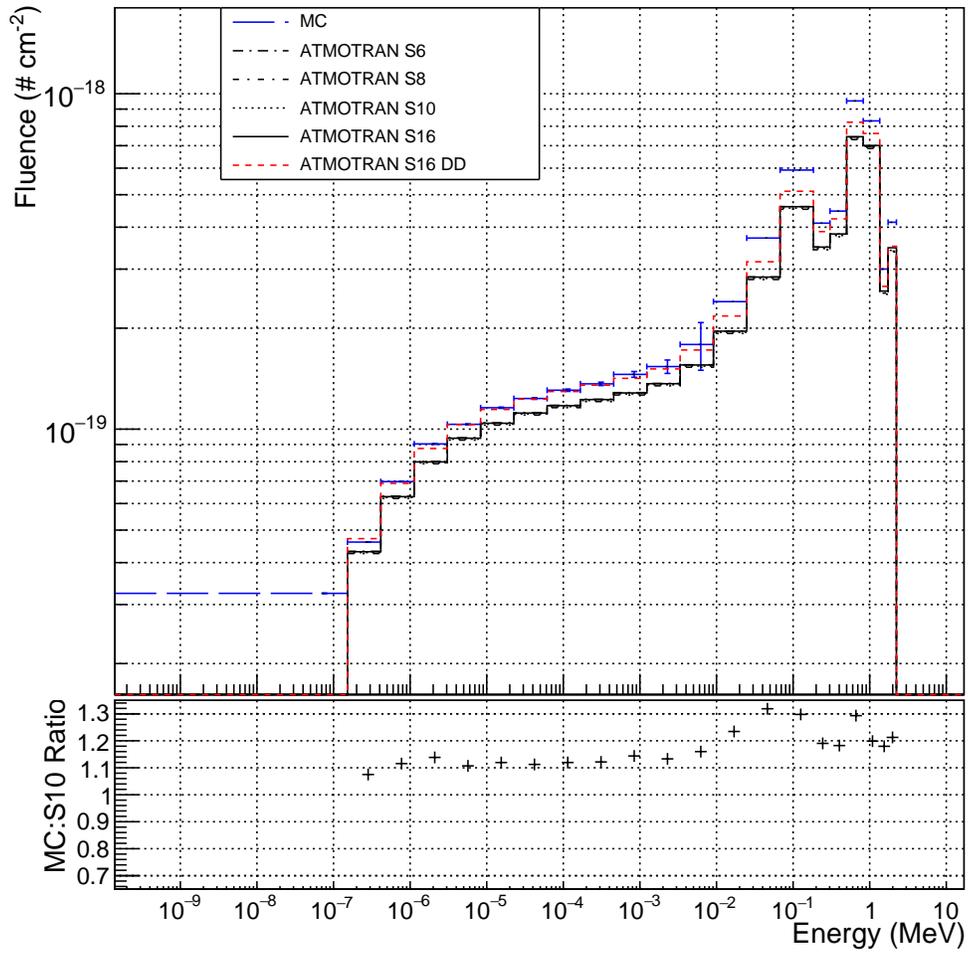


Figure 44: Energy spectrum at 1000 km $\theta = 0$, for 2 MeV source at 20 km

1MeV_300km

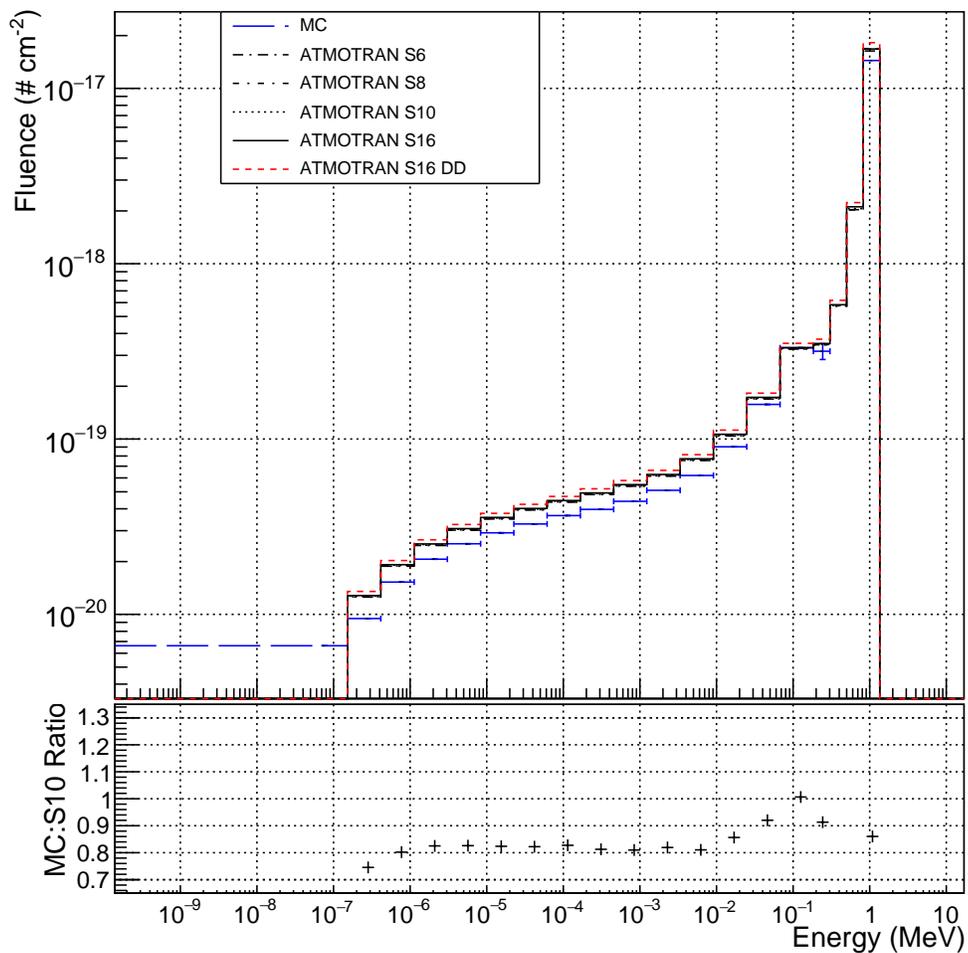


Figure 45: Energy spectrum at 1000 km $\theta = 0$, for 1 MeV source at 300 km

1MeV_100km

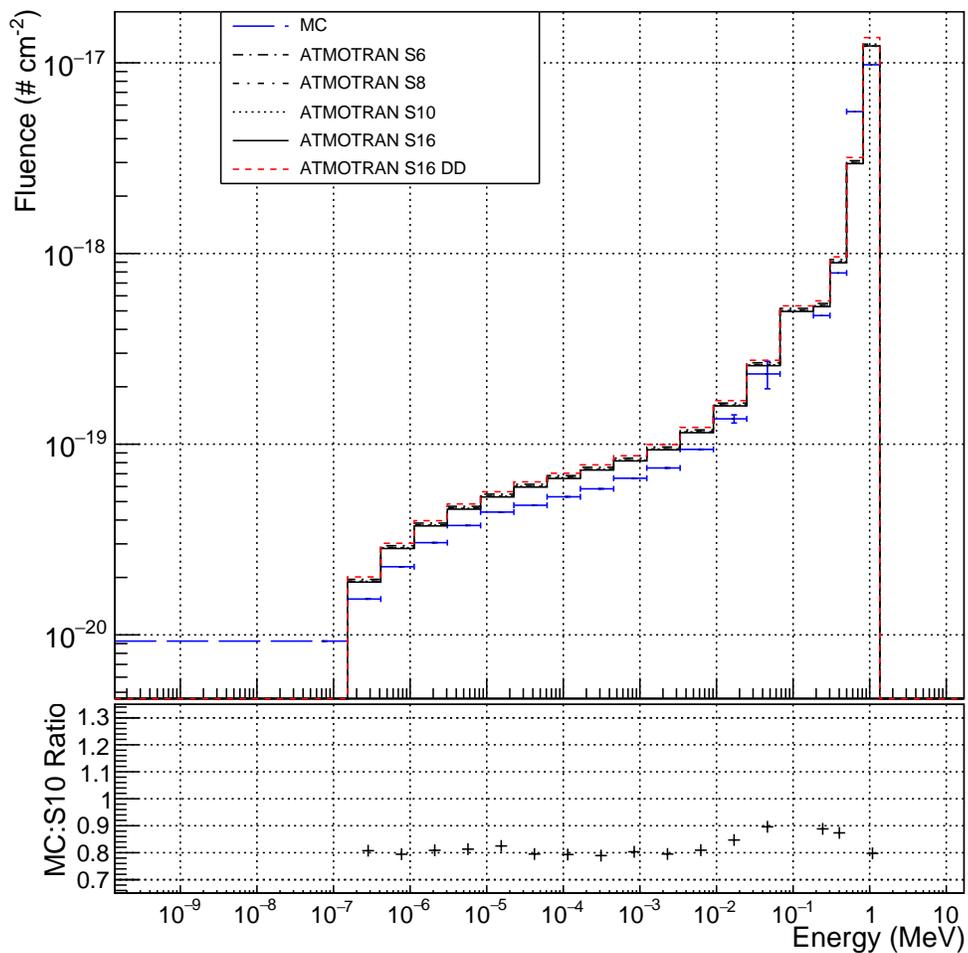


Figure 46: Energy spectrum at 1000 km $\theta = 0$, for 1 MeV source at 100 km

1MeV_30km

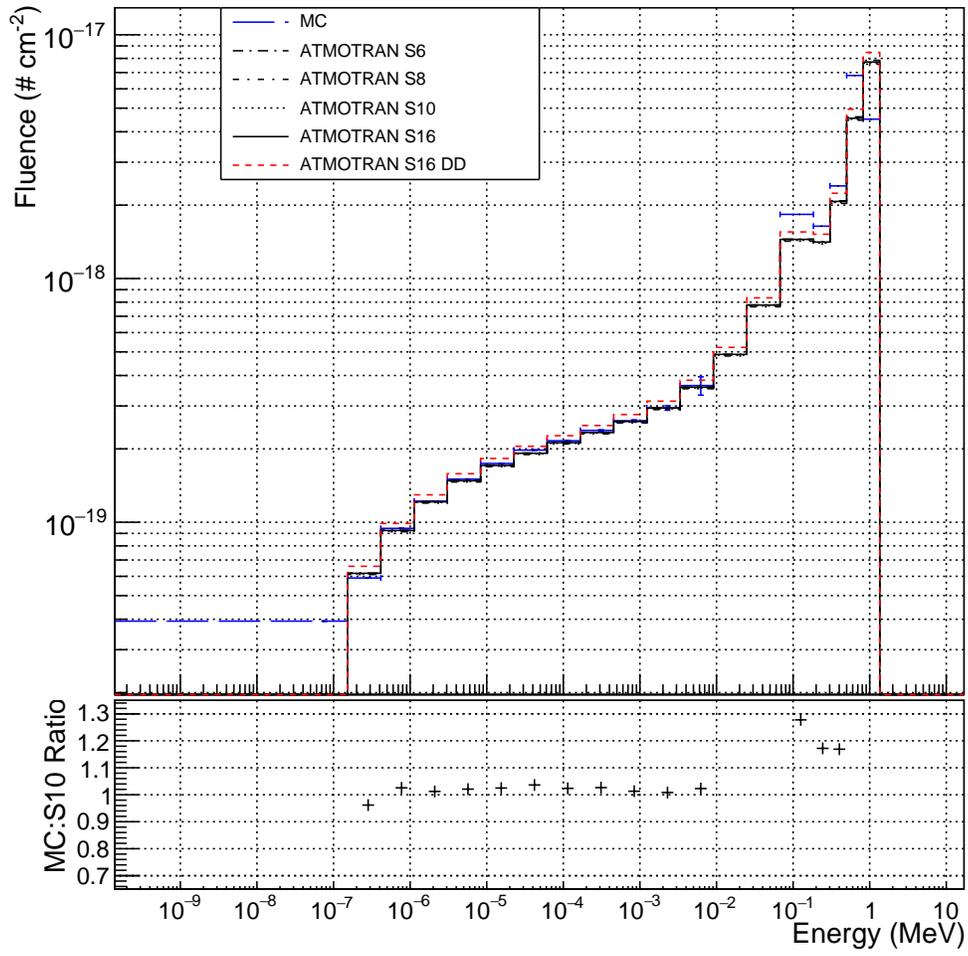


Figure 47: Energy spectrum at 1000 km $\theta = 0$, for 1 MeV source at 30 km

1MeV_20km

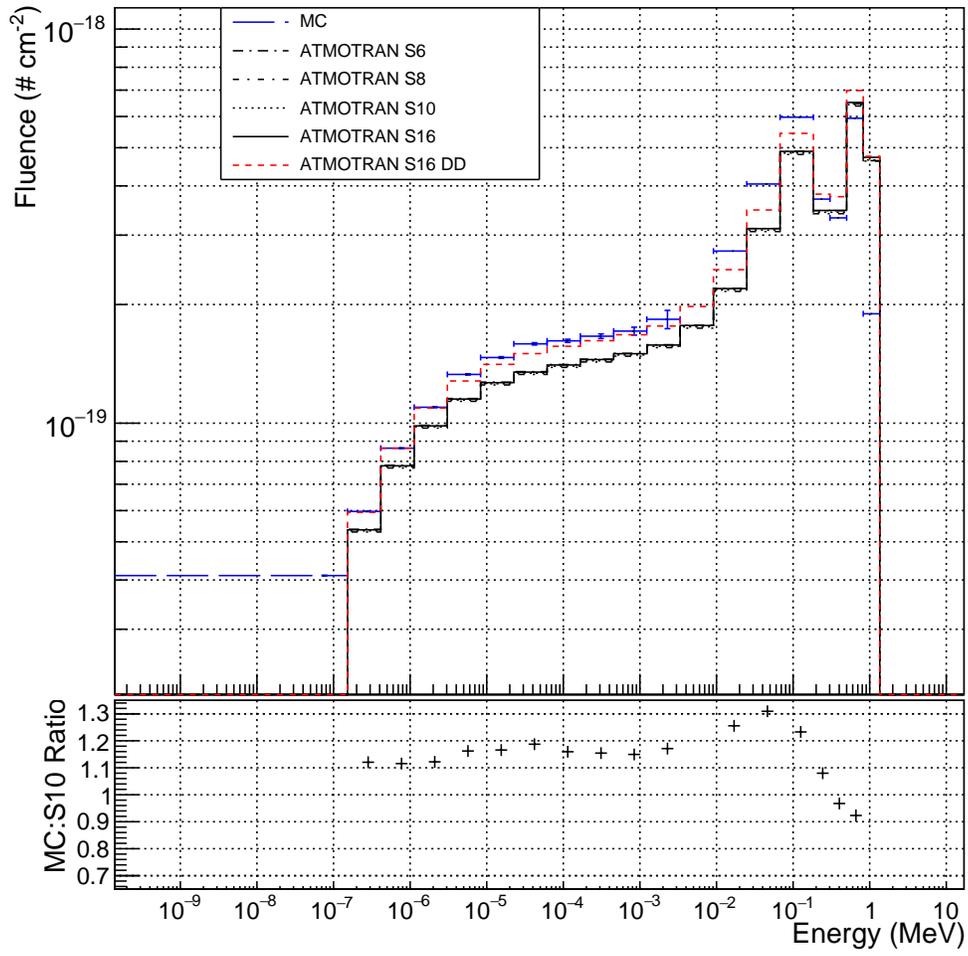


Figure 48: Energy spectrum at 1000 km $\theta = 0$, for 1 MeV source at 20 km

7.3 Fluence vs Angle Comparison

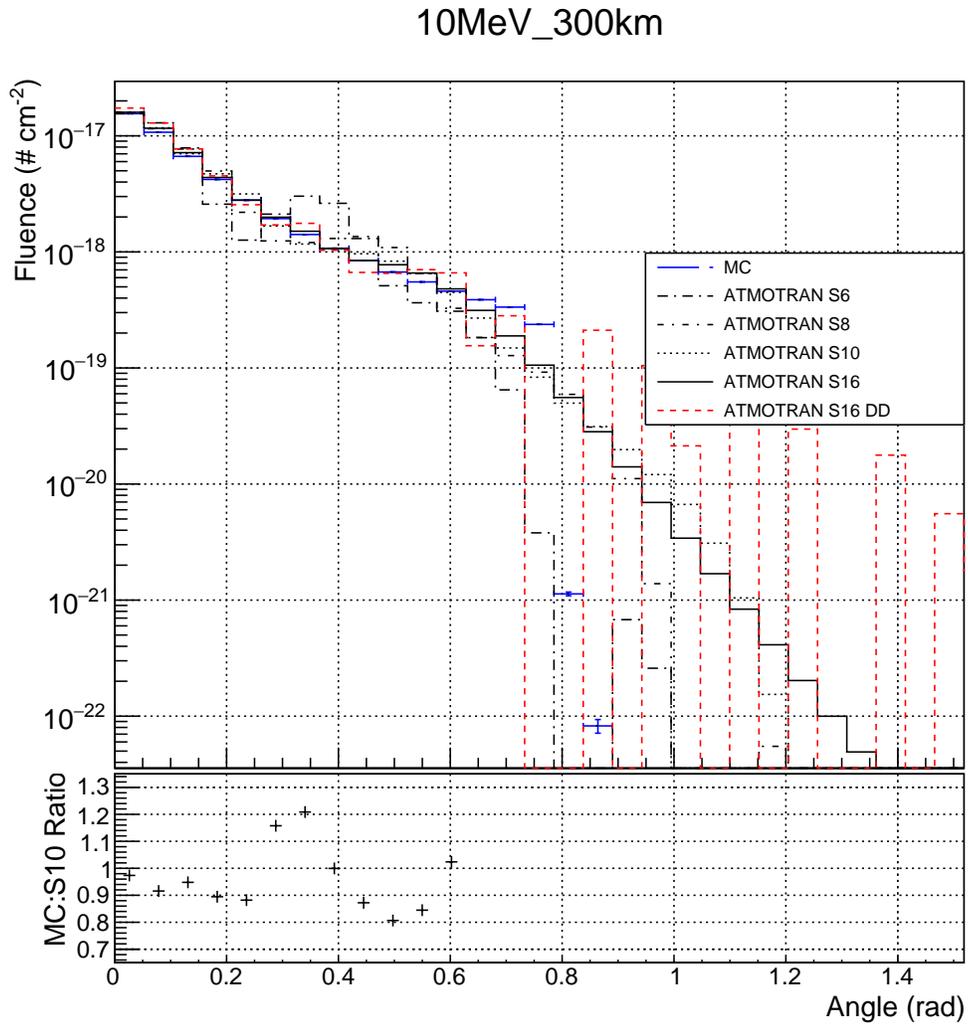


Figure 49: Fluence vs polar angle at 1000 km, for 10 MeV source at 300 km

10MeV_100km

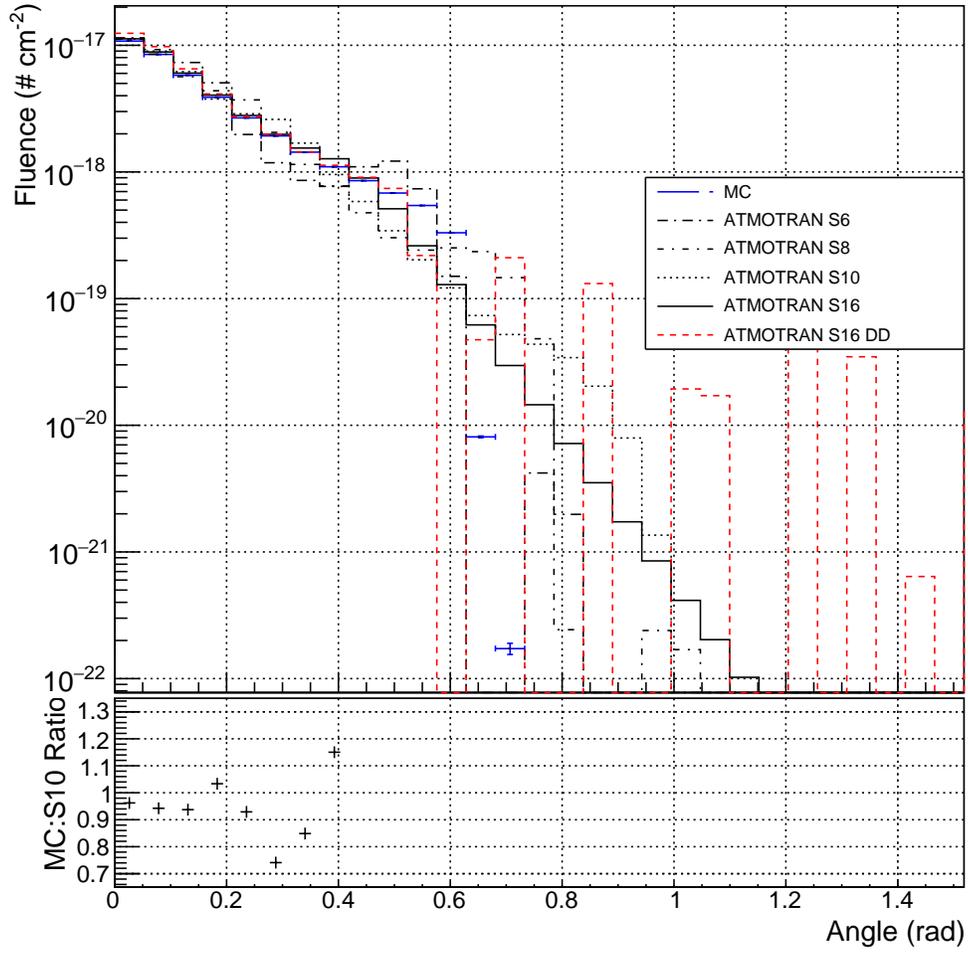


Figure 50: Fluence vs polar angle at 1000 km, for 10 MeV source at 100 km

10MeV_30km

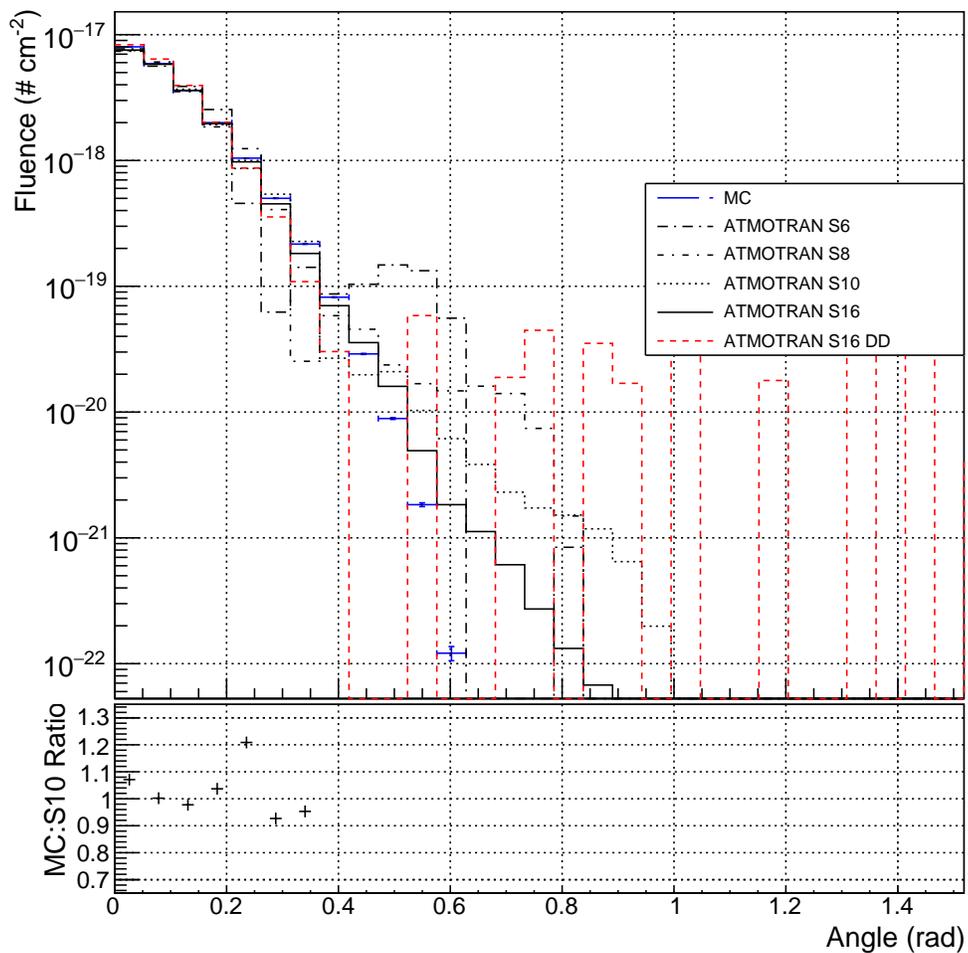


Figure 51: Fluence vs polar angle at 1000 km, for 10 MeV source at 30 km

