

wiscobolt

A free, open-source, deterministic photon-electron Boltzmann transport solver

Muhsin H. Younis
Department of Medical Physics
University of Wisconsin – Madison
Madison, Wisconsin.

First published 10-21-2023
Latest version: 0.1, published 10-21-2023

Disclosures

wiscobolt was created with the support of the National Cancer Institute of the National Institutes of Health under Award Number T32CA009206, administered through the Radiological Sciences Training Program of the University of Wisconsin – Madison. The content in this document is solely the responsibility of the author and does not necessarily represent the official views of the National Institutes of Health.

License

Copyright (C) 2023. Muhsin H. Younis.

wiscobolt is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. wiscobolt is distributed in the hope that it will be useful, but without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose. See the GNU General Public License, located at `LICENSE/gpl-3.0.txt` in the main wiscobolt directory or <https://www.gnu.org/licenses/> for more details.

Acknowledgements

There are several people and entities the author would like to thank for their contribution and assistance.

- Dan Younis, for providing advice with Fortran, numerical methods, and physics more generally, at a critical stage in this program's development.
- The authors of ELSEPA (ELastic Scattering of Electrons and Positrons by neutral Atoms, molecules, and positive ions) – Francesc Salvat, Aleksander Jablonski and Cedric J. Powell. Cedric J. Powell brought ELSEPA to the author's attention. Aleksander Jablonski provided a helpful description of how to operate ELSEPA. Francesc Salvat provided answers to questions the author had relating to ongoing research in electron scattering cross sections. Francesc Salvat also created PENELOPE, a Monte Carlo Boltzmann transport solver. Documentation relating to PENELOPE's 2018 release has been useful in the creation of wiscobolt's cross section generating code.
- The authors of Sandia National Labs' CEPXS program, a photon-electron-positron cross-section generating code, in particular for their 'physics guide,' which has been useful in the creation of wiscobolt's cross section generating code.
- The authors of Sandia National Labs' SCEPTR program, as well as Los Alamos National Labs' Attila program, and finally Varian's Acuros XB program. Publications relating to these solvers and their algorithms have been tremendously influential on wiscobolt.

- The authors of the Geant4DNA project, whose documentation was very informative in relation to low energy electron scattering cross sections.
- The Medical Physics department at the University of Wisconsin – Madison, in particular Weibo Cai and Bryan Bednarz.
- All institutions from which physics data used by wiscobolt has been obtained: The National Institute of Standards and Technology (NIST), the International Atomic Energy Agency (IAEA), and Lawrence Livermore National Laboratories (LLNL).

Foreword

The purpose of this document is to describe wiscobolt, a free and open-source deterministic Boltzmann transport solver, written in the language Fortran. At the moment, wiscobolt is specialized for photon and/or electron transport, with the ability to handle external beams and coupled photon-electron transport. The Boltzmann transport equation provides essentially the density of radiation particles at some location (3 coordinates) and with some momentum (3 coordinates), or equivalently some direction (2 coordinates) and energy (1 coordinate). This is the so-called ‘equation of state’ of radiological physics. Such a thorough description of a system of radiation particles is very useful across a range of fields, most notably for us, in radiology, wherein it can be used to calculate a number of interesting quantities, including energy deposition following radiation therapy and transmission factors used for radiation safety. However, this is a six-dimensional problem, and as such it is computationally demanding. In the early days of computing, solutions were most feasibly obtained using so-called ‘infinite slab geometry,’ wherein one assumes that there is no variation in the source and medium except in one dimension, that being the exact representation of an infinitely wide planar beam incident upon an infinitely wide slab. For practical purposes, this is a great representation of the behavior of the radiation particle density along the central axis in a beam and medium that is laterally much bigger than the range of the particles being transported. Nevertheless, this reduces the problem to one dimension in space and two dimensions in momentum (one dimension in angle, and one dimension in energy), making solution much more feasible. This method of solution was popular in the neutron transport community. On the other hand, the Monte Carlo method of radiation transport, developed by physicists at Los Alamos National Labs, rapidly became the method of choice for research in transport of high energy (\sim MeV) photons and electrons. This method involves random sampling of scattering probability distributions in order to simulate the interaction of radiation particles with a medium. Due to the simplicity of this method, and the wealth of research dedicated to its optimization over the last \sim fifty years, Monte Carlo remains the method of choice in medical physics.

In the last twenty or so years, with the development of faster computers, larger storage availability, and more efficient numerical methods, the study of deterministic, fully six-dimensional Boltzmann transport has grown. Most notably, treatment of the three spatial dimensions with the versatile and efficient ‘finite element’ method has become a cornerstone of modern deterministic Boltzmann transport calculations. At the moment, however, the author is not aware of any free, open-source, finite element photon-electron Boltzmann transport solvers. Such is the motivation for wiscobolt. However, it must be noted that there are a handful that are not open-source and not generally available to one such as the author. These include Sandia National Labs’ SCEPTRE, a very efficient, extremely versatile research program which has been used extensively not only in medical physics, but also the study of the effects of radiation in electronics. Additionally, there is the clinically-oriented Varian Acuros XB, and its progenitor Attila. Notably, Varian Acuros XB is one example of an ongoing shift in radiation therapy treatment planning, from approximate forms of dose calculation such as convolution-superposition, to more sophisticated forms, such as Monte Carlo and deterministic Boltzmann transport, which more appropriately treat material in-

homogeneities as well as transport in a magnetic field (useful for MR-guided radiotherapy) and heavy ion therapy (proton therapy is a great example).

The typical user will likely find the **wiscobolt implementation** document much more useful, in which we describe primarily: 1) the user input, 2) the output of the program, and 3) the exhaustive details of each module and each subroutine contained within the program. There also exists the **wiscobolt physics** document, which describes the physics data and models used to determine, essentially, the interaction probabilities for a given material that are involved in the Boltzmann transport equation. Then, the **wiscobolt verification** document describes a number of numerical studies that have been performed to demonstrate the mathematical *and* physical integrity of the solver. That is, in that document, we verify that we are capable of numerically performing the mathematical operations that we claim here to construct, as well as quantify how well our solver works under some kinds of stress tests, such as a problem whose (known) solution has sharp spatial/energy gradients, or one with particular boundary conditions. The document also demonstrates that we can reproduce experimental results relating to the physics data and models we use, materials, etc., which is important in demonstrating that the solver is using reliable physics data and models that have also been implemented correctly.

In all, wiscobolt is intended to be used by researchers in any field to which MeV photon and/or electron Boltzmann transport can be applied. We hope that wiscobolt, due to its accessibility, will satisfy a need in the research community. We intend to generalize it soon to more particle types (first up: positrons and protons), as well as allow for more generality in the problem geometry. A big drawback of our solver is that it currently can not treat any non-convex geometries, including objects with cavities. While this is not necessarily a fundamental feature of the program, we are not sure when we will have the time to implement this capability (requests will go a long way). Another feature with an undecided ETA is the ability to use elements other than tetrahedra. For this, hexahedra will likely be the first to be included.

The author invites all questions, feedback, comments, or problems, and can be reached most easily at: **myounis@wisc.edu**. It is inevitable that errors will arise during day-to-day use. Therefore, contacting the author about these errors will tremendously benefit wiscobolt. Additionally, this document is not peer-reviewed, and so corrections based on statements made in this document are not only encouraged, but requested. Our confidence in this document is derived from the research from which it was created, as well as the results in the **wiscobolt verification** document.

Table of Contents

1	INTRODUCTION TO BOLTZMANN TRANSPORT	1
1.1	Background	1
1.1.1	The continuity equation	2
1.1.2	Augmenting the continuity equation	5
1.2	Sources	13
1.3	Coupling particles	15
2	DISCRETIZATION	17
2.1	Series expansions	18
2.2	Spatial discretization	21
2.2.1	The finite element method	22
2.2.2	Creating the weak form	23
2.2.3	Applying the FEM	25
2.2.4	Evaluation of spatial inner products for tetrahedra	32
2.3	Angular discretization – S_N	36
2.3.1	Angular quadrature	36
2.3.2	Sweep	40
2.3.3	Boundary conditions in the sweep	43
2.3.4	Summary	45
2.4	Angular discretization – P_N	45
2.4.1	Boundary conditions	49
2.4.2	Summary	50
2.5	Energy discretization – multi-group formalism	50
2.6	Energy discretization – finite element method	54
3	SOLUTION	58
3.1	Iterative methods	59
3.1.1	Energy iteration	59
3.1.2	Source iteration	61
3.1.3	Generalized minimal residual method	65
3.2	Acceleration across solution and discretization methods	66
3.2.1	Acceleration of SI	66
3.2.2	Acceleration of GMRES	67
3.2.3	Diffusion synthetic acceleration	68
3.3	Approximations	68
3.3.1	Neglecting the positron fluence and the electron-photon source	68

3.3.2	The restricted continuous slowing down approximation	69
3.3.3	The extended transport correction	73
3.3.4	Integration of uncollided fluence	78
A	Angular inner products	80
A.1	Analytical solution	86
	A.1.1 Summary	91
A.2	Quadrature solution	92
References		94

1 INTRODUCTION TO BOLTZMANN TRANSPORT

1.1 Background

For a great introduction to radiological physics at a practical level, the reader may be referred to any of the following texts: [1, 2]. For a thorough introduction to the topic of dose calculation, including convolution-superposition methods, Monte Carlo, and Boltzmann transport, the user is strongly recommended to read [3], which was the author’s first introduction to Boltzmann transport. Our goal here is to describe the ‘equation of state’ in radiological physics, and derive it through intuition. This is not a very rigorous treatment, but the hope is that it provides the reader with a basic understanding of what Boltzmann transport *is*, what it can *do*, and why we need it. Notably, we will not discuss methods of Boltzmann transport other than deterministic. That is to say, Monte Carlo methods, which are currently the method of choice in radiological physics [3, 4], will not be described in this document. For those interested in Monte Carlo methods, the reader is referred to [3–6], and there are many more resources that can be found with little effort.

In principle, the ‘equation of state,’ or, the statement which suffices to describe all of the information in our system, should be one that describes the evolution of the density of radiation particles at a position \mathbf{r} and with momentum \mathbf{p} (or, equivalently, direction $\hat{\mathbf{k}}$ and energy E) over time t . This may seem a bit cumbersome, but it really is intuitive: you should (classically) be able to choose a particular region in space, and identify the number particles that span a given solid angle and have a certain range of energy. Of course, this description fully neglects Heisenberg uncertainty, or in other words the quantum mechanical nature of matter. The acceptability of this depends both on the spatial and energetic regime in which we wish to operate, that is, we can not go to energies that are too low nor spatial scales that are too small [5, 7], but a more thorough discussion is reserved for the **wiscobolt physics** document. Nevertheless, the composite position-momentum setting of this problem is referred to as **phase space**. The quantity we want, the particle number density $n(\mathbf{r}, \mathbf{p}, t)$, is simply the density of particles at phase space location (\mathbf{r}, \mathbf{p}) at some time t . It will have units of inverse volume multiplied by inverse cubic momentum. But consider briefly that our use of ‘phase space’ may be a bit of a misnomer, so that we may better understand this quantity as fundamentally classical. In classical physics, if we have a system of N particles, then ‘phase space’ may more generally refer to a $6N$ dimensional coordinate space, where each *point* fully describes every particle’s position and momentum, and thus this one point entirely describes our system [8]. That is because the i th particle, assumed to be arbitrarily localized in classical physics, will have three position coordinates \mathbf{r}_i , with $i = 1, \dots, N$, and three momentum coordinates \mathbf{p}_i . The phase space is then $(\mathbf{r}_1, \dots, \mathbf{r}_N, \mathbf{p}_1, \dots, \mathbf{p}_N)$. A number of methods exist to determine, from initial conditions, the motion of this point for all time, including Newton’s laws, the principle of least action, Hamiltonian mechanics, etc., all of which are equivalent. Now, the direct connection between Boltzmann transport and these methods of classical physics, is that we consider a density $n(\mathbf{r}, \mathbf{p}, t)$ of unbound (radiation)

particles for some position \mathbf{r} and momentum \mathbf{p} , which is most generally:

$$n(\mathbf{r}, \mathbf{p}, t) = \sum_{i=1}^N \delta^3[\mathbf{r} - \mathbf{r}_i(t)] \delta^3[\mathbf{p} - \mathbf{p}_i(t)] \quad (1.1.1)$$

where $\delta^3(\mathbf{r})$ is a three-dimensional Dirac delta function. This statement is, in fact, fully equivalent to the conceptualization of a system as a point (or trajectory, when t runs) in a $6N$ dimensional space. But in a system of $N \approx 10^{23}$, or even more particles, such a statement is not only impractical, but also unnecessary. When these delta functions are placed very close to one another in phase space, such that the location of a given particle in phase space is not known, nor necessary to know up to some resolution, you end up with essentially a smooth, continuous *distribution* of particles that can be used to determine the macroscopic properties of your system.

Having this down, the BTE is simply the equation which governs the macroscopic changes in $n(\mathbf{r}, \mathbf{p}, t)$ for radiation particles, under a number of assumptions. It was originally formulated by Ludwig Boltzmann, a name more familiar to statistical physics than radiological physics, to give a thorough classical treatment to the motion of gas molecules as part of his formulation of kinetic theory. As you will soon see, it can be given a remarkably simple, even *heuristic* construction. However, as we frequently find in physics, simple rules can lead to arbitrarily complicated problems. We will find that for our purposes the BTE is an *integro-differential* equation, which calls upon a number of quantities that can be either experimentally or theoretically obtained and fed into the equation. Nevertheless, a computational solution is quite feasible, and the BTE has found its way into prominence in nuclear physics [9] (largely in neutron transport), condensed matter physics [10] (including but not limited to *phonon* transport), astronomy [11, 12], and medical physics [1, 3]. For instance, in medical physics, it can be used to calculate the energy deposition per unit mass for a given source of particles, otherwise known as **dose**, through the relationship:

$$D(\mathbf{r}) = \int dE S_{col}(\mathbf{r}, E) \varphi(\mathbf{r}, E) \quad (1.1.2)$$

where $\varphi(\mathbf{r}, E)$ is the **fluence** of particles, a quantity related to n , and $S_{col}(\mathbf{r}, E)$ is the **mass collisional stopping power** of the relevant particles of energy E and in the material located about position \mathbf{r} . Dose is a physically and metrically accessible surrogate for cell-killing, and therefore, it is of great importance in radiation therapy treatment planning. This equation, though we will not derive it and have yet to understand the form of $S_{col}(\mathbf{r}, E)$, is a great demonstration of the power of the Boltzmann transport equation. If we were to expand it a little bit, it would clearly represent the tallying of energy-depositing interactions within a medium, weighted by the energy they actually deposit, in a region of space \mathbf{r} .

1.1.1 The continuity equation

We will attempt to look for an equation for n with a simple model system and then generalize. While the result of this section, the continuity equation, is very simple and found in a number

of fields, it is worth discussing here because it is a good way to obtain an intuition for a number of radiometric quantities which we will be important in solving the BTE.

First, we note that energy is an important quantity in the context of medical physics, so we will here switch to phase space coordinates $(\mathbf{r}, \hat{\mathbf{k}}, E)$, where $\hat{\mathbf{k}} = \hat{\mathbf{p}}$, and now n has units of inverse volume-steradian-energy. Now, let's suppose we have radiation particles that can not be absorbed, scattered, nor created. This means that $\hat{\mathbf{k}}$ and E have no way to change, and so the following construction can just as well be carried out for a typical, 3D mass-, number-, or charge density. In fact, what follows is a generic treatment of the continuity equation that may be found in a textbook on fluid dynamics or electrodynamics. For instance, an equivalent, but not as simple, treatment can be found in Chapter 4 of [13]. This treatment, however, follows [3]. Nevertheless, begin with a small volume element centered at \mathbf{r} , with side lengths δx , δy , and δz , and total volume $\delta V = \delta x \delta y \delta z$. Now, we suggest that the dimensions of the box are small enough that, if we know the value of n at \mathbf{r} (for any \mathbf{k} and t), then we know the value of n at the walls of the box, via a Taylor series expansion centered about \mathbf{r} . For instance, we can write n at the wall at $x + \delta x/2$:

$$n(x + \delta x/2, y, z, \hat{\mathbf{k}}, E, t) = n(\mathbf{r}, \hat{\mathbf{k}}, E, t) + \frac{1}{2} \delta x \frac{\partial n}{\partial x}(\mathbf{r}, \hat{\mathbf{k}}, E, t) \quad (1.1.3)$$

And this can be done for each wall. Particles will flow into and out of this volume element. The total number of particles (per unit wavenumber) in this box is:

$$\int_{-\delta x/2}^{\delta x/2} dx' \int_{-\delta y/2}^{\delta y/2} dy' \int_{-\delta z/2}^{\delta z/2} dz' n(\mathbf{r}', \hat{\mathbf{k}}, E, t) = n(\mathbf{r}, \hat{\mathbf{k}}, E, t) \delta x \delta y \delta z = n(\mathbf{r}, \hat{\mathbf{k}}, E, t) \delta V \quad (1.1.4)$$

But we have not allowed for there to be any mechanisms for any particles to change energy nor direction, and so the number of particles of a given energy and direction will not change. Now, we will define **flow** as the rate of change in the total number of particles in a given volume. That is, in this volume, flow δI is defined as:

$$\delta I = \frac{\partial n}{\partial t} \delta V \quad (1.1.5)$$

Now, within the volume element, change in n can only occur due to travel in and out of the volume element through one of the surfaces. We call this **flux** over a given surface. More specifically, we say that flux means the rate of particles escaping the surface per unit area (so, it is also the flow per unit area due to a given surface). How can we express this quantity in terms of n and the given geometry? Over a surface of area A with normal $\hat{\mathbf{n}}$, or with area vector $\mathbf{A} = A\hat{\mathbf{n}}$, and over a time Δt , a volume $|\mathbf{A} \cdot \mathbf{v}| \Delta t$ is spanned by the path length of particles that have velocity \mathbf{v} . The number of particles in this volume is therefore the number of particles that have escaped the surface, and mathematically, this is the density multiplied by the volume. Thus, the change in the number of particles is $n|\mathbf{A} \cdot \mathbf{v}| \Delta t$. The flux has therefore been defined as this quantity divided by the area and time, or $n|\hat{\mathbf{n}} \cdot \mathbf{v}|$. A sign or direction can be given to this quantity with regards to what we consider positive flux (particles entering) and negative flux (particles leaving), in which case we would ascribe the unit vector $\hat{\mathbf{n}}$ to this quantity. Nevertheless, if we are interested in the flux across, say, $x + \delta x/2$, then our flux is $n(x + \delta x/2, y, z)v_x$, and we have a contribution to the overall flow

which is this multiplied by the area of the surface. By another Taylor expansion of n , we can write:

$$\delta I_{x+\delta x/2} = \left[nv_x + \frac{1}{2} \delta x \frac{\partial(nv_x)}{\partial x} \right] \delta y \delta z \quad (1.1.6)$$

The other sides contribute similarly. Putting all of the six contributions together, we have:

$$\delta I = - \left[\frac{\partial(nv_x)}{\partial x} + \frac{\partial(nv_y)}{\partial y} + \frac{\partial(nv_z)}{\partial z} \right] \delta V \quad (1.1.7)$$

Equating to (1.1.5) yields:

$$\frac{\partial n}{\partial t} + \frac{\partial(nv_x)}{\partial x} + \frac{\partial(nv_y)}{\partial y} + \frac{\partial(nv_z)}{\partial z} = 0 \quad (1.1.8)$$

But taking into consideration that v_i are independent of r_i since they are coordinates of our system, we finally have:

$$\frac{\partial n}{\partial t}(\mathbf{r}, \hat{\mathbf{k}}, E, t) + \mathbf{v}(\hat{\mathbf{k}}, E) \cdot \nabla n(\mathbf{r}, \hat{\mathbf{k}}, E, t) = 0 \quad (1.1.9)$$

We recognize this equation as a **continuity equation**, but it does differ from the typical continuity equation one encounters in electrostatics and fluid dynamics with the charge/mass density. In particular, \mathbf{v} is taken as a *variable* here by virtue of its relation to the variables $\hat{\mathbf{k}}$ and E , whereas, it is frequently considered a constant or a vector field. That is, say we have some charge Q_V in a volume V . The charge will decrease at a rate given by the flux across the boundary of V , or ∂V . Therefore, we would say:

$$\frac{dQ_V}{dt} = - \oint_{\partial V} d\mathbf{A} \cdot \mathbf{v}(\mathbf{r}, t) \rho(\mathbf{r}, t) \quad (1.1.10)$$

But dQ_V/dt is also:

$$\frac{dQ_V}{dt} = \frac{d}{dt} \int_V d^3\mathbf{r} \rho(\mathbf{r}, t) \quad (1.1.11)$$

The continuity equation is the form that we obtain when we use the divergence theorem to write (1.1.10) in terms of a volume integral, then throw off the integration, to find:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\mathbf{v} \rho) = 0 \quad (1.1.12)$$

where we then usually define $\mathbf{v}\rho \equiv \mathbf{J}(\mathbf{r}, t)$ as the current density. But our case is actually quite different because we are working in a position-velocity (momentum) phase space, so there is no implicit dependence of \mathbf{v} on \mathbf{r} . Nevertheless, we find it interesting to now define our analog of the current density, the **current density distribution**:

$$\mathbf{j}(\mathbf{r}, \hat{\mathbf{k}}, E, t) = n(\mathbf{r}, \hat{\mathbf{k}}, E, t) \mathbf{v}(\hat{\mathbf{k}}, E) \quad (1.1.13)$$

and you can make the appropriate substitution for \mathbf{v} . Now, we can easily interpret this quantity. If we have some surface $d\mathbf{A}$ at \mathbf{r} , then $\mathbf{j}(\mathbf{r}, \hat{\mathbf{k}}, t) \cdot d\mathbf{A} d\Omega dE dt$ describes the *net*

number of particles (in minus out) crossing the surface with direction within $d\Omega$ of $\hat{\mathbf{k}}$, energy within dE of E , between times t and $t + dt$. However, a physically closer analog of current density in the space-only continuity equation would be something like:

$$\mathbf{J}(\mathbf{r}, t) = \int dEd\Omega \mathbf{j}(\mathbf{r}, \hat{\mathbf{k}}, E, t) \quad (1.1.14)$$

The magnitude of the particle current density distribution is the **angular fluence rate per unit energy**:

$$\dot{\psi}(\mathbf{r}, \hat{\mathbf{k}}, E, t) = v(E)n(\mathbf{r}, \hat{\mathbf{k}}, E, t) \quad (1.1.15)$$

which we will soon find is the most convenient surrogate of $n(\mathbf{r}, \hat{\mathbf{k}}, E, t)$ in the Boltzmann transport equation. Integrating over the non-spatial phase space coordinates gives us the typical **fluence rate**:

$$\dot{\Phi}(\mathbf{r}, t) = \int dEd\Omega \dot{\psi}(\mathbf{r}, \hat{\mathbf{k}}, E, t) \quad (1.1.16)$$

Importantly, the quantity $\dot{\Phi}dA dt$ is the *total* number of particles crossing the surface $d\mathbf{A}$ between time t and $t + dt$. However, the quantity $\mathbf{J} \cdot d\mathbf{A} dt$ is the *net* number of particles crossing the surface. So, the former is “in plus out,” but the later is “in minus out.” Lastly, if we integrate $\dot{\psi}$ over only the angular directions of the particle, we get the typical **fluence rate per unit energy** required for dose (rate):

$$\dot{\varphi}(\mathbf{r}, E, t) = \int d\Omega \dot{\psi}(\mathbf{r}, \hat{\mathbf{k}}, E, t) \quad (1.1.17)$$

and of course, integrating a ‘rate’ with respect to time over an interval gives the total amount of that quantity over that interval. Note that we will later exclusively refer to ψ as the angular fluence and φ as the fluence, despite that they are distributed in energy.

1.1.2 Augmenting the continuity equation

The continuity equation is entirely too simple for our purposes. However, it does describe directional, straight-line, ray-like travel that we know particles undergo inbetween scattering events or after creation. So we seek to retain it as a starting point for the BTE, and augment it with terms allowing for particle creation, absorption, and scattering. Our design rationale is as follows: the function n describes the density of particles in a given point in phase space. Over time, particles appear and disappear at a given point in phase space when they change their: position, direction, or energy. Consequently, we must describe the rate of all mechanisms which cause a temporal change in n . The sum of each of these rates will yield the total time derivative of n , which is $\partial n / \partial t$. Note that the following derivation follows very closely the one found in [3].

1. **Streaming.** This accounts for typical, directional, continuity-equation-like propagation. It is:

$$\left(\frac{\partial n}{\partial t} \right)_{\text{str}} = -v(E)\hat{\mathbf{k}} \cdot \nabla n(\mathbf{r}, \hat{\mathbf{k}}, E, t) \quad (1.1.18)$$

2. **Scattering *into* a phase space location.** Note here that “scattering” is considered to be a change in particle energy and/or direction, in a manner which may be kinematically related. However, we are not, in this term, choosing the *initial* energy or direction of a single particle, but rather we must account for all of them. The change occurs locally, so that we don’t consider scattering to change \mathbf{r} , although a scattered particle will in principle travel to a different \mathbf{r} through streaming. As a whole, we simply say that, a particle once had momentum $(\hat{\mathbf{k}}', E')$, and now it has $(\hat{\mathbf{k}}, E)$. Let’s focus on a single interaction with a focused, nearly monoenergetic beam of particles of incoming direction and energy within $\Delta\Omega'$ and $\Delta E'$ of $(\hat{\mathbf{k}}', E')$. The volume density of particles in this beam is, of course, $\Delta n = n(\hat{\mathbf{k}}', E')\Delta E'\Delta\Omega'$. Scattering theory provides us with the understanding of how to determine the change in n per unit path length, which we can multiply by the velocity of the particles (the exchange rate between position and time) to obtain the change per unit time.

That is, in scattering theory, we focus on the ‘cross section’ (see **wiscobolt physics** or a book on scattering theory, classical [14] or quantum [15]). The cross section for scattering to occur per scattering center (such as an atom or bound electron) and per incident particle, that maps $(\hat{\mathbf{k}}', E')$ to $(\hat{\mathbf{k}}, E)$, is $(d\sigma_s/dEd\Omega_s)(\hat{\mathbf{k}}' \rightarrow \hat{\mathbf{k}}, E' \rightarrow E)$, where Ω_s is the scattering solid angle (the significance of a scattering solid angle, particularly in this quantity, will be made clear soon). The fraction of particles which undergo this scattering interaction, per unit path length, is then ρ_s multiplied by this cross section, where $\rho_s = \rho_s(\mathbf{r})$ is the density of scattering centers around \mathbf{r} . Multiplied by n , this gives the particle density per unit path length lost by $(\hat{\mathbf{k}}', E')$ and gained by $(\hat{\mathbf{k}}, E)$ (we will account for the loss in particle density in the originating phase space location shortly). That is to say something like:

$$\Delta \left[\frac{\partial n}{\partial \ell}(\hat{\mathbf{k}}, E) \right]_{\text{in}} = \rho_s n(\hat{\mathbf{k}}', E') \Delta E' \Delta \Omega' \frac{d\sigma_s}{dEd\Omega_s}(\hat{\mathbf{k}}' \rightarrow \hat{\mathbf{k}}, E' \rightarrow E) \quad (1.1.19)$$

Again, the rate of change of n is then given by multiplication with $v(E')$:

$$\Delta \left[\frac{\partial n}{\partial t}(\hat{\mathbf{k}}, E) \right]_{\text{in}} = \rho_s v(E') n(\hat{\mathbf{k}}', E') \Delta E' \Delta \Omega' \frac{d\sigma_s}{dEd\Omega_s}(\hat{\mathbf{k}}' \rightarrow \hat{\mathbf{k}}, E' \rightarrow E) \quad (1.1.20)$$

Now, to generalize to the contribution from all points in phase space, we sum over all partial contributions, i.e., over all incident directions and energies. Fully assembled, the term we seek is:

$$\left[\frac{\partial n}{\partial t}(\hat{\mathbf{k}}, E) \right]_{\text{in}} = \rho_s \int dE' d\Omega' \frac{d\sigma_s}{dEd\Omega_s}(\hat{\mathbf{k}}' \rightarrow \hat{\mathbf{k}}, E' \rightarrow E) v(E') n(\hat{\mathbf{k}}', E') \quad (1.1.21)$$

Note that our scattering cross sections assume that the medium is at rest, which means that they never allow radiation particles to scatter from lower energy to higher energy. Thus, one could (and will later) take the lower bound of the E' integral to be E itself. Additionally, we will be considering systems for which we can be confident that no particles exist with an energy exceeding some constant E_{\max} . Thus, we could take the upper bound of the E' integral to be E_{\max} . Also, a note on position dependence is

warranted here. Cross sections are generally obtained through quantum mechanical scattering theory, meaning that to ascribe them with a position and momentum dependence simultaneously is meaningless. Yet, we will in fact consider our cross sections to have position dependence in our calculations by virtue of what material is ‘at’ or ‘near’ a given location, and the density therein. Thus, we will include \mathbf{r} in the arguments of these quantities to emphasize this note. Now, we will also define the **differential linear attenuation coefficient of scattering**:

$$\Sigma_s(\mathbf{r}, \hat{\mathbf{k}}' \rightarrow \hat{\mathbf{k}}, E' \rightarrow E) \equiv \rho_s(\mathbf{r}) \frac{d\sigma_s}{dE d\Omega_s}(\mathbf{r}, \hat{\mathbf{k}}' \rightarrow \hat{\mathbf{k}}, E' \rightarrow E) \quad (1.1.22)$$

Thus:

$$\left(\frac{\partial n}{\partial t} \right)_{\text{in}} = \int dE' d\Omega' \Sigma_s(\mathbf{r}, \hat{\mathbf{k}}' \rightarrow \hat{\mathbf{k}}, E' \rightarrow E) v(E') n(\mathbf{r}, \hat{\mathbf{k}}', E') \quad (1.1.23)$$

It is finally worth noting that the angular dependence of the cross sections we will be using can be written in terms of one variable, the **scattering angle cosine** μ_s . When a scattering interaction occurs (at least, in the scattering mechanisms we’ll be interested in, which average over polarization/spin implicitly), the only quantity which matters to the interaction is the difference in angle between the incident and outgoing particles. This can be described by:

$$\mu_s \equiv \hat{\mathbf{k}} \cdot \hat{\mathbf{k}}' \quad (1.1.24)$$

This represents only one degree of freedom, as it will have the same value for every set of $\hat{\mathbf{k}}'$ vectors that are related to one another by rotation about $\hat{\mathbf{k}}$. This is to say, the set of $\hat{\mathbf{k}}'$ that are along the same cone whose axis is $\hat{\mathbf{k}}$ will yield the same μ_s , and physically the μ_s is all that the interaction distinguishes without knowing both incident and outgoing polarization/spin states that otherwise provide the interaction with some particular ‘handedness.’ With this coordinate, we can take:

$$\frac{d\sigma_s}{d\Omega_s} = \frac{1}{2\pi} \frac{d\sigma_s}{d\mu_s} \quad (1.1.25)$$

which comes from:

$$d\Omega_s = 2\pi d\mu_s \quad (1.1.26)$$

where we have presumably integrated over some ‘scattering polar angle,’ over which our cross sections will be isotropic. Then, it is also notable that we sometimes (but not always) find that the energy dependence of Σ_s is separable from its angular dependence [5, 6, 16]. We would then take:

$$\frac{d\sigma_s}{dE' d\Omega_s} = \frac{1}{2\pi} \frac{d\sigma_s}{dE'} \frac{df}{d\mu_s} \quad (1.1.27)$$

where $df/d\mu_s$ is an angular distribution normalized over μ_s alone.