10MeV_20km

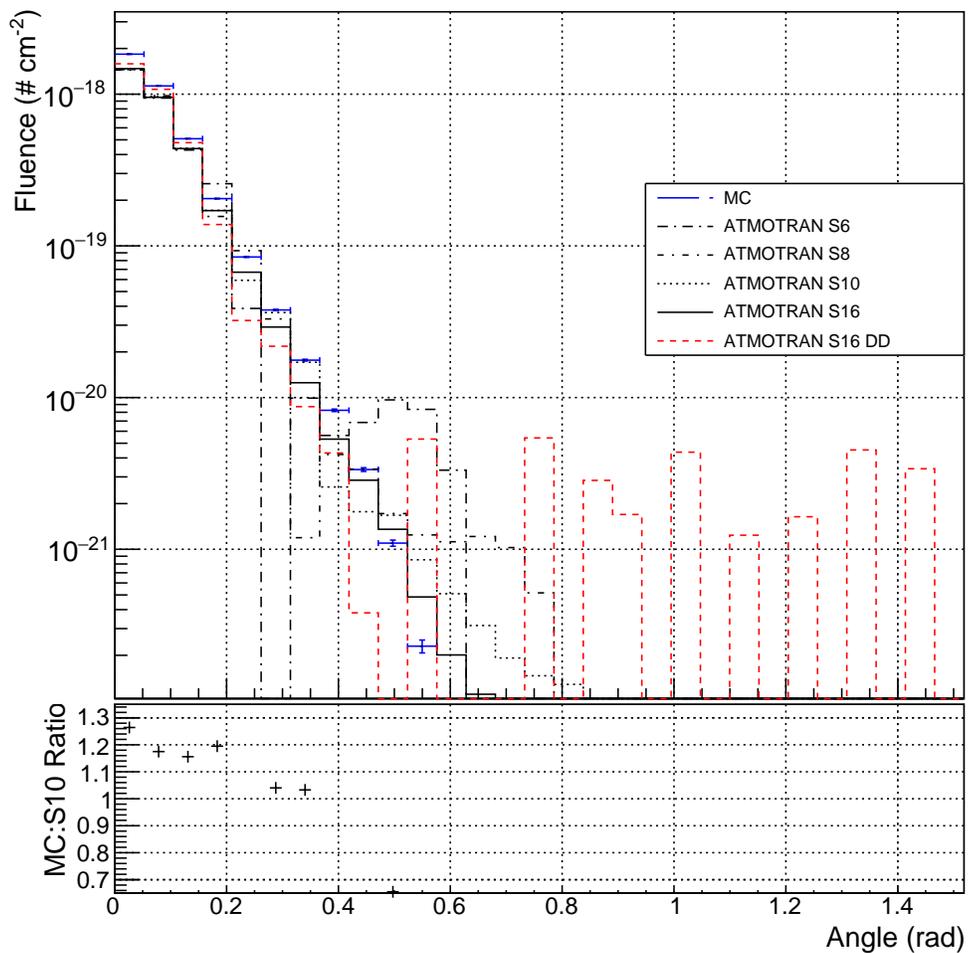


Figure 52: Fluence vs polar angle at 1000 km, for 10 MeV source at 20 km

5MeV_300km

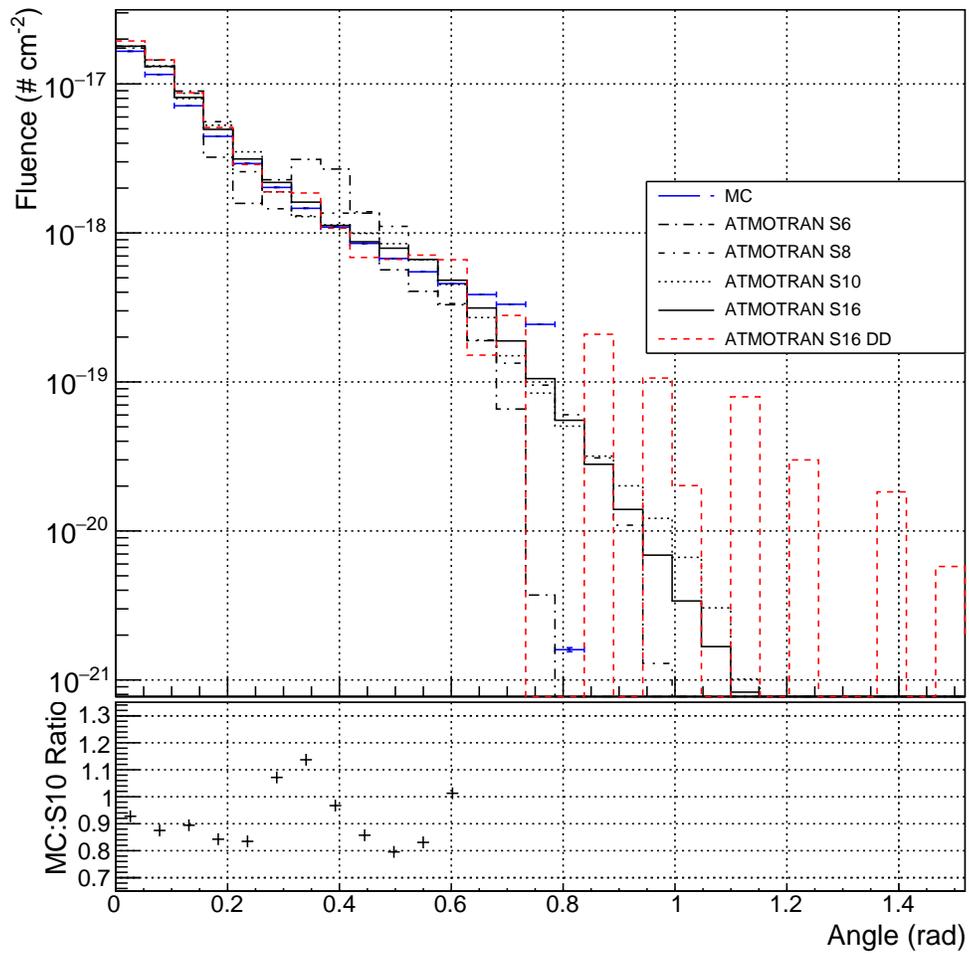


Figure 53: Fluence vs polar angle at 1000 km, for 5 MeV source at 300 km

5MeV_100km

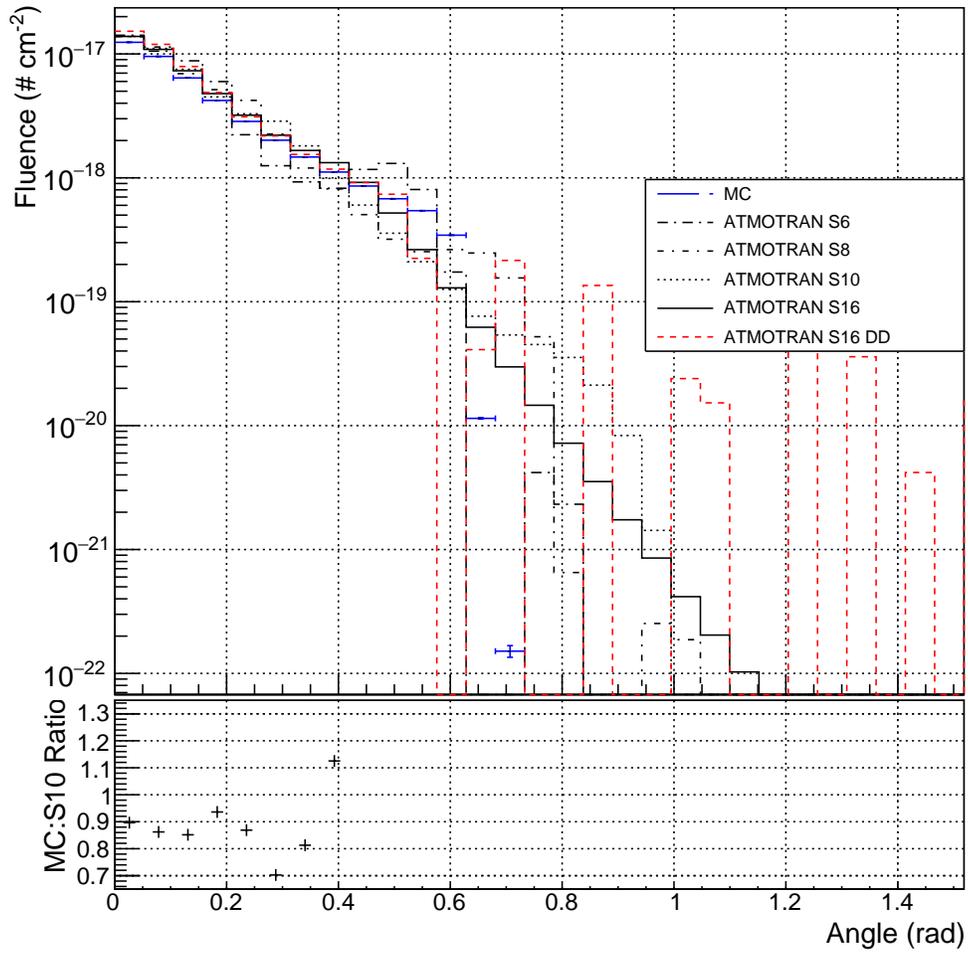


Figure 54: Fluence vs polar angle at 1000 km, for 5 MeV source at 100 km

5MeV_30km

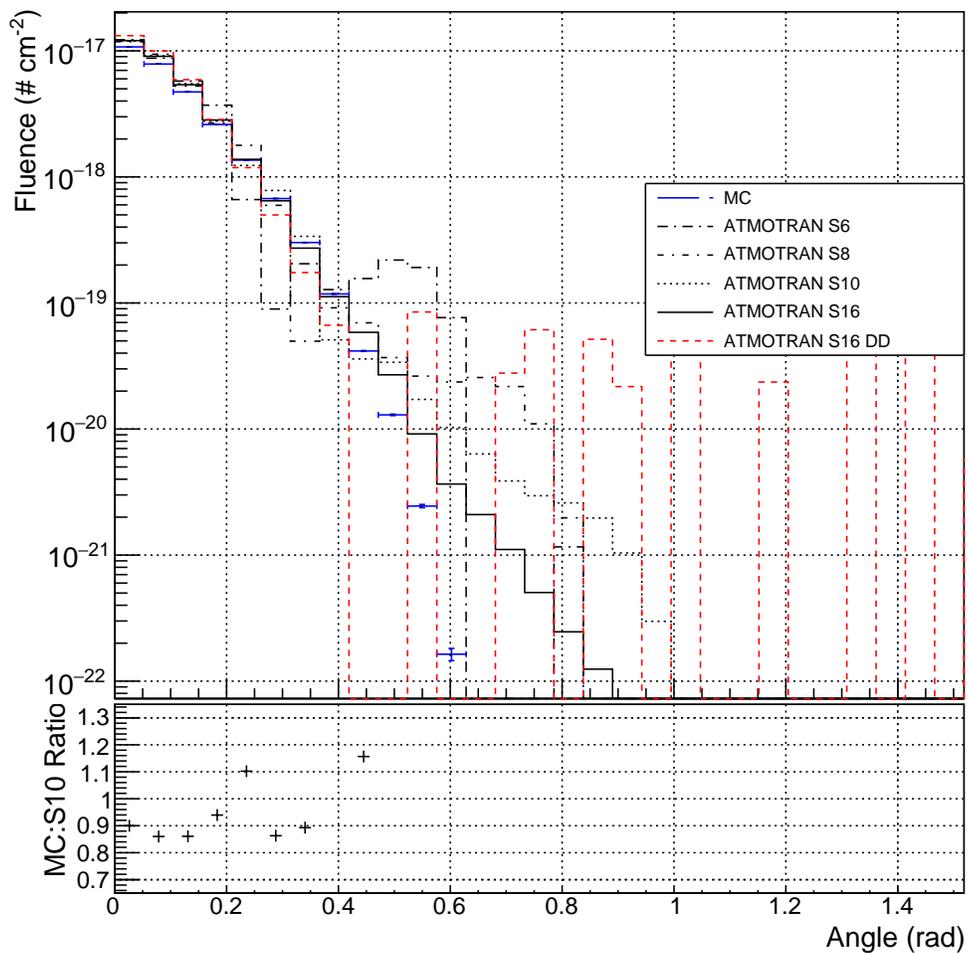


Figure 55: Fluence vs polar angle at 1000 km, for 5 MeV source at 30 km

5MeV_20km

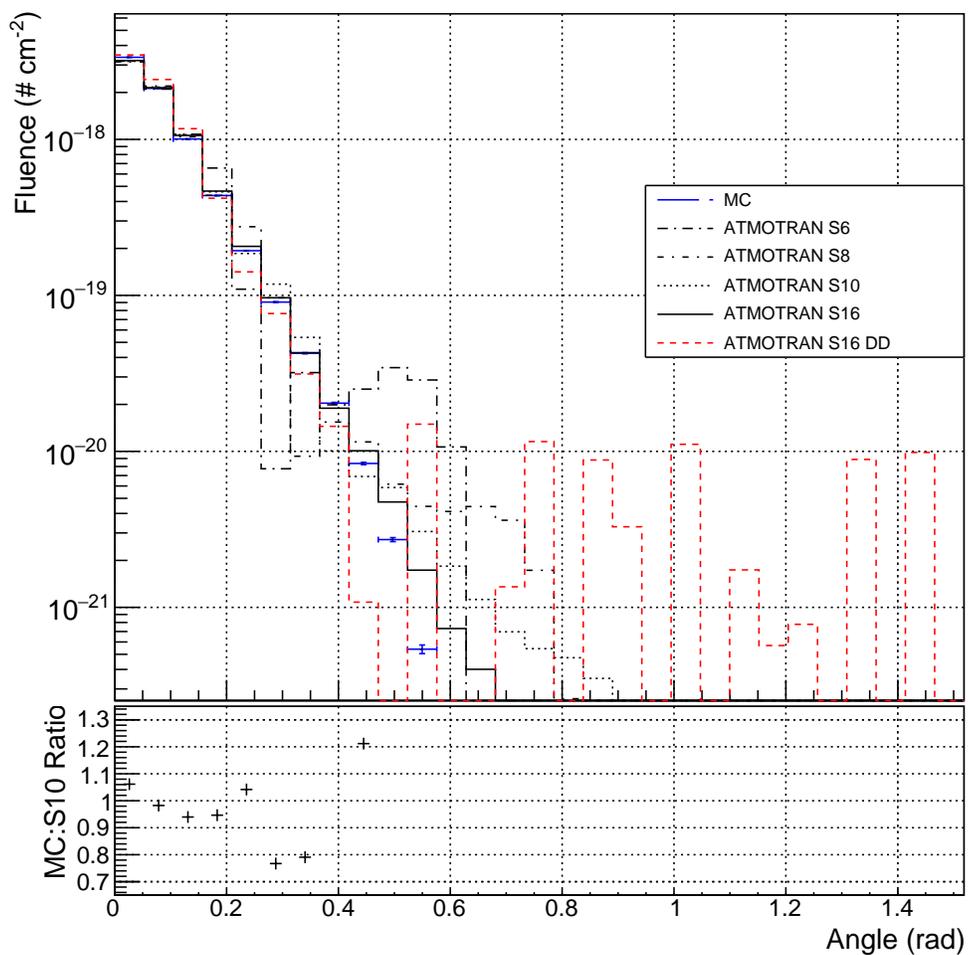


Figure 56: Fluence vs polar angle at 1000 km, for 5 MeV source at 20 km

2MeV_300km

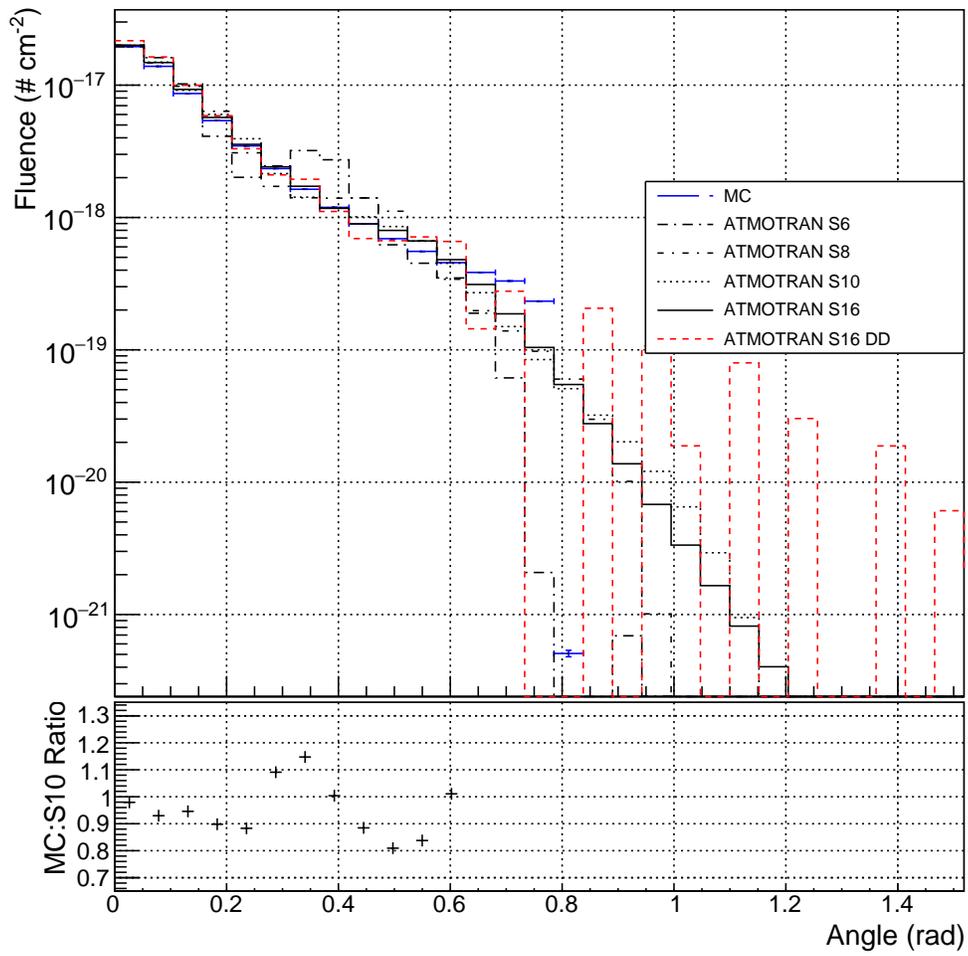


Figure 57: Fluence vs polar angle at 1000 km, for 2 MeV source at 300 km

2MeV_100km

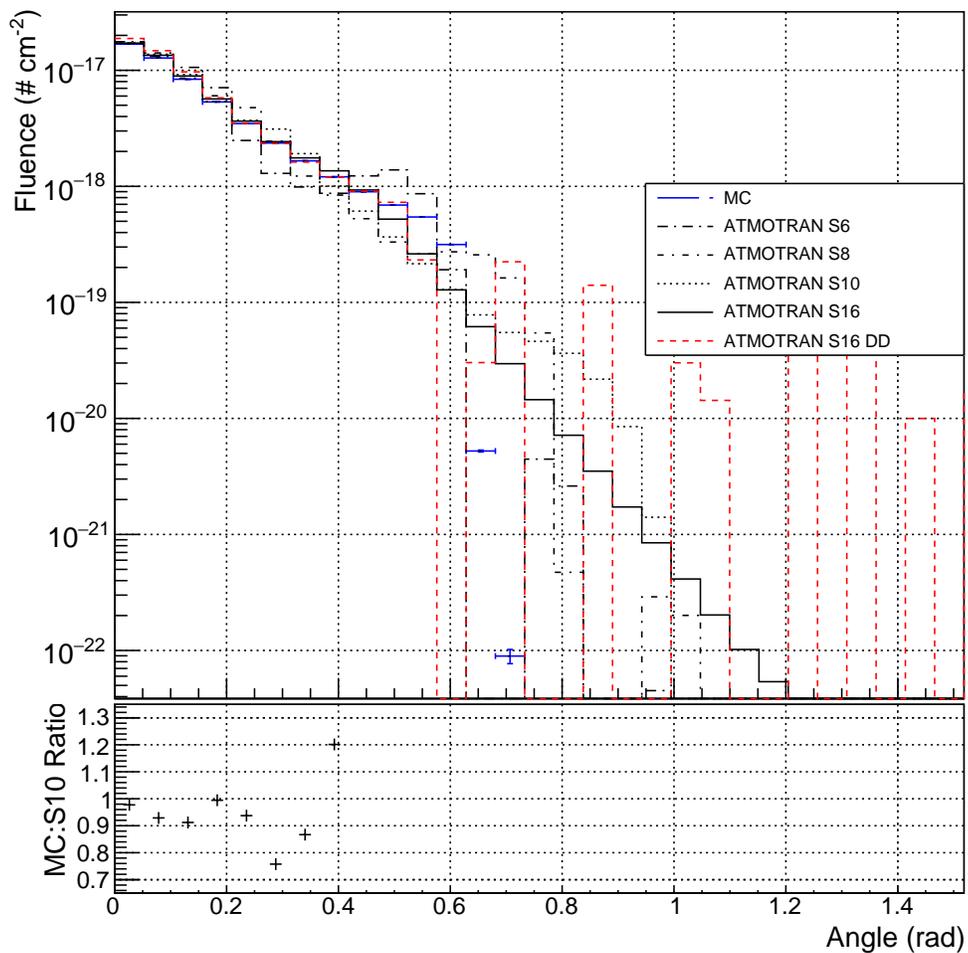


Figure 58: Fluence vs polar angle at 1000 km, for 2 MeV source at 100 km

2MeV_30km

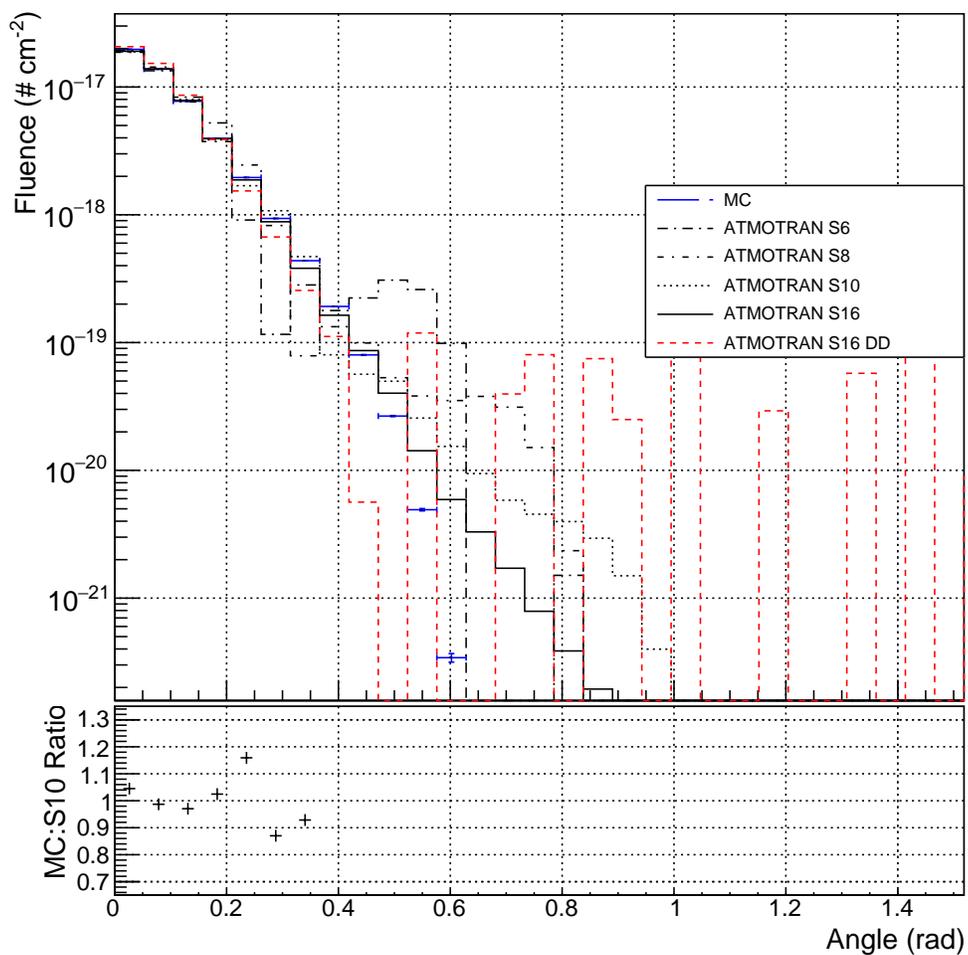


Figure 59: Fluence vs polar angle at 1000 km, for 2 MeV source at 30 km

2MeV_20km

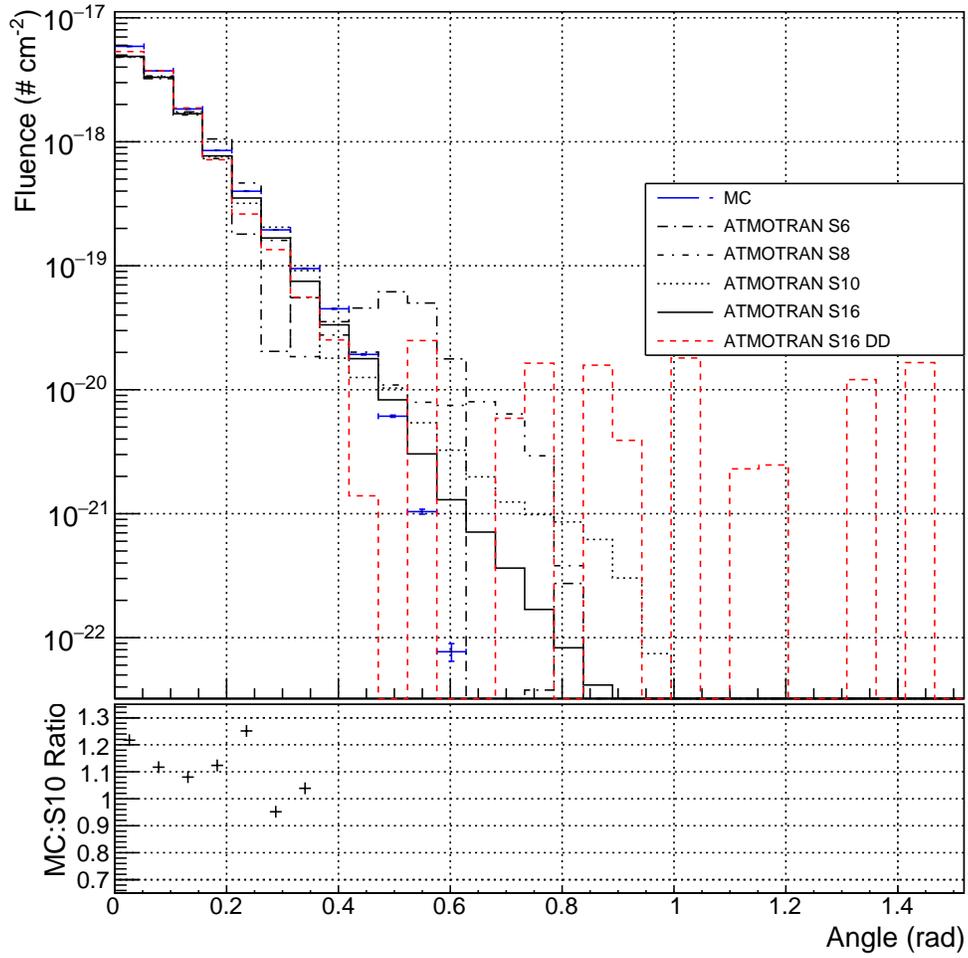