3. **Scattering *out of* a phase space location.** We created a lot of particles in the phase space location $(\hat{\mathbf{k}}, E)$, due to in-scattering from all other directions and energies. Now, we must account for the loss of particles at a given phase space location such as $(\hat{\mathbf{k}}, E)$ due to out-scattering to all other directions and energies. The cross section for scattering to occur per scattering center per unit incident particle that maps $(\hat{\mathbf{k}}, E)$ to $(\hat{\mathbf{k}}', E')$ is now $(d\sigma_s/dE'd\Omega_s)(\hat{\mathbf{k}} \rightarrow \hat{\mathbf{k}}', E \rightarrow E')$, where we have swapped the primed and un-primed indices. So the number of particles created around the location $(\hat{\mathbf{k}}', E')$ is given by:

$$\left[\frac{\partial n}{\partial t}(\hat{\mathbf{k}}, E) \right]_{\text{out}} = \rho_s \int dE' d\Omega' \frac{d\sigma_s}{dE d\Omega_s}(\hat{\mathbf{k}} \rightarrow \hat{\mathbf{k}}', E \rightarrow E') v(E) n(\hat{\mathbf{k}}, E) \quad (1.1.28)$$

Notably, the primed (integrated) and un-primed variable pairs were swapped, then the primed variables were integrated. This follows from reasoning similar to what we employed for the scattering-in term. The term $v(E) n(\hat{\mathbf{k}}, E)$ is pulled out of the integral, and we find that we've just fully integrated the differential scattering cross section. With momentary degeneracy in our notation, we define the **linear attenuation coefficient of scattering**:

$$\Sigma_s(\mathbf{r}, E) \equiv \rho_s(\mathbf{r}) \sigma_s(\mathbf{r}, E) \quad (1.1.29)$$

where σ_s is the integrated differential scattering cross section. We'd then conclude:

$$\left(\frac{\partial n}{\partial t} \right)_{\text{out}} = -v(E) \Sigma_s(\mathbf{r}, E) n(\mathbf{r}, \hat{\mathbf{k}}, E, t) \quad (1.1.30)$$

4. **Absorption.** This follows a similar treatment as scattering out, but with the absorption cross section we don't need to integrate anything as there are no outgoing energies. We simply leverage $\Sigma_a(\mathbf{r}, E) \equiv \rho_s(\mathbf{r}) \sigma_a(\mathbf{r}, E)$ as the **linear attenuation coefficient of absorption**, which has the interpretation that it describes the loss per unit path length of particles due specifically to absorption. From this intuitive description we can write:

$$\left(\frac{\partial n}{\partial t} \right)_{\text{abs}} = -v(E) \Sigma_a(\mathbf{r}, E) n(\mathbf{r}, \hat{\mathbf{k}}, E, t) \quad (1.1.31)$$

Now we will combine the results of scattering out and absorption to write:

$$\Sigma_t(\mathbf{r}, E) \equiv \Sigma_s(\mathbf{r}, E) + \Sigma_a(\mathbf{r}, E) \quad (1.1.32)$$

and we will refer to this as the **total linear attenuation** term.

$$\left(\frac{\partial n}{\partial t} \right)_{\text{attn}} \equiv \left(\frac{\partial n}{\partial t} \right)_{\text{out}} + \left(\frac{\partial n}{\partial t} \right)_{\text{abs}} = -v(E) \Sigma_t(\mathbf{r}, E) n(\mathbf{r}, \hat{\mathbf{k}}, E, t) \quad (1.1.33)$$

5. **Sources.** We want to be able to treat particle sources without having to model their intricate physical origins. The source term is the only term we'll just *define*:

$$\left(\frac{\partial n}{\partial t} \right)_{\text{src}} \equiv \dot{s}(\mathbf{r}, \hat{\mathbf{k}}, E, t) \quad (1.1.34)$$

In principle, the source term will populate our phase space with particles, and then their interactions will dictate n . We will thoroughly discuss construction of these source terms, which is fully consistent with the above definition, soon.

Now let's assemble the equation of state we seek. It is just the statement that the rate of change of n is the sum of the partial contributions from different processes we have just outlined:

$$\boxed{\frac{\partial n}{\partial t} = -v(E)\hat{\mathbf{k}} \cdot \nabla n(\mathbf{r}, \hat{\mathbf{k}}, E, t) + \int dE' d\Omega' \Sigma_s(\mathbf{r}, \hat{\mathbf{k}}' \cdot \hat{\mathbf{k}}, E' \rightarrow E)v(E')n(\mathbf{r}, \hat{\mathbf{k}}', E', t) - \Sigma_t(\mathbf{r}, E)v(E)n(\mathbf{r}, \hat{\mathbf{k}}, E, t) + \dot{s}(\mathbf{r}, \hat{\mathbf{k}}, E, t)} \quad (1.1.35)$$

This is the **time-dependent Boltzmann transport equation (TDBTE)**. It is a linear, first-order, integro-partial-differential equation. Let us also consider what physical cases justify $\partial n/\partial t \approx 0$. If the source is steady, and the medium in which this all takes place is not changing, then we can safely suggest that n does not change with time. We can then get rid of the LHS. Now, we also find a potentially tedious factor of v on every n littered in this equation, so we take a hint to make $vn = \dot{\psi}$ the object of our interest. Lastly, we integrate the equation with respect to t to directly turn $\dot{\psi}$ into ψ . We finally arrive at the **time-independent Boltzmann transport equation** (which we'll refer to as BTE from this point on), formulated with respect to the **angular fluence** ψ :

$$\boxed{\hat{\mathbf{k}} \cdot \nabla \psi(\mathbf{r}, \hat{\mathbf{k}}, E) + \Sigma_t(\mathbf{r}, E)\psi(\mathbf{r}, \hat{\mathbf{k}}, E) - \int dE' d\Omega' \Sigma_s(\mathbf{r}, \hat{\mathbf{k}}' \cdot \hat{\mathbf{k}}, E' \rightarrow E)\psi(\mathbf{r}, \hat{\mathbf{k}}', E') = s(\mathbf{r}, \hat{\mathbf{k}}, E)} \quad (1.1.36)$$

This can also be formulated as a linear operator problem. If we define:

$$\hat{T}\psi \equiv \hat{\mathbf{k}} \cdot \nabla \psi + \Sigma_t \psi \quad (1.1.37)$$

as the **transport operator**, and:

$$\hat{K}\psi \equiv \int dE' d\Omega' \Sigma_s(\mathbf{r}, \hat{\mathbf{k}}' \cdot \hat{\mathbf{k}}, E' \rightarrow E)\psi(\mathbf{r}, \hat{\mathbf{k}}', E') \quad (1.1.38)$$

as the **scattering operator**, and finally:

$$\hat{L} \equiv \hat{T} - \hat{K} \quad (1.1.39)$$

as the **BTE operator**, then the BTE is:

$$\hat{L}\psi = (\hat{T} - \hat{K})\psi = s \quad (1.1.40)$$

This form, and these defined operators, will be important to us when formulating a numerical solution. We will also find a way to interpret them, upon which we will rely when we formulate this solution.

With the BTE derived, let's talk about some of the mathematical requirements for a solution. Having removed its time dependence, the BTE requires a boundary condition [3, 9, 17]. The boundary condition we will use is particularly suitable for radiation therapy in a convex volume. Specifically, this condition is the non-re-entrant boundary condition which states that we know ψ over the entire boundary of our problem volume *only* for $\hat{\mathbf{k}}$ pointing *into* our volume. More formally, if V is the domain of our volume, Γ is its surface, and $\hat{\mathbf{n}}(\mathbf{r})$ is the normal vector of the surface position \mathbf{r} (meaningful only when $\mathbf{r} \in \Gamma$), then:

$$\psi(\mathbf{r}, \hat{\mathbf{k}}, E) = \bar{\psi}(\mathbf{r}, \hat{\mathbf{k}}, E), \forall \hat{\mathbf{k}} \cdot \hat{\mathbf{n}}(\mathbf{r}) < 0 \& \mathbf{r} \in \Gamma \quad (1.1.41)$$

where $\bar{\psi}$ is assumed to be known. To be more formal (and brief), we will let Γ_{\downarrow} will represent the part of the combined angular and spatial domains such that $\hat{\mathbf{k}} \cdot \hat{\mathbf{n}}(\mathbf{r}) < 0 \& \mathbf{r} \in \Gamma$, and then Γ_{\uparrow} will be $\hat{\mathbf{k}} \cdot \hat{\mathbf{n}}(\mathbf{r}) > 0 \& \mathbf{r} \in \Gamma$. To give you an example of a boundary condition, if we are irradiating a volume with a single photon beam of a particular configuration, then we simply must specify the beam's in-going fluence at the beam-facing surface of the volume. Thus, $\bar{\psi}$ is the angular fluence of primary (or 'uncollided') particles at the surface. On the other hand, we may have an internal source, and it may appear uncertain what to take for $\bar{\psi}$. The answer is a matter of practicality depending on our problem. If our volume is considered to be inside of a vacuum (or, for instance, an extremely low density material such as air), then we simply take the 'vacuum boundary conditions' $\bar{\psi} = 0$ because there will be no particles entering our volume. However, if our volume is, say, surrounded by a medium which reflects all particles about the normal, then we can take the 'reflective boundary conditions,' which are not explicitly provided but nonetheless provide the final constraint needed to solve the BTE:

$$\bar{\psi}(\mathbf{r}, \hat{\mathbf{k}}, E) = \psi[\mathbf{r}, \hat{\mathbf{k}} - 2(\hat{\mathbf{k}} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}, E] \quad (1.1.42)$$

where the argument for the angular coordinate in the RHS is just the ray whose reflection about $\hat{\mathbf{n}}$ is $\hat{\mathbf{k}}$. Notably, implementation of reflective boundary conditions will not be discussed here, and they are not implemented in wiscobolt. However, they are discussed in a number of papers, including [18, 19]. In wiscobolt, we focus most generally on boundary conditions wherein $\bar{\psi}$ is known analytically, which is highly suitable for external beam problems.

An integral form of the BTE exists, but references the solution on both sides of the equation, except if we were to somehow take $\Sigma_s = 0$. That is, if we take this seemingly un-physical assumption, the BTE would read:

$$\hat{\mathbf{k}} \cdot \nabla \psi + \Sigma_t \psi = s \quad (1.1.43)$$

Despite that it would be unintuitive for a particle to be attenuated but not scattered to a different phase space element, this equation can be used, for instance, to describe the fluence of completely uncollided particles from the source, and therefore our boundary condition. These particles can be attenuated while not showing up in a different location in phase space because, simply by definition, they are *uncollided* particles. The explicit solution to this equation is too involved for us to derive, so we will state it directly:

$$\psi(\mathbf{r}, \hat{\mathbf{k}}, E) = \int_0^{\infty} d\ell s(\mathbf{r} - \ell \hat{\mathbf{k}}, \hat{\mathbf{k}}, E) e^{-\int_0^{\ell} d\ell' \Sigma_t(\mathbf{r} - \ell' \hat{\mathbf{k}}, E)} \quad (1.1.44)$$

meaning that \hat{T}^{-1} is actually explicitly known, and it is given by the equation above, or more generally:

$$\hat{T}^{-1}f(\mathbf{r}, \hat{\mathbf{k}}, E) = \int_0^\infty d\ell f(\mathbf{r} - \ell\hat{\mathbf{k}}, \hat{\mathbf{k}}, E) e^{-\int_0^\ell d\ell' \Sigma_t(\mathbf{r} - \ell'\hat{\mathbf{k}}, E)} \quad (1.1.45)$$

We can also define the optical path length between two points \mathbf{r} and \mathbf{r}' :

$$\tau(\mathbf{r}, \mathbf{r}', E) \equiv \int_0^{|\mathbf{r} - \mathbf{r}'|} d\ell \Sigma_t \left(\mathbf{r} - \ell \frac{\mathbf{r} - \mathbf{r}'}{|\mathbf{r} - \mathbf{r}'|}, E \right) \quad (1.1.46)$$

The placement of this quantity in \hat{T} suggests that $e^{-\tau(\mathbf{r}, \mathbf{r}', E)}$ is the fraction of particles with energy E that get attenuated when travelling from \mathbf{r}' to \mathbf{r} . The form of τ is easier to understand in homogeneous media; the quantity $e^{-\Sigma_t \Delta\ell}$ is the fractional attenuation over the path length $\Delta\ell$. This is just Beer's law. In the above integral, we've just parametrized the distance between \mathbf{r}' and \mathbf{r} with ℓ , and integrated. This allows us to write:

$$\hat{T}^{-1}f(\mathbf{r}, \hat{\mathbf{k}}, E) = \int_0^\infty d\ell f(\mathbf{r} - \ell\hat{\mathbf{k}}, \hat{\mathbf{k}}, E) e^{-\tau(\mathbf{r}, \mathbf{r} - \ell\hat{\mathbf{k}}, E)} \quad (1.1.47)$$

A volume integral form of this equation also exists [3], and we will actually generally prefer it:

$$\hat{T}^{-1}f(\mathbf{r}, \hat{\mathbf{k}}, E) = \int d^3\mathbf{r} \frac{1}{|\mathbf{r} - \mathbf{r}'|^2} f(\mathbf{r}', \hat{\mathbf{k}}, E) e^{-\tau(\mathbf{r}, \mathbf{r}', E)} \delta^2 \left(\hat{\mathbf{k}} - \frac{\mathbf{r} - \mathbf{r}'}{|\mathbf{r} - \mathbf{r}'|} \right) \quad (1.1.48)$$

Now, if this is applied to the original BTE, one can write ψ solely in terms of integrals, however, these are integrals which involve ψ itself. Nevertheless, the application of \hat{T}^{-1} can reproduce the inverse square law, and contains Beer's law of exponential attenuation. Given a reasonably simple source and problem geometry, ψ here can typically be constructed heuristically. This will be demonstrated in the next section. However, for now, we actually will quickly equip this expression to modify the BTE that we will be discretizing. Specifically, before discretization, we'll want to briefly ask: what kind of sources might we encounter? Suppose we focus on the simplest source: a spherical point source located at the origin of our coordinate system. It will come in the form:

$$s(\mathbf{r}, \hat{\mathbf{k}}, E) = \dot{s}_0 \Delta t \delta^3(\mathbf{r}) g(\hat{\mathbf{k}}) f(E) \quad (1.1.49)$$

where \dot{s}_0 is the rate at which particles are emitted from the beam, Δt is how long we consider the beam to be emitting particles at this rate, we also use $s_0 = \dot{s}_0 \Delta t$, $f(E)$ is a normalized angular distribution, $g(\hat{\mathbf{k}})$ is a normalized angular distribution telling us which directions have more particles, and $\delta^2(\hat{\mathbf{k}} - \hat{\mathbf{r}})$ is a term which states that particles stream radially outward from the source. Note that integration over energy, space, and angle produces $\dot{s}_0 \Delta t$, which amounts to counting up all of the particles which you have allowed to populate phase space. With this, the **uncollided angular fluence** is $\hat{T}^{-1}s$, which is more conveniently stated as:

$$\tilde{\psi}^0(\mathbf{r}, \hat{\mathbf{k}}, E) \equiv \hat{T}^{-1}s = \tilde{\varphi}^0(\mathbf{r}, E) \delta^2(\hat{\mathbf{k}} - \hat{\mathbf{r}}) \quad (1.1.50)$$

where $\tilde{\varphi}^0$ is the **uncollided fluence**, given for this case as:

$$\tilde{\varphi}^0(\mathbf{r}, E) = \frac{1}{r^2} \dot{s}_0 \Delta t g(\mathbf{r}) f(E) e^{-\tau(\mathbf{r}, 0, E)} \quad (1.1.51)$$

with minor modifications to its form for other cases (to be more thoroughly discussed in the next section). Now, this can always, where it is needed, be constructed heuristically by using the inverse square law and Beer's law, the latter of which requires ray-tracing (we will discuss this in great depth later on, but for now, let $\tilde{\varphi}^0$ suffice). Now, we must recognize and accept that, if the true solution ψ involves $\tilde{\psi}^0$, then ψ involves a singularity, by virtue of $\delta^2(\hat{\mathbf{k}} - \hat{\mathbf{f}})$. This is no good, because we want to discretize ψ . Thus, we will try to write $\tilde{\psi}^0$ out of our solution entirely. Consider first that, by definition of $\tilde{\psi}^0$, the BTE is:

$$(\hat{T} - \hat{K})\psi = \hat{T}\tilde{\psi}^0 \quad (1.1.52)$$

Suppose we now expand ψ as:

$$\boxed{\psi \equiv \tilde{\psi} + \tilde{\psi}^0} \quad (1.1.53)$$

where $\tilde{\psi}$ is the **collided angular fluence**, then:

$$\begin{aligned} (\hat{T} - \hat{K})\psi &= \hat{T}\tilde{\psi}^0 \\ (\hat{T} - \hat{K})(\tilde{\psi} + \tilde{\psi}^0) &= \hat{T}\tilde{\psi}^0 \\ (\hat{T} - \hat{K})\tilde{\psi} &= \hat{T}\tilde{\psi}^0 - \hat{T}\tilde{\psi}^0 + \hat{K}\tilde{\psi}^0 \\ \hat{L}\tilde{\psi} &= \hat{K}\tilde{\psi}^0 \end{aligned} \quad (1.1.54)$$

which suggests, intuitively, that the scattering of uncollided particles is a source of collided particles. This itself is, of course, a BTE. Only, the particles of our interest are those which have been scattered at least once, and they are sourced by the first collision of incident particles. We will take:

$$\boxed{\tilde{s} \equiv \hat{K}\tilde{\psi}^0} \quad (1.1.55)$$

which is known as long as $\tilde{\psi}^0$ is known, and $\tilde{\psi}^0$ is known as long as s is known. And thus, the BTE is once again cast in a simple linear form:

$$\hat{L}\tilde{\psi} = \tilde{s} \quad (1.1.56)$$

which is just the BTE with a different fluence and source. Yet, this is not trivial rearrangement, and it is helpful during discretization. This method is known as the **first-collision scattering source (FCS)** method, and it is described thoroughly in [20]. It does, however, create a slight numerical non-conservation of particle number if it is wielded without caution. The correction for this is described as well by [20], as well as in **wiscobolt implementation**.

Note lastly that there exists an alternative, second-order form of the BTE which can also be solved, called the **self-adjoint angular flux equation (SAAFE)** [18, 19, 21, 22]. To obtain this equation, one takes the BTE and applies the inverse of $\hat{S} \equiv \Sigma_t - \hat{K}$. Then, ψ is able to be written as $-\hat{S}^{-1}\hat{\mathbf{k}} \cdot \nabla \psi + \hat{S}^{-1}\psi$, at which point one can apply $\hat{L}\psi = s$ to find the SAAFE:

$$(-\hat{\mathbf{k}} \cdot \nabla \hat{S}^{-1}\hat{\mathbf{k}} \cdot \nabla + \hat{S})\psi = s - \hat{\mathbf{k}} \cdot \nabla \hat{S}^{-1}s \quad (1.1.57)$$

The utility of this formulation is not obvious at this point, but in any case, wiscobolt does not use the SAAFE, preferring instead the direct $\hat{L}\psi = s$ form of the BTE, for reasons which may be understood by comparing solve times and storage against the first-order BTE [21].

1.2 Sources

Let's now talk about the sources we might encounter in radiation transport. First, note that it is convenient to write \mathbf{r} in Cartesian coordinates, but $\hat{\mathbf{k}}$ in spherical coordinates. That is to say, we take:

$$\hat{\mathbf{k}}(\theta, \phi) = \cos \phi \sin \theta \hat{\mathbf{x}} + \sin \phi \sin \theta \hat{\mathbf{y}} + \cos \theta \hat{\mathbf{z}} \quad (1.2.1)$$

Where we can not confuse θ and ϕ with coordinates corresponding to position-space, but rather they are exclusive to $\hat{\mathbf{k}}$ -space. In this case, the solid angle element takes the form:

$$d\Omega = \sin \theta d\theta d\phi \quad (1.2.2)$$

And integration over all $\hat{\mathbf{k}}$ -space runs over:

$$\Omega = \{(\theta, \phi) \mid 0 \leq \theta \leq \pi, 0 \leq \phi < 2\pi\} \quad (1.2.3)$$

It is here also useful to define the **polar angle cosine**:

$$\mu = \cos \theta \quad (1.2.4)$$

and then:

$$d\Omega = d\mu d\phi \quad (1.2.5)$$

$$\Omega = \{(\mu, \phi) \mid -1 \leq \mu \leq 1, 0 \leq \phi < 2\pi\} \quad (1.2.6)$$

We can build up sources from the spherical point source that we described in the previous section. First, what if the source is not localized at the origin, but rather it is localized at \mathbf{r}_0 ? Then, we apply a translation to \mathbf{r}_0 :

$$s(\mathbf{r}, \hat{\mathbf{k}}, E) = s_0 \delta^3(\mathbf{r} - \mathbf{r}_0) g(\hat{\mathbf{k}}) f(E) \quad (1.2.7)$$

We can define:

$$\boldsymbol{\varkappa} \equiv \mathbf{r} - \mathbf{r}_0 \quad (1.2.8)$$

Credit for the $\boldsymbol{\varkappa}$ glyph is due to Griffiths' Electrodynamics book [23]. This leads to:

$$\tilde{\psi}^0(\mathbf{r}, \hat{\mathbf{k}}, E) = \tilde{\varphi}^0(\mathbf{r}, E) \delta^2(\hat{\mathbf{k}} - \hat{\boldsymbol{\varkappa}}) \quad (1.2.9)$$

where:

$$\tilde{\varphi}^0(\mathbf{r}, E) = \frac{1}{\varkappa^2} s_0 g(\boldsymbol{\varkappa}) f(E) e^{-\tau(\mathbf{r}, \mathbf{r}_0, E)} \quad (1.2.10)$$

What if we have two point sources – one at \mathbf{r}_1 with anisotropy g_1 and the other at \mathbf{r}_2 with anisotropy g_2 ? Then due to the linearity of the BTE, we can just add the sources together, yielding:

$$s(\mathbf{r}, \hat{\mathbf{k}}, E) = s_0 f(E) [g_1(\hat{\mathbf{k}}) \delta^3(\mathbf{r} - \mathbf{r}_1) + g_2(\hat{\mathbf{k}}) \delta^3(\mathbf{r} - \mathbf{r}_2)] \quad (1.2.11)$$

$$\tilde{\psi}^0(\mathbf{r}, \hat{\mathbf{k}}, E) = \tilde{\varphi}_1^0(\mathbf{r}, E) \delta^2\left(\hat{\mathbf{k}} - \frac{\mathbf{r} - \mathbf{r}_1}{|\mathbf{r} - \mathbf{r}_1|}\right) + \tilde{\varphi}_2^0(\mathbf{r}, E) \delta^2\left(\hat{\mathbf{k}} - \frac{\mathbf{r} - \mathbf{r}_2}{|\mathbf{r} - \mathbf{r}_2|}\right) \quad (1.2.12)$$

What if we have a distribution of sources in our volume V , $\rho_S(\mathbf{r})$? Then we have a continuous sum of point sources:

$$s(\mathbf{r}, \hat{\mathbf{k}}, E) = s_0 g(\hat{\mathbf{k}}) f(E) \int d^3 \mathbf{r}' \delta^3(\mathbf{r} - \mathbf{r}') \rho_S(\mathbf{r}') \quad (1.2.13)$$

But, this is just:

$$s(\mathbf{r}, \hat{\mathbf{k}}, E) = s_0 \rho_S(\mathbf{r}) g(\hat{\mathbf{k}}) f(E) \quad (1.2.14)$$

so now, there is no singularity in s . This won't need to receive special treatment during discretization, and we wouldn't really need to construct $\tilde{\varphi}^0$ heuristically. Sources such as these, i.e., internal sources, are not yet implemented in wiscobolt. The user may, however, circumvent this if they are able to construct this source in a particular format and have wiscobolt read the file in, while wiscobolt typically creates the source based on user input.

We next consider: what if our source is a plane wave, i.e., really far away relative to the size of V , or an “infinite” plane of spherical point sources – how do we construct our uncollided particles/boundary condition, $\tilde{\varphi}^0$? It is useful to first prove that the term $\delta^2(\hat{\mathbf{k}} - \hat{\mathbf{z}})$ goes to $\delta(\hat{\mathbf{k}} - \hat{\mathbf{k}}_0)$. Suppose we have a spherical source at \mathbf{r}_0 . Then, we abstract it in some $\hat{\mathbf{k}}_0$ direction. So, we will take our beam origin to be $\mathbf{r}_0 - \ell \hat{\mathbf{k}}_0$, where ℓ is the distance over which we abstract the point source. Now, $\hat{\mathbf{z}}$ is replaced by:

$$\hat{\mathbf{z}}' = \frac{\mathbf{r} - (\mathbf{r}_0 - \ell \hat{\mathbf{k}}_0)}{|\mathbf{r} - (\mathbf{r}_0 - \ell \hat{\mathbf{k}}_0)|} = \frac{\hat{\mathbf{z}} \ell / \ell + \hat{\mathbf{k}}_0}{|\hat{\mathbf{z}} \ell / \ell + \hat{\mathbf{k}}_0|} \quad (1.2.15)$$

where in the equation above, $\hat{\mathbf{z}}$ retains its definition as the unit vector pointing from \mathbf{r}_0 to \mathbf{r} , but $\hat{\mathbf{z}}'$ is now the vector referenced in $\delta^2(\hat{\mathbf{k}} - \hat{\mathbf{z}}')$. Now, if we take ℓ to be much larger than the distance ℓ , meaning we have pulled the beam origin very far away from the location at which we're evaluating the source or fluence, we have $\ell / \ell \rightarrow 0$, or:

$$\hat{\mathbf{z}}' \rightarrow \hat{\mathbf{k}}_0 \quad (1.2.16)$$

Thus:

$$\delta^2(\hat{\mathbf{k}} - \hat{\mathbf{z}}) \rightarrow \delta^2(\hat{\mathbf{k}} - \hat{\mathbf{k}}_0) \quad (1.2.17)$$

Now, we may suppose that some beam angular distribution $g(\hat{\mathbf{k}})$ is able to produce some relative distribution of particles on the surface of the plane. However, the inverse square law should, in principle, go to zero. How can this be treated so that we may deduce a form for our uncollided fluence/boundary condition, $\tilde{\varphi}^0$? We can suppose that the number of particles in the beam, s_0 is correspondingly dialed to cancel out the arbitrarily small inverse square factor. Then, we can merge the terms $s_0 g(\hat{\mathbf{k}}) / |\mathbf{r} - \mathbf{r}_0|^2$ into a new term $s_{0,\text{eff}} \rho_S(\mathbf{r})$, where $\rho_S(\mathbf{r})$ is some normalized areal density and $s_{0,\text{eff}}$ is, by definition, the total number of particles in the plane. When not dealing with an infinitely large plane, we imagine that the plane is blocked or ‘collimated,’ using some arbitrarily-shaped cutout. With ρ_S normalized over the plane, the burden of describing the total number of particles in the beam *remains* upon $s_{0,\text{eff}}$, as is most convenient for the user and the developer. If the user intends to describe such a

source, wiscobolt accommodates it nicely, and this is described thoroughly in the user-input section of **wiscobolt implementation**. If, however, the user wishes to use an uncollimated beam, at the moment wiscobolt does not assume anything but a uniform distribution, which can't be normalized, so the user must apply their own factor which describes the areal density of particles. This is also described in **wiscobolt implementation**.

It is interesting to note that, in every case, we have reduced $\tilde{\psi}^0$ to the form:

$$\tilde{\psi}^0(\mathbf{r}, \hat{\mathbf{k}}, E) = \tilde{\varphi}^0(\mathbf{r}, E) \delta^2[\hat{\mathbf{k}} - \hat{\mathbf{k}}_0(\mathbf{r})] \quad (1.2.18)$$

as long as we have:

$$\hat{\mathbf{k}}_0(\mathbf{r}) = \begin{cases} \hat{\mathbf{k}}_0, & \text{planar beam} \\ \hat{\mathbf{z}}, & \text{spherical beam} \end{cases} \quad (1.2.19)$$

1.3 Coupling particles

Radiation therapy, whether external or internal, with photons, electrons, alpha particles, protons, neutrons, etc., is not a problem of one particle alone [1–3]. Generally, uncharged particles like photons do not ‘deposit dose,’ but rather it is secondary charged particles like electrons and positrons which deposit dose, collision-by-collision. This discussion is central to an understanding of dosimetry, and involves the distinction between various forms of energy transfer. The reader is referred to [1]. Nevertheless, it is therefore relevant to construct a set of BTE’s wherein our particles all ‘talk’ to each other. For instance, in the case of external photon radiotherapy, the photons, electrons, and positrons need to be able to create one another. The rationale is simple: particle α can act as a source for particle β through a variety of interactions, in a term similar to the ‘scattering in’ term we’ve constructed. Let’s consider the interactions of interest to us, namely all of the prominent cross sections allowing for conversion between photons, electrons, and positrons. In media composed of atoms and molecules, and in the MeV energy regime, we have the following set of interactions, where the symbol like $\alpha \rightarrow \beta$ indicates a scattering interaction where the incident particle is α and the outgoing particle is β [2]:

γ	\rightarrow	γ	Compton effect, fluorescence
		e^- e^+	Compton effect, Auger effect, pair production, photoelectric effect Pair production
e^-	\rightarrow	γ	Bremsstrahlung, fluorescence
		e^- e^+	Elastic scattering, inelastic scattering, Auger effect None
e^+	\rightarrow	γ	Bremsstrahlung, pair annihilation
		e^- e^+	Bhaba scattering (ionization) Elastic scattering, Bhaba scattering (energy loss)

A thorough discussion on these cross sections is the focus of the **wiscobolt physics** document. Note that some, but not all, of these interactions involve loss of the incident particle (i.e., absorption, for example, the photoelectric effect), while some involve creation of an

extra particle (for example electron inelastic scattering, which technically comprises energy loss of the incident particle, and then sometimes ionization to produce another particle). It is also worth discussing fluorescence and the Auger effect in a bit more depth as each appears twice but they are relatively unique processes when caused by photons and electrons [2, 16, 24]. A photon can first create an inner shell atomic vacancy thru the photoelectric effect or Compton scattering. This means the emission of an electron (ionization), and constitutes the $\gamma \rightarrow e$ interactions thru either the photoelectric effect or Compton scattering. However, following this, electrons from a higher atomic shell can drop down into the vacancy, emitting a photon in the process of doing so. This photon is then fluorescence as per $\gamma \rightarrow \gamma$ in the diagram above. Yet again, this photon can create an atomic vacancy, freeing yet another electron. *This* electron is considered an Auger electron as $\gamma \rightarrow e$, considered to be produced by the *original* γ . The photon emitted by this electron is then considered produced by the original γ as well, rather than produced by an $e \rightarrow \gamma$ mechanism. The ensuing photons and electrons are, respectively, fluorescence photons and Auger electrons thru $\gamma \rightarrow \gamma$ and $\gamma \rightarrow e$. But an electron creates an inner shell atomic vacancy thru impact ionization. All electrons, including the first liberated, are referred to as Auger electrons as $e \rightarrow e$. The ensuing photons I will still refer to as fluorescence, despite that the term ‘fluorescence’ isn’t typically considered a process of $e \rightarrow \gamma$, but rather something like $\gamma \rightarrow \gamma$ or $\gamma \rightarrow e \rightarrow \gamma$. In truth, we are considering interactions like $e \rightarrow \gamma \rightarrow e \rightarrow \gamma$ to be fluorescence. Nevertheless, it will be necessary to treat each of these four processes separately later on. Pathways and probabilities are documented in [24], and will be discussed more thoroughly in **wiscobolt physics**.