Figure 60: Fluence vs polar angle at 1000 km, for 2 MeV source at 20 km

1MeV_300km

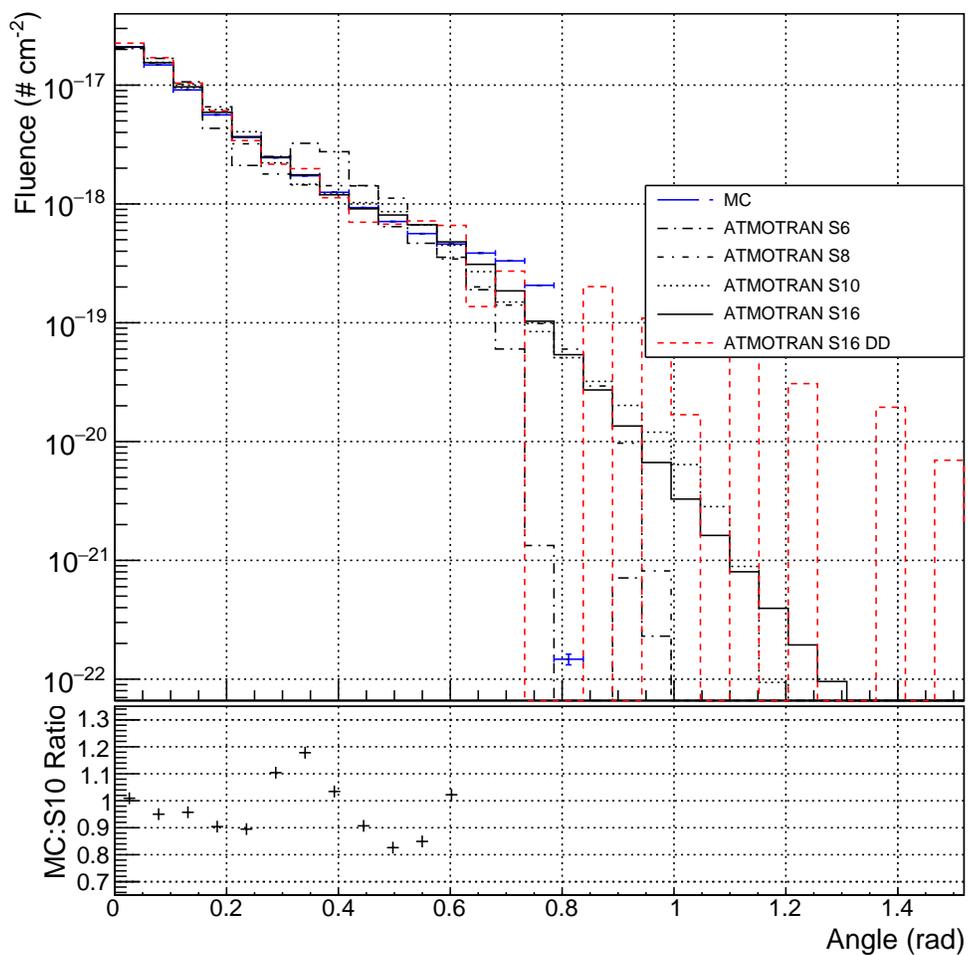


Figure 61: Fluence vs polar angle at 1000 km, for 1 MeV source at 300 km

1MeV_100km

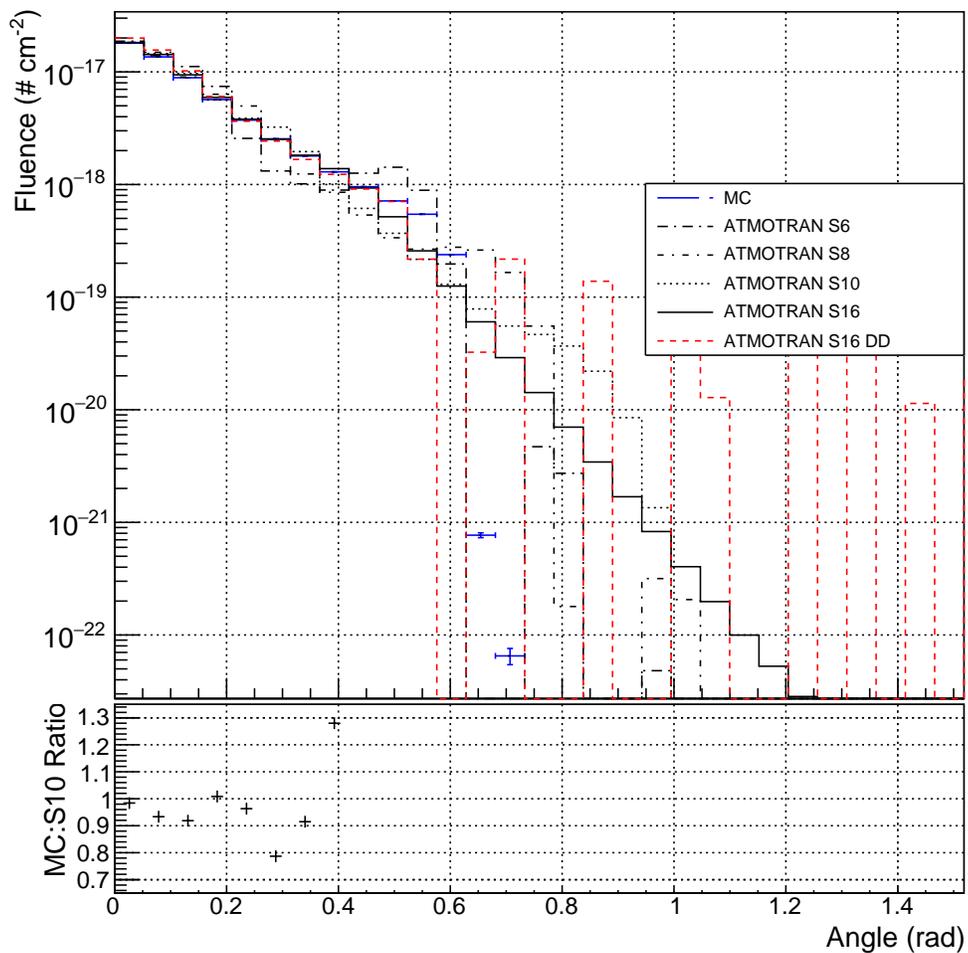


Figure 62: Fluence vs polar angle at 1000 km, for 1 MeV source at 100 km

1MeV_30km

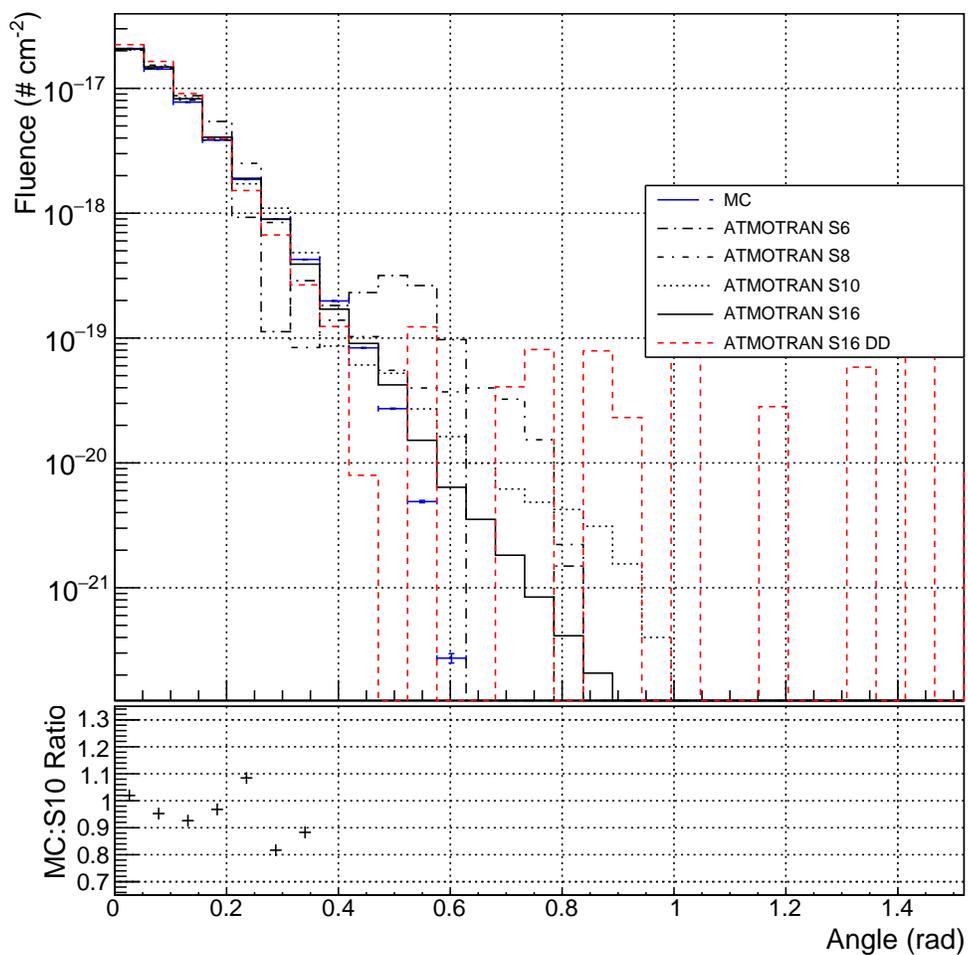


Figure 63: Fluence vs polar angle at 1000 km, for 1 MeV source at 30 km

1MeV_20km

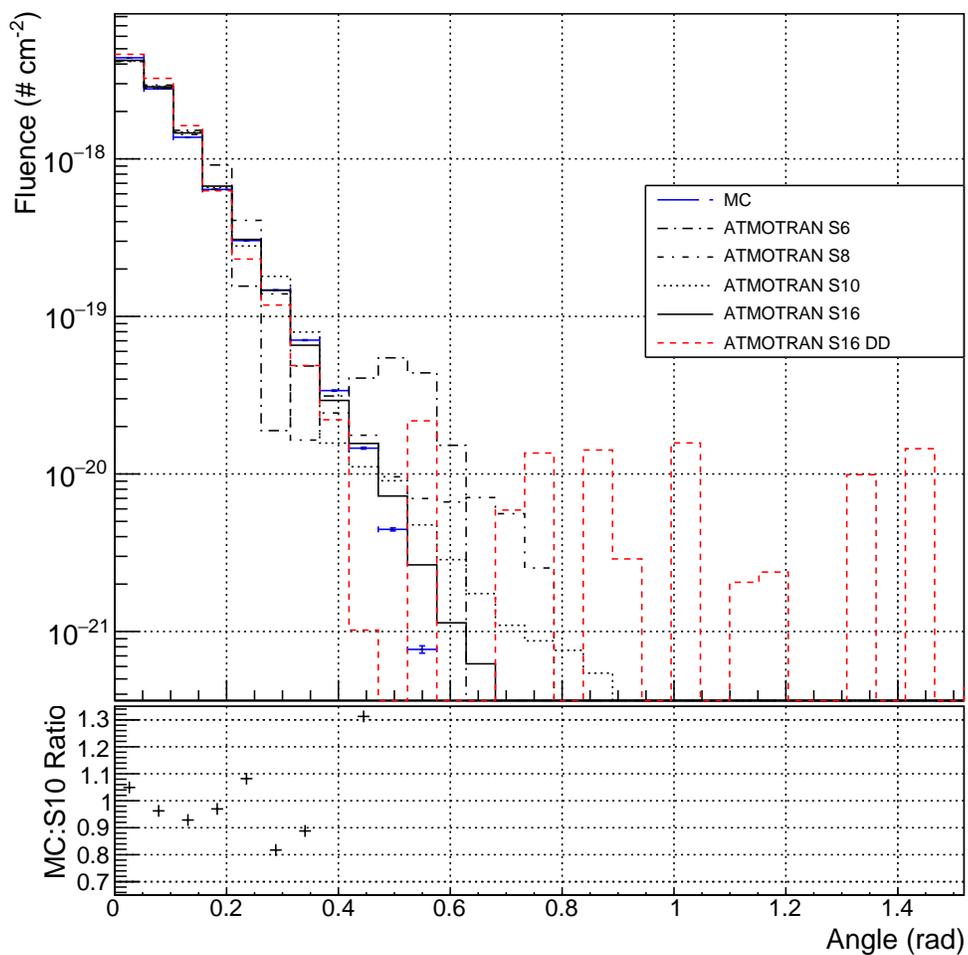


Figure 64: Fluence vs polar angle at 1000 km, for 1 MeV source at 20 km

7.4 Appendix C - NJOY Input Files

Multigroup cross sections are included in ATMOTRAN via direct read-in of NJOY output files. NJOY produces pointwise and multigroup cross sections and related quantities from evaluated data in the ENDF format. The currently available version is NJOY 2016, maintained and distributed by LANL. The NJOY source code was accessed online at www.njoy.lanl.gov. The ENDF evaluated raw data files were downloaded from

t2.lanl.gov/nis/data/endl/endlvii.1-n.html. From this site the “raw eval” data file can be downloaded for any isotope. Several isotopes are required to build a realistic atmosphere, as detailed in the atmospheric geometry section. This site also provides the material number for each isotope. The material number is required within the NJOY input file. NJOY is run with the command `njoy < input` indicating a text file runcard to be used as an input argument. On the second line is the number 70, referencing the ENDF raw eval file, that in this case would need to be named `tape70`. This program reads in these data and makes them available to the `geometry` program, which combines these data with the physical geometry to arrive at useful cross sections. In the following code listings are included the input files to generate multigroup data for a nitrogen (Listing 1) and oxygen (Listing 2) atmosphere. The energy scale used by ATMOTRAN and required for NJOY is shown in Table 24.

Listing 1: NJOY input file to create ATMOTRAN multigroup data for ^{14}N

```
moder
70 -21
reconr
-21 -22
'pendf tape for 14N' /
725 3/
.005 0.0 0.0 0.0/
'14N from ENDF'/
'proc by njoy'/
'see orig endf/b-vii'/
0/
broadr
-21 -22 -23
725 1 0 0 0 /
.005/
0 / temperature
0/
groupr
-21 -23 0 -26
725 3 3 3 3 1 1 1 /
'14N '/
0 / temperature
1.0e10 /
3 2/
6 2/
0/
0/
stop
```

Listing 2: NJOY input file to create ATMOTRAN multigroup data for ^{16}O

```
moder
70 -21
reconr
-21 -22
'pendf tape for 160' /
825 3/
.005 0.0 0.0 0.0/
'16I from ENDF'/
'proc by njoy'/
'see orig endf/b-vii'/
0/
broadr
-21 -22 -23
825 1 0 0 0 /
.005/
0 / temperature
0/
groupr
-21 -23 0 -26
825 3 3 3 3 1 1 1 /
'160 '/
0 / temperature
1.0e10 /
3 2/
6 2/
0/
0/
stop
```

Table 24: Energy scale used in ATMOTRAN

Bin edge	Energy (eV)
0	1.390e-04
1	1.520e-01
2	4.140e-01
3	1.130e+00
4	3.060e+00
5	8.320e+00
6	2.260e+01
7	6.140e+01
8	1.670e+02
9	4.540e+02
10	1.235e+03
11	3.350e+03
12	9.120e+03
13	2.480e+04
14	6.760e+04
15	1.840e+05
16	3.030e+05
17	5.000e+05
18	8.230e+05
19	1.353e+06
20	1.738e+06
21	2.232e+06
22	2.865e+06
23	3.680e+06
24	6.070e+06
25	7.790e+06
26	1.000e+07
27	1.200e+07
28	1.350e+07
29	1.500e+07
30	1.700e+07