Now, consider the conversion of particle α to β to add a source term into the BTE for β which looks like:

$$s^{\alpha\beta}(\mathbf{r}, \hat{\mathbf{k}}^\beta, E^\beta) = \int dE^\alpha d\Omega^\alpha \Sigma_s^{\alpha\beta}(\mathbf{r}, \hat{\mathbf{k}}^\alpha, \hat{\mathbf{k}}^\beta, E^\alpha, E^\beta) \psi^\alpha(\mathbf{r}, \hat{\mathbf{k}}^\alpha, E^\alpha) \equiv \hat{K}^{\alpha\beta} \psi^\alpha \quad (1.3.1)$$

where $\hat{\mathbf{k}}^\beta$ is the direction of particle β , and E^β is the energy of particle β , while $\hat{K}^{\alpha\beta}$ is the scattering operator to map ψ^α to a source of ψ^β , $\Sigma_s^{\alpha\beta}(\hat{\mathbf{k}}^\alpha, \hat{\mathbf{k}}^\beta, E^\alpha, E^\beta)$ is the linear attenuation coefficient of conversion of α with incoming direction and energy $\hat{\mathbf{k}}^\alpha$ and E^α to β with outgoing direction and energy $\hat{\mathbf{k}}^\beta$ and E^β . Now, our BTE is a coupled integro-differential equation, using cross sections described above:

$$\begin{aligned} \hat{L}^\gamma \psi^\gamma &= s^\gamma + \hat{K}^{e-\gamma} \psi^{e-} + \hat{K}^{e+\gamma} \psi^{e+} \\ \hat{L}^{e-} \psi^{e-} &= s^{e-} + \hat{K}^{\gamma e-} \psi^\gamma + \hat{K}^{e+e-} \psi^{e+} \\ \hat{L}^{e+} \psi^{e+} &= s^{e+} + \hat{K}^{\gamma e+} \psi^\gamma \end{aligned} \quad (1.3.2)$$

with the correct coordinate systems implied, and where γ describes photons, e^- describes electrons, and e^+ describes positrons. Another way to depict this as a coupled set of equations, we can also write:

$$\begin{pmatrix} \hat{L}^\gamma & -\hat{K}^{e-\gamma} & -\hat{K}^{e+\gamma} \\ -\hat{K}^{\gamma e-} & \hat{L}^{e-} & -\hat{K}^{e+e-} \\ -\hat{K}^{\gamma e+} & 0 & \hat{L}^{e+} \end{pmatrix} \begin{pmatrix} \psi^\gamma \\ \psi^{e-} \\ \psi^{e+} \end{pmatrix} = \begin{pmatrix} s^\gamma \\ s^{e-} \\ s^{e+} \end{pmatrix} \quad (1.3.3)$$

This is quite a complicated coupled problem, but not entirely impossible to solve. This system of equations can be directly discretized and solved algebraically with many (realistic) approximations.

2 DISCRETIZATION

There are a number of ways the BTE can be discretized, with each physically unique set of coordinates from phase space receiving its own treatment, and interaction between the different discretization methods yielding radically different problems. For example, the spatial coordinates can be discretized by the **finite element method (FEM)**, which can certainly be argued to be superior to simpler spatial differencing techniques. In wiscobolt, we exclusively use the FEM for space. But, depending on whether one discretizes the angular coordinates with **discrete ordinates** (S_N , where we will consider N as the number of angles dedicated to the polar angle) or **real spherical harmonics** expansions (P_N , where N is the order of real spherical harmonics up to which we assume the moments of ψ are other than zero), the angular discretization method combines with the spatial FEM to produce some peculiar properties that must be taken into consideration when solving the problem. The energy coordinate is usually either discretized with the **multigroup approximation** or the 1D FEM. Note, however, that the angular dependence of the scattering terms is almost always treated with a P_N series expansion, regardless of whether the rest of the BTE is treated with S_N or P_N . Following discretization, we can introduce iterative numerical methods that are used to arrive at a solution, as well as acceleration methods to act upon these methods. Then, we will discuss approximations that can be made so that a precise solution can be practically obtained. These will be the primary concern of the next chapter. This chapter will be divided into six sub-sections:

1. Series expansions
2. Spatial discretization
3. Angular discretization – S_N
4. Angular discretization – P_N
5. Energy discretization – multi-group formalism
6. Energy discretization – finite-element method

Our BTE will take the generic form:

$$(\hat{T} - \hat{K})\psi = s \quad (2.0.1)$$

And much of our focus will be discretization on an operator-to-operator basis. But we will also make a special note for sources of the form \tilde{s} . Now, one recognizes that coupling of different particle types is relatively simple, not notwithstanding a small point that must be made during energy discretization.

2.1 Series expansions

A major difficulty in solving the Boltzmann transport equation is the presence of integral operators. We seek to alleviate the trouble these give us. For the angular integrals, this can be done by expanding the quantities in the integrand, namely ψ and Σ_s , in terms of real spherical harmonics and Legendre polynomials, respectively. This sets the stage for the ‘ P_N scattering’ treatment, which we take regardless of angular discretization.

To begin, it is necessary we outline the conventions we will be using, which are not always consistent among the literature. Real spherical harmonics of degree ℓ and order m y_ℓ^m can be given with respect to the typical, complex spherical harmonics Y_ℓ^m , with the given conventions [25]:

$$Y_\ell^m(\hat{\mathbf{k}}) = \sqrt{\frac{(2\ell+1)}{4\pi} \frac{(\ell-m)!}{(\ell+m)!}} P_\ell^m(\mu) e^{im\phi} \quad (2.1.1)$$

with the Condon-Shortley phase present in the associated Legendre polynomials $P_\ell^m(\mu)$, which are given by:

$$P_\ell^m(\mu) = (-1)^m (1 - \mu^2)^{m/2} \frac{d^m}{d\mu^m} [P_\ell(\mu)] \quad (2.1.2)$$

and then $P_\ell(\mu)$ is a Legendre polynomial, given by:

$$P_\ell(\mu) = \frac{1}{2^\ell \ell!} \frac{d^\ell}{d\mu^\ell} [(\mu^2 - 1)^\ell] = P_\ell^0(\mu) \quad (2.1.3)$$

With these definitions, we can define the set of real functions [26]:

$$y_\ell^m(\hat{\mathbf{k}}) \equiv \begin{cases} \frac{1}{\sqrt{2}} [Y_\ell^{-m}(\hat{\mathbf{k}}) + (-1)^m Y_\ell^m], & m > 0 \\ Y_\ell^0(\hat{\mathbf{k}}), & m = 0 \\ \frac{i}{\sqrt{2}} [Y_\ell^m(\hat{\mathbf{k}}) - (-1)^m Y_\ell^{-m}(\hat{\mathbf{k}})], & m < 0 \end{cases} \quad (2.1.4)$$

or, equivalently, we can say:

$$y_\ell^m(\hat{\mathbf{k}}) = C_\ell^m P_\ell^m(\mu) S^m(\phi) \quad (2.1.5)$$

wherein we have defined:

$$S^m(\phi) \equiv \begin{cases} \cos(m\phi), & m \geq 0 \\ \sin(|m|\phi), & m < 0 \end{cases} \quad (2.1.6)$$

$$C_\ell^m \equiv (-1)^{3m/2-|m|/2} \sqrt{\frac{2\ell+1}{2\pi(1+\delta_{m0})} \frac{(\ell-m)!}{(\ell+m)!}} \quad (2.1.7)$$

The factor of $(-1)^{3m/2-|m|/2}$ is $(-1)^m$ if $m \geq 0$ and 1 if $m < 0$. I prefer to write it this way, which goes against frequent conventions for these real spherical harmonics, because I want y_ℓ^m in terms of P_ℓ^m rather than $P_\ell^{|m|}$ for reasons to be seen in **Appendix A**. Now, the utility of the real spherical harmonics in the context of the BTE is evident primarily in the P_N discretization method. In the S_N discretization method, the expansions we make will conveniently collapse and become identical whether they were made with y_ℓ^m or Y_ℓ^m , and this

is a deliberate consequence of the construction of y_ℓ^m . Namely, both the real and complex forms of the spherical harmonics obey the following important identities [25]:

$$\int_{4\pi} d\Omega y_\ell^{m*}(\theta, \phi) y_{\ell'}^{m'}(\theta, \phi) = \delta_{\ell\ell'} \delta_{mm'} \quad (2.1.8)$$

$$P_\ell(\hat{\mathbf{k}} \cdot \hat{\mathbf{k}}') = \frac{4\pi}{2\ell+1} \sum_{m=-\ell}^{\ell} y_\ell^m(\hat{\mathbf{k}}) y_\ell^{m*}(\hat{\mathbf{k}}') \quad (2.1.9)$$

$$\int d\Omega' y_\ell^m(\hat{\mathbf{k}}') P_{\ell'}(\hat{\mathbf{k}} \cdot \hat{\mathbf{k}}') = \frac{4\pi}{2\ell+1} y_\ell^m(\theta, \phi) \delta_{\ell\ell'} \quad (2.1.10)$$

$$\sum_{\ell=0}^{\infty} \frac{2\ell+1}{2} P_\ell(\mu) P_\ell(\mu') = \delta(\mu - \mu') \quad (2.1.11)$$

$$\delta^2(\hat{\mathbf{k}} - \hat{\mathbf{k}}') = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} y_\ell^m(\hat{\mathbf{k}}) y_\ell^{m*}(\hat{\mathbf{k}}') = \frac{1}{2\pi} \delta(\hat{\mathbf{k}} \cdot \hat{\mathbf{k}}' - 1) \quad (2.1.12)$$

where $y_\ell^{m*} = y_\ell^m$, of course, but this is written so we can replace y_ℓ^m with Y_ℓ^m and have the same identities. Now, the first of these identities allows us to expand a function on the unit sphere $f(\hat{\mathbf{k}})$ like:

$$f(\hat{\mathbf{k}}) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} f_\ell^m y_\ell^m(\hat{\mathbf{k}}) \quad (2.1.13)$$

which leads to:

$$f_\ell^m = \int d\Omega' f(\hat{\mathbf{k}}') y_\ell^m(\hat{\mathbf{k}}) \quad (2.1.14)$$

If, however, we expand $f(\hat{\mathbf{k}})$ in spherical harmonics Y_ℓ^m , we will get a different set of coefficients. Let's temporarily call them \hat{f}_ℓ^m . In general, \hat{f}_ℓ^m are complex. They are only real if $f(\phi) = f(2\pi - \phi)$, i.e., if there is azimuthal symmetry about $\phi = \pi$. This would also mean $f(\hat{\mathbf{k}})$ can be written as a weighted sum of $\cos(n\phi)$ over n . This is not the case for the functions we will be expanding. In order to avoid complex coefficients \hat{f}_ℓ^m , which are computationally feasible but unfavorable due to storage and multiplication considerations, we use real spherical harmonics.

Now, the Legendre polynomials themselves will be interesting as well. They happen to obey [25]:

$$\int_{-1}^1 d\mu P_\ell(\mu) P_{\ell'}(\mu) = \frac{2}{2\ell+1} \delta_{\ell\ell'} \quad (2.1.15)$$

Since Legendre polynomials are, by definition, not normalized (rather, they are defined to obey $P_\ell(1) = 1$), there is more than one convention for expansion of a function $g(\mu)$. We use the following convention:

$$g(\mu) = \sum_{\ell=0}^{\infty} g_\ell P_\ell(\mu) \quad (2.1.16)$$

Which then implies:

$$g_\ell = \frac{2\ell+1}{2} \int_{-1}^1 d\mu g(\mu) P_\ell(\mu) \quad (2.1.17)$$

Other conventions may absorb $\sqrt{(2\ell+1)/2}$ into P_ℓ (thereby normalizing it), or leave $(2\ell+1)/2$ in the sum in (2.1.16). Some also involve factors of $1/2\pi$ [16].

Consider the following interpretation of a Legendre polynomial expansion (which can easily be generalized, in the two-dimensional case, to spherical harmonics). The $\ell = 0$ term is just proportional to $\int_{-1}^1 d\mu g(\mu)$. If we consider $g(\mu)$ some kind of distribution in μ , then this tells us the ‘mass’ of this distribution (the total number of whatever g describes). The $\ell = 1$ term is proportional to $\int_{-1}^1 d\mu \mu g(\mu)$, which itself is related to the *average value* of μ in $g(\mu)$. In general, the ℓ term is related to $\int_{-1}^1 d\mu \mu^\ell g(\mu)$, which is the ℓ th ‘moment’ in g . Thus, the quantity g_ℓ informs the asymmetry of the function g up to the order ℓ . If all g except g_0 are very close to zero, then g is very close to isotropic. If, on the other hand, g is extremely anisotropic, we will need an exceedingly high ℓ to accurately describe the landscape of g . The significance of this discussion follows: we will now expand $\Sigma_s(\hat{\mathbf{k}} \cdot \hat{\mathbf{k}}')$, in Legendre polynomials with variable $\hat{\mathbf{k}} \cdot \hat{\mathbf{k}}'$ and ψ in real spherical harmonics with variable $\hat{\mathbf{k}}$, yielding a new scattering operator:

$$\hat{K}\psi = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \sum_{\ell'=0}^{\infty} \int d\Omega' y_\ell^m(\theta', \phi') P_{\ell'}(\mu_s) \int dE' \Sigma_{s,\ell'}(\mathbf{r}, E', E) \psi_\ell^m(\mathbf{r}, E') \quad (2.1.18)$$

Employing the identity (2.1.10), we can simplify:

$$\hat{K}\psi = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \frac{4\pi}{2\ell+1} y_\ell^m(\theta, \phi) \int dE' \Sigma_{s,\ell}(\mathbf{r}, E', E) \psi_\ell^m(\mathbf{r}, E') \quad (2.1.19)$$

Thus far, there is no approximation, and the above happens to be a convenient form of the scattering operator in general. However, to make this computationally useful, we must truncate the ℓ sum to a finite degree, which we will write as L . We have a good rationale for choosing L . We should choose L such that $\Sigma_{s,L+1} \approx 0$, which we know is to say that our scattering is not significantly anisotropic beyond degree L .

Now, we turn to ψ_ℓ^m . When we later formulate a P_N solution, we will pickup with the scattering operator in the form above. For generality, however, we will return ψ to the quantity of our interest by undoing the expression ψ_ℓ^m . Before doing so, it is interesting to note that the $\ell = m = 0$ term of the expansion of ψ is closely related to the fluence:

$$\psi_0^0 = \frac{1}{\sqrt{4\pi}} \varphi$$

(2.1.20)

Nevertheless, using ψ again:

$$\hat{K}\psi = \sum_{\ell=0}^L \sum_{m=-\ell}^{\ell} \frac{4\pi}{2\ell+1} y_\ell^m(\hat{\mathbf{k}}) \int dE' \Sigma_{s,\ell}(\mathbf{r}, E', E) \int d\Omega' \psi(\mathbf{r}, \hat{\mathbf{k}}', E') y_\ell^m(\theta', \phi') \quad (2.1.21)$$

We can now treat sources of the form found in \tilde{s} , i.e., sources constructed by applying a scattering operator to some fluence. Recall:

$$\tilde{s} = \int dE' d\Omega' \Sigma_s \tilde{\varphi}^0 \delta^2 [\hat{\mathbf{k}}_0(\mathbf{r}) - \hat{\mathbf{k}}'] \quad (2.1.22)$$

If we attempt to work out the integral as much as we can, we will still want to have Σ_s expanded in Legendre moments. When this is done, we have:

$$\tilde{s} = \sum_{\ell=0}^L \int dE' \Sigma_{s,\ell} \tilde{\varphi}^0 \int d\Omega' P_\ell(\hat{\mathbf{k}} \cdot \hat{\mathbf{k}}') \delta^2 [\hat{\mathbf{k}}_0(\mathbf{r}) - \hat{\mathbf{k}}'] \quad (2.1.23)$$

At this stage, we could consider proceeding with $P_\ell[\hat{\mathbf{k}} \cdot \hat{\mathbf{k}}_0(\mathbf{r})]$, but in my calculations, that seems to be computationally more difficult than expanding $\delta^2[\hat{\mathbf{k}}_0(\hat{\mathbf{r}}) - \hat{\mathbf{k}}']$ in real spherical harmonics. It may be rationalized on the basis of matrix-less multiplication. Nevertheless, we have:

$$\int d\Omega' P_\ell(\hat{\mathbf{k}} \cdot \hat{\mathbf{k}}') \delta^2 [\hat{\mathbf{k}}_0(\mathbf{r}) - \hat{\mathbf{k}}'] = \sum_{\ell'=0}^L \sum_{m=-\ell'}^{\ell'} y_{\ell'}^{m'} [\hat{\mathbf{k}}_0(\mathbf{r})] \int d\Omega' y_{\ell'}^{m'}(\hat{\mathbf{k}}') P_\ell(\hat{\mathbf{k}} \cdot \hat{\mathbf{k}}') \quad (2.1.24)$$

An earlier identity yields:

$$\int d\Omega' P_\ell(\hat{\mathbf{k}} \cdot \hat{\mathbf{k}}') \delta^2 [\hat{\mathbf{k}}_0(\mathbf{r}) - \hat{\mathbf{k}}'] = \frac{4\pi}{2\ell+1} \sum_{m=-\ell}^{\ell} y_\ell^m [\hat{\mathbf{k}}_0(\mathbf{r})] y_\ell^m(\hat{\mathbf{k}}) \quad (2.1.25)$$

and so we finally have:

$$\tilde{s} = \sum_{\ell=0}^L \sum_{m=-\ell}^{\ell} \frac{4\pi}{2\ell+1} y_\ell^m(\hat{\mathbf{k}}) y_\ell^m [\hat{\mathbf{k}}_0(\mathbf{r})] \int dE' \Sigma_{s,\ell}(\mathbf{r}, E', E) \tilde{\varphi}^0(\mathbf{r}, E') \quad (2.1.26)$$

2.2 Spatial discretization

In order to discretize the spatial part of the phase space, we will use the **Galerkin method** paired with the **finite element method (FEM)**. Unfortunately, the intricate details of the FEM are *far* too involved to fully treat here, and I simply don't have enough expertise to do so anyway. However, this method is easily one of the most powerful computational methods, found in a number of fields of engineering, and so one will have no trouble finding several resources which go into varying levels of depth on the topic. The reader is referred to [27]. In regards to finite elements in the Boltzmann transport equation specifically, the reader is referred to [28, 29]. Thus, I will prioritize less a pedagogical treatment in the following section, and prefer instead to provide the information which is needed to understand wiscobolt.

First, we will introduce the FEM briefly. Then, we will switch gears and reformulate the BTE in a form to which the FEM can be applied – in terms of integral quantities over small volumes. Then, the FEM will be applied.

What follows is derived from the following sources: [3, 27–29]. However, the interested reader is preferentially recommended to read [28] and/or [29], as they are very rigorous, and I have found [3] confusing on this topic. Nevertheless, we will refrain from citations except for statements that are specifically due to a particular source or set of sources.

2.2.1 The finite element method

Despite its technical complexity, the rationale for the FEM is quite simple: we will convert a problem which takes place throughout a large, continuous volume, to the composition of potentially co-dependent problems which take place in small *pieces* of the volume. These ‘small pieces’ are referred to as **elements**. To form elements, we first need to construct a **mesh**, which is a collection of points or **nodes** that are interconnected to form the aforementioned elements. The mesh should meet some criteria for the given problem. The nodes are connected by lines, multiple lines enclose a face, and multiple faces enclose an element. The number of elements in our problem, and the size of each element, are two choices we have to make, and they should be made on a problem-by-problem basis. There exist a number of programs which create meshes from user-defined geometry. These will be a topic of much deeper discussion in **wiscobolt implementation**. An important note to be made here and in **wiscobolt implementation** is how materials are defined. Of course, material definition is a trait that is not general in the FEM. But, it is of great importance in the BTE. The rationale when constructing a mesh is to first define ‘geometry.’ For instance, suppose I have a gold sphere embedded within a cubical water tank. My geometry has thus been described. Then, a mesh is created in this geometry. So, certain elements will correspond to certain materials, in this case, gold and water. It is then noteworthy that we should use a mesh that is refined such that the size of a given element is not significantly greater than the ‘mean free path’ of the particle in question travelling in the material in question. That is because the mean free path is a metric of the length scale over which our particle experiences scattering. If many scattering events are occurring over a certain distance, then we will need greater spatial over that distance. Additionally, near material interfaces, especially those with disparate mean free paths, we should use smaller elements [30], as there will generally be a relatively sharp gradient in the solution due to the interruption of scattering behavior near the interface (such as backscattering).

Now, we start by calling our volume a 3D domain V . Then, we take V and subdivide it into non-overlapping sub-volumes, like V^e for the e th element. V is then said to be given by:

$$V = \bigcup_{e=1}^{N_E} V^e \quad (2.2.1)$$

Where the symbol \bigcup means “union,” and N_E is the number of elements in our mesh. One can use a variety of types of elements, including within the same problem, and each type of element is characterized by the number of nodes (vertices), edges, and faces. For instance, triangles (2D elements) have 3 nodes, 3 edges, and 1 face. Tetrahedra (3D) have 4 nodes, 6 edges, and 4 faces. Hexahedra (3D) have 8 nodes, 12 edges, and 6 faces. So-called structured meshes have a regular structure, i.e., elements of identical type, shape, and volume, while unstructured meshes have non-uniform elements. In structured meshes, the problem can

be simplified, but we will not be concerned with these, as we hope to use larger elements in regions which require less resolution (low Z materials) and smaller elements in regions which require great resolution (high Z materials, material interfaces). Additionally, with unstructured meshes we will need to use what is referred to as the **discontinuous finite element method (DFEM)**, wherein we suggest that the approximate solution we obtain for $\psi(\mathbf{r})$ does not need to agree with the value of ψ at the same physical point but referenced in another element. However, in the end of the computation, one will need to have a method to decide upon one value of ψ at a particular physical point. For instance, one may visit all elements sharing a given node, and take the maximum or minimum value. In some kinds of calculations, including S_N calculations, one may choose to take the ‘upstream’ value (the word ‘upstream’ may appear meaningless in this context, but would make sense to the reader after reading **Section 2.3.2**). We choose to simply average the value of ψ at all nodes corresponding to the same physical point.

2.2.2 Creating the weak form

Before we will see the utility of the FEM, we must reformulate the BTE. In particular, we now seek to write the **weak form** of the BTE, an equation obtained by multiplying through with some arbitrary **test function** $v(\mathbf{r})$ and integrating. In particular the function we seek $\psi(\mathbf{r})$ is one which satisfies, for any square-integrable function $v(\mathbf{r})$, the equation:

$$\int_V d^3\mathbf{r} v(\mathbf{r}) \hat{L}\psi(\mathbf{r}) = \int_V d^3\mathbf{r} v(\mathbf{r}) s(\mathbf{r}) \quad (2.2.2)$$

Or, in “inner product notation”:

$$\langle v, \hat{L}\psi \rangle_V = \langle v, s \rangle_V \quad (2.2.3)$$

where V is the volume in which we seek a solution to ψ . In principle, one should impose this equality by testing *every* v in an infinite-dimensional ‘function space.’ Due to the linearity of the integral operator and \hat{L} , this arbitrarily difficult task can be simplified to examining those v which correspond to a particular *basis set*. If ψ and s themselves permit well-behaved expansions with respect to the same basis set, and the basis set in question plays nicely with the operator \hat{L} such that the integral on the LHS can be explicitly calculated for v and ψ replaced by any two components of the basis, then a straightforward, *exact*, solution may actually be available. For instance, if we expand ψ and s in some basis $\{\phi_n(\mathbf{r})\}$, with coefficients ψ_n and s_n , then take $v = \phi_n(\mathbf{r})$ and require equality for all n , our problem would look like:

$$\sum_{m=1}^{\infty} \psi_m \langle \phi_n, \hat{L}\phi_m \rangle = \sum_{m=1}^{\infty} s_m \langle \phi_n, \phi_m \rangle \quad (2.2.4)$$

If our basis is orthonormal, then we have a neat (albeit infinitely large) matrix equation packaged for us:

$$\sum_{m=1}^{\infty} \psi_m \langle \phi_n, \hat{L}\phi_m \rangle = s_n \quad (2.2.5)$$

and we could define $L_{nm} \equiv \langle \phi_n, \hat{L} \phi_m \rangle$ as the ‘matrix elements’ of \hat{L} . Then:

$$\sum_{m=1}^{\infty} L_{nm} \psi_m = s_n \quad (2.2.6)$$

A matrix problem, with a deeper structure very similar to that in many physics problems. In essence, what we’ve done is swapped the ‘variable’ with which we express ψ , from \mathbf{r} to n . Some of the simpler problems in physics are actually *soluble* in the form above. Others can be readily solved up to finite order. Our problem, however, is not so kind. In order to find a useful equation similar to (2.2.6), we will have to resort to the **Galerkin method**, in which one decides to be satisfied with a solution valid for a certain set of test functions in a finite-dimensional subspace. We will use a spatial mesh not only to discretize the spatial part of phase space, but also to allow ourselves to pick different test functions within different element, making for a more efficient and modular problem. It can be noted as well that the Galerkin method on a mesh, just like all ‘good’ discretization methods, converges to the exact solution as the discretization becomes more refined, by virtue of our problem looking more like (2.2.6), although, this will not be proven here.

We will process the weak form of the BTE further before applying it to a mesh or picking test functions. The whole weak form could be written:

$$\int_V d^3\mathbf{r} (v\hat{\mathbf{k}} \cdot \nabla \psi + v\Sigma_t \psi) = \int_V d^3\mathbf{r} vs + \int_V d^3\mathbf{r} v\hat{K}\psi \quad (2.2.7)$$

We leave $\hat{K}\psi$ in this form because \hat{K} does not mix the spatial dependence of ψ , so we will find its treatment very simple. The most important term to consider is $\hat{\mathbf{k}} \cdot \nabla \psi$. We will work this integral through so that we can peel off the ∇ from ψ . This is because we will soon want to expand ψ to a finite order, which constitutes an approximation, one which is made worse by the application of a derivative. As a matter of fact, it is one which is made *much* worse by the application of a derivative. We will later see that if we were to proceed with this form, we would disallow particles the ability to stream between two disparate locations in space. Nevertheless, we will use the chain rule to write:

$$v\nabla\psi = \nabla(v\psi) - \psi\nabla v \quad (2.2.8)$$

We have no issue taking ∇v because v is a test function, and the weak form of the BTE is not an approximation itself. Now, the streaming term in the weak BTE is:

$$\int_V d^3\mathbf{r} v\hat{\mathbf{k}} \cdot \nabla \psi = \int_V d^3\mathbf{r} \hat{\mathbf{k}} \cdot \nabla(v\psi) - \int_V d^3\mathbf{r} \psi\hat{\mathbf{k}} \cdot \nabla v \quad (2.2.9)$$

The first term, which still contains ∇ acting on ψ , should be worked out using **Gauss’ theorem**, which relates the volume integral of a divergence of a vector field (in this case $\hat{\mathbf{k}}v\psi$) to the flux of the vector field over the boundary of the volume. That is, we now have:

$$\int_V d^3\mathbf{r} \hat{\mathbf{k}} \cdot \nabla(v\psi) = \oint_{\Gamma} dA (\hat{\mathbf{k}} \cdot \hat{\mathbf{n}}) v\psi \quad (2.2.10)$$

This integral may be split into two utilizing the boundary condition $\bar{\psi}$. If $\hat{\mathbf{k}}$ is pointing into the volume, we can use $\psi = \bar{\psi}$. Now, this means we are taking, within the integrand:

$$\psi = \begin{cases} \psi, & (\mathbf{r}, \hat{\mathbf{k}}) \in \Gamma_{\uparrow} \\ \bar{\psi}, & (\mathbf{r}, \hat{\mathbf{k}}) \in \Gamma_{\downarrow} \end{cases} \quad (2.2.11)$$

Given that Γ_{\uparrow} is when $\hat{\mathbf{k}} \cdot \hat{\mathbf{n}} > 0$, and Γ_{\downarrow} is when $\hat{\mathbf{k}} \cdot \hat{\mathbf{n}} < 0$, this can be written in one line:

$$(\hat{\mathbf{k}} \cdot \hat{\mathbf{n}})\psi = \frac{1}{2}(\hat{\mathbf{k}} \cdot \hat{\mathbf{n}} + |\hat{\mathbf{k}} \cdot \hat{\mathbf{n}}|)\psi + \frac{1}{2}(\hat{\mathbf{k}} \cdot \hat{\mathbf{n}} - |\hat{\mathbf{k}} \cdot \hat{\mathbf{n}}|)\bar{\psi} \quad (2.2.12)$$

For notational brevity, make the definitions:

$$\begin{aligned} \varsigma_{\uparrow}(\hat{\mathbf{n}}, \hat{\mathbf{k}}) &\equiv \frac{1}{2}(\hat{\mathbf{k}} \cdot \hat{\mathbf{n}} + |\hat{\mathbf{k}} \cdot \hat{\mathbf{n}}|) \\ \varsigma_{\downarrow}(\hat{\mathbf{n}}, \hat{\mathbf{k}}) &\equiv \frac{1}{2}(\hat{\mathbf{k}} \cdot \hat{\mathbf{n}} - |\hat{\mathbf{k}} \cdot \hat{\mathbf{n}}|) \end{aligned} \quad (2.2.13)$$

Now, we can write the surface integral of the streaming term:

$$\oint_{\Gamma} dA (\hat{\mathbf{k}} \cdot \hat{\mathbf{n}})v\psi = \int_{\Gamma_{\uparrow}} dA \varsigma_{\uparrow}v\psi + \int_{\Gamma_{\downarrow}} dA \varsigma_{\downarrow}v\bar{\psi} \quad (2.2.14)$$

and this is as far as we will simplify the weak form of the BTE. To restate the weak form of the BTE, *with no approximation*, we have:

$$\int_{\Gamma_{\uparrow}} dA \varsigma_{\uparrow}v\psi + \int_{\Gamma_{\downarrow}} dA \varsigma_{\downarrow}v\bar{\psi} - \int_V d^3\mathbf{r} \psi \hat{\mathbf{k}} \cdot \nabla v + \int_V d^3\mathbf{r} v \Sigma_t \psi = \int_V d^3\mathbf{r} vs + \int_V d^3\mathbf{r} v \hat{K} \psi$$

$$(2.2.15)$$

This abstract form is not meant to be easy to digest, it simply serves as a starting point for us to determine an approximate matrix problem like in (2.2.6).

2.2.3 Applying the FEM

Now, we decide to expand ψ in a basis function. In the FEM, we start by suggesting that the basis functions we choose are those which correspond to interpolation of our function between the nodes of an element. Thus, ψ throughout a whole element is obtained by interpolating with ψ known only at the nodes. Now, this *does not* need to be linear interpolation, however, we will stick with linear interpolation with first order elements because as of this version, wiscobolt is only capable of linear elements (ETA for higher order elements is unknown – requests go a long way though). Quadratic interpolation with higher order elements can offer tremendous benefits: it shifts part of the burden of what follows from computation to explicit mathematics and clever algorithms. In doing so, it does also allow one to use a less refined mesh but obtain a solution which is often on-par or more accurate than solutions obtained with linear interpolation on a heavily refined mesh [29]. However, when considering the FEM with S_N angular discretization, one particular aspect of the computation is rather difficult to work through and implement with quadratic interpolation. It is known as the

“sweep,” though for reasons we will understand only when we reach **Section 2.3**. Still, the difficulty hasn’t stopped others from attempting solutions or workarounds, and they have done so with great success. For those who are interested in the use of quadratic interpolation in solving the BTE, I highly recommend reading the paper from Haut et al. on the subject [29].

The scheme for linear interpolation always goes as follows: consider an element e with N_K^e nodes. To get the value of ψ at some \mathbf{r} strictly within this element knowing only ψ at the nodes of the element, that is, $\psi(\mathbf{r}_k^e)$, we say:

$$\boxed{\psi(\mathbf{r}) \approx \sum_{k=1}^{N_K^e} u_k^e(\mathbf{r}) \psi_k^e, \mathbf{r} \in V^e} \quad (2.2.16)$$

where:

$$\boxed{\psi_k^e \equiv \psi(\mathbf{r}_k^e)} \quad (2.2.17)$$

The function $u_k^e(\mathbf{r})$ is the **shape function** of the element e and node k . To get a slightly better understanding of what the k th shape function really *does*, let’s consider the simplest case of interpolation – 1D linear interpolation. The formula for 1D linear interpolation of the function $y(x)$ between points x_1 and x_2 is:

$$y(x) \approx \frac{y(x_2) - y(x_1)}{x_2 - x_1} (x - x_1) + y(x_1), \quad x \in [x_1, x_2] \quad (2.2.18)$$

If we work this out to write it in terms of a weighted sum of $y(x_1)$ and $y(x_2)$, we see:

$$y(x) \approx \left(1 - \frac{x - x_1}{x_2 - x_1}\right) y(x_1) + \frac{x - x_1}{x_2 - x_1} y(x_2) \quad (2.2.19)$$

The shape functions are therefore:

$$\begin{aligned} u_1(x) &= 1 - \frac{x - x_1}{x_2 - x_1} = \frac{x_2 - x}{x_2 - x_1} \\ u_2(x) &= \frac{x - x_1}{x_2 - x_1} \end{aligned} \quad (2.2.20)$$

We also here see another common method of interpolation, wherein the line between $y(x_1)$ and $y(x_2)$ is parametrized, and if we let t describe the fraction of length along the line inbetween these points, we can range t from 0 to 1 to perform interpolation with:

$$y[x(t)] \approx (1 - t)y(x_1) + t y(x_2) \quad (2.2.21)$$

Evidently, $t(x) = (x - x_1)/(x_2 - x_1)$ (we will actually use shape functions of this form in **Section 2.6**). Any way you look at it, the shape functions take geometric information about the inputs x_i , *not* the outputs y_i , and give us weights which sum to 1, as:

$$\boxed{\sum_{k=1}^{N_K^e} u_k^e(\mathbf{r}) = 1, \mathbf{r} \in V^e} \quad (2.2.22)$$

which also follows from considering the above as ‘interpolation of $f(\mathbf{r}) = 1$.’ They also satisfy:

$$u_k^e(\mathbf{r}_{k'}^e) = \delta_{kk'} \quad (2.2.23)$$

which follows from evaluating some $f(\mathbf{r})$ at $\mathbf{r}_{k'}^e$. Both of these equations are useful to verify that one is able to accurately construct shape functions. For linear interpolation in 3D, shape functions most generally come in the form:

$$u_k^e(\mathbf{r}) = A_k^e + B_k^e x + C_k^e y + D_k^e z \quad (2.2.24)$$

where the coefficients depend on all \mathbf{r}_k^e for the e in question.