References

- [1] K. D. Lathrop and F. W. Brinkley, “TWOTRAN SPHERE: A FORTRAN Program to Solve the Multigroup Transport Equation in Two-Dimensional Spherical Geometry”, Los Alamos Report LA-4567, 1970.
- [2] R. E. Alcouffe, R. S. Baker, J. A. Dahl, S.A. Turner, and Robert Ward, “PARTISN: A Time-Dependent, Parallel Neutral Particle Transport Code System”, Los Alamos Report LA-UR-08-07258, 2008.
- [3] T. W. Armstrong, K. C. Chandler, and J. Barish, “Calculations of Neutron Flux Spectra Induced in the Earth’s Atmosphere by Galactic Cosmic Rays”, *Journal of Geophysical Research*, **78**(16), pp. 2715-2726, 1973.
- [4] M. Z. Jacobson, “Fundamentals of atmospheric modeling”, Cambridge University Press, Second Edition, New York, 2005.
- [5] J. M. Picone et al, “NRLMSISE-00 empirical model of the atmosphere: Statistical comparisons and scientific issues”, *J. Geophys. Res.*, **107**(A12), 2002.
- [6] A. A. Artamonov et al., “Neutron monitor yield function for solar neutrons: A new computation”, *J. Geophys. Res. Space Phys.*, **121**, pp. 117-128, 2016.