Currently, wiscobolt exclusively deals with tetrahedral elements. Thus, $N_K^e = 4$, and $N_F^e = 4$ (the number of faces). The shape functions for tetrahedral elements are given without proof as:

$$u_k^e(\mathbf{r}) = \frac{1}{6V^e}(a_k^e + b_k^e x + c_k^e y + d_k^e z) \quad (2.2.25)$$

where:

$$V^e = \frac{1}{6} \det S^e \quad (2.2.26)$$

is the volume of the element (we’re recycling the symbol for the element domain), and:

$$a_k^e = \text{cofactor}(S^e)_{k1} \quad (2.2.27)$$

$$b_k^e = \text{cofactor}(S^e)_{k2} \quad (2.2.28)$$

$$c_k^e = \text{cofactor}(S^e)_{k3} \quad (2.2.29)$$

$$d_k^e = \text{cofactor}(S^e)_{k4} \quad (2.2.30)$$

while the matrix S^e is:

$$S^e \equiv \begin{pmatrix} 1 & x_1^e & y_1^e & z_1^e \\ 1 & x_2^e & y_2^e & z_2^e \\ 1 & x_3^e & y_3^e & z_3^e \\ 1 & x_4^e & y_4^e & z_4^e \end{pmatrix} \quad (2.2.31)$$

We will conventionally define a vector containing the last three coefficients in the shape function, as:

$$\mathbf{b}_k^e \equiv (b_k^e, c_k^e, d_k^e) \quad (2.2.32)$$

and so it could be said:

$$u_k^e(\mathbf{r}) = \frac{1}{6V^e}(a_k^e + \mathbf{b}_k^e \cdot \mathbf{r}) \quad (2.2.33)$$

Now, we can insert these shape functions into our BTE by first considering what happens to the volume integrals that go over the entire mesh:

$$\int_V d^3\mathbf{r} \rightarrow \sum_{e=1}^{N_E} \int_{V^e} d^3\mathbf{r} \quad (2.2.34)$$

and then, within each element's integrand, ψ can be safely expanded in terms of shape functions corresponding to that element. For instance:

$$\int_V d^3\mathbf{r} \psi v = \sum_{e=1}^{N_E} \sum_{k'=1}^{N_K^e} \psi_{k'}^e \int_{V^e} d^3\mathbf{r} u_{k'}^e v \quad (2.2.35)$$

The *surface integrals*, on the other hand, are a bit less straightforward. Using Stokes' theorem, we can reason that one surface integral extending over Γ can be considered the sum of surface integrals of arbitrarily sized and shaped pieces inside the volume V . That is to say, we can integrate over all of the faces of all of our mesh elements and add the surface integrals together. This is not an approximation. This is the mathematical justification for breaking our entire problem into several problems inside the volume, such that we are going to be able to consider a single element at a time as long as we are still integrating over its entire surface (as we should expect). Now, we must learn how to speak about element faces. We will later want to index element faces on a local basis (i.e., for tetrahedral elements, $f = 1, 2, 3, 4$, more generally, $f = 1, \dots, N_F^e$), but right now we will define any face within the mesh by stating the two elements which share the face, as in (e, e') . In that case, we take:

$$\oint_{\Gamma} dA \rightarrow \sum_{e=1}^{N_E} \sum_{e'} \int_{\Gamma^{ee'}} dA \quad (2.2.36)$$

where the sum over e' is not stated as running from 1 to N_E , because there will be in general only N_F^e faces for the element e , and so only as many *viable* pairs (e, e') . We *do* consider (e, e') and (e', e) as distinct faces in this sum, and in fact, the two faces have different normal vectors anyway. We don't consider $e = e'$ to have any meaning in the surface integral, so such terms will just be zero. Now, however, we discuss what is being integrated. We have split the Γ integral into an integral for Γ_{\uparrow} , involving ψ , and one for Γ_{\downarrow} , involving $\bar{\psi}$. Now that our surface integrals run over *the elements themselves*, the “boundary condition” for a given element is the value of ψ in the neighboring element, thus we'd expand ψ not as $\psi_{k'}^e$ but as $\psi_{k'}^{e'}$. This distinction matters in the DFEM! The Γ_{\downarrow} integrand is only nonzero when $\hat{\mathbf{k}}$ is *incident* upon the volume, thus, a $\Gamma_{\downarrow}^{ee'}$ integrand is similarly only nonzero when $\hat{\mathbf{k}}$ is incident on the face (e, e') according to e . This is also enforced by the presence of $\varsigma_{\uparrow}^{ee'}$ and $\varsigma_{\downarrow}^{ee'}$, so there is no need to reiterate that integration occurs only over $\Gamma_{\uparrow}^{ee'}$ or $\Gamma_{\downarrow}^{ee'}$. Physically, this means we are allowing fluence from e' to propagate into e when e' is ‘upstream’ of e . We can write out the whole area integral:

$$\oint_{\Gamma} dA (\hat{\mathbf{k}} \cdot \hat{\mathbf{n}}) v \psi = \sum_{e=1}^{N_E} \sum_{e'} \sum_{k'=1}^{N_K^e} \left(\psi_{k'}^e \int_{\Gamma^{ee'}} dA \varsigma_{\uparrow}^{ee'} u_{k'}^e v + \psi_{k'}^{e'} \int_{\Gamma^{ee'}} dA \varsigma_{\downarrow}^{ee'} u_{k'}^{e'} v \right) \quad (2.2.37)$$

If e lies on the boundary of V , then we use $\psi_{k'}^{e'} = \bar{\psi}(\mathbf{r}_{k'}^e)$, i.e, just use the boundary condition at the physical point $\mathbf{r}_{k'}^e$. The equation above will be made easier to interpret once we carry out angular discretization, but we will discuss it briefly first so that you can recognize the significance of this perhaps unintuitive approach. In P_N , we will find that the above leads to some tricky matrix elements in (e, e') , but ones that generally allow propagation between

two elements regardless of angular index. In S_N , however, this equation will lead to a very important step in S_N -based solution methods, referred to as the sweep, wherein we are actually able to choose specific $\hat{\mathbf{k}}$ values and determine which elements depend on which other elements.

Now, we have expanded ψ and transformed our integrals to the mesh. If we write out our discretized weak form, we find a persistent sum over e on both sides of the equation. This sum can be thrown off while maintaining equality, meaning that we have a system of N_E equations. We find:

$$\sum_{k'=1}^{N_K^e} \left(- \int_{V^e} d^3 \mathbf{r} u_{k'}^e \hat{\mathbf{k}} \cdot \nabla v + \sum_{e'} \varsigma_{\uparrow}^{ee'} \int_{\Gamma^{ee'}} dA u_{k'}^e v + \Sigma_t^e \int_{V^e} d^3 \mathbf{r} u_{k'}^e v \right) \psi_{k'}^e = \\ \int_{V^e} d^3 \mathbf{r} v s + \sum_{k'=1}^{N_K^e} \left(\int_{V^e} d^3 \mathbf{r} u_{k'}^e v \right) \int dE' d\Omega' \Sigma_s^e \psi_{k'}^e - \sum_{k'=1}^{N_K^e} \sum_{e'} \left(\varsigma_{\downarrow}^{ee'} \int_{\Gamma^{ee'}} dA u_{k'}^{e'} v \right) \psi_{k'}^{e'} \quad (2.2.38)$$

Since we use first-order elements, the $\varsigma^{ee'}$ terms have been removed from integration. Now, the last step is to perform the **Galerkin method**. Rather than search for a solution to the equation above over every possible function v , for a given e we search for a solution which satisfies the equation in a subspace that is relevant to that e , namely, the subspace formed by shape functions. This means we can take $v = u_k^e$, and impose that the above is satisfied for every k in an element e , finally forming our matrix problem. Thus, the use of a mesh not only discretizes our problem, but also allows us to avoid the use of a global test function subspace, greatly simplifying the problem, one of the many motivations of the finite element method. Nevertheless, let's here define the various inner products that we appear to need:

$$I_{1,kk'}^e \equiv \int_{V^e} d^3 \mathbf{r} u_{k'}^e u_k^e \quad (2.2.39)$$

$$I_{2\uparrow,kk'}^{ee'} \equiv \int_{\Gamma^{ee'}} dA u_{k'}^e u_k^e \quad (2.2.40)$$

$$I_{2\downarrow,kk'}^{ee'} \equiv \int_{\Gamma^{ee'}} dA u_{k'}^{e'} u_k^e \quad (2.2.41)$$

$$\mathbf{I}_{3,kk'}^e \equiv \int_{V^e} d^3 \mathbf{r} u_{k'}^e \nabla u_k^e \quad (2.2.42)$$

With these definitions, our problem can be written a bit more succinctly:

$$\sum_{k'=1}^{N_K^e} \left(- \hat{\mathbf{k}} \cdot \mathbf{I}_{3,kk'}^e + \sum_{e'} \varsigma_{\uparrow}^{ee'} I_{2\uparrow,kk'}^{ee'} + \Sigma_t^e I_{1,kk'}^e \right) \psi_{k'}^e = \\ \int_{V^e} d^3 \mathbf{r} u_k^e s + \sum_{k'=1}^{N_K^e} I_{1,kk'}^e \int dE' d\Omega' \Sigma_s^e \psi_{k'}^e - \sum_{k'=1}^{N_K^e} \sum_{e'} \varsigma_{\downarrow}^{ee'} I_{2\downarrow,kk'}^{ee'} \psi_{k'}^{e'} \quad (2.2.43)$$

Now, the inner product $I_{1,kk'}^e$ is rather persistent, and I prefer to explicitly invert it on both sides before proceeding. Noting that the units of $I_{1,kk'}^e$ are volume, a factor introduced during creation of the weak form, I see this as a restoration of the original BTE. With this inversion, the matrices multiplying ψ_k^e once again have units of inverse length, and more importantly, the scattering matrix (which is not *supposed to* map particles between two positions) will not have any dependence on the node, only the element generally. The inner product operators, with I_1^{-1} applied, will be designated by the same symbols, except with a prime next to their numbering. Now, to borrow notation from [29], we will define:

$$G_{kk'}^e \equiv -\hat{\mathbf{k}} \cdot \mathbf{I}_{3',kk'}^e \quad (2.2.44)$$

$$F_{\uparrow,kk'}^e \equiv \sum_{e'} \varsigma_{\uparrow}^{ee'} I_{2',\uparrow,kk'}^{ee'} \quad (2.2.45)$$

$$F_{\downarrow,kk'}^{ee'} \equiv \varsigma_{\downarrow}^{ee'} I_{2',\downarrow,kk'}^{ee'} \quad (2.2.46)$$

$$M_{kk'}^e \equiv \Sigma_t^e \delta_{kk'} \quad (2.2.47)$$

It is worth noting that all but the last of these matrix elements are continuous functions of $\hat{\mathbf{k}}$.

Now, we will briefly discuss integrals of s and the shape function u_k^e in the FCS method. If not using the FCS method, the following note is not really relevant unless we wish to determine and plot $\hat{T}^{-1}s$ numerically, in which case the object that we obtain for node (e, k) is actually the integral of $\hat{T}^{-1}s$ and u_k^e , which is only (up to application of I_1^{-1}) identical to $(\hat{T}^{-1}s)(\mathbf{r}_k^e)$ if this quantity varies almost linearly across the element. Nevertheless, in the FCS method, s is actually \tilde{s} , whose spatial dependence is essentially that of $\tilde{\varphi}^0$. It could be said that \tilde{s} can be expanded in shape functions, then the integral becomes matrix multiplication of $\tilde{s}(\mathbf{r}_k^e)$ with I_1 . However, sometimes \tilde{s} isn't linear, even with an almost impractical number of elements. Now, we know that $\tilde{\varphi}^0$ is related to the exponential of the optical path length from an external beam source to the point \mathbf{r} . This is noteworthy in the context of electron transport, for which the attenuation coefficients Σ_t are rather large. They quantify exponential decay, as per the distance $1/\Sigma_t$, which is often on the order of microns. Therefore, to approximate $\tilde{\varphi}^0$ as being linear with depth, we would need to have an extremely refined mesh towards the surface of our volume that is facing the beam. This is not practical in general, but especially, for instance, in the case of a thin but wide volume which is totally inside of the beam. Yet, linearity may remain a good approximation for both ψ and $\tilde{\psi}$. So, the solution is to perform the integrals we reference in a more sophisticated manner, rather than using an expansion of $\tilde{\varphi}^0$. That is to say, we will take:

$$\tilde{s}_k^e \equiv \sum_{k'=1}^{N_K^e} (I_1^{-1})_{kk'}^e \int_{V^e} d^3\mathbf{r} u_{k'}^e(\mathbf{r}) \tilde{s}(\mathbf{r}) \quad (2.2.48)$$

rather than $\tilde{s}(\mathbf{r}_k^e)$. In principle, $\tilde{s}(\mathbf{r})$ is able to be determined at any point in the volume. This may not be obvious pending other forms of discretization, but this is because we need only to determine $\tilde{\varphi}^0$ at some given point in the volume. The form of $\tilde{\varphi}^0(\mathbf{r})$ is, as we discussed in **Section 1.2**, able to be determined analytically for any choice of \mathbf{r} . So, rather than expand \tilde{s} , we will want to discuss the case and methodology wherein one must evaluate this integral

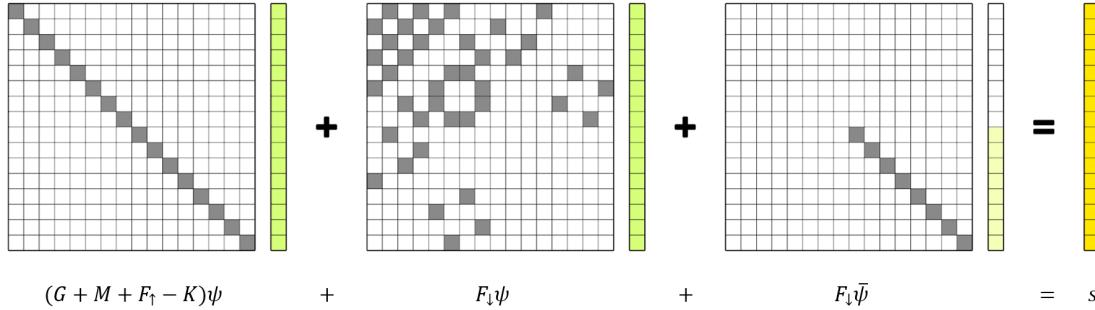


Figure 1: A hypothetical problem structure for a tetrahedral mesh of 16 elements. Each matrix entry corresponds to a 4×4 matrix, potentially with operators such as $\int dE d\Omega$, while each vector entry corresponds to a column vector of length 4. A discussion on how this problem can be approached is relegated to **Section 2.3.2**.

numerically. We will opt to put off this discussion until **Section 3.3.4**, because it will involve integration techniques such as quadrature, which we have yet to discuss. Additionally, it is only a relevant note for massive charged particles, such as electrons, rather than photons.

Now, we can publish the final form of our spatially discretized BTE:

$$\sum_{k'=1}^{N_K^e} \left(G_{kk'}^e + M_{kk'}^e + F_{\uparrow, kk'}^e \right) \psi_{k'}^e + \sum_{k'=1}^{N_K^e} \sum_{e'} F_{\downarrow, kk'}^{ee'} \psi_{k'}^{e'} = s_k^e + \int dE' d\Omega' \Sigma_s^e \psi_k^e \quad (2.2.49)$$

Recognize something very important here: this equation describes a matrix problem in (k, k') (i.e., two nodes), *and* in (e, e') , i.e., two elements. Think of this problem as a block-matrix problem, where (e, e') indexes blocks and (k, k') indexes elements within a block. Since all interior elements only share N_F^e faces with their neighbors, the row e (if it describes an interior element) will only have $N_F^e + 1$ nonzero blocks, where N_F^e are due to off-diagonal blocks (only $F_{\downarrow, kk'}^{ee'}$ has these) and 1 is due to diagonal terms. Now, if the element has some number of faces on the boundary, there will be an additive vector corresponding to these elements with the last term constructed using boundary conditions. In **Figure 1**, we show the hypothetical structure of this matrix problem for an actual, representative, 16 element mesh that was created. Note that, without other forms of discretization, this ‘matrix problem’ still involves many continuous operators. Nevertheless, it is worth noting that this matrix will generally be *huge*, and its inversion is very nontrivial. We will learn more about how this can be made feasible when we discretize angle and discuss iteration methods.

Note finally that, in the above equation, we could define the portion of the transport operator which is diagonal as:

$$T_{kk'}^e \equiv G_{kk'}^e + M_{kk'}^e + F_{\uparrow, kk'}^e \quad (2.2.50)$$

which we’ll *call* the transport matrix, but this isn’t really the spatially discretized form of \hat{T} . The whole discretized form of \hat{T} is not diagonal in e , and would include the $F_{\downarrow, kk'}^{ee'}$ (from the RHS). More specifically, $T_{kk'}^e$ is missing part of \hat{T} ’s streaming operator.

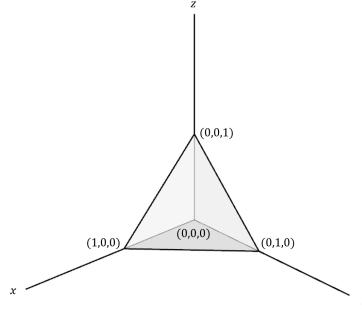


Figure 2: Linear tetrahedron.

2.2.4 Evaluation of spatial inner products for tetrahedra

We can now discuss solutions of the inner products $I_{1,kk'}^e$, $I_{2\uparrow,kk'}^e$, $I_{2\downarrow,kk'}^{ee'}$, and $\mathbf{I}_{3,kk'}^e$ put explicitly in terms of \mathbf{r}_k^e . We restate each inner product here:

$$I_{1,kk'}^e = \langle u_k^e, u_{k'}^e \rangle_{V^e} \quad (2.2.51)$$

$$I_{2\uparrow,kk'}^e = \langle u_k^e, u_{k'}^e \rangle_{\Gamma^{ee'}} \quad (2.2.52)$$

$$I_{2\downarrow,kk'}^{ee'} = \langle u_k^e, u_{k'}^{e'} \rangle_{\Gamma^{ee'}} \quad (2.2.53)$$

$$\mathbf{I}_{3,kk'}^e = \langle u_{k'}^e, \nabla u_k^e \rangle_{V^e} \quad (2.2.54)$$

Let's begin with the first inner product. One way to work with the 3D integrals is, for a given element, to *transform* the coordinate system to one in which the integral becomes easier to solve [3, 29]. This method is used for various types of elements, regardless of order [29]. With a first-order tetrahedron, we transform an element to a ‘linear tetrahedron.’ This is depicted in **Figure 2**. A linear tetrahedron *always* has the following structure:

k	x'_k	y'_k	z'_k
1	0	0	0
2	1	0	0
3	0	1	0
4	0	0	1

while its shape functions, i.e. $u_k[\mathbf{r}(\mathbf{r}')]$, are consequently:

$$\begin{aligned} u_1(\mathbf{r}') &= 1 - x' - y' - z' \\ u_2(\mathbf{r}') &= x' \\ u_3(\mathbf{r}') &= y' \\ u_4(\mathbf{r}') &= z' \end{aligned} \quad (2.2.55)$$

These are used to write the coordinate transformations. We can exactly expand, say, $x(\mathbf{r}')$ in terms of these shape functions, giving:

$$x(\mathbf{r}') = \sum_{k=1}^{N_K^e} x_k^e u_k(\mathbf{r}') \quad (2.2.56)$$

and likewise for y and z . This expansion is valid within the element e only. One then converts an integral in \mathbf{r} to one in \mathbf{r}' using the Jacobian:

$$J^e(\mathbf{r}') = \begin{pmatrix} \partial x / \partial x' & \partial y / \partial x' & \partial z / \partial x' \\ \partial x / \partial y' & \partial y / \partial y' & \partial z / \partial y' \\ \partial x / \partial z' & \partial y / \partial z' & \partial z / \partial z' \end{pmatrix} \quad (2.2.57)$$

With x , y , and z expressed in e . Generally, one major utility of the Jacobian is that:

$$\int_{V^e} d^3\mathbf{r} f(\mathbf{r}) = \int_{V^e} d^3\mathbf{r}' \det [J^e(\mathbf{r}')] f[\mathbf{r}(\mathbf{r}')] \quad (2.2.58)$$

Of course, this holds for a general coordinate transformation, but ours is affine (all derivatives are constant), so we know $\det J^e$ is constant, so it can be removed from integration. As a matter of fact, one can quite easily derive the precise value $\det J^e$ should take. Suppose we are interested in $f(\mathbf{r}) = 1$:

$$\int_{V^e} d^3\mathbf{r} = \det J^e \int_{V^e} d^3\mathbf{r}' \quad (2.2.59)$$

Since the volume of a linear tetrahedron is $1/6$:

$$\det J^e = 6V^e \quad (2.2.60)$$

This result is geometrically quite elegant. The Jacobian itself is a transformation matrix for infinitesimal displacements localized at a given point. Thus, the determinant of this transformation matrix is the ratio, at any given point in the new system, of a volume element in the old system to the new system. In the case of a linear transformation, the structure of a volume element is constant. The volume (though it's unitless) of a linear tetrahedron is $1/6$, the volume of a tetrahedron is V^e . Hence, $V^e/(1/6) = \det J^e$. Thus, we will eliminate all use of $\det J^e$, preferring to use $6V^e$.

Now, let's perform our integral in the linear tetrahedron:

$$I_{1,kk'}^e = 6V^e \int_{V^e} d^3\mathbf{r}' u_k(\mathbf{r}') u_{k'}(\mathbf{r}') \quad (2.2.61)$$

Which is a simple volume integration problem. Expressing the volume of integration, the bounds come to be:

$$\int_{V^e} d^3\mathbf{r}' u_k(\mathbf{r}') u_{k'}(\mathbf{r}') = \int_0^1 dx' \int_0^{1-x'} dy' \int_0^{1-x'-y'} dz' u_k(\mathbf{r}') u_{k'}(\mathbf{r}') \quad (2.2.62)$$

This integration is easily carried out for any combination of (k, k') . The final result is:

$$I_{1,kk'}^e = \frac{1}{20} V^e (1 + \delta_{kk'}) \quad (2.2.63)$$

From here we can show:

$$(I_1^{-1})_{kk'}^e = \frac{4}{V^e} (-1 + 5\delta_{kk'}) \quad (2.2.64)$$

Next, we'll deal with $I_{2\uparrow, kk'}^{ee'}$. Here, we will begin to adopt the convention that faces are labelled by (e, f) , where f is an index of the faces belonging to e , instead of describing a face by the two elements which share it, (e, e') . Now, we could then say that we can determine e' by specifying e and f , i.e., $e' = e'(e, f)$. We could also determine the face's index in e' , i.e., $f' = f'(e, f)$. The meaning of these maps is simple: $e'(e, f)$ says that if you pick an element e , and a face within that element f , then $e'(e, f)$ is the element sharing that face. Then, $f'(e, f)$ is the index of this face within e' . This leads to a useful identity:

$$\begin{aligned} e'[e'(e, f), f'(e, f)] &= e \\ f'[e'(e, f), f'(e, f)] &= f \end{aligned} \quad (2.2.65)$$

Construction of these maps is *tremendously important* in wiscobolt for reasons we will see during angular discretization, and will be described in adequate depth in the **wiscobolt implementation** document. Nevertheless, for a surface integral, our coordinate transformation takes a slightly different form. Namely, rather than transform to a linear tetrahedron, we will consider transformations to map a given face to a simple unit right triangle. Our coordinates will be x' and y' . Now, we can do this because, if shape functions are evaluated on $\mathbf{r} \in \Gamma^{ef}$, then the shape function for the node(s) which are not on the face will be zero entirely (in the case of a tetrahedron, only one node lies outside of a chosen face). Note that we could still use a linear tetrahedron, but this will require an asymmetric treatment given to the face opposite of node 1, which would have to be cancelled out by a tedious convention on our 'surface Jacobian' which will be introduced shortly. This way is easier. Now, before proceeding, let's define face indices with respect to node indices. Let the face f be the one which contains all nodes except node f , i.e., the face opposite to node f . Our integrals will obey the transformation relationship:

$$dA = |\mathbf{J}_{s,f}^e(\mathbf{r}')| dx' dy' \quad (2.2.66)$$

Where $\mathbf{J}_{s,f}^e(\mathbf{r}')$ is the surface Jacobian, a vector given by:

$$\mathbf{J}_{s,f}^e = \left(\frac{\partial x}{\partial x'}, \frac{\partial y}{\partial x'}, \frac{\partial z}{\partial x'} \right) \times \left(\frac{\partial x}{\partial y'}, \frac{\partial y}{\partial y'}, \frac{\partial z}{\partial y'} \right) \quad (2.2.67)$$

Which is useful when working with nonlinear shape functions. However, we *are not* working with nonlinear shape functions, so we can forego this technical description and devise a simpler one. Let's look for something similar to $\det J^e = 6V^e$. With linear shape functions, the derivatives featured in (2.2.67) come out to be constant again. Recognize then that:

$$\int_{\Gamma^{ef}} dA = |\mathbf{J}_{s,f}^e(\mathbf{r}')| \int_{\Gamma^{ref}} dx' dy' \quad (2.2.68)$$

On the LHS, we have the area of a face f in element e . On the RHS, we have the magnitude of the surface Jacobian multiplied by the area of our unit right triangle, which is just $1/2$. Therefore:

$$|\mathbf{J}_{s,f}^e| = 2A_f^e \quad (2.2.69)$$

As with the analogous relationship for the Jacobian, this does *not* generally hold for nonlinear shape functions. A simple formula for the areas, which can be supplied by knowledge of the

mesh, is Heron's formula, which gives the area of a tetrahedral face based solely on its side lengths:

$$A = \sqrt{s(s - \alpha)(s - \beta)(s - \gamma)} \quad (2.2.70)$$

Where α , β , and γ describe the side lengths of the tetrahedral face, and:

$$s = \frac{1}{2}(\alpha + \beta + \gamma) \quad (2.2.71)$$

Using our convention for naming faces, we would have something like:

$$\begin{aligned} \alpha_f^e &= |\mathbf{r}_{k'}^e - \mathbf{r}_{k''}^e| \\ \beta_f^e &= |\mathbf{r}_{k'}^e - \mathbf{r}_{k'''}^e| \\ \gamma_f^e &= |\mathbf{r}_{k''}^e - \mathbf{r}_{k'''}^e| \end{aligned} \quad (2.2.72)$$

Where $k', k'', k''' \neq f$. We can then proceed to perform integration. The shape functions are the same as for the linear tetrahedron, but $z' = 0$, and we eliminate $u_4(\mathbf{r}')$ entirely. Working it through yields an answer that is intuitive in form:

$$I_{2\uparrow, kk'}^{ef} = \frac{1}{12} A_f^e (1 - \delta_{kf})(1 - \delta_{k'f})(1 + \delta_{kk'}) \quad (2.2.73)$$

Specifically, we see that if the face f does not contain either the node k or k' (i.e., if $f = k$ or if $f = k'$), then the entire inner product is zero, as this term accounts for transport occurring from k' to k only if both are on face f . Note as well that this was not *assumed*, the shape functions $u_k^e(\mathbf{r} \in \Gamma^{ef})$ are, in fact, zero when $k = f$. If $k = k'$, the inner product is doubled. Nevertheless, we use this form to write the next inner product, $I_{2\downarrow, kk'}^{ef}$. Writing out that integral, we find a bit of difficulty in matching up the coordinate systems of e and $e'(e, f)$ for transformation. Yet, we recognize that if the node k , which is in element e , refers to a node not on the face f (i.e., $k = f$), then the inner product should be zero. Similarly, if the node k' , which is in element $e'(e, f)$, refers to a node not on the face $f'(e, f)$ (i.e., $k' = f'(e, f)$), then again the inner product should be zero. Lastly, if the nodes k and k' refer to the same physical node (though they may *not* be the same index, as e and e' will generally have different labeling), then the inner product should double, as it does above. Let's say we have some node, \mathbf{r}_k^e . The element neighboring e on f is $e'(e, f)$. What is the index k' such that $\mathbf{r}_k^e = \mathbf{r}_{k'}^{e'(e, f)}$? Whatever it may be, we define it as $k'(e, f, k)$. Intuitively, if you choose an element, a face, and a node within that face, you will be able to find the index corresponding to the same node in the element sharing that face in particular. Again, construction is left to implementation. Now, if $k' = k'(e, f, k)$, then in fact we are referring to transport between the same physical node across two elements, and the inner product above doubles. We finally have:

$$I_{2\downarrow, kk'}^{ef} = \begin{cases} I_{2\uparrow, kk'}^{ef}, & (e, f) \text{ on boundary} \\ \frac{1}{12} A_f^e (1 - \delta_{kf}) [1 - \delta_{k'f'(e, f)}] [1 + \delta_{k'k'(e, f, k)}], & \text{else} \end{cases} \quad (2.2.74)$$

These constructs all require a deep knowledge of one's mesh, which typically can only be practically obtained if either the mesh is extremely simple and constructed with a high degree

of symmetry, or if one can devise efficient algorithms which search the typically-large array containing the mesh for the correct connections.

We lastly treat $\mathbf{I}_{3,kk'}^e$. If we were dealing with higher-order elements, we would once again immediately use a coordinate transformation. With our elements, we can put that off. First, look at ∇u_k^e . If $u_k^e = (1/6V^e)(a_k^e + \mathbf{b}_k^e \cdot \mathbf{r})$, then easily $\nabla u_k^e = (1/6V^e)\mathbf{b}_k^e$, which can be removed from integration as it does not involve \mathbf{r} . Now we are treating:

$$\mathbf{I}_{3,kk'}^e = \frac{1}{6V^e} \mathbf{b}_k^e \int_{V^e} d^3\mathbf{r} u_{k'}^e(\mathbf{r}) \quad (2.2.75)$$

Since the integral on the RHS is linear in \mathbf{r} , and the volume integral of \mathbf{r} is proportional to the center of gravity of the given integration volume (plus another constant), the above integral can be performed in a tetrahedron without a transformation. The result is that the integral is simply $V^e/4$, for any k' . Thus:

$$\boxed{\mathbf{I}_{3,kk'}^e = \frac{1}{24} \mathbf{b}_k^e} \quad (2.2.76)$$

2.3 Angular discretization – S_N

Up to now, we've simplified the angular integrals in our problem, and discretized space in a manner very intricately related to angular discretization. The first angular discretization method we look at is **discrete ordinates**, or S_N [3, 28, 29, 31, 32], which refers to discretizing $\hat{\mathbf{k}}$ simply by only considering our functions evaluated at discrete values of $\hat{\mathbf{k}}$. This could seem unsophisticated, but you'll notice that in our spatial discretization, we littered $\hat{\mathbf{k}}$ in many places where it would be perfectly convenient to choose a specific value of $\hat{\mathbf{k}}$ to plug in. That is, the transport operator doesn't mix angular coordinates. On the other hand though, the scattering operator does mix angular coordinates. But, if we pair discrete ordinates with **quadrature** [3, 32], we can treat both the transport and scattering operators at specific values of $\hat{\mathbf{k}}$. Quadrature is the use of knowledge of a function only at certain points in order to evaluate an integral. In other words, we evaluate an integral as a weighted sum of our function evaluated at particular points (note that Riemann sums are a special case of quadrature). If we can accurately employ quadrature, then we can treat our integrals, and that fully justifies the use of discrete ordinates.