- [7] S. V. Poluianov et al., “Production of cosmogenic isotopes ^7Be , ^{10}Be , ^{14}C , ^{22}Na , and ^{36}Cl in the atmosphere”, *J. Geophys. Res. Atmo.*, **121**(13), pp. 8125-8136, 2016.
- [8] A. Mishev and I. Usoskin, “Numerical model for computation of effective and ambient dose equivalent at flight altitudes: Application for dose assessment during GLEs”, *J. Space Weather and Space Climate*, **5**(A10), 2015).
- [9] A. V. Grigoriev et al., “Thunderstorm neutrons in near space: Analyses and numerical simulation”, *J. Geophys. Res: Space Phys.*, **115**(A8), 2010.
- [10] A. Drozdov, A. Grigoriev, and Y. Malyskin, “Assessment of thunderstorm neutron radiation environment at altitudes of aviation flight”, *J. Geophys. Res: Space Phys.*, **118**(2), pp. 947-955, 2013.
- [11] B. E. Carlson, N. G. Lehtinen, and U. S. Inan, “Neutron production in terrestrial gamma ray flashes”, *J. Geophys. Res: Atmospheres*, **115**, pp. 1-24, 2010.
- [12] X-5 Monte Carlo Team, “MCNP - A General Monte Carlo N-Particle Transport Code, Version 5.” Los Alamos Report LA-UR-03-1987, 2003.