2.3.1 Angular quadrature

Typically, quadrature goes as:

$$\int_{-1}^1 dx f(x) = \int_{-1}^1 dx w(x)g(x) \approx \sum_{i=1}^n w_{ni}g(x_{ni}) \quad (2.3.1)$$

where we recognize that a coordinate transformation can phrase any definite integral with finite bounds as one over $x \in [-1, 1]$. The function $w(x)$ is one which represents a particular form of quadrature, then $g(x) \equiv f(x)/w(x)$, while n is the quadrature order, w_{ni} is the i th

quadrature weight of the given form of quadrature of the n th order, and x_{ni} is the i th abscissa of the n th order quadrature. Now, the use of $g(x) = f(x)/w(x)$ may appear unintuitive, but it is a matter of great importance, an investigation of which will give us good quadrature sets for our problem. The various quadrature schemes rely on the *exact* integration of some set of basis polynomials up to some order depending on n (but not necessarily n). After choosing some basis set, one can determine the weights and abscissae by enforcing its exact integration. Now, polynomial bases are always defined by some weight function which only defines the inner product with which the basis is constructed [33]. A quadrature scheme exactly integrates a polynomial $g(x)$ up to some order, paired with a weight function $w(x)$, and the weight function does not need to be a polynomial, nor anything nearly resembling one. If one finds themselves with a particular $f(x)$ which can be factored into a weight function $w(x)$ and another function $g(x)$ which can exactly or closely be represented by a polynomial to the order exactly integrated by n th order quadrature, quadrature can be an incredible convenience.

To get a better understanding of this, let's look at Gauss-Legendre and Gauss-Chebyshev quadrature, both of which we will see are quite well suited to our problem. In Gauss-Legendre quadrature, we take $w(x) = 1$. This weight function suffices to derive all Legendre polynomials as orthogonal polynomials over $x \in [-1, 1]$ normalized to $P_n(1) = 1$. It could then be determined that:

$$w_{ni} = \frac{2}{(1 - x_{ni}^2) [P'_n(x_{ni})]^2} \quad (2.3.2)$$

Where x_{ni} is the i th root of the Legendre polynomial $P_n(x)$, thus given by:

$$P_n(x_{ni}) = 0 \quad (2.3.3)$$

Note that because of the above equation, $|x_{ni}| \neq 1$. If for whatever reason one is determined to use ± 1 as abscissae, they should resort to an adjusted form of the Gauss-Legendre quadrature scheme, known as Gauss-Lobatto quadrature [32]. The functions $f(x)$ which can be well treated by Gauss-Legendre quadrature are those which can be approximated by a polynomial of order $2n - 1$ (though Gauss-Lobatto quadrature sacrifices an extra two orders). In Gauss-Chebyshev quadrature, we take $w(x) = 1/\sqrt{1 - x^2}$. The corresponding polynomials are the somewhat less well-known Chebyshev polynomials of the first kind (interestingly, they are given by a polynomial in $\cos x$, $T_n(\cos x) = \cos(nx)$). Again, it could be determined that:

$$w_{ni} = \frac{\pi}{n} \quad (2.3.4)$$

$$x_{ni} = \cos\left(\frac{2i - 1}{2n}\pi\right) \quad (2.3.5)$$

This is a scheme which integrates exactly those $g(x)$ which involve Chebyshev polynomials up to order $n - 1$. But why would such a strange $w(x)$ ever be chosen? Consider a function $g(\phi)$ on a unit circle being integrated over a half-circle arc, $\phi \in [0, \pi]$. In order to transform this into an integral over -1 to 1 , we can use the variable substitution $\nu = \cos \phi$ (an intuitive substitution in its own right). The transformation yields:

$$d\phi = -\frac{d\nu}{\sin \phi} = -\frac{1}{\sqrt{1 - \nu^2}} d\nu \quad (2.3.6)$$

$$\int_0^\pi d\phi \rightarrow \int_{-1}^1 d\nu \frac{1}{\sqrt{1-\nu^2}} \quad (2.3.7)$$

So our integral over $g(\phi)$ becomes:

$$\int_0^\pi d\phi g(\phi) = \int_{-1}^1 d\nu \frac{1}{\sqrt{1-\nu^2}} g[\phi(\nu)] \quad (2.3.8)$$

Being able to distinguish $w(x) = 1/\sqrt{1-\nu^2}$ as a weight function is then natural and convenient, because this kind of quadrature yields:

$$\int_{-1}^1 d\nu \frac{1}{\sqrt{1-\nu^2}} g[\phi(\nu)] \approx \frac{\pi}{n} \sum_{i=1}^n g[\phi(\nu_{ni})] \quad (2.3.9)$$

The basics of quadrature have been outlined, and now we discuss the application of quadrature to our problem. We only intend to work with integrals of the form:

$$\int d\Omega f(\theta, \phi) = \int_{-1}^1 d\mu \int_0^{2\pi} d\phi f[\theta(\mu), \phi] \quad (2.3.10)$$

Where $\mu = \cos \theta$ is the angle cosine of the polar angle θ . An integral over μ can be treated by Gauss-Legendre quadrature quite straightforwardly:

$$\int_{-1}^1 d\mu f(\mu, \phi) \approx \sum_{i=1}^{N_\mu} w_i f(\mu_i, \phi) \quad (2.3.11)$$

Where N_μ is the number of polar angles we choose, as determined by the quadrature set μ_i . But what about the integral over ϕ ? We have an issue with immediately picking a quadrature scheme because ϕ runs over the whole unit circle, while $\nu = \cos \phi$ is only one-to-one across half of the unit circle. That is to say, if we always define $\nu = \cos \phi$, we can not always say $\phi = \arccos \nu$, because ϕ and $2\pi - \phi$ will return the same ν despite being distinct on the unit sphere. We can, however, begin to treat this problem by splitting our ϕ integral into two pieces. That is to say, we look at:

$$\int_0^{2\pi} d\phi g(\phi) = \int_0^\pi d\phi g(\phi) + \int_\pi^{2\pi} d\phi g(\phi) \quad (2.3.12)$$

Then, each integral can be treated separately. Let's agree that, always, $\nu = \cos \phi$. However, we map from ϕ to ν uniquely in the first integral simply with $\phi = \arccos \nu$. This implies a Gauss-Chebyshev quadrature scheme identical to that shown in (2.3.9). In the second integral, we can map from ϕ to ν uniquely with $\phi = 2\pi - \arccos \nu$. This yields something different:

$$\int_\pi^{2\pi} d\phi g(\phi) = \int_{-1}^1 d\nu \frac{1}{\sqrt{1-\nu^2}} g(2\pi - \arccos \nu) \quad (2.3.13)$$

Thus, our quadrature scheme utilizes:

$$\int_0^{2\pi} d\phi g(\phi) = \int_{-1}^1 d\nu \frac{1}{\sqrt{1-\nu^2}} [g(\arccos \nu) + g(2\pi - \arccos \nu)] \quad (2.3.14)$$

Now, via Fourier analysis we know we can always expand $g(\phi)$, at the very least, as a weighted sum of $\cos(n\phi)$ and $\sin(n\phi)$. Supposing that all coefficients of the $\sin(n\phi)$ terms are zero, then $g(\phi)$ will satisfy $g(x) = g(2\pi - x)$, and the two terms in the above integral can be merged. This, in the end, would result in the ability to use fewer angular nodes while still obtaining the same amount of information in the solution. Unfortunately, we can not make this simplification. I will note, however, that [3] seems to make this assumption, while arriving at the end to the same result as we will arrive, at least symbolically. The difference is that we will eventually state that we need discrete ordinates over the whole azimuthal interval $[0, 2\pi)$. Nevertheless, I have not found this assumption to be true, and if I perform calculations with it, I obtain asymmetric fluences in symmetric geometries. So, instead, we perform the integral shown above with separate quadrature nodes:

$$\int_0^{2\pi} d\phi g(\phi) = \frac{2\pi}{N_\phi} \sum_{i=1}^{N_\phi/2} g(\phi_i) + \frac{2\pi}{N_\phi} \sum_{i=1}^{N_\phi/2} g(\phi_{i+N_\phi/2}) \quad (2.3.15)$$

where N_ϕ is necessarily even, otherwise we are giving preference to either the interval $[0, \pi]$ or $[\pi, 2\pi]$. Supposing we shift the index of the second sum, as $i \rightarrow i - N_\phi/2$, so that the lower limit becomes $N_\phi/2 + 1$ and the upper limit becomes N_ϕ , we can just say:

$$\int_0^{2\pi} d\phi g(\phi) = \frac{2\pi}{N_\phi} \sum_{i=1}^{N_\phi} g(\phi_i) \quad (2.3.16)$$

Finally, the integration scheme is:

$$\int_{4\pi} d\Omega f(\hat{\mathbf{k}}) \approx \frac{2\pi}{N_\phi} \sum_{i=1}^{N_\mu} \sum_{j=1}^{N_\phi} w_i f(\hat{\mathbf{k}}_{ij}) \quad (2.3.17)$$

In general, the index i will correspond to the polar angle cosine and the index j will correspond to the azimuthal angle cosine.

We finally apply the scheme of discrete ordinates and angular quadrature to the BTE, beginning with the scattering term:

$$(\hat{K}\psi)_{ij} = \frac{2\pi}{N_\nu} \sum_{i'=1}^{N_\mu} \sum_{j'=1}^{N_\nu} \sum_{\ell=0}^L \sum_{m=-\ell}^{\ell} \frac{4\pi}{2\ell+1} y_{\ell ij}^m y_{\ell i' j'}^m w_{i'} \int dE' \Sigma_{s,\ell}(\mathbf{r}, E', E) \psi_{i' j'}(\mathbf{r}, E') \quad (2.3.18)$$

If we are using the first-collision scattering source, there is no quadrature anyway:

$$\tilde{s}_{ij} = \sum_{\ell=0}^L \sum_{m=-\ell}^{\ell} \frac{4\pi}{2\ell+1} y_{\ell ij}^m y_{\ell}^m [\hat{\mathbf{k}}_0(\mathbf{r})] \int dE' \Sigma_{s,\ell}(\mathbf{r}, E', E) \tilde{\varphi}^0(\mathbf{r}, E') \quad (2.3.19)$$

The terms above are easily coupled with the spatial FEM. In $(\hat{K}\psi)_{ij}$, no matrix multiplication occurs in e nor k . Instead, Σ_s is given a superscript e , which just says that a given element has its own cross sections corresponding to the material in that element, all \mathbf{r} –dependent quantities are evaluated at \mathbf{r}_k^e .

Before discretizing the transport operator, we will attempt to simplify the sums of the form:

$$S_m(\hat{\mathbf{k}}, \hat{\mathbf{k}}') \equiv \sum_{m=-\ell}^{\ell} y_{\ell}^m(\hat{\mathbf{k}}) y_{\ell}^{m*}(\hat{\mathbf{k}}') \quad (2.3.20)$$

The justification for this is not necessarily intuitive: it is related to our preference to use matrix-less multiplication. That is, it so happens that forming the scattering terms as in (2.3.18) and (2.3.19) is computationally not as efficient as forming the scattering terms in a manner that breaks the co-dependence of the indices (ℓ, m, i, j) . Now, whether we are using spherical harmonics or real spherical harmonics, we find:

$$\sum_{m=-\ell}^{\ell} y_{\ell}^m(\hat{\mathbf{k}}) y_{\ell}^{m*}(\hat{\mathbf{k}}') = \frac{2\ell+1}{4\pi} \sum_{m=0}^{\ell} \frac{(\ell-m)!}{(\ell+m)!} P_{\ell}^m(\mu) P_{\ell}^m(\mu') c^m(\phi - \phi') \quad (2.3.21)$$

Where:

$$c^m(\phi - \phi') \equiv 2 \cos[m(\phi - \phi')] - \delta_{m,0} \quad (2.3.22)$$

This all allows us to write:

$$(\hat{K}\psi)_{ij} = \frac{2\pi}{N_{\nu}} \sum_{i'=1}^{N_{\mu}} \sum_{j'=1}^{N_{\nu}} \sum_{\ell=0}^L \sum_{m=0}^{\ell} \frac{(\ell-m)!}{(\ell+m)!} P_{\ell i}^m P_{\ell i'}^m c_{jj'}^m w_{i'} \int dE' \Sigma_{s,\ell}(\mathbf{r}, E', E) \psi_{i'j'}(\mathbf{r}, E') \quad (2.3.23)$$

as well as:

$$\tilde{s}_{ij} = \sum_{\ell=0}^L \sum_{m=0}^{\ell} \frac{(\ell-m)!}{(\ell+m)!} P_{\ell i}^m P_{\ell}^m [\hat{\mathbf{k}}_0(\mathbf{r}) \cdot \hat{\mathbf{z}}] c_j^m [\hat{\mathbf{k}}_0(\mathbf{r})] \int dE' \Sigma_{s,\ell}(\mathbf{r}, E', E) \tilde{\varphi}^0(\mathbf{r}, E') \quad (2.3.24)$$

where:

$$c_j^m [\hat{\mathbf{k}}_0(\hat{\mathbf{r}})] \equiv 2 \cos[m(\phi_j - \phi_{\hat{\mathbf{k}}_0(\mathbf{r})})] - \delta_{m,0} \quad (2.3.25)$$

2.3.2 Sweep

As for the transport matrix, coupling with the FEM deserves quite a bit of attention, as it introduces a peculiar structure to our spatial block matrices, which is to be taken into account if one wishes to invert anything involving \hat{T} . I will tell you now that, in all solution methods which are discretized by S_N , we will want to know how to invert \hat{T} alone. Now, return to (2.2.49). The terms G , F_{\uparrow} , and F_{\downarrow} remain to be understood. Easily, G is treated by introducing discrete ordinates:

$$G_{ijkk'}^e = -\hat{\mathbf{k}}_{ij} \cdot \mathbf{I}_{3',kk'}^e \quad (2.3.26)$$

But the F terms involve the quantities ς_{\uparrow} and ς_{\downarrow} . When we plug in $\hat{\mathbf{k}}_{ij}$, we will find the terms:

$$\begin{aligned} \varsigma_{\uparrow,ij}^{ef} &= \frac{1}{2} (\hat{\mathbf{k}}_{ij} \cdot \hat{\mathbf{n}}_f^e + |\hat{\mathbf{k}}_{ij} \cdot \hat{\mathbf{n}}_f^e|) \\ \varsigma_{\downarrow,ij}^{ef} &= \frac{1}{2} (\hat{\mathbf{k}}_{ij} \cdot \hat{\mathbf{n}}_f^e - |\hat{\mathbf{k}}_{ij} \cdot \hat{\mathbf{n}}_f^e|) \end{aligned} \quad (2.3.27)$$

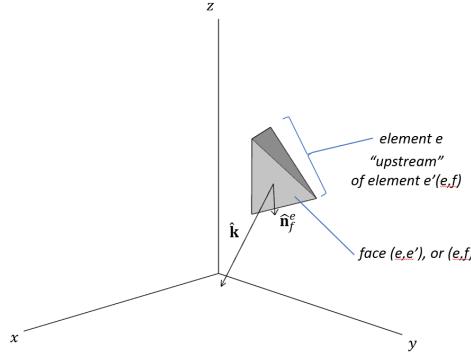


Figure 3: Depiction of the definition of an ‘upstream’ element, whose neighbor on the given face is then the ‘downstream’ element, for a chosen angle $\hat{\mathbf{k}}$.

$$(G + M + F_\uparrow)\psi + F_\downarrow\psi + F_\downarrow\bar{\psi} = s$$

Figure 4: The problem of $\hat{T}\psi = s$ over all angles simultaneously. Each matrix element is actually a 4×4 matrix that is continuously parametrized by $\hat{\mathbf{k}}$. Notably, the two known constants are s and $F_\downarrow\bar{\psi}$

Importantly, $\zeta_{\downarrow,ij}^{ef}$ is zero for particular combinations of (i, j, e, f) . Specifically, those that describe a face which is pointing *along* the chosen direction $\hat{\mathbf{k}}$, i.e., one that points ‘downstream’ as depicted in **Figure 3**. This indicates that the element e is ‘upstream’ of element $e'(e, f)$, so element e should not receive any fluence from $e'(e, f)$. Think about how this affects the connectivity of our problem. Consider a modified form of the block-structure conceptualization of our spatial matrix problem. Recognize that, if we consider only the discretization of the transport operator \hat{T} , each (i, j) can now index one of these block-matrices. That is, we now have **Figure 4**. Interestingly, for a given (i, j) , F_\downarrow will be entirely zero if $\zeta_{\downarrow,ij}^{ef} = 0$, i.e., if the face on the given element is pointing downstream. What this means is that the solution to the given element is not *getting anything* from its downstream neighbor. Now, suppose we do in fact choose a specific angular coordinate, such that we can evaluate a-priori all ζ_{\downarrow}^{ef} for this coordinate and kill off terms in the F_\downarrow matrix. Intuitively, we will be able to find elements e on the surface of the mesh for which all upstream faces are considered boundary faces, and thus accounted for in the boundary source term $F_\downarrow\bar{\psi}$. That leaves $F_{\downarrow}^{ee'} = 0$ for all e' . Thus, the row e would be soluble. Then, the solution ψ^e would be used to form terms like $F_{\downarrow}^{e'e}\psi^e$ for all e' , allowing for one to solve for element e' if *all* of the elements

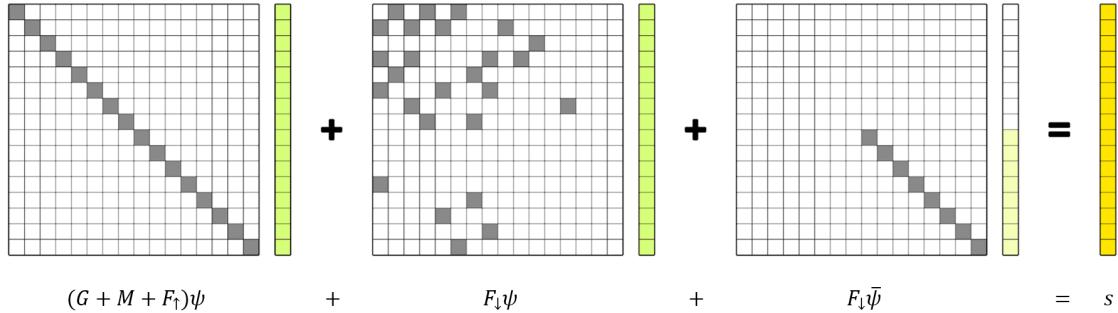


Figure 5: The problem of $\hat{T}\psi = s$ for a particular angle $\hat{\mathbf{k}}$. The choice of angle kills several elements of F_\downarrow , specifically, those which would otherwise describe flow of particles *against* $\hat{\mathbf{k}}$. This problem is soluble by determining rows e for which the whole matrix problem reduces to a single block-matrix problem, such as $(G^e + M^e + F_\uparrow^e)\psi^e = s^e - (F_\downarrow\psi)^e$. Then, this solution ψ^e can be supplied to solve for rows e' for which the matrix problem reduces to something like $(G^{e'} + M^{e'} + F_\uparrow^{e'})\psi^{e'} = s^{e'} - (F_\downarrow\bar{\psi})^{e'} - F_\downarrow^{e'e}\psi^e$, where we here recognize that ψ^e is now known.

$F_\downarrow^{e'e''}\psi^{e''}$ are known. More generally, the block will look something like **Figure 5**, where F_\downarrow is no longer symmetric. This is interesting, because it provides a procedure to invert \hat{T} . Find a row with only the diagonal element. This corresponds to elements on the boundary which also have no upstream neighboring faces. No such rows existed in the general block matrix we discovered earlier, because all boundary elements shared at least one face with another element. But now that we've specified $\hat{\mathbf{k}}$, we're able to eliminate off-diagonal terms corresponding to downstream faces, so there exist some boundary elements with nothing but the diagonal element. Now, if we write the matrix equation corresponding to these elements, we are actually able to directly produce a solution by simply inverting the $N_K^e \times N_K^e$ matrix described by the diagonal block. That is, we have a problem of the form:

$$\sum_{k'=1}^{N_K^e} T_{kk'}^e x_{k'}^e = b_k^e \quad (2.3.28)$$

where x^e and b^e are just some column vectors, and where e describes an element on the boundary and $T_{kk'}^e$ is the transport matrix we defined in (2.2.50). Note that this simplified form implicitly includes the boundary condition term within b_k^e . Now, we can quite easily perform this inversion, in fact, for a tetrahedron for which $T_{kk'}^e$ is just 4×4 , this inversion can be performed explicitly. Then, we'd be able to solve for the full element through:

$$x_k^e = \sum_{k'=1}^{N_K^e} (T^{-1})_{kk'}^e b_{k'}^e \quad (2.3.29)$$

We reiterate the important part of this equation: the downstream neighbors of element e actually *depend* on x^e . Supposing we solve for every element that had only a diagonal element, we can now search our system matrix for elements which have only their diagonal element, *and off-diagonal elements corresponding to already solved elements*. That is, suppose

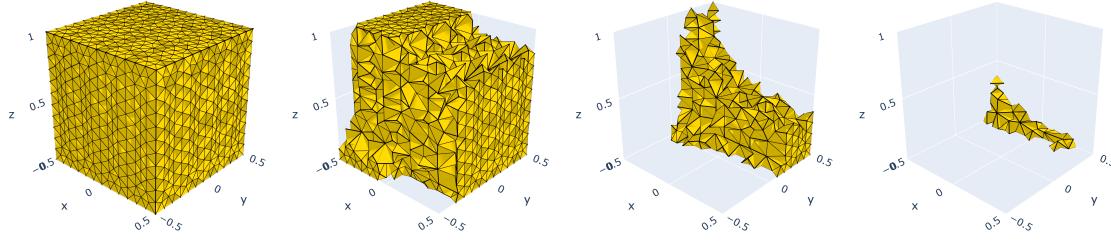


Figure 6: 3D tetrahedral mesh of a cube being swept for a given discrete ordinate roughly corresponding to the camera angle. From left to right, elements are being removed as they become soluble. Mesh was created with the Gmsh software [34].

that in our system matrix, we find x^1 and x^2 as elements that can be solved immediately as above. Then, we find some x^3 , which has off-diagonal elements such that:

$$\sum_{k'=1}^{N_K^e} F_{\downarrow, kk'}^1 x_{k'}^1 + \sum_{k'=1}^{N_K^e} F_{\downarrow, kk'}^2 x_{k'}^2 + \sum_{k'=1}^{N_K^e} T_{kk'}^3 x_{k'}^3 = b_k^e \quad (2.3.30)$$

The LHS is just the full matrix multiplication, $\sum_{e' k'} T_{kk'}^{ee'} x_{k'}^{e'}$, except with faces that are *downstream* of $e = 3$ removed, while the faces that are *upstream* of $e = 3$ are $(e, e') = (3, 1)$ and $(3, 2)$. But, we *already know* x^1 and x^2 . So, in order to solve for x^3 , we can just use:

$$x_k^3 = \sum_{k'=1}^{N_K^e} (T^{-1})_{kk'}^3 \left(b_{k'}^3 - \sum_{k''=1}^{N_K^e} F_{\downarrow, k' k''}^1 x_{k''}^1 - \sum_{k''=1}^{N_K^e} F_{\downarrow, k' k''}^2 x_{k''}^2 \right) \quad (2.3.31)$$

Essentially, we have updated the source and applied the inverse of the diagonal element. Then, we can repeat: we search for elements that have their diagonal and off-diagonal terms that use already-solved elements. We repeat this until we have solved for all x^e . This procedure is referred to as a **sweep**. Determination of a sweep order for any given mesh and $\hat{\mathbf{k}}$ is not necessarily trivial. However, details on how it will be constructed are in the **wiscobolt implementation** document. A brief depiction of how the sweep may act on a three dimensional mesh, in a sweep order determined by wiscobolt, is shown in **Figure 6**.

2.3.3 Boundary conditions in the sweep

A brief note about boundary conditions and singular sources is here warranted. Now, it should be relatively well understood how boundary conditions feature in our problem; in **Figure 4**, we had depicted the boundary condition vector being multiplied by a diagonal F_{\downarrow} matrix. That is, for boundary elements we must multiply boundary conditions by a particular matrix prior to inversion of the diagonal part of the system matrix. Note that since we are using the *discontinuous* finite element method, we do not necessarily impose boundary conditions as *the* solution to the (global) boundary nodes themselves, rather, the (local) nodes within a boundary element treat their neighbors as being determined by boundary conditions.

It is worth discussing what to take as s_k^e and what to take as $\psi^{e'=0}$ (the boundary condition), in a given context. Since the sweep occurs only when one is explicitly inverting \hat{T} , which is obviously very distinct from inverting \hat{L} , we will discuss everything in the context of BTEs with no scattering part, so that we may get a sense of the general prescription when given particular sources and boundary conditions.

Consider first the generic case, $\hat{T}f = s$, where s is a singular, external beam source. Our boundary condition is \bar{f} and it is given by tracing the beam source to the boundary of the problem, as we used to construct $\tilde{\psi}^0$ in the FCS method. Now, s is singular, and is nonzero only outside of the problem volume. So, what do we take as s_k^e in this case? Actually, we *do* take $s_k^e = 0$. But this problem remains distinct from the trivial problem $\hat{T}f = 0$. That is because the *boundary conditions* act as the source for the first sweep. That is to say, the sweep will begin at the boundary elements, where although $s_k^e = 0$, the boundary condition term given by:

$$s_{\partial,ijk}^e \equiv - \sum_{k'=1}^{N_K^e} \sum_{f=1}^{N_F^e} F_{\downarrow,ijkk'}^{ef} \bar{f}_{ijk'}^{e'(e,f)} \quad (2.3.32)$$

will be nonzero in general. Therefore, the solution to the boundary elements will itself be nonzero, and it will propagate downstream as per the sweep.

On the other hand, in the FCS problem, we would deal with essentially $\hat{T}\tilde{f} = \tilde{s}$, where $\tilde{s} = \hat{K}\tilde{\psi}^0$. This source is, unlike s alone, a singular, and not generally zero everywhere inside the problem volume. Thus, it will be able to be used as \tilde{s}_k^e . But what is the boundary condition? We can reason that the boundary condition is actually zero. In the generic BTE, we formulated the boundary condition as being determined from $\hat{T}\tilde{\psi}^0 = s$ on the basis that particles which have yet to enter the volume are not able to be scattered, and so $\hat{K} = 0$ (in principle, $\Sigma_t = 0$ too, though it is not assumed in forming $\tilde{\psi}^0$, yet the above is in fact equivalent to the solution of $\hat{T}f = s$ with $\Sigma_t = 0$ on the boundary). In formulating the FCS BTE, we reasoned that the boundary condition to $\tilde{\psi}$ was zero, because we had a solution of the form $\tilde{\psi} + \tilde{\psi}^0$, and $\tilde{\psi}^0$ fully satisfies the boundary condition, so $\tilde{\psi}$ must be zero on the boundary.

The more general rationale to determining a boundary condition will be important to us to treat particle-particle scattering sources, as well as in **Section 3.3.3**, when we discuss a tweak to the FCS method that involves re-defining the ‘uncollided’ particles. Therefore, we discuss it here. Suppose our problem geometry was expanded normally by some small increment, δ . In this new shell of volume, we will say that our attenuation coefficients are zero, i.e., $\Sigma_s = 0$ and $\Sigma_t = 0$, or more generally, $\hat{K} = 0$. Therefore, this problem is the same as the original problem. Now, in the original problem, the boundary elements require the solution of the BTE to the upstream elements. In the new problem, the original problem’s boundary elements (which are no longer boundary elements) require the solution of the BTE to the new boundary elements, which are necessarily zero by virtue of $\hat{K} = 0$. Thus, in order to determine boundary conditions for the old problem, we can simply take evaluate the solution on the boundary with $\hat{K} = 0$. This is fully consistent with the FCS problem’s boundary conditions, wherein $\tilde{s}(\mathbf{r} \in \partial V) = 0$, as well as the external beam problem, wherein $\hat{T}\psi = s$ on the boundary. It also teaches us that for a particle-particle scattering source,

where the outgoing particle is not sourced by an external beam, we must take boundary conditions of zero.

Thus, we can identify a source and a boundary condition in any problem. In general, the approach is that we can treat every sweep problem as one that involves the total source $s_k^e + s_{\partial,k}^e$, with $s_{\partial,k}^e$ given by generalization of (2.3.32). Notably, at most, $s_{\partial,k}^e$ is populated *only* for elements on the boundary of the volume, generally for all angles, via the inaccuracy introduced by discretization of the singularity in $\bar{\psi} = \bar{\psi}^0$. This unified treatment allows us to perform the sweep operation with a given source vector to produce a given product vector, regardless of whether we are using the FCS, or whatever boundary conditions we have.

2.3.4 Summary

We have formulated the simultaneous space-angle BTE as:

$$\boxed{\sum_{k'=1}^{N_K^e} \left(G_{ijkk'}^e + M_{kk'}^e + F_{\uparrow,ijkk'}^e \right) \psi_{ijk'}^e = s_{ijk}^e - \sum_{k'=1}^{N_K^e} \sum_{f=1}^{N_F^e} F_{\downarrow,ijkk'}^e \psi_{ijk'}^{e'(e,f)} + \sum_{i'=1}^{N_\mu} \sum_{j'=1}^{N_\phi} K_{ii'jj'}^e \psi_{i'j'k}^e} \quad (2.3.33)$$

Where:

$$\boxed{\begin{aligned} G_{ijkk'}^e &= -\hat{\mathbf{k}}_{ij} \cdot \mathbf{I}_{3',kk'}^e \\ M_{kk'}^e &= \Sigma_t^e \delta_{kk'} \\ F_{\uparrow,ijkk'}^e &= \sum_{f=1}^{N_F^e} \varsigma_{\uparrow,ij}^{ef} I_{2'\uparrow,kk'}^{ef} \\ F_{\downarrow,ijkk'}^e &= \varsigma_{\downarrow,ij}^{ef} I_{2'\downarrow,kk'}^{ef} \\ K_{ii'jj'}^e &= \frac{2\pi}{N_\nu} \sum_{i'=1}^{N_\mu} \sum_{j'=1}^{N_\nu} \sum_{\ell=0}^L \sum_{m=0}^{\ell} \frac{(\ell-m)!}{(\ell+m)!} P_{\ell i}^m P_{\ell i'}^m c_{jj'}^m w_{i'} \int dE' \Sigma_{s,\ell}^e(E', E) \end{aligned}} \quad (2.3.34)$$

We may also define a ‘sweep source’:

$$\boxed{s_{F,ijk}^e \equiv s_{ijk}^e - \sum_{k'=1}^{N_K^e} \sum_{f=1}^{N_F^e} F_{\downarrow,ijkk'}^e \psi_{ijk'}^{e'(e,f)}} \quad (2.3.35)$$

We also know that we can use the first collision source method with $\psi \rightarrow \tilde{\psi}$ and $s \rightarrow \tilde{s}$, where:

$$\boxed{\tilde{s}_{ijk}^e = \sum_{\ell=0}^L \sum_{m=0}^{\ell} \frac{(\ell-m)!}{(\ell+m)!} P_{\ell i}^m P_{\ell}^m [\hat{\mathbf{k}}_0(\mathbf{r}_k^e) \cdot \hat{\mathbf{z}}] c_j^m [\hat{\mathbf{k}}_0(\mathbf{r}_k^e)] \int dE' \Sigma_{s,\ell}^e(E', E) \tilde{\varphi}_k^{0,e}(E')} \quad (2.3.36)$$

2.4 Angular discretization – P_N

Now, let’s talk about a P_N solution. The manner by which one turns $\hat{L}\psi = s$ into a matrix problem in (ℓ, ℓ') and (m, m') is not immediately clear following series expansions. In the

series expansions section, we only discussed the terms involving \hat{K} . We did not try to expand, say, $\hat{\mathbf{k}} \cdot \nabla \psi$. The reason for this is that we *shouldn't* just expand ψ like ψ_ℓ^m here. We would need to treat the $\hat{\mathbf{k}} \cdot \nabla$ operator as well. One option is to look at the streaming operator with respect to the current density:

$$\hat{\mathbf{k}} \cdot \nabla \psi = \nabla \cdot \mathbf{j} \quad (2.4.1)$$

And then you could try to make meaning of \mathbf{j}_ℓ^m in isolation. You can attempt to relate these moments to ψ_ℓ^m , but will find difficulty unless you assume a specific form of \mathbf{j} , such as one that approximates streaming by some other mechanism [9]. For instance, diffusion [3, 9, 35–37]. Diffusion suggests:

$$\mathbf{j} = -C(\mathbf{r}, E) \nabla \psi \quad (2.4.2)$$

Where $C(\mathbf{r}, E)$ is the diffusion coefficient. This is the average path length between collisions divided by the dimensionality of the space, so in our case it would be given by $1/3\Sigma_t$. Using this, we would be suggesting that particles do not ‘stream,’ i.e., that particles with direction $\hat{\mathbf{k}}$ do not travel in direction $\hat{\mathbf{k}}$ (which can actually be a pretty good approximation in some cases [9]). Now, this would permit an extraordinarily clean P_N discretization, because it wouldn't be a matrix problem in (ℓ, ℓ') and (m, m') , it would actually be a vector problem in ℓ and m with no mixing between elements. However, in most cases, this approximation is too inaccurate. That said, we will find a utility for diffusion in **Section 3.2**.

Having seen that through, we can construct a neat P_N equation by reformulating the weak form of the BTE, but with a composite space-angle test function $w(\mathbf{r}, \hat{\mathbf{k}}) \equiv v(\mathbf{r}) y_\ell^m(\hat{\mathbf{k}})$, with y_ℓ^m being the **real spherical harmonic** of moment m and order ℓ . The purpose of using real spherical harmonics here is to avoid complex values in our soon-to-be-formulated matrix problem. Since real spherical harmonics span the entire space of test functions, we can form a perfectly viable weak form provided we enforce it for all (ℓ, m) (and then we will truncate it to a finite set as an approximation). We discussed what conventions and identities exist for our real spherical harmonics y_ℓ^m in **Section 2.1**. However, we will introduce one more convention: we will label our real spherical harmonics *not* with (ℓ, m) , but rather with a new, singular index: q . The map to q is:

$$q = \ell(\ell + 1) + m + 1 \quad (2.4.3)$$

And thus, q can uniquely map to ℓ and m , so long as we agree that $-\ell \leq m \leq \ell$, $\ell \geq 0$, and both ℓ and m are integers (verify, if you like, that this makes sense, starting from $(0, 0)$, then on to $(1, -1)$, $(1, 0)$, $(1, 1)$, and so on). The typical sums we encounter like $\sum_{\ell' m'} M_{\ell' m'} v_{\ell'}^{m'}$ now become matrix multiplication like $\sum_{q'} M_{qq'} v_{q'}$. For ℓ that is truncated to L , q becomes truncated to:

$$Q = (L + 1)^2 \quad (2.4.4)$$

which emphasizes that storage of matrices like $M_{qq'}$ will scale as L^4 . Additionally, it is useful to know how to map from q to ℓ and m . There is a rather general method to deducing these relationships, but I will simply provide them:

$$\ell(q) = \lceil \sqrt{q} \rceil - 1 \quad (2.4.5)$$

$$m(q) = q - \ell(q)[\ell(q) + 1] - 1 \quad (2.4.6)$$

Now, pickup with the discretized spatial weak-form BTE in (2.2.49). As a refresher:

$$\sum_{k'=1}^{N_K^e} \left(G_{kk'}^e + M_{kk'}^e + F_{\uparrow, kk'}^e \right) \psi_{k'}^e + \sum_{k'=1}^{N_K^e} \sum_{f=1}^{N_F^e} F_{\downarrow, kk'}^{ef} \psi_{k'}^{e'(e,f)} = s_k^e + \int dE' d\Omega' \Sigma_s^e \psi_k^e \quad (2.4.7)$$

Now, to get the weak form with the $w(\mathbf{r}, \hat{\mathbf{k}})$ we desire, we can start here and multiply through with y_ℓ^m then integrate. Unlike the FEM, there is no such thing as an angular ‘element,’ we are using the 2D basis set y_ℓ^m directly which, if $L \rightarrow \infty$, would formulate a perfectly true matrix equation. After multiplying and integrating, we expand ψ_k^e as:

$$\psi_k^e(\hat{\mathbf{k}}) = \sum_{q=1}^Q y_q(\hat{\mathbf{k}}) \psi_{kq}^e \quad (2.4.8)$$

where:

$$\psi_{kq}^e = \int d\Omega y_q(\hat{\mathbf{k}}) \psi_k^e(\hat{\mathbf{k}}) \quad (2.4.9)$$

And similarly for s_k^e . Additionally, let’s write out \hat{K} as we had it during **Section 2.1**, or:

$$\int dE' d\Omega' \Sigma_s^e \psi_k^e = \sum_{q=1}^Q \frac{4\pi}{2\ell(q) + 1} y_q(\hat{\mathbf{k}}) \int dE' \Sigma_{s,\ell(q)}^e \psi_{kq}^e \quad (2.4.10)$$

Now we’ll go term-by-term:

$$\int d\Omega y_q(\hat{\mathbf{k}}) G_{kk'}^e(\hat{\mathbf{k}}) \psi_{k'}^e(\hat{\mathbf{k}}) = - \sum_{q'} \psi_{k'q'}^e \left[\int d\Omega \hat{\mathbf{k}} y_q(\hat{\mathbf{k}}) y_{q'}(\hat{\mathbf{k}}) \right] \cdot \mathbf{I}_{3',kk'}^e \quad (2.4.11)$$

Define the ‘whole-sphere angular inner product’:

$$\int d\Omega \hat{\mathbf{k}} y_q(\hat{\mathbf{k}}) y_{q'}(\hat{\mathbf{k}}) \equiv \mathbf{A}_{qq'} \quad (2.4.12)$$

And the P_N form of the G term:

$$G_{kk'qq'}^e \equiv -\mathbf{A}_{qq'} \cdot \mathbf{I}_{3',kk'}^e \quad (2.4.13)$$

Next combine the $M_{kk'}^e$ term and the \hat{K} term (both of which are diagonal in spatial indices):

$$\int d\Omega y_q(\hat{\mathbf{k}}) (\Sigma_t^e - \hat{K}) \psi_k^e(\hat{\mathbf{k}}) = \left[\Sigma_t^e - \frac{4\pi}{2\ell(q) + 1} \int dE' \Sigma_{s,\ell(q)}^e \cdot \right] \psi_{kq}^e \quad (2.4.14)$$

Therefore, this entire term above is diagonal in both spatial and angular indices (the latter is a departure from what we may intuitively expect, and what we found in S_N , it is to say

that scattering does not mix angular moments of ψ). We can now define the P_N scattering and M terms:

$$M_{kk'qq'}^e \equiv \Sigma_t^e \delta_{kk'} \delta_{qq'} \quad (2.4.15)$$

$$K_q^e \equiv \frac{4\pi}{2\ell(q) + 1} \int dE' \Sigma_{s,\ell(q)}^e. \quad (2.4.16)$$

which is not written as a diagonal matrix, while M is, because \hat{K} is not generally grouped with terms in \hat{T} (this is a conceptual and computational convenience).

Now, the F_\uparrow and F_\downarrow terms are not trivial at all. We should, in an isolated setting, consider first:

$$\int d\Omega y_q(\hat{\mathbf{k}}) \zeta_\uparrow^{ef}(\hat{\mathbf{k}}) \psi_k^e(\hat{\mathbf{k}}) = \sum_{q'=1}^Q \psi_{kq}^e \int d\Omega \zeta_\uparrow^{ef}(\hat{\mathbf{k}}) y_q(\hat{\mathbf{k}}) y_{q'}(\hat{\mathbf{k}}) \quad (2.4.17)$$

So, what do we make of the integral on the RHS? Well, for the given face described by (e, f) , ζ_\uparrow^{ef} is zero for such values of $\hat{\mathbf{k}}$ that $\hat{\mathbf{k}} \cdot \hat{\mathbf{n}}_f^e \leq 0$. Therefore, ζ_\uparrow^{ef} can be considered to simply *clip* the angular integral over this region. Nevermind how difficult it will be to formulate these integrals (and it will be), we have:

$$\sum_{q'=1}^Q \psi_{kq}^e \int d\Omega \zeta_\uparrow^{ef}(\hat{\mathbf{k}}) y_q(\hat{\mathbf{k}}) y_{q'}(\hat{\mathbf{k}}) = \sum_{q'=1}^Q \psi_{kq}^e \int_{\hat{\mathbf{k}} \cdot \hat{\mathbf{n}}_f^e > 0} d\Omega \hat{\mathbf{k}} \cdot \hat{\mathbf{n}}_f^e y_q(\hat{\mathbf{k}}) y_{q'}(\hat{\mathbf{k}}) \quad (2.4.18)$$

We will be interested in the following angular inner products:

$$\mathbf{A}_{\uparrow,qq'}(\hat{\mathbf{n}}) \equiv \int_{\hat{\mathbf{k}} \cdot \hat{\mathbf{n}} > 0} d\Omega \hat{\mathbf{k}} y_q(\hat{\mathbf{k}}) y_{q'}(\hat{\mathbf{k}}) \quad (2.4.19)$$

And a similar thing can be done for the F_\downarrow term. In the end we'd end up defining:

$$\mathbf{A}_{\downarrow,qq'}(\hat{\mathbf{n}}) \equiv \int_{\hat{\mathbf{k}} \cdot \hat{\mathbf{n}} < 0} d\Omega \hat{\mathbf{k}} y_q(\hat{\mathbf{k}}) y_{q'}(\hat{\mathbf{k}}) \quad (2.4.20)$$

We can then define the P_N forms of the F_\uparrow and F_\downarrow terms:

$$F_{\uparrow,kk'qq'}^e = \sum_{f=1}^{N_F^e} \hat{\mathbf{n}}_f^e \cdot \mathbf{A}_{\uparrow,qq'}^{ef} I_{2'\uparrow,kk'}^{ef} \quad (2.4.21)$$

$$F_{\downarrow,kk'qq'}^e = \hat{\mathbf{n}}_f^e \cdot \mathbf{A}_{\downarrow,qq'}^{ef} I_{2'\downarrow,kk'}^{ef} \quad (2.4.22)$$

Now, the full matrix problem is:

$$\sum_{q'=1}^Q \sum_{k'=1}^{N_K^e} \left(G_{kk'qq'}^e + M_{kk'qq'}^e + F_{\uparrow,kk'qq'}^e \right) \psi_{k'q'}^e + \sum_{q'=1}^Q \sum_{k'=1}^{N_K^e} \sum_{f=1}^{N_F^e} F_{\downarrow,kk'qq'}^e \psi_{k'q'}^{e'(e,f)} = s_{kq}^e + K_q^e \psi_{kq}^e \quad (2.4.23)$$

This looks easy enough, in fact, as a whole, it may look even easier than S_N given that \hat{K} has such a simple form. It may seem like we can easily lump that in with the transport matrix and do a sweep like in S_N , as long as we can invert simultaneously the k and q –dependence. But there *is no sweep!* If we try to think of this as a block-matrix problem in e again, where each block is now a k and q matrix, we recognize that the elements will always talk to their neighbors, and there is no such notion of ‘upstream’ or ‘downstream,’ because they are mixed in the integrals $\mathbf{A}_{\uparrow,qq'}^{ef}$ and $\mathbf{A}_{\downarrow,qq'}^{ef}$. That is, for a given (e,f) , the angular inner products are not generally zero. In fact, the only relationship they can be said to satisfy is:

$$\mathbf{A}_{qq'} = \mathbf{A}_{\uparrow,qq'}^{ef} + \mathbf{A}_{\downarrow,qq'}^{ef} \quad (2.4.24)$$

Which is not actually zero for all (q,q') as we would want it to be in order to perform a sweep. So, in fact, we can not avoid treating the $P_N L$ operator as a matrix in (e,e') . This is a consideration which is tremendously important in the solution methods we will soon discuss.

The formulation of the inner products is rather involved. The calculations can actually be done explicitly, at least, in the case of linear elements (though they need not be tetrahedra). However, this is a very tedious process, and in my attempts at implementing the solution, I have found it to be slower than a sophisticated integral quadrature procedure. Still, the quadrature procedure is quite slow, and one may suspect that a more optimal implementation of the explicit method would be faster. Nevertheless, the description of both methods is reserved for **Appendix A**. Briefly, the angular integrals $\mathbf{A}_{qq'}$ are always performed explicitly, and they produce expansion coefficients of $\hat{\mathbf{k}}y_q$ in spherical harmonics. In other words, the angular function $k_i y_q$ is given by a linear combination of $y_{q'}$ with weight $A_{i,qq'}$. This is used to expand $\mathbf{A}_{\uparrow,qq'}$ and $\mathbf{A}_{\downarrow,qq'}$ and reduce everything to an integral of $y_q y_{q'}$ over one of the two halves of the unit sphere. Then, identities are used to reduce everything to an integral of $y_q y_{q'}$ over just one half of the unit sphere (depending still on $\hat{\mathbf{n}}$). Additionally, because $\hat{\mathbf{n}}^{ee'} = -\hat{\mathbf{n}}^{e'e}$, we reduce the total number of integrals we must perform for each of $\mathbf{A}_{\uparrow,qq'}$ and $\mathbf{A}_{\downarrow,qq'}$ to Q^2 multiplied by the number of ‘global faces’ N_F in the volume, not $N_E N_F^e$. A global face has an ambiguous normal vector up to a factor of (-1) , and is described by the pair (e,e') regardless of order, unlike a ‘local face.’ A global face, of course, can not be described by (e,f) . Nevertheless, *at this point*, integrals are performed either explicitly or via quadrature (and it is still slow).

2.4.1 Boundary conditions

As there is no sweep, boundary conditions are a bit more straightforward to be implemented in the P_N method. All we really need to do is write the boundary source term:

$$s_{\partial,kq}^e \equiv - \sum_{k'=1}^{N_K^e} \sum_{f=1}^{N_F^e} \sum_{q'=1}^Q F_{\downarrow,kk'qq'}^{ef} \bar{f}_{k'q'}^{e'(e,f)} \quad (2.4.25)$$

2.4.2 Summary

We have formulated the simultaneous space-angle BTE as:

$$\boxed{\sum_{k'=1}^{N_K^e} \sum_{q'=1}^Q \left(G_{kk'qq'}^e + M_{kk'qq'}^e + F_{\uparrow, kk'qq'}^e \right) \psi_{k'q'}^e = s_{qk}^e - \sum_{k'=1}^{N_K^e} \sum_{q'=1}^Q \sum_{f=1}^{N_F^e} F_{\downarrow, kk'qq'}^{ef} \psi_{k'q'}^{e'(e,f)} + \sum_{q'=1}^Q K_{qq'}^e \psi_{kq'}^e} \quad (2.4.26)$$

Where:

$$\boxed{\begin{aligned} G_{kk'qq'}^e &= -\mathbf{A}_{qq'} \cdot \mathbf{I}_{3',kk'}^e \\ M_{kk'qq'}^e &= \Sigma_t^e \delta_{kk'} \delta_{qq'} \\ F_{\uparrow, kk'qq'}^e &= \sum_{f=1}^{N_F^e} \hat{\mathbf{n}}_f^e \cdot \mathbf{A}_{\uparrow,qq'}^{ef} I_{2'\uparrow,kk'}^{ef} \\ F_{\downarrow, kk'qq'}^{ef} &= \hat{\mathbf{n}}_f^e \cdot \mathbf{A}_{\downarrow,qq'}^{ef} I_{2'\downarrow,kk'}^{ef} \\ K_q^e &\equiv \frac{4\pi}{2\ell(q) + 1} \int dE' \Sigma_{s,\ell(q)}^e. \end{aligned}} \quad (2.4.27)$$

We may also define a ‘sweep source’:

$$\boxed{s_{F,kq}^e \equiv s_{kq}^e - \sum_{k'=1}^{N_K^e} \sum_{q'=1}^Q \sum_{f=1}^{N_F^e} F_{\downarrow, kk'qq'}^{ef} \psi_{k'q'}^{e'(e,f)}} \quad (2.4.28)$$

We also know that we can use the first collision source method with $\psi \rightarrow \tilde{\psi}$ and $s \rightarrow \tilde{s}$, where:

$$\boxed{\tilde{s}_{kq}^e = \frac{4\pi}{2\ell(q) + 1} \int dE' \Sigma_{s,\ell(q)}^e(E', E) \tilde{\varphi}_k^{0,e}(E')} \quad (2.4.29)$$

2.5 Energy discretization – multi-group formalism

To discretize the energy coordinate of the phase space, we can use the multi-group approximation. In this, we *integrate* energy-dependent quantities over discrete energy groups. First, we must pick a particle energy range that is relevant to the problem, and then we will divide it into G (not necessarily uniform) intervals. We will let E_1 be the maximum and E_{G+1} be the minimum, as energy *loss* is typically an interesting quantity when picking the energy range (this convention becomes much more convenient later on when we look at ‘energy iteration’). Typically E_{G+1} is a cutoff energy after which we are not interested in the interactions of the particle in question. The particles in group g are those within interval $\Delta E_g = E_g - E_{g+1}$. Note that ΔE_g is not generally constant, however, it should be chosen such that energy-dependent quantities, in particular cross sections, are relatively unchanging within. For example, with photons and when the photoelectric effect is relevant, it is typically customary to define some satisfactory energy grouping (possibly with a constant or

logarithmic ΔE_g , or based on the spectrum of particles in the boundary conditions) and then insert new energy groups which cover the edges in the photoelectric absorption spectrum. It is also typical to have different energy groups for different particle types. Clearly, increasing G gives a better calculation, but a longer computation. However, we will later find that G is one of the more important parameters to leverage for a reasonable computation time. To begin discretization, let's look at the integration of ψ over E (in this subsection, we throw off the designations of collided and uncollided beams until the end):

$$\psi_g(\mathbf{r}, \hat{\mathbf{k}}) \equiv \int_{E_{g+1}}^{E_g} dE \psi(\mathbf{r}, \hat{\mathbf{k}}, E) \quad (2.5.1)$$

For brevity, we say:

$$\int_{E_{g+1}}^{E_g} dE f(E) \equiv \int_g dE f(E) \quad (2.5.2)$$

And thus, we can say:

$$\int_0^\infty dE f(E) \approx \sum_{g=1}^G \int_g dE f(E) \quad (2.5.3)$$

Assuming that $f(E > E_1)$ and $f(E < E_{G+1})$ are both close to zero. Now, integrate the BTE term-by-term. Consider the total attenuation term first:

$$\int_g dE \Sigma_t(E) \psi(E) = ? \quad (2.5.4)$$

We would *like* to make this into a term of the form $\Sigma_{t,g} \psi_g$. However, this would require:

$$\Sigma_{t,g} \stackrel{?}{=} \frac{1}{\psi_g} \int_g dE \Sigma_t(E) \psi(E) \quad (2.5.5)$$

Which seems a bit trivial. So we'll need to work with some approximations. Using the above, we can start by defining the following:

$$\frac{\psi(\mathbf{r}, \hat{\mathbf{k}}, E)}{\psi_g(\mathbf{r}, \hat{\mathbf{k}})} \equiv F_g(E) \quad (2.5.6)$$

Which we *assume* has no \mathbf{r} nor $\hat{\mathbf{k}}$ dependence. If we did not make this assumption, then we would be requiring that $\Sigma_{t,g}$ has dependence on $\hat{\mathbf{k}}$ (as well as a more complicated dependence on \mathbf{r}). Though this is, in principle, *generally true* if we are defining $\Sigma_{t,g}$ as such, it is very troublesome, and not necessary in the end. This approximation is referred to as the **energy separability approximation**. We therefore conclude that we *can* say:

$$\Sigma_{t,g} = \int_g dE \Sigma_t(E) F_g(E) \quad (2.5.7)$$

The function $F_g(E)$ is now referred to as the **spectral weighting function**. It has units of inverse energy. What should be picked for this function? Once again we balance computational simplicity with faithfulness to the original problem. A reasonable, *approximate*

expression should be determined for $F_g(E)$, which does *simplify* our problem. It first appeared in a statement of the form:

$$\psi(E) = F_g(E)\psi_g, \quad E_{g+1} \leq E \leq E_g \quad (2.5.8)$$

Since ψ_g is, up to a factor of ΔE_g , the average of ψ over the group g , if g is narrow enough, then $F_g(E)$ could carry the burden of interpolating from ψ_{g+1} to ψ_g . We can also expect that $\psi(E)$ stays near ψ_g , so that $F_g(E) = 1/\Delta E_g$. We will stick with this. Now we state the final form of the energy-grouped attenuation we will use:

$$\boxed{\Sigma_{t,g}(\mathbf{r}) = \frac{1}{\Delta E_g} \int_g dE \Sigma_t(\mathbf{r}, E)} \quad (2.5.9)$$

We note here that Σ_t itself is to be derived from our scattering cross sections Σ_s . In particular:

$$\Sigma_t(E) = 2\pi \int_{E'_{\min}(E)}^{E'_{\max}(E)} dE' \int_{-1}^1 d\mu \Sigma_s(\mu, E \rightarrow E') \quad (2.5.10)$$

Now let's look at the scattering terms. We can first discretize the integrals present already:

$$\int_0^{E_1} dE' \Sigma_s(E', E) \psi(E') = \sum_{g'=1}^G \int_{g'} dE' \Sigma_s(E', E) \psi(E') \quad (2.5.11)$$

Note that we modified the lower bound of the LHS integral to be zero rather than E . This appears to allow scattering from low to high energies, but this will be rectified soon by a condition on the summation index g' once we apply the BTE-wide energy average shortly, but the scattering cross sections Σ_s will, once we construct them, enforce that such scattering does not occur *anyway*. This also appears to imply that the RHS lacks a term making the integral from 0 to E_{G+1} . However, we are picking E_{G+1} such that $\psi(E') \approx 0$ here. We're now dealing with:

$$\int_g dE \int_E^{E_1} dE' \Sigma_s(E', E) \psi(E') = \sum_{g'=1}^g \int_g dE \int_{g'} dE' \Sigma_s(E', E) \psi(E') \quad (2.5.12)$$

Noting that the g' sum runs to g , not G . Within the sum on the RHS we want to take $\Sigma_{s,g'g} \psi_{g'}$. Thus, we must have:

$$\Sigma_{s,g'g} = \frac{1}{\psi_{g'}} \int_g dE \int_{g'} dE' \Sigma_s(E', E) \psi(E') \quad (2.5.13)$$

We once again invoke the energy separability approximation and the whole discussion that followed, to conclude:

$$\boxed{\Sigma_{s,g'g}(\mathbf{r}) = \frac{1}{\Delta E_{g'}} \int_g dE \int_{g'} dE' \Sigma_s(\mathbf{r}, E', E)} \quad (2.5.14)$$

The generic BTE with only energy grouping reads:

$$[\hat{\mathbf{k}} \cdot \nabla + \Sigma_{t,g}(\mathbf{r})] \psi_g = \sum_{g'=1}^g \int d\Omega' \Sigma_{s,g'g} \psi_{g'} + s_g \quad (2.5.15)$$

The prescription is, intuitively, to replace energy integrals of $f(E', E)h(E')$ into sums like $\int_{g'g} f_{g'g} h_{g'}$. Now, with respect to the first-collision scattering source, we would of course have:

$$\tilde{s}_g = \sum_{g'=1}^g \int d\Omega' \Sigma_{s,g'g} \tilde{\varphi}_{g'} \delta^2[\hat{\mathbf{k}}' - \hat{\mathbf{k}}_0(\mathbf{r})] \quad (2.5.16)$$

We will here ask: how should we get $\tilde{\varphi}_g$? The intuition would be to integrate the expression we had earlier for $\tilde{\varphi}^0$, which involved an energy distribution $f(E)$ and a Beer's law-derived exponent involving Σ_t . But, this can be a bit impractical (not impossible, though). We can, however, do something simpler. Looking at (2.5.15), we realize that the quantity which we would want to represent $\tilde{\varphi}_g^0$, i.e., the quantity which would allow us to transform the BTE in terms of $\tilde{\psi}_g$ and \tilde{s}_g , would in principle be derived from eliminating the scattering mechanism, $\Sigma_{s,g'g}$ and swapping ψ_g for $\tilde{\psi}_g^0$. That is simply applying the definition of $\tilde{\psi}_g^0$ to a slightly modified BTE. It seems then that the order of operations, i.e., the order in which we apply the multigroup approximation and transform the BTE in terms of collided particles, actually matters. This is because, if we first apply the multigroup approximation, the uncollided particles are determined by:

$$[\hat{\mathbf{k}} \cdot \nabla + \Sigma_{t,g}(\mathbf{r})] \tilde{\psi}_g^0 = s_g \quad (2.5.17)$$

Which is, as a matter of fact, a generic uncollided particle BTE. Notably, however, it is *monoenergetic*, i.e., it has a *constant* linear attenuation coefficient. That means we know the solution is:

$$\tilde{\psi}_g^0 = f_g e^{-\int d\ell \Sigma_{t,g}[\mathbf{r}(\ell)]} g(\mathbf{r}, \hat{\mathbf{k}}) \delta^2[\hat{\mathbf{k}} - \hat{\mathbf{k}}_0(\mathbf{r})] \quad (2.5.18)$$

where $g(\mathbf{r}, \hat{\mathbf{k}})$ is a function of space and angle that depends on the kind of source being used. Pay attention only to the two factors: f_g and $e^{-\int d\ell \Sigma_{t,g}[\mathbf{r}(\ell)]}$. Now, had we energy-grouped the polychromatic expression for $\tilde{\varphi}^0$, we would find the corresponding single term: $\int_g dE f(E) e^{-\int d\ell \Sigma_t[\mathbf{r}(\ell), E]}$. The only difference is that, in the previous equation, Σ_t was not assumed to have any energy dependence, and it took the value $\Sigma_{t,g}$. This is, after all, the essence of the multigroup approximation *anyway*. But, insofar as we are solving (2.5.15) now, *not* the original BTE, we will just take the constant $\Sigma_{t,g}$. Otherwise, if we are using an approximation in $\tilde{\psi}$ but not $\tilde{\psi}^0$, we will have a (probably small) excess/deficit of particles that is not due to the approximation method itself, but rather, this lopsidedness of the approximation.

We now make a note concerning energy grouping. There are two useful ways to construct energy groups. One is to use 'linear' energy groups, meaning that ΔE_g is constant. That would mean:

$$E_g = -\frac{1}{G}(E_{\max} - E_{\min})g + \left(1 + \frac{1}{G}\right)E_{\max} - \frac{1}{G}E_{\min} \quad (2.5.19)$$

We can also use ‘logarithmic’ energy groups. This means that $\ln(E_g/E_{g+1})$ is constant. So, on a logarithmic scale, our energy groups are equally size.

$$E_g/\text{MeV} = \ln \left[-\frac{1}{G} (e^{E_{\max}/\text{MeV}} - e^{E_{\min}/\text{MeV}})g + \left(1 + \frac{1}{G}\right) e^{E_{\max}/\text{MeV}} - \frac{1}{G} e^{E_{\min}/\text{MeV}} \right] \quad (2.5.20)$$

Now, the decision of which structure to take can be motivated by particle type, but I mostly prefer to stick with logarithmic energy groups. These provide resolution to lower energies where in many cases it is needed. Especially when dealing with electrons, the curvature of the energy spectra typically present is very large at low energies. One may also want to keep in mind the need to manually insert energy group bounds. For instance, when solving for photons, one should insert an energy bound at $2m_e c^2$, which is the cutoff below which pair production can not occur. This makes it so that photons of energy less than $2m_e c^2$ won’t be said to contribute to pair production, as we are not lumping together photons of energies below and above this value. Additionally, shell binding energies may be important due to the structure of the photoelectric effect cross sections. Much more on this in **wiscobolt physics**, but also, wiscobolt is equipped with settings to manually impose these ‘exceptions’ to the user-selected energy group structure, more on this in **wiscobolt implementation** (the user-input chapter fully describes this).

2.6 Energy discretization – finite element method

Now we discuss a method of energy discretization alternative to multigroup formalism, using the FEM. We refer to this as FEXS (finite element cross sections). This has not been used as extensively as multigroup formalism – it has been developed recently and was implemented for the SCEPTRÉ Boltzmann transport solver, by Drumm et al. [38, 39] (although, note that I do not have access to the document [39], only I know it is related to generation of finite element cross sections for SCEPTRÉ, which we attempt to do ourselves in **wiscobolt physics**). The FEM in energy happens to be advantageous in various areas. For one, it relies on the approximation that ψ varies as a polynomial within a group, rather than that ψ is close to flat, as is assumed by MGXS. Additionally, we will much later introduce an operator to the electron BTE, and discretization of this operator is rather crude with MGXS, but quite straightforward with FEXS. The validity of discretization of this operator with FEXS has been demonstrated by [38]. Finally, it so happens that MGXS is a special case of FEXS – that is, one where the shape functions are flat, either 1 or 0 depending on the node. We will discuss this briefly at the end of this section.

We’ve already learned about the FEM in the context of spatial discretization. Now, we will apply it to a 1D problem. We will take the linear FEM exclusively, because this is all wiscobolt uses at the moment, but the generalization to higher order meshes should be simple. Specifically, we take for energy shape functions:

$$\begin{aligned} \Lambda_1^g(E) &\equiv \frac{E - E_2^g}{\Delta E_g} \\ \Lambda_2^g(E) &\equiv 1 - \frac{E - E_2^g}{\Delta E_g} \end{aligned} \quad (2.6.1)$$

so that:

$$\Lambda_n^g(E_{n'}) = \delta_{nn'} \quad (2.6.2)$$

Where g indexes the element ($g = 1, \dots, G$, so, you can still consider it to refer to a group) and n indexes a node ($n = 1, 2$). We still take E_1^1 to be the maximum energy, equivalent to E_1 , and E_2^G to be the minimum energy, equivalent to E_{G+1} . We will actually retain the entire set of E_g as the global nodes (g in that context did not mean ‘group’ either, since $g = 1, \dots, G + 1$). Notably, ΔE_g here retains the exact same meaning as in the multi-group method. Another way, more convenient for implementation, is to write:

$$\Lambda_n^g(E) = (-1)^n \frac{E_{3-n}^g - E}{\Delta E_g}$$

(2.6.3)

A simple map between E_n^g and the E_g we defined in MGXS is $E_n^g = E_{g+n-1}$ ($g+n-1$ is, in essence, an element of a 1D connectivity array). Now, we have:

$$\psi(E) = \sum_{n=1}^2 \Lambda_n^g(E) \psi(E_n^g), \quad E_2^g \leq E \leq E_1^g \quad (2.6.4)$$

We will let $\psi_n^g = \psi(E_n^g)$, while $\psi_g = \psi(E_g)$, where $g = 1, \dots, G + 1$ in both conventions.