- [13] D. R. Culp et al., “Monte Carlo Simulation of Atmospheric Neutron Transport at High Altitudes Using MCNP”, Air Force Institute of Technology Report AFIT/EN-TR-90-5, 1990.
- [14] R. C. Byrd, “Atmospheric transport of neutrons and gamma-rays from a high-altitude nuclear detonation”, Los Alamos Report LA-12962-MS, 1995.
- [15] R. C. Byrd and B. D. Heerema, “Atmospheric transport of neutrons and gamma-rays from near-horizon nuclear detonations”, Los Alamos Report LA-13099-MS, 1996.
- [16] T. E. Booth, “MCNP Variance Reduction Examples”, Los Alamos Report LA-UR-12-25907, 2004.
- [17] W. C. Sailor, “The MCD code system for atmospheric and exo-atmospheric time of flight simulation.”, Paper 203346, *Proceedings of Nuclear and Emerging Technologies for Space*, Atlanta GA, 2009.
- [18] K. D. Lathrop and F. W. Brinkley, “Theory and Use of the General-Geometry TWOTRAN Program”, Los Alamos Report LA-4432, 1970.
- [19] K. D. Lathrop and F. W. Brinkley, “TWOTRAN-II: An Interfaced, Exportable Version of the TWOTRAN Code for Two-Dimensional Transport”, Los Alamos Report, LA-4848, 1973.
- [20] J. E. Morel and G. R. Montry, “Analysis and elimination of the discrete-ordinates flux dip”, *Transport Theory and Statistical Physics*, **13**(5), pp. 615-633, 1984.
- [21] T. R. Hill and W. H. Reed, “TIMEX: a time dependent explicit discrete ordinates program for the solution of multigroup transport equations with delayed neutrons”, Los Alamos Report LA-6201-MS, 1976.
- [22] W. R. Martin, C. E. Yehnert, L. Lorence and J. J. Duderstadt, “Phase-space finite element methods applied to the first-order form of the transport equation”, *Annals of Nuclear Energy*, **8**, pp. 633-646, 1981.
- [23] T. S. Palmer and M. L. Adams, “Curvilinear Geometry Transport Discretizations in the Thick Diffusion Limit”, *Proceedings of the International Conference Mathematical Methods and Supercomputing in Nuclear Application*, 1993.
- [24] E. E. Lewis and W. F. Miller Jr., “Computational Methods of Neutron Transport”, Wiley-Interscience, New York, 1984.
- [25] B. G. Carlson, “Solution of the Transport Equation by S_N Approximation”, Los Alamos Report LA-1599, 1953.

- [26] K. D. Lathrop, “Ray Effects in Discrete Ordinates Equations”, *Nuclear Science and Engineering*, **32**(3), pp. 357-369, 1968.
- [27] K. D. Lathrop and B. G. Carlson, “Discrete Ordinates Angular Quadrature of the Neutron Transport Equation”, Los Alamos Report LA-3186, 1965.
- [28] B. W. Patton and J. P. Holloway, “Application of preconditioned GMRES to the numerical solution of the neutron transport equation”, *Annals of Nuclear Energy*, **29**, pp. 109-136, 2002.
- [29] B. W. Patton and J. P. Holloway, “Some Remarks on GMRES for Transport Theory”, *Nuclear Mathematical and Computational Sciences: A Century in Review, A Century Anew*, Gatlinburg, Tenn., American Nuclear Society, 2003.
- [30] T. R. Hill, “ONETRAN: A Discrete Ordinates Finite Element Code for the Solution of the One-Dimensional Multigroup Transport Equation”, Los Alamos Report LA-5990-MS, 1975.
- [31] E. Machorro, “Discontinuous Galerkin finite element method applied to the 1-D spherical neutron transport equation”, *Journal of Computational Physics*, **223**(1), pp. 67-81, 2007.
- [32] M. L. Adams, “Discontinuous Finite Element Transport Solutions in Thick Diffusive Problems”, *Nuclear Science and Engineering*, (137), pp. 298-333, 2001.
- [33] J.E. Morel, “A Hybrid Collocation-Galerkin- S_N method for Solving the Boltzmann Transport Equation”, *Nuclear Science and Engineering*, **101**(1), pp. 72-87, 1989.
- [34] W. F. Walters, “Use of Chebyshev-Legendre Quadrature Set in Discrete-Ordinates Codes”, *Proceedings of the Conference on Numerical Methods in High Temperature Physics*, Los Alamos Report LA-11342-C, 1988.
- [35] W. F. Walters and J. E. Morel, “Investigation of Linear-Discontinuous Angular Differencing for the 1-D Spherical Geometry S_N Equations”, *Proceedings of the International Topical Meeting on Advances in Mathematics, Computation and Reactor Physics*, 1990.
- [36] E. W. Larsen, “The Asymptotic Diffusion Limit of Discretized Transport Problems”, *Nuclear Science and Engineering*, **112**, pp. 336-346, 1992.
- [37] S. Hamilton, M. Benzi, and J. Warsa, “Negative Flux Fixups in Discontinuous Finite Element S_N Transport”, *Proceedings of International Conference on Mathematics, Computational Methods & Reactor Physics*, 2009.
- [38] K. D. Lathrop, “Spatial Differencing of the Transport Equation: Positivity vs. Accuracy”, *J. Comp. Phys.*, **4**(2), pp. 237-248, 1970.

- [39] W. H. Reed, T. R. Hill, F. W. Brinkley, K. D. Lathrop, “TRIPLET: A Two-Dimensional Multigroup, Triangular-Mesh, Planar Geometry, Explicit Transport Code”, Los Alamos Report LA-5428-MS, 1973.
- [40] E. W. Larsen, J.E. Morel, W. F. Miller Jr., “Asymptotic Solutions of Numerical Transport Problems in Optically Thick, Diffusive Regimes”, *J. Comp. Phys.*, **69**(2), pp. 283-324, 1987.
- [41] J. S. Warsa and A. K. Prinja, “p-Adaptive Numerical Methods for Particle Transport”, *Transport Theory and Statistical Physics*, **28**(3), pp. 229-270, 1999.
- [42] E. W. Larsen, “Diffusion-synthetic Acceleration Methods for Discrete Ordinates Problems”, *Transport Theory and Statistical Physics*, **13**(1-2), pp. 107-126, 1983.
- [43] T. Ohnishi, “Finite element method applied to reactor physics problems”, *J. Nuc. Sci. and Tech.*, **8**, pp. 717-720, 1971.
- [44] C. C. Pain, M. D. Eaton, R .P. Smedley-Stevenson, A. J. H. Goddard, M. D. Piggott, and C. R. E. de Oliveira, “Streamline upwind Petrov-Galerkin methods for the steady-state Boltzmann transport equation”, *Computer methods in applied mechanics and Engineering*, **195**, pp. 4448-4472, 2006.
- [45] G. Gunnebaud, B. Jacob, “Eigen v3”, Available online at: <http://eigen.tuxfamily.org>, 2010.
- [46] R. Brun and F. Rademakers, “ROOT - An object-oriented data analysis framework”, *Nuclear Instruments and Methods Section A*, **389**(1-2), pp. 81-86, 1997.
- [47] J. D. Hunter, “Matplotlib: A 2D Graphics Environment”, *Computing in Science and Engineering*, **9**(3), pp. 90-95, 2007.
- [48] Free Software Foundation, “GNU Automake Manual”, Available online at: <https://www.gnu.org/software/automake/manual>, 2020.
- [49] R. MacFarlane, D. W. Muir, R. M. Boicourt, A. C. Kahler III and J. L. Conlin, “The NJOY Nuclear Data Processing System, Version 2016”, Los Alamos Report LA-UR-17-20093, 2017.
- [50] M.B. Chadwick, M. Herman, P. Obloinsk, M.E. Dunn, Y. Danon, A.C. Kahler, D.L. Smith, B. Pritychenko, G. Arbanas, R. Arcilla, R. Brewer, D.A. Brown, R. Capote, A.D. Carlson, Y.S. Cho, H. Derrien, K. Guber, G.M. Hale, S. Hoblit, S. Holloway, T.D. Johnson, T. Kawano, B.C. Kiedrowski, H. Kim, S. Kunieda, N.M. Larson, L. Leal, J.P. Lestone, R.C. Little, E.A. McCutchan, R.E. MacFarlane, M. MacInnes, C.M. Mattoon, R.D. McKnight, S.F. Mughabghab, G.P.A. Nobre, G. Palmiotti, A. Palumbo, M.T. Pigni, V.G. Pronyaev, R.O. Sayer, A.A. Sonzogni, N.C. Summers, P. Talou, I.J. Thompson, A. Trkov, R.L. Vogt, S.C.

- van der Marck, A. Wallner, M.C. White, D. Wiarda, P.G. Young, "ENDF/B-VII.1: Nuclear Data for Science and Technology: Cross Sections, Covariances, Fission Product Yields and Decay Data", *Nuclear Data Sheets*, **112**, p. 2887, 2011.
- [51] W. F. Walters, "Use of the Chebyshev-Legendre Quadrature Set in Discrete Ordinates Codes", Los Alamos Report LA-UR-87-3621, 1987.
- [52] E.W. Larsen and J.B. Keller, "Asymptotic Solution of Neutron Transport Problems for Small Mean Free Paths", *Journal of Mathematical Physics*, **15**, pp. 75-81, 1974.
- [53] C. Lingus, "Analytical Test Cases for Neutron and Radiation Transport Codes", *Proc. Second Conf. Transport Theory*, CONF-710107, pp. 655-659, Los Alamos Scientific Laboratory, 1971.
- [54] The Geant4 Collaboration, "GEANT4 - a simulation toolkit", *Nuclear Instruments and Methods Section A*, **506**, pp. 250-303, 2003.
- [55] K. G. Thompson, J. S. Warsa, K. G. Budge and J. H. Chang, "Capsaicin: Deterministic Thermal Radiative Transfer", Los Alamos Report LA-UR-06-3378, 2006.