Now, we make the weak form of the BTE by multiplying through with $\Lambda_n^g(E)$ and integrating over the group g , then expanding ψ in $\Lambda_{n'}^g$:

$$\int_g dE \Lambda_n^g \hat{\mathbf{k}} \cdot \nabla \psi = \hat{\mathbf{k}} \cdot \nabla \sum_{n'=1}^2 \langle \Lambda_n^g, \Lambda_{n'}^g \rangle_g \psi_{n'}^g \quad (2.6.5)$$

$$\int_g dE \Lambda_n^g \Sigma_t \psi = \sum_{n'=1}^2 \langle \Lambda_n^g, \Sigma_t \Lambda_{n'}^g \rangle_g \psi_{n'}^g \quad (2.6.6)$$

To treat the scattering term, we first recognize:

$$\int dE' \Sigma_{s,\ell}(E' \rightarrow E) \psi = \sum_{g'=1}^g \sum_{n'=1}^2 \langle \Sigma_{s,\ell}, \Lambda_{n'}^{g'} \rangle_{g'} \psi_{n'}^{g'} \quad (2.6.7)$$

And then:

$$\int_g dE \Lambda_n^g(E) \int dE' \Sigma_{s,\ell}(E' \rightarrow E) \psi(E') = \sum_{g'=1}^g \sum_{n'=1}^2 \langle \Lambda_n^g, \langle \Sigma_{s,\ell}, \Lambda_{n'}^{g'} \rangle_{g'} \rangle_g \psi_{n'}^{g'} \quad (2.6.8)$$

We now make the following definitions:

$$\lambda_{n'n}^g \equiv \langle \Lambda_n^g, \Lambda_{n'}^g \rangle_g \quad (2.6.9)$$

$$\Sigma_{t,n'n}^g \equiv \langle \Lambda_n^g, \Sigma_t \Lambda_{n'}^g \rangle_g \quad (2.6.10)$$

$$\Sigma_{s,\ell,n'n}^{g'g} \equiv \langle \Lambda_n^g, \langle \Sigma_{s,\ell}, \Lambda_{n'}^{g'} \rangle_{g'} \rangle_g \quad (2.6.11)$$

Briefly, the source should receive separate treatment depending on the problem at hand. If our source is given explicitly and has no singularities, we should use:

$$\int_g dE \Lambda_n^g s = \sum_{n'=1}^2 \lambda_{n'n}^g s_{n'}^g \quad (2.6.12)$$

where $s_n^g = s(E_n^g)$. But if our source is written in the form $\hat{K}\psi$, we would write:

$$\int_g dE \Lambda_n^g s^\beta = \int d\Omega' \int dE' \int_g dE \Lambda_n^g(E) \Sigma_s^{\alpha\beta}(E' \rightarrow E) \psi^\alpha(E') \quad (2.6.13)$$

where ψ^α is the fluence of the particle being scattered to create the source. Now, we would expand ψ^α as usual, then we'd find:

$$\int_g dE \Lambda_n^g s^\beta = \int d\Omega' \sum_{g'=1}^g \sum_{n'=1}^2 \psi_{n'}^{\beta,g'} \int_{g'} dE' \int_g dE \Lambda_n^g(E) \Sigma_s^{\alpha\beta}(E' \rightarrow E) \Lambda_{n'}^{g'}(E') \quad (2.6.14)$$

rather than 2.6.12. We note

Now our problem reads:

$$\sum_{n'=1}^2 (\lambda_{n'n}^g \hat{\mathbf{k}} \cdot \nabla + \Sigma_{t,n'n}^g) \psi_{n'}^g - \sum_{g'=1}^g \sum_{n'=1}^2 \int d\Omega \Sigma_{s,n'n}^{g'g} \psi_{n'}^{g'} = \int_g dE \Lambda_n^g s \quad (2.6.15)$$

The analog of $\lambda_{n'n}^g$ in the spatial FEM is $I_{1,kk'}^e$. We determined that I_1^e was more conveniently inverted on both sides. We will do this once again for yet another major simplification, and we will define:

$$\begin{aligned} \Sigma_{t',n'n}^g &= \sum_{n''=1}^2 (\lambda^{-1})_{nn''}^g \Sigma_{t,n'n''}^g \\ \Sigma_{s',n'n}^{g'g} &= \sum_{n''=1}^2 (\lambda^{-1})_{nn''}^g \Sigma_{s,n'n''}^{g'g} \\ s_n^g &= s(E_n^g) = \sum_{n''=1}^2 (\lambda^{-1})_{nn''}^g \int_g dE \Lambda_{n''}^g s \end{aligned} \quad (2.6.16)$$

Where, in the last line we are technically performing the same approximation regardless of whether s is explicitly known or calculated from 2.6.12. Nevertheless, it is easy to show that:

$$\boxed{\lambda_{n'n}^g = \frac{1}{6} \Delta E_g (1 + \delta_{nn'})} \quad (2.6.17)$$

And also then:

$$\boxed{(\lambda^{-1})_{nn'}^g = \frac{2}{\Delta E_g} (-1 + 3\delta_{nn'})} \quad (2.6.18)$$

Now, our problem is decidedly a matrix problem in the index pairs (g, g') and (n, n') , analogous to the problem we found in the spatial FEM in the pairs (e, e') and (k, k') . It doesn't have a sweep (yet), but, it does have off-diagonal terms in the scattering operator. The problem now stands as:

$$\sum_{n'=1}^2 (\delta_{nn'} \hat{\mathbf{k}} \cdot \nabla + \Sigma_{t',n'n}^g) \psi_{n'}^g - \sum_{g'=1}^g \sum_{n'=1}^2 \int d\Omega \Sigma_{s',n'n}^{g'g} \psi_{n'}^{g'} = s_n^g \quad (2.6.19)$$

Or we can make the associations:

$$T_{nn'}^g \equiv \delta_{nn'} \hat{\mathbf{k}} \cdot \nabla + \Sigma_{t',n'n}^g \quad (2.6.20)$$

$$K_{n'n}^{g'g} \equiv \int d\Omega \Sigma_{s',n'n}^{g'g}. \quad (2.6.21)$$

And this yields:

$$\sum_{n'=1}^2 T_{nn'}^g \psi_{n'}^g - \sum_{g'=1}^g \sum_{n'=1}^2 K_{n'n}^{g'g} \psi_{n'}^{g'} = s_n^g \quad (2.6.22)$$

Unlike in MGXS, we will not discuss the use of the FCS with FEXS. This is because, a FCS in FEXS would require analytical solution of:

$$\sum_{n'=1}^2 T_{nn'}^g \tilde{\psi}_{n'}^{0,g} = s_n^g \quad (2.6.23)$$

or:

$$\begin{pmatrix} \hat{\mathbf{k}} \cdot \nabla + \Sigma_{t,11} & \Sigma_{t,21} \\ \Sigma_{t,12} & \hat{\mathbf{k}} \cdot \nabla + \Sigma_{t,22} \end{pmatrix} \begin{pmatrix} \tilde{\psi}_1^0 \\ \tilde{\psi}_2^0 \end{pmatrix} = \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} \quad (2.6.24)$$

where as MGXS only requires analytical inversion of $\hat{\mathbf{k}} \cdot \nabla + \Sigma_{t,g}$, which is doable via ray-tracing as discussed previously. The issue with the above equation is the mixing of $\tilde{\psi}_1^0$ and $\tilde{\psi}_2^0$, due to the coefficients $\Sigma_{t,12}$ and $\Sigma_{t,21}$. One way that FCS can still be achieved is to determine $\tilde{\psi}_n^{0,g} = \tilde{\psi}^0(E_n^g)$ from the analytical, undiscretized form of $\tilde{\psi}^0$. However, this means construction (or at least ability to evaluate at a given E) of $\Sigma_t(E)$. If this is done however, it must be noted that $\Sigma_t(E)$ contains some terms which are not straightforward to write analytically for electrons, and so a re-definition of the uncollided fluence will need to be made which is with respect to only those terms in $\Sigma_t(E)$ which can be written (this will become clearer in **Section 3.3.2** when we see the term which is not straightforwardly included in $\Sigma_t(E)$, despite that $\Sigma_{n'n}^g$ is still easy to form).

Lastly, we note that a direct connection exists between the FEXS and MGXS discretization methods. Specifically, MGXS is a special case of FEXS, wherein one takes $\Lambda_1^g = 1$ and $\Lambda_2^g = 0$. This leaves only matrix elements with $(n, n') = (1, 1)$, and we had solutions of the form $\psi_g = \int_g dE \Lambda_1^g \psi$.

3 SOLUTION

We have formulated a rather large matrix problem in terms of discrete quantities related in some manner to the continuous angular fluence. We have formulated essentially four ‘pathways’ of discretization, which have unique properties in how they may be approached: S_N MGXS, S_N FEXS, P_N MGXS, and P_N FEXS. For a single particle, the number of unknowns that must be solved for a given pathway is given in the following list (assuming that every element in our spatial mesh has the same number of nodes, and all elements in either the spatial or energetic FEM are linear):

- S_N MGXS: $N_E N_K^e N_\mu N_\phi G$
- S_N FEXS: $2N_E N_K^e N_\mu N_\phi G$
- P_N MGXS: $N_E N_K^e (L + 1)^2 G$
- P_N FEXS: $2N_E N_K^e (L + 1)^2 G$

Noting that S_N still treats scattering with P_L , but this is not supposed to be the number of operations performed, so L does not feature in the first two items. Now, the matrix problems which produce these unknowns contain as many elements as the number of unknowns *squared*. So, is a naïve approach of simply solving a linear system feasible? Let’s take a rather coarse spatial discretization with reasonable angular discretization. Say we have $N_E = 10,000$, $N_K^e = 4$, $N_\mu = N_\phi = 16$, $L = 16$, $G = 25$. Then, we’d have:

- S_N MGXS: $2.56 \cdot 10^8$ unknowns / $6.5536 \cdot 10^{16}$ total matrix elements
- S_N FEXS: $5.12 \cdot 10^8$ unknowns / $2.62144 \cdot 10^{17}$ total matrix elements
- P_N MGXS: $2.89 \cdot 10^8$ unknowns / $8.3521 \cdot 10^{16}$ total matrix elements
- P_N FEXS: $5.78 \cdot 10^8$ unknowns / $3.34084 \cdot 10^{17}$ total matrix elements

These estimates, obviously, are prohibitively large. There is no computer capable of working with $\sim 10^{16}$ operations in a reasonable amount of time, nevermind inverting a system of that size. Now, to be fair, we *know* that this system will be extremely sparse in the spatial part. So, $\sim 10^{16}$ shouldn’t be an accurate estimate of the number of operations we must perform. However, even with sparseness, we have very many unknowns. And remember, this is a rather coarse discretization. Typical problems encountered may involve between 100,000 and 1,000,000 elements, substantially more difficult than 10,000. So, our problem, though discretized, needs further application of *approximations*, as well as clever solution methods that take advantage of the structure of the system we have formed. This is to make our solution *possible*. As for making the solution *practical*, we can only go so far. We may be concerned primarily with two matters: speed and storage. As far as speed is concerned, we can actually do surprisingly well to produce our solutions rather quickly on a modest personal computer. This is a matter of, obviously, minimizing the number of floating point operations, but also less intuitively, managing data access issues which will show up for one

of our solution methods. Note, we have detailed precisely how this is handled by wiscobolt in the **wiscobolt implementation** document. The problem is, regardless of how quickly we can make our solver, storage is an issue which may quickly overwhelm us. This is made quite clear by the number of unknowns in our coarse discretization. Knowing that, we must not only store our solution vector (at least while we are performing the solution), but also other arrays of similar and in some cases larger size, we can obtain an estimate of the kind of storage we will need. For *single precision* and the coarse $N_E = 10,000$, the amount of unknowns in S_N MGXS amounts to about 1 GB (note, as we have discussed in the **wiscobolt implementation** document, we frequently store our solutions in the *global* mesh, meaning, rather than $N_E N_K^e$ spatial unknowns, we only have N_K , which is *drastically* fewer. However, it is still necessary to construct some arrays which are based on the elemental mesh). This isn't so bad, until we decide to increase the number of elements to $N_E = 100,000$. Then, we are dealing with 10 GB, all for the *solution alone*. If we then want to change, say, the number of energy groups to $G = 50$ (which is not at all excessive), we already have 20 GB (altogether more RAM than what my personal computer has at the time of writing this). So, while we can make efforts to minimize our storage consumption, especially at the cost of computation speed, we can not eliminate storage issues entirely, at least not for the more interesting calculations one may want to do.

Nevertheless, we will outline some solution methods that are general enough to treat a generic BTE. Then, we will discuss some of the physics involved in order to make approximations and simplify the coupled nature of the problem we outlined at the beginning of this document. Finally, we will discuss very briefly how the solutions may be accelerated. We go into it only briefly because an excellent publication by Drumm and Fan [21] already exists describing in depth and benchmarking acceleration across discretization and solution methods, in SCEPTRE.

3.1 Iterative methods

3.1.1 Energy iteration

We stated earlier that we will not consider our scattering mechanisms to be able to create (an) outgoing particle(s) with a greater energy than that of the incoming particle. We are taking the perspective that particles in our target material, such as atoms, are at rest (or, maybe more rigorously, in the 'lab frame' they have very little energy in comparison to the incident, and typically relativistic, projectiles). Nevertheless, what this means for us is that there is no transport of particles from an energy group (or element) g' to an energy group $g < g'$. Remember, *larger* energy group index g corresponds to *lower* energy. We have essentially formulated a matrix problem which has as indices g' and g . Our matrix has a term diagonal in g , i.e., T_g , and a term which handles the mixture of particles across different groups, $K_{g'g}$. Now, to say that no particles travel to higher energy groups is to say that:

$$K_{g' \rightarrow g} = 0, \quad g < g' \quad (3.1.1)$$

Which follows from $\Sigma_{g' \rightarrow g} = 0$, $g < g'$. Note that the objects like $K_{g' \rightarrow g}$ and T_g are written to suppress all other indices, potentially including the indices (n, n') from the FEXS energy

discretization method. Multiplication by, say, T_g , will be taken to imply matrix multiplication in all other indices as per a block matrix problem, or a continuous operator which has only had energy discretization (the result, symbolically, is the same). Now, if this condition is satisfied for $K_{g'g}$, then it is also satisfied for $L_{gg'}$. That allows us to formulate the following matrix problem:

$$\begin{pmatrix} L_{11} & 0 & \dots & 0 \\ L_{21} & L_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ L_{G1} & L_{G2} & \dots & L_{GG} \end{pmatrix} \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_G \end{pmatrix} = \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_G \end{pmatrix} \quad (3.1.2)$$

Inversion of such a matrix is straightforward iteratively, in a method referred to as ‘forward iteration.’ Simply *write out* the matrix equation line-by-line:

$$\begin{aligned} L_{11}\psi_1 &= s_1 \\ L_{21}\psi_1 + L_{22}\psi_2 &= s_2 \\ &\vdots \\ L_{G1}\psi_1 + L_{G2}\psi_2 + \dots + L_{GG}\psi_G &= s_G \end{aligned} \quad (3.1.3)$$

One can solve for ψ_1 , and then use that to obtain ψ_2 , and so on. This can be seen by writing each line as:

$$\sum_{g' < g} L_{gg'}\psi_{g'} + L_{gg}\psi_g = s_g \quad (3.1.4)$$

At which point we can just rearrange:

$$\psi_g = (L_{gg})^{-1}(s_g - \sum_{g' < g} L_{gg'}\psi_{g'}) \quad (3.1.5)$$

But then, with $T_{gg'} = T_g\delta_{gg'}$, we recognize that, $L_{gg'} = -K_{gg'}$ for $g' \neq g$. Thus:

$$\psi_g = (L_{gg})^{-1}(s_g + \sum_{g' < g} K_{gg'}\psi_{g'}) \quad (3.1.6)$$

This is a *great simplification* because not only have we manually inverted the (g, g') dependence in L , but we also find that we only need to solve G monoenergetic Boltzmann transport problems in space and angle, specifically, only those in the diagonal in (g, g') .

Energy iteration will be applied to our following iteration methods, but before proceeding, we should look at some of its implications right now. First, we will decide that:

$$S_g \equiv s_g + \sum_{g' < g} K_{gg'}\psi_{g'} \quad (3.1.7)$$

Immediately, we will then have the following Boltzmann transport problem:

$$(T_g - K_{gg})\psi = S_g \quad (3.1.8)$$

This becomes interesting when we look at this as a BTE in its continuous form. That is:

$$\hat{\mathbf{k}} \cdot \nabla \psi_g + \left(\Sigma_{t,g} - \int d\Omega \Sigma_{s,gg} \cdot \right) \psi_g = S_g \quad (3.1.9)$$

In MGXS, this is a problem with a constant linear attenuation coefficient, and a scattering mechanism which is differential only in angle. Now, the angular integral still makes this a bit difficult to solve, of course. But nevertheless, it is interesting to note that, especially for massive particles, the within-group (or low energy transfer) interactions represented by $\Sigma_{s,gg}$ happen to be the most forward-peaked interactions. We will take advantage of this fact to make an extremely valuable simplification to the BTE in **Section 3.3.3**.

3.1.2 Source iteration

The next iterative method we discuss is **source iteration (SI)** [3, 29, 40]. We look back to a definition made much earlier:

$$\psi = \tilde{\psi} + \tilde{\psi}^0 \quad (3.1.10)$$

which separates our beam into a collided part $\tilde{\psi}$ and an uncollided part $\tilde{\psi}^0$. We did this because we knew that, for external beams, $\tilde{\psi}^0$ has a unique form which is both unsuitable for discretization and suitable for explicit calculation of $\hat{K}\tilde{\psi}^0$. Nevertheless, here we can achieve a simpler form of the BTE if we generalize this separation, and actually distinguish the ‘ p th’ collided part of the beam. Specifically, let $\tilde{\psi}^p$ be the fluence of particles which have been collided *exactly* p times. Then, we can say:

$$\psi = \sum_{p=0}^{\infty} \tilde{\psi}^p \quad (3.1.11)$$

We will let ψ^M be the fluence of particles which have been collided up to M times, meaning:

$$\psi^M = \sum_{p=0}^M \tilde{\psi}^p \quad (3.1.12)$$

Now, in principle, $\tilde{\psi}^{p+1}$ is sourced by scattering of $\tilde{\psi}^p$, but it does not have a scattering source term corresponding to scattering into itself, because when $\tilde{\psi}^{p+1}$ particles scatter they are no longer $\tilde{\psi}^{p+1}$ particles, but rather $\tilde{\psi}^{p+2}$ particles. Intuitively, this means:

$$\hat{T}\tilde{\psi}^{p+1} = \hat{K}\tilde{\psi}^p \quad (3.1.13)$$

for $p \geq 0$. This says that the source of $p+1$ particles is p particles alone. But we can arrive at this result more rigorously. Suppose that M is arbitrarily large such that $\psi \approx \psi^M$ (this is an assumption, but it can be proven that this is a valid approximation for some M , i.e., that this kind of expansion converges). This becomes an equality if we tend M to infinity. If we apply \hat{L} to ψ^M , we get:

$$\hat{L}\psi^M = (\hat{T} - \hat{K}) \sum_{p=0}^M \tilde{\psi}^p = s \quad (3.1.14)$$

Using $\hat{K}\tilde{\psi}^M \approx 0$, we have:

$$\hat{T} \sum_{p=0}^M \tilde{\psi}^p = \hat{K} \sum_{p=0}^{M-1} \tilde{\psi}^p + s \quad (3.1.15)$$

Then, with $\hat{T}\tilde{\psi}^0 = s$, we can say:

$$\hat{T} \sum_{p=1}^M \tilde{\psi}^p = \hat{K} \sum_{p=0}^{M-1} \tilde{\psi}^p \quad (3.1.16)$$

Shifting the p index of the sum on the LHS, we have:

$$\hat{T} \sum_{p=0}^{M-1} \tilde{\psi}^{p+1} = \hat{K} \sum_{p=0}^{M-1} \tilde{\psi}^p \quad (3.1.17)$$

Now the sum can be thrown off:

$$\boxed{\hat{T}\tilde{\psi}^{p+1} = \hat{K}\tilde{\psi}^p, \quad p = 0, 1, \dots} \quad (3.1.18)$$

starting, of course, from $\hat{T}\tilde{\psi}^0 = s$, or equivalently, from $\hat{T}\tilde{\psi}^1 = \tilde{s}$ if one prefers to provide that and begin the equation above from $p = 1$. If we prefer to build up ψ^M successively we could identically write:

$$\boxed{\hat{T}\psi^{p+1} = \hat{K}\psi^p + s} \quad (3.1.19)$$

Note that this is just a method to solve a BTE. If \hat{K} is monoenergetic provided the right source S , as per energy iteration, the above simplifies straightforwardly.

We will find that SI is very practical with S_N angular discretization. First recognize that SI attempts to solve the BTE exclusively through inversion of \hat{T} . Recall that in both S_N and P_N , we found off-diagonal (e, e') block matrix elements in the matrix T corresponding to \hat{T} , corresponding to dependence of one element e on all of its neighbors e' . But in S_N , the matrix T does not mix angle, thus, it is simply indexed by angular indices (i, j) . For one choice of (i, j) , which corresponds to selection of a single, discrete angle, some off-diagonal (e, e') elements will actually be zero! Specifically, if one selects the matrix T_{ij} corresponding to some discrete ordinate $\hat{\mathbf{k}}_{ij}$, then the matrix element (e, e') , corresponding to two neighboring finite elements, is zero when the element e is ‘upstream’ of e' . This is seen by looking at F_{\downarrow} , which is the exclusively off-diagonal component of T . Since SI asserts we only invert \hat{T} , we end up with a sequence of inversions of the spatial part of T , that are indexed by (i, j) . For a given (i, j) , we’d then be able to find first all elements with no upstream neighbors, solve them through explicitly inverting their diagonal block matrices (as they will have no off-diagonal components in \hat{T}), and then use their solution to supply their neighbors that can ultimately be inverted by updating the source and inverting their diagonal block matrices (similar to forward substitution in EI). This procedure is referred to as the ‘sweep,’ and we described it to some detail in **Section 2.3.2**. But for P_N , the matrix T *does* mix angular indices (q, q') . So, in general, every element will depend on all of its neighbors, and we can not perform inversion of T iteratively in such a manner. We could

still iteratively solve $Tx = b$, but doing so is not very practical in comparison to iteratively solving $Lx = b$ as a whole, which we will describe in the next section.

For S_N , SI outlines an iterative procedure to obtain any $\tilde{\psi}^p$. We still must, however, ask two questions: 1) is this problem convergent, i.e., does this necessarily converge as $p \rightarrow \infty$? 2) Does this come reasonably close to convergence for p that is not excessively large? We will demonstrate shortly that this converges as $\psi = \psi^{M \rightarrow \infty}$, and is in fact identical to a more mathematically motivated form of matrix inversion via iteration. But we'll first discuss: if we want to take $\psi \approx \psi^M$ with a given convergence criterion, what M should we expect? Photons tend not to scatter as much as, say, electrons or hadrons. In water, the range of photons of energy 1 MeV is about 16 cm [41]. Thus, in radiation therapy, a single photon will likely not scatter too many times before being absorbed or exiting the patient entirely. So for a typical situation and a reasonable convergence criterion we really won't anticipate $p \gg 10$ or so. Thus, it happens to be extremely feasible for photons to use S_N with SI. We can assess convergence in two ways. One is to evaluate the residual of the problem at every iterate, i.e.:

$$r^p \equiv |s - \hat{L}\psi^p|_2 \quad (3.1.20)$$

where $|x|_2$ is the L2-norm of the vector x . Or, with what is much less computationally expensive, we can look at the ratio of the L2-norm of the fluence at the next iterate to the fluence up to the previous iterate.

$$\epsilon^p \equiv \frac{|\tilde{\varphi}^p|_2}{|\varphi^{p-1}|_2} = \frac{|\varphi^p - \varphi^{p-1}|_2}{|\varphi^{p-1}|_2} \quad (3.1.21)$$

This is formally called the ‘spectral radius’ [35]. This gives us an idea of how much ‘stuff’ is being added to our total solution. In wiscobolt, the user has the option to calculate r^p at every iterate, but we rely on ϵ^p to determine convergence up to a user-defined criterion (typically, 10^{-8} is safe).

Now, let's briefly demonstrate that source iteration is identical to $\hat{L}\psi = s$ when we take $M \rightarrow \infty$. Start with the original problem:

$$\hat{L}\psi = s \quad (3.1.22)$$

Iteration procedures exist wherein \hat{L} is split into the sum of two or more operators. They are designed such that one can obtain a solution iteratively while only needing to invert one, simpler operator. For instance, consider the simple algebraic matrix problem:

$$Ax = b \quad (3.1.23)$$

Gauss-Seidel iteration, a method which has also been applied to solving the BTE, involves decomposing A into the sum of a lower triangular matrix L and an upper triangular matrix U (with either containing the diagonal). Then, one can write:

$$Lx = b - Ux \quad (3.1.24)$$

The RHS is then “lagged,” meaning one iterates as:

$$\begin{aligned} Lx^0 &= b \\ Lx^1 &= b - Ux^0 \\ &\vdots \\ Lx^{M+1} &= b - Ux^M \end{aligned} \tag{3.1.25}$$

We have seen how inversion of a lower triangular matrix can be straightforward through iteration. These methods are known to converge if the matrices involved are either symmetric and positive-definite, or strictly diagonally dominant. Instead, however, our iteration method asserted that our original operator \hat{L} is broken down into $\hat{T} - \hat{K}$, with $\hat{K}\psi$ being lagged. This is obvious if we consider fully writing out the first few terms of this iteration:

$$\begin{aligned} \psi^0 &= \hat{T}^{-1}s \\ \psi^1 &= \hat{T}^{-1}s + \hat{T}^{-1}\hat{K}\hat{T}^{-1}s \\ \psi^2 &= \hat{T}^{-1}s + \hat{T}^{-1}\hat{K}\hat{T}^{-1}s + \hat{T}^{-1}\hat{K}\hat{T}^{-1}\hat{K}\hat{T}^{-1}s \\ &\vdots \end{aligned} \tag{3.1.26}$$

We recognize that we could define from scratch:

$$\begin{aligned} \tilde{\psi}^0 &\equiv \hat{T}^{-1}s \\ \tilde{\psi}^1 &\equiv \hat{T}^{-1}\hat{K}\hat{T}^{-1}s \\ \tilde{\psi}^2 &\equiv \hat{T}^{-1}\hat{K}\hat{T}^{-1}\hat{K}\hat{T}^{-1}s \\ &\vdots \end{aligned} \tag{3.1.27}$$

so that:

$$\begin{aligned} \psi^0 &= \tilde{\psi}^0 \\ \psi^1 &= \tilde{\psi}^0 + \tilde{\psi}^1 \\ \psi^2 &= \tilde{\psi}^0 + \tilde{\psi}^1 + \tilde{\psi}^2 \\ &\vdots \end{aligned} \tag{3.1.28}$$

We recognize that this is perfectly consistent with our earlier assertion:

$$\psi^M = \sum_{p=0}^M \tilde{\psi}^p \tag{3.1.29}$$

We talked about reasonable numbers of scattering events for photons, but we glossed over electrons (or more generally, massive particles). What about them? Electrons love to scatter, and they have ranges in water between hundreds of microns to tens of nanometers [1, 2]. This is *extremely* impractical with SI. Thus, we have three practical choices: 1) attempt to modify our interpretation of what it means to ‘scatter’ with respect to our iteration [42–44], 2) attempt to accelerate the convergence of SI [21], or 3) use a different solution method altogether [45]. The first choice is the topic of **Section 3.3.3**. The second is the topic of **Section 3.2**. The third is the topic of the next section.

3.1.3 Generalized minimal residual method

The linear system $\hat{L}\psi = s$, most notably, is extremely large when discretized, but also extremely sparse. Additionally, it is not symmetric nor necessarily positive-definite. For situations such as this, an incredibly practical and successful algorithm was developed by Saad and Schultz [45], known as the **generalized minimum residual method (GMRES)**. A complete description will be found in the primary source or in **wiscobolt implementation**, however, a basic description will be given here. In GMRES, we form an approximate solution to the linear system $Ax = b$ using only multiplication with A , some generic, much smaller sized matrix multiplications, and one backwards substitution. The solution that GMRES gives us is exactly the one which minimizes the residual $b - Ax$ in a t -dimensional **Krylov subspace**, using only t multiplications with A . The Krylov subspace is defined by what kinds of vectors prop it up, or *span* it:

$$K_t(A, x_0) = \text{span} \left\{ v_0, Av_0, A^2v_0, \dots, A^{t-1}v_0 \right\} \quad (3.1.30)$$

Where x_0 is an initial guess solution, and $v_0 = (b - Ax_0)/|b - Ax_0|$, i.e., the normalized residual of the initial guess. Note that the Krylov subspace does not specify *anything* about the structure of the multiplication Av_0 , although for our purposes, it will be matrix multiplication. But what t is taken? This algorithm iteratively appends vectors and orthogonalizes them to the Krylov subspace until such a t is reached that we *know* we will find a residual below some given convergence criterion. That said, the choice of x_0 typically has a significant impact on the convergence of the problem. I very strongly recommend reading [45] in order to properly appreciate the simplicity and power of this method.

Since most terms in L are diagonal in (e, e') (only F_\downarrow is not), multiplication by L is not exceedingly difficult. With S_N though, we will prefer to modify the problem as so:

$$(1 - \hat{T}^{-1}\hat{K})\psi = \hat{T}^{-1}s \quad (3.1.31)$$

and so to form $\hat{T}^{-1}\hat{K}\psi$, one would first form the vector $\hat{K}\psi$, then use a sweep to apply \hat{T}^{-1} (with boundary condition zero). This is advantageous because the linear system $1 - \hat{T}^{-1}\hat{K}$ will always converge more quickly than $\hat{T} - \hat{K}$ (at least slightly), since we have already done part of the work in inverting \hat{T} . This approach is most generally called **preconditioning** [21, 46], and we can come up with alternative ways to precondition that may also be applied to P_N . Let's suppose that we have some easily invertible operator \hat{M} , such that:

$$\hat{M}\psi \approx s \quad (3.1.32)$$

(We are being loose with the definition of ‘ \approx ’). Then, we would be able to say something like:

$$\hat{M}^{-1}\hat{L} \approx 1 \quad (3.1.33)$$

If we attempted to solve the linear system:

$$\hat{M}^{-1}\hat{L}\psi = \hat{M}^{-1}s \quad (3.1.34)$$

with $(\hat{M}^{-1}\hat{L})$ as our new system operator and $\hat{M}^{-1}s$ as our new source, then we would expect faster GMRES convergence (i.e., smaller Krylov subspace dimension t) depending

on our choice of \hat{M} . Now recognize that \hat{T} in the S_N case is ‘easily invertible’ and also ‘approximate’ to \hat{L} . Thus, application of \hat{T}^{-1} is a preconditioning strategy. But other preconditioners can be investigated in addition to this, and would be particularly useful for P_N (where each application of \hat{T}^{-1} would require its own entire GMRES procedure since we can not perform a sweep). A common method is **incomplete factorization (IF)** [21, 46], wherein we can perform a matrix decomposition on L which retains the same sparsity pattern. That is, if we tried to perform, say, an LU-decomposition [46], which factors a matrix A into a product of lower triangular and upper triangular matrices (both of which are easily iteratively inverted, hence the utility of this method), then we may expect that the decomposition is drastically less sparse than A . This is not acceptable because L has far too many elements. We are aware, however, that we can store and subsequently multiply its elements using sparse storage and multiplication methods, and so if we would form a decomposition which is forced to have the same sparsity pattern as L , then we can use that as an approximate inverse and thus a preconditioner. Another preconditioner we can use is actually a S_N SI system itself! We will discuss this further in the next section.

3.2 Acceleration across solution and discretization methods

It is possible to accelerate a slowly converging solution/discretization method with a coarse solver with a different solution/discretization method. This is described in tremendous depth in [21], involving both a description of the mathematical problem as well as thorough benchmarks with various combinations of acceleration methods and parameters in particular problems. What follows, outside of a brief discussion of diffusion synthetic acceleration that is derived from other sources, is derived from this paper.

3.2.1 Acceleration of SI

Acceleration of SI involves applying a correction to ψ^M following every iterate which is an estimate of the true error at this iterate. First, suppose that we now perform source iteration in the classical form. The residual at this iterate is:

$$r = s - (\hat{T} - \hat{K})\psi^p = \hat{K}(\psi^p - \psi^{p-1}) \quad (3.2.1)$$

The error is:

$$\varepsilon = \psi - \psi^p \quad (3.2.2)$$

We can show that:

$$(\hat{T} - \hat{K})\varepsilon = r \quad (3.2.3)$$

And therefore:

$$\psi = \psi^p + \varepsilon \quad (3.2.4)$$

This tells us something almost obvious: if you want to know the error, by which you can find the true solution, you *have* to solve the BTE. A clever thing we could do here is *approximate* ε , add it to ψ^p , then continue iterating as such. We would no longer maintain the interpretation of ψ^p , that it represents particles which have scattered up to p times, but

despite this we are still able to iterate with the formerly-defined ψ^p added to ε to get closer and closer to ψ . So let's define the following iteration scheme:

$$\boxed{\begin{aligned} \hat{T}\psi^{p+1/2} &= K\psi^p + s \\ r^p &= \hat{K}(\psi^{p+1/2} - \psi^p) \\ (\hat{T} - \hat{K})\varepsilon^p &= r^p \\ \psi^{p+1} &= \psi^{p+1/2} + \varepsilon^p \\ \varepsilon^p &= \frac{|\varphi^p - \varphi^{p-1}|_2}{|\varphi^{p-1}|_2} \end{aligned}} \quad (3.2.5)$$

Now, we would need to conveniently obtain ε to make this procedure worthwhile. This can be achieved with an entire solution that is discretized more *coarsely*. That is, we boot up an entire SI *or* GMRES with ε as ψ and r as s , and with a smaller $N_\mu N_\nu$ and larger convergence criterion. Believe it or not, this can *significantly* accelerate the solution of the original problem. We can approximately map from a coarse set of discrete ordinates to a refined one or vice versa thru a Galerkin quadrature map. For example, suppose that the coarse set of angles are indexed by (i', j') and the refined set by (i, j) . We would say map from refined to coarse with:

$$v_{i'j'} = \sum_q v_q y_{qi'j'} \quad (3.2.6)$$

where:

$$v_q = \sum_{ij} w_i w_j v_{ij} y_{qij} \quad (3.2.7)$$

And then:

$$v_{ij} = \sum_q v_q y_{qij} \quad (3.2.8)$$

where:

$$v_q = \sum_{i'j'} w_{i'} w_{j'} v_{i'j'} y_{qi'j'} \quad (3.2.9)$$

where the sum over q must run to at least $2(N_\mu + 1)$, where N_μ would be the number of polar angles in our refined or coarse discrete ordinates set, respectively. But, we could also obtain a coarse solution to, say, v_q . That is to say, we can not only use two different solution methods to accelerate one another, but we can also use two different discretization methods to do so.

3.2.2 Acceleration of GMRES

In the context of GMRES, we perform acceleration not by estimating a residual, but rather by applying a different solution/discretization method to the vector, say, Lv_0 , at every iteration. That is to say, we wish to consider our GMRES problem not $L\psi = s$, but rather $M^{-1}L\psi = M^{-1}s$, where M is an approximation of L . This 'approximation' we can take to mean a more coarsely discretized L , with the proper coarse-to-refined and refined-to-coarse

transformations implied. For example, suppose we wish to accelerate GMRES with a coarse SI solve. When we apply L throughout the algorithm, we produce, some vector Lv_0 . We then make this coarse and boot up an entire SI solve wherein this is the source, and the solution we seek will be $M^{-1}Lv_0$ after it is made refined. It also goes without saying that we'd have to construct our initial source as $M^{-1}s$. A thorough benchmark

3.2.3 Diffusion synthetic acceleration

We mentioned in **Section 2.4** that one way to form a P_N solution method was to approximate the current density with the corresponding form which follows from diffusion [3, 9]. We can actually approximate the error in SI/pre-condition GMRES with the following approximate BTE operator:

$$\hat{M} = \hat{D} - \hat{K} \quad (3.2.10)$$

where:

$$\hat{D}\psi = -\nabla \cdot C(\mathbf{r}, E)\nabla\psi \quad (3.2.11)$$

SI error would be determined as $\hat{M}\epsilon = r$, and GMRES pre-conditioning as $\hat{M}^{-1}\hat{L}\psi = \hat{M}^{-1}s$. Notably, no sweep is implied by \hat{D} . However, we will not discuss discretization of this operator.

3.3 Approximations

3.3.1 Neglecting the positron fluence and the electron-photon source

Our first two approximations are specific to coupled photon-electron transport problems, for which this solver was originally intended. They modify the coupling between particle types [3]. Furthermore, one is a necessity for wiscobolt, because at the moment positrons are not implemented at all. First, we will treat pair production as a process which produces two electrons rather than one positron and one electron, allowing us to neglect the positron fluence entirely. Though one may suspect that this is wildly unphysical, it is quite reasonable when looking at the behavior of positrons in media (one must only be careful when computing charge deposition). Specifically, like electrons, positrons also deposit their energy through inelastic interactions, while also undergoing tortuous elastic scattering, until they reach very low energies. They do sometimes annihilate sooner, but it is typically at this point that they annihilate. So, we are essentially neglecting the annihilation, which is reasonable for us since the two 0.511 MeV photons emitted by annihilation are not substantial in deposition of energy. Second, we neglect transport of photons that are created by electrons. There are two noteworthy ways for electrons to produce photons in this energy regime: Bremsstrahlung and fluorescence following impact ionization. Now, Bremsstrahlung will still be considered to slow down electrons, it's just that we will never see what Bremsstrahlung photons do after their creation. On the other hand, attenuation associated with fluorescence is due to impact ionization, and so neglecting transport of fluorescence photons following impact ionization means neglecting it entirely. We now have a problem which is coupled only in the electron

fluence:

$$\boxed{\begin{aligned}\hat{L}^\gamma \psi^\gamma &= s^\gamma \\ \hat{L}^e \psi^e &= s^e + s^{\gamma e}\end{aligned}} \quad (3.3.1)$$

where we use the symbol ‘e’ to represent the electron field, rather than e^- , since there is no longer ambiguity with positrons. These equations can be solved sequentially. Note that both equations are technically cast in the form $\hat{L}\psi = s$, with particle-particle scattering included in s , making each a generic BTE. Nevertheless, taking these approximations we now have the schematic:

γ	\rightarrow	γ	Compton effect, fluorescence
		e^-	Compton effect, Auger effect, pair production, photoelectric effect
		e^+	<u>Pair production</u>
e^-	\rightarrow	γ	<u>Bremsstrahlung, fluorescence</u>
		e^-	Elastic scattering, inelastic scattering, Auger effect
		e^+	<u>None</u>
e^+	\rightarrow	γ	<u>Bremsstrahlung, pair annihilation</u>
		e^+	<u>Bhaba scattering (ionization)</u>
		e^+	Elastic scattering, Bhaba scattering (energy loss)

3.3.2 The restricted continuous slowing down approximation

Next, we consider a modification to our scattering treatment for electrons only. We already know we’ll be dealing with elastic electron scattering and inelastic electron scattering (both radiative and collisional). This derivation follows exactly [38]. Now, we will find that some cross sections for inelastic electron scattering exhibit singularities or inaccuracies for arbitrarily low energy transfers, i.e., for $E' = E$. It is therefore important to split the treatment for ‘soft’ and ‘hard’ scattering, which are to be distinguished according to a small cutoff energy Δ (generally around 10 – 20 keV). That is to say, revisit the integrals present in our scattering and attenuation operators, focusing solely on inelastic scattering, and then split the integrals as so:

$$\begin{aligned}\int_E^{E_{\max}} dE' \int d\Omega' \Sigma_s(\mu_s, E' \rightarrow E) \psi(\hat{\mathbf{k}}', E') &= \\ \int_E^{E+\Delta} dE' \int d\Omega' \Sigma_s(\mu_s, E' \rightarrow E) \psi(\hat{\mathbf{k}}', E') + \int_{E+\Delta}^{E_{\max}} dE' \int d\Omega' \Sigma_s(\mu_s, E' \rightarrow E) \psi(\hat{\mathbf{k}}', E')\end{aligned} \quad (3.3.2)$$

for the scattering, and:

$$\begin{aligned}\int_{E_{\min}}^E dE' \int d\Omega_s \Sigma_s(\mu_s, E \rightarrow E') &= \\ \int_{E-\Delta}^E dE' \int d\Omega_s \Sigma_s(\mu_s, E \rightarrow E') + \int_{E_{\min}}^{E-\Delta} dE' \int d\Omega_s \Sigma_s(\mu_s, E \rightarrow E')\end{aligned} \quad (3.3.3)$$

for the attenuation. The soft term is the first term on the RHS in both of the equations above. These are the terms which will be treated differently. First, we will assume that soft scattering is associated with no change in direction. That is to say, we assume that the scattering with $E' \approx E$ is associated with $\mu_s \approx 1$. We will find later that the kinematic restraints present in inelastic electron scattering justify this approximation quite well, provided that we are also assuming that no ionization occurs due to soft collisions. Now, this allows us to rephrase our problem by eliminating all angular integrals. We now define:

$$s_R \equiv \int_E^{E+\Delta} dE' \Sigma_s(E' \rightarrow E) \psi(E') - \psi(E) \int_{E-\Delta}^E dE' \Sigma_s(E \rightarrow E') \quad (3.3.4)$$

Where:

$$\Sigma_s(E' \rightarrow E) = \int d\Omega_s \Sigma_s(\mu_s, E' \rightarrow E) \quad (3.3.5)$$

And we are suppressing $\hat{\mathbf{k}}$ in $\psi(\hat{\mathbf{k}}, E)$. The term s_R is simply $(\hat{K} - \Sigma_t)\psi$ for soft inelastic scattering only. We seek to make a simplification to the cross section's behavior that relies on the fact that $E' - E$ (or $E - E'$ in the second term) will be small. Now, suppose we define:

$$\varepsilon(E' \rightarrow E) \equiv E' - E \quad (3.3.6)$$

Then, we could write the first term in s_R in terms of $\varepsilon(E', E)$, and the second in terms of $\varepsilon(E, E')$. We would then be left with:

$$s_R = \int_0^\Delta d\varepsilon \Sigma_s(E + \varepsilon, E) \psi(E + \varepsilon) - \int_0^\Delta d\varepsilon \Sigma_s(E \rightarrow E - \varepsilon) \psi(E) \quad (3.3.7)$$

Now, since ε in either integrand is just a dummy index, we can merge the integrands:

$$s_R = \int_0^\Delta d\varepsilon [\Sigma_s(E + \varepsilon, E) \psi(E + \varepsilon) - \Sigma_s(E \rightarrow E - \varepsilon) \psi(E)] \quad (3.3.8)$$

A useful expression for the integrand can be given by considering Taylor expansions for E' about E . Specifically, consider the function:

$$f(E', \varepsilon) \equiv \Sigma_s(E' \rightarrow E' - \varepsilon) \psi(E') \quad (3.3.9)$$

Which is just the first integrand in s_R , expressed with different coordinates. This is equivalent to:

$$f[E'(E, \varepsilon), \varepsilon] = \Sigma_s(E + \varepsilon \rightarrow E) \psi(E + \varepsilon) \quad (3.3.10)$$

If $E'(E, \varepsilon) = E + \varepsilon$. Now, let's make the first approximation. Expand $f(E', \varepsilon)$ in E' about E :

$$f(E', \varepsilon) \approx f(E, \varepsilon) + \varepsilon \frac{\partial}{\partial E'} [f(E', \varepsilon)] \bigg|_{E'=E} \quad (3.3.11)$$

The derivative in the second term is not dependent on E' , so we write it to make this fact more clear:

$$f(E', \varepsilon) \approx f(E, \varepsilon) + (E' - E) \frac{\partial}{\partial E} [f(E, \varepsilon)] \quad (3.3.12)$$

Now, this appears to suggest:

$$f(E', \varepsilon) - f(E, \varepsilon) \approx (E' - E) \frac{\partial}{\partial E} [f(E, \varepsilon)] \quad (3.3.13)$$

We can use (3.3.10) for the first term on the LHS and (3.3.9) for the second:

$$\Sigma_s(E + \varepsilon \rightarrow E) \psi(E + \varepsilon) - \Sigma_s(E \rightarrow E - \varepsilon) \psi(E) \approx \varepsilon \frac{\partial}{\partial E} [f(E, \varepsilon)] \quad (3.3.14)$$

The left hand side is our integrand in s_R , so we really have:

$$s_R \approx \int_0^\Delta d\varepsilon \varepsilon \frac{\partial}{\partial E} [f(E, \varepsilon)] \quad (3.3.15)$$

Or we can pull the E derivative outside the integral, since ε is a dummy index:

$$s_R \equiv \frac{\partial}{\partial E} \left[\psi(E) \int_0^\Delta d\varepsilon \varepsilon \Sigma_s(E \rightarrow E - \varepsilon) \right] \quad (3.3.16)$$

We can also transform our integration once again to the variable $E'(E, \varepsilon)$. This yields:

$$s_R = \frac{\partial}{\partial E} \left[\psi(E) \int_{E-\Delta}^E dE' (E - E') \Sigma_s(E \rightarrow E') \right] \quad (3.3.17)$$

We recognize the term in the integral looks a lot like the typical linear energy transfer:

$$L(E) = \int_{E/2}^E dE' (E - E') \Sigma_s(E \rightarrow E') \quad (3.3.18)$$

The only difference is the fact that it only considers interactions of outgoing energy down to $E - \Delta$. So, we call what we have the **restricted linear energy transfer**:

$$L_\Delta(E) \equiv 2\pi \int_{-1}^1 d\mu \int_{E-\Delta}^E dE' (E - E') \Sigma_s(\mu, E \rightarrow E')$$

(3.3.19)

Where we reintroduce angular coordinates to emphasize that we had to integrate over all scattering angles earlier. Finally, we are left with:

$$s_R = \frac{\partial}{\partial E} [L_\Delta(E) \psi(E)] \quad (3.3.20)$$

We define:

$$\hat{R}\psi \equiv \frac{\partial}{\partial E} (L_\Delta \psi) = s_R$$

(3.3.21)

Note that this term has consumed the attenuation term corresponding to soft scattering, and so in Σ_t one only needs to consider hard scattering. This goes for the first collision source method as well. That is, we *once again* change the physical significance of $\tilde{\varphi}^0$ (and we are actually not finished doing so), in preference of its mathematical significance. Suppose

that $\Sigma_{t,\Delta}$ is the linear attenuation coefficient without soft collisions, we construct $\tilde{\varphi}^0$ with this quantity, and not Σ_t . This is to say that we treat \hat{R} , if not as its own unique term, in a manner much more similar to \hat{K} than \hat{T} , despite that \hat{R} is a combination of terms from both. Note, however, that in MGXS exclusively, we will see a bit of a restoration as we will split the \hat{R} operator into a scattering and attenuation coefficient once again.

Nevertheless, we must now discretize \hat{R} . In general, one may approach this term as another operator which maps between energy groups/elements. Since this mapping occurs without change in direction, the angular dependence of this term may either be treated distinctly from how the angular dependence of \hat{K} is treated (more on this in **wiscobolt implementation**), or one may use Legendre moments corresponding to $\delta(\mu - 1)/2\pi$:

$$\frac{1}{2\pi} \delta(\mu - 1) = \sum_{\ell=0}^{\infty} \frac{2\ell + 1}{4\pi} P_{\ell}(\mu) \quad (3.3.22)$$

Truncation of this term to finite order yields very poor approximations. It is quite preferable to simply include another scattering operator term which applies a mapping in energy without mapping in angle (more in this in **Section 3.3.3**). Now, energy discretization must, of course, be treated separately for MGXS and FEXS. The FEXS development is a generic FEM derivation, but credit for this precise formula is due to [38]. We first integrate over the shape function Λ_n^g , and apply integration by parts with $u = \Lambda_n^g$ and $v = L_{\Delta}\psi$. We find:

$$\int_g dE \Lambda_n^g \frac{\partial}{\partial E} (L_{\Delta}\psi) = (\Lambda_n^g L_{\Delta}\psi) \Big|_{\partial(g)} - \int_g dE \frac{\partial \Lambda_n^g}{\partial E} L_{\Delta}\psi \quad (3.3.23)$$

where $\partial(g)$ is the ‘boundary’ of the domain for group g . The reason we don’t simply substitute upper and lower group bounds is to draw a close analogy to the sweep term we discovered when discretizing $\hat{\mathbf{k}} \cdot \nabla$ in the spatial FEM. There, we did essentially a higher dimensional integration by parts using the divergence theorem to produce an area integral over all faces of a spatial element. We supposed for a choice of $\hat{\mathbf{k}}$ that we could know the solution to the integral over an ‘upstream’ face as being derived from the upstream element. So, here we will make a similar suggestion, although it is much simpler to comprehend in 1D. We suppose that we know the solution to $(\Lambda_n^g L_{\Delta}\psi)$ for the ‘upstream energy element,’ i.e., $g - 1$, assuming $g > 1$. Then, $(\Lambda_n^g L_{\Delta}\psi)$ at the higher energy boundary is given by $(\Lambda_n^g L_{\Delta}\psi)(E_2^{g-1})$, i.e., evaluated in $g - 1$ at the lower energy boundary. After that, we say $(\Lambda_n^g L_{\Delta}\psi)$ at the lower energy boundary remains unknown, or $(\Lambda_n^g L_{\Delta}\psi)(E_2^g)$. For $g = 1$, we use a ‘boundary condition’ of ψ for the leading term, which will always be taken as zero since we always choose E_1 as the largest energy present in the source spectrum. Therefore, we are left with:

$$\int_g dE \Lambda_n^g \frac{\partial}{\partial E} (L_{\Delta}\psi) = (1 - \delta_{g,1}) \Lambda_n^g (E_2^{g-1}) L_{\Delta}(E_2^{g-1}) \psi_2^{g-1} - \Lambda_n^g (E_2^g) L_{\Delta}(E_2^g) \psi_2^g - \int_g dE \frac{\partial \Lambda_n^g}{\partial E} L_{\Delta}\psi \quad (3.3.24)$$

The factor $(1 - \delta_{g,1})$ is used to more conveniently enforce the boundary condition, i.e., that $\psi_2^0 = 0$. We use $\Lambda_n^g (E_2^{g-1}) = \Lambda_n^g (E_1^g) = \delta_{n1}$ to write:

$$\int_g dE \Lambda_n^g \frac{\partial}{\partial E} (L_{\Delta}\psi) = (1 - \delta_{g,1}) \delta_{n1} L_{\Delta}(E_2^{g-1}) \psi_2^{g-1} - \delta_{n2} L_{\Delta}(E_2^g) \psi_2^g - \int_g dE \frac{\partial \Lambda_n^g}{\partial E} L_{\Delta}\psi \quad (3.3.25)$$

We can assume that the group $g = 1$ has no ψ^{g-1} term, like taking the boundary condition $\psi(E) = 0$ for $E > E_1^1$. Now, for the remaining integral on the RHS, we expand ψ , to find:

$$\int_g dE \frac{\partial \Lambda_n^g}{\partial E} L_\Delta \psi = \sum_{n'=1}^2 \psi_{n'}^g \left\langle \frac{\partial \Lambda_n^g}{\partial E}, L_\Delta \Lambda_{n'}^g \right\rangle_g \quad (3.3.26)$$

We will now define:

$$\boxed{\left\langle \frac{\partial \Lambda_n^g}{\partial E}, L_\Delta \Lambda_{n'}^g \right\rangle_g \equiv L_{n'n}^g} \quad (3.3.27)$$

For linear shape functions:

$$L_{n'n}^g = \frac{(-1)^{n+n'+1}}{\Delta E_g^2} \int_g dE L_\Delta(E) (E_{3-n'}^g - E) \quad (3.3.28)$$

We finally have the matrix elements of the RCSDA operator:

$$\boxed{R_{n'n}^{g'g} \equiv \delta_{g',g-1} (1 - \delta_{g,1}) \delta_{n1} \delta_{n'2} L_\Delta(E_{n'}^{g'}) - \delta_{g'g} [\delta_{n2} \delta_{n'2} L_\Delta(E_{n'}^{g'}) + L_{n'n}^g]} \quad (3.3.29)$$

Which is written to emphasize the off-diagonal and on-diagonal elements of this operator. This is also prior to applying λ^{-1} . With λ^{-1} we just take $R \rightarrow (\lambda^{-1})R$.

On the other hand, the MGXS derivation is a bit more involved, and even its statement is a bit beyond the scope of this document [16, 43], however, it is described in the aforementioned sources as well as the **wiscobolt physics** document.

3.3.3 The extended transport correction

We lastly consider a modification to our scattering treatment which helps us deal with highly forward-peaked scattering. This is an interesting method that has been understood for a while in the context of elastic electron scattering, providing drastically faster convergence, but it can also be applied to inelastic scattering [16, 42, 43]. For a thorough analysis on its validity when applied to either form of scattering, I recommend the reader visit [43], and try to understand the discussion in Section III ('Moments Analysis'). This paper shows that the ETC very well treats elastic scattering without excessively high Legendre order, and it also somewhat well treats inelastic scattering.

We will first treat elastic scattering. Suppose that we are dealing with a scattering cross section precisely of the form $\Sigma_s(\mu) = C\delta(\mu - 1)/2\pi$, where C is some quantity not dependent on μ . This permits a simpler scattering cross section, as $\mu = 1$ corresponds essentially to $\mathbf{k} = \hat{\mathbf{k}}'$ when integrated over $\hat{\mathbf{k}}'$ (or $\hat{\mathbf{k}}$). But, simple or not, if we attempt to truncate our scattering operator to order L , we will find a generally very poor approximation of $\delta(\mu - 1)$. This is very sensible, because the distribution $\delta(\mu - 1)$ is *arbitrarily* anisotropic, so accurate representation should require *arbitrarily* high orders of Legendre polynomials. Among the side effects of this poor approximation is the littering of negative values which, for L that is not impractically large, can survive into quantities of interest, such as the angular fluence and even its distributions (though, one can demonstrate that these negatives should

not survive into any angular-integrated quantities, like the fluence). Now, we will see in the **wiscobolt physics** document that we will frequently deal with distributions like $\delta(\mu - \bar{\mu})$, for some function of energy $\bar{\mu}$. These kinds of distributions are simply statements of kinematics, derived from conservation of momentum. However, since our entire matrix problem is formulated with respect to various energy and angle integrals, these singular distributions are mostly smeared out, and we will not have to worry about them. This is especially true in the case of photon scattering, wherein Compton scattering is the only scattering mechanism of our interest that has such a constraint. But if $\bar{\mu}$ is a function with weak dependence on energy, such as one that is mostly $\bar{\mu} \approx 1$, corresponding to forward scattering, then these singular distributions, as we have discussed, will be a problem. We will also find exactly $\delta(\mu - 1)$ in the treatment of the aforementioned RCSDA. As for the case of elastic electron scattering, the distributions we find are not kinematically restrained as per $\delta(\mu - \bar{\mu})$, but in any case they are *very* highly forward peaked. It is not appropriate, of course, to suggest that forward peaked scattering *exclusively* occurs. In the case of elastic scattering, forward scattering of course means no scattering at all. Therefore, it is necessary to come up with a way to treat forward peaked scattering without necessitating prohibitively large L . As it turns out, we can actually very efficiently treat forward peaked scattering without arbitrarily large L . Such drastically forward peaked scattering is conceptually simple, so why shouldn't it be computationally simple? We describe the so-called **delta-down approximation** or **extended transport correction (ETC)**. The rationale is as follows: with elastic scattering specifically, define a modified attenuation coefficient $\tilde{\Sigma}_{el}$ that is the typical attenuation coefficient, but without contribution from the most forward-peaked scattering. That is, we will say:

$$\Sigma_{el} \equiv \tilde{\Sigma}_{el} + \frac{1}{2\pi} C \delta(\mu - 1) \quad (3.3.30)$$

Now, what happens to the portion of the total attenuation coefficient belonging to elastic scattering? It becomes:

$$\Sigma_{t(el)} = \tilde{\Sigma}_{t(el)} + C \quad (3.3.31)$$

where:

$$\tilde{\Sigma}_{t(el)} \equiv 4\pi \tilde{\Sigma}_{el,0} \quad (3.3.32)$$

which is just the total attenuation coefficient corresponding to the ‘scattering mechanism’ $\tilde{\Sigma}_{el}$. Here we find a fascinating trick. Let’s now investigate what happens to the terms in the BTE when we use the expressions above for our elastic scattering and attenuation terms:

$$\begin{aligned} \hat{K}_{el}\psi &= \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \frac{4\pi}{2\ell+1} \tilde{\Sigma}_{el,\ell} y_{\ell}^m \int d\Omega' y_{\ell}^m \psi + C\psi \\ \Sigma_{t(el)}\psi &= \tilde{\Sigma}_{t(el)}\psi + C\psi \end{aligned} \quad (3.3.33)$$

Note that the above equations are *not* approximations. One may suggest that (3.3.30) is, at the very least, an assumption, but in fact it is not, as there will always exist a $\tilde{\Sigma}_{el}(\mu, E)$ that can be given by rearrangement. Of course, it could be singular, and a meaningless re-write of the problem when applied without consideration, but it is nevertheless exact. Now, the trick becomes apparent when we lump together the scattering and attenuation terms of the

BTE. That is, consider:

$$\hat{S}_{el}\psi \equiv \Sigma_{t(el)}\psi - \hat{K}_{el}\psi \quad (3.3.34)$$

We can write this out in terms of $\tilde{\Sigma}_{el}$:

$$\begin{aligned} \hat{S}_{el}\psi &= \Sigma_{t(el)}\psi - \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \frac{4\pi}{2\ell+1} \Sigma_{el,\ell} y_{\ell}^m \int d\Omega' y_{\ell}^m \psi \\ &= \tilde{\Sigma}_{t(el)}\psi + C\psi - \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \frac{4\pi}{2\ell+1} \tilde{\Sigma}_{el,\ell} y_{\ell}^m \int d\Omega' y_{\ell}^m \psi - C\psi \end{aligned} \quad (3.3.35)$$

In the last line, we note that $C\psi$ is cancelled out. Thus, we have:

$$\begin{aligned} \hat{S}_{el}\psi &= \Sigma_{t(el)}\psi - \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \frac{4\pi}{2\ell+1} \Sigma_{el,\ell} y_{\ell}^m \int d\Omega' y_{\ell}^m \psi \\ &= \tilde{\Sigma}_{t(el)}\psi - \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \frac{4\pi}{2\ell+1} \tilde{\Sigma}_{el,\ell} y_{\ell}^m \int d\Omega' y_{\ell}^m \psi \end{aligned} \quad (3.3.36)$$

Immediately, we recognize that the problem is perfectly symmetric with respect to swapping the unmodified attenuation coefficients with the modified attenuation coefficients. Intuitively, the term C represents, apparently, scattering which occurs with no energy loss as well as no change in direction. This should *not* be considered scattering in the first place! Yet, notwithstanding that, for all ℓ , the term $\Sigma_t - 4\pi\Sigma_{s,\ell}/(2\ell+1)$ is still identical to $\tilde{\Sigma}_t - 4\pi\tilde{\Sigma}_{s,\ell}/(2\ell+1)$. So how does this make any difference to our problem? Consider for instance SI, where we *don't* ever use \hat{S}_{el} all at once. Instead, we serially apply \hat{T} , which contains Σ_t , then \hat{K} , which contains Σ_s . It is known that the ratios of $4\pi\Sigma_{s,\ell}/(2\ell+1)$ to Σ_t , a.k.a. the ‘scattering ratios,’ dictate the convergence properties of source iteration [37]. That is to say, if our particles are being created by $\Sigma_{s,\ell}$ as rapidly as they are being swallowed by Σ_t , such that the scattering ratios are near unity (they should never exceed it), then that means our particles are *scattering* a lot. As we’ve established, the operator $\hat{T}^{-1}\hat{K}$ applied to some fluence $\tilde{\psi}^p$ determines the share of particles which scatter exactly one more time than p times. So, due to the large number of scattering events which occur, we will have successively $\tilde{\psi}^p$, $\tilde{\psi}^{p+1}$, $\tilde{\psi}^{p+2}$, ..., that are competitive with regard to the total number of particles they describe. If, however, we have determined a kind of mechanism, such as $\tilde{\Sigma}_s$, which has smaller Legendre moments, then we can expect convergence much quicker. I hesitate to call it a *mechanism*, though that’s what it represents. Or, you can think of it like removing the scattering events which have no change in energy nor angle, which are represented by C , and are thus computationally wasteful. In the end, the problem we are trying to solve with $\tilde{\Sigma}_s$ is *exactly the same* as it is with Σ_s , but, the use of the former is essentially a mathematical trick which plays more nicely with our iteration methods. We must then abandon the perspective that $\tilde{\psi}^p$ represents the particles that have scattered p -times (though, energy iteration already nullifies this perspective, and practical storage limits make it uninteresting anyway).

In fact, we will even abandon the notion that $\tilde{\psi}^0$ represents uncollided particles. The reason this is greatly advantageous is that, when we go to construct $\tilde{\varphi}^0$, we are including

a Beer's law term involving the factor $e^{-\Sigma_t \ell}$ to describe the fraction of particles that are uncollided after travelling a distance ℓ in the medium. Now, the distance scale $1/\Sigma_t$ describes how rapidly this exponent goes to zero. Specifically, $1/\Sigma_t$ is the so-called 'mean free path' (MFP) in 1D, and after a handful of MFP's, our Beer's law term reaches very close to zero. Note that the MFP is also the average distance between collisions. Now, even at large energies, the MFP for electrons is somewhere between μm and nm . So, unless we want to use nanometer mesh elements at the entrance of the beam path, we will be vastly undersampling $\tilde{\varphi}^0$ in space, and it will practically be nonzero only for points at the very surface. This leads to *over-estimation* of the number of particles in our problem, given that we generally assume a linear variation of $\tilde{\varphi}^0$ (except for when we construct $\tilde{\varphi}_k^{0,e}$ as per **Section 3.3.4**, but this remains impractical with such large Σ_t). Additionally, it leads to *slightly* slower convergence since less of the work has been done in constructing $\tilde{\varphi}^0$. However, the MFP constructed with $\tilde{\Sigma}_{el}$ is dramatically larger, and these problems are then mostly alleviated by using it to construct $\tilde{\varphi}^0$, which is no longer the 'uncollided' fluence.

On the other hand, what if we use GMRES? In GMRES, our system matrix is either $\hat{T} - \hat{K}$ in the P_N case or $(1 - \hat{T}^{-1}\hat{K})$ in the S_N case, with the corresponding sources s or $\hat{T}^{-1}s$, respectively. Whether we use the modified attenuation coefficients or not, $\hat{T} - \hat{K}$ is unchanged because \hat{S}_{el} is contained wholly within $\hat{T} - \hat{K}$. So, the utility of the ETC in this case for P_N is solely when we are using the first collision scattering source method, because it makes our uncollided particles much less attenuated, and so we'll have a source which will converge faster. But, for S_N , $(1 - \hat{T}^{-1}\hat{K})$ becomes a much better preconditioned operator.

We shall now determine a value for C . We decide that it should enforce that $\tilde{\Sigma}_{el,L+1} = 0$, so that we gain a 'free' extra Legendre order by only using up to L . This means that truncation of the original problem to $L + 1$ is equivalent to truncation of the new problem to L . Setting $\Sigma_{el,L+1} = 0$ yields:

$$C = \frac{4\pi}{2L+3} \Sigma_{el,L+1} \quad (3.3.37)$$

And thus:

$$\tilde{\Sigma}_{el,\ell} = \Sigma_{el,\ell} - \frac{2\ell+1}{2L+3} \Sigma_{el,L+1}$$

(3.3.38)

In the literature describing the ETC you may find different factors for $\Sigma_{el,L+1}$. In fact, it is typically -1 alone, but this is merely a consequence of the different conventions we have used for Legendre expansions.

Now, should this be extended to inelastic (particularly hard collisional) scattering? First, know that inelastic electron collisions come in the form:

$$\Sigma_{inel} = S\delta(\mu - \bar{\mu}) \quad (3.3.39)$$

for some functions of incident and outgoing energy, S and $\bar{\mu}$. The near-universal expression for $\bar{\mu}$ derives from relativistic kinematics, and it's:

$$\bar{\mu}(x \rightarrow y) = \sqrt{\frac{y(x+2)}{x(y+2)}} \quad (3.3.40)$$

where x is the incident energy and y is the outgoing energy, both in units of electron masses. What kind of values suggest $\bar{\mu} \approx 1$? Generally, interactions with low energy transfer (we assumed this in **Section 3.3.2**, we now have the tools to demonstrate this). For instance, suppose we have an incident electron of 1 MeV ($\sim 1.96 m_e c^2$). In **Figure 7**, we plot $\bar{\mu}(1 \text{ MeV} \rightarrow 1 \text{ MeV} - \Delta)$ vs. Δ , where Δ is the energy loss of the incident electron. We have a little bit of breathing room to say $\bar{\mu} \approx 1$. Consider $\Delta = 0.02 \text{ MeV}$, which is a reasonable cutoff energy for the RCSDA. This actually corresponds to $\bar{\mu} = 0.995$, which is far enough from 1 that we can not suggest exactly forward-scattering, but is close enough that we can apply the ETC with worthwhile benefit (this is shown in [43]). At higher energies, we have more breathing room for $\bar{\mu} \approx 1$. Thus, it is reasonable to apply the ETC, and we should expect that removal of a $\delta(\mu - 1)$ component of scattering will meaningfully. However, we note that unlike with elastic scattering, the applying the ETC here does *not* leave the BTE invariant. Additionally, for MGXS, there will be no analogous cancellation for the total attenuation coefficient. That is because in MGXS, hard collisional scattering is considered to be exclusively the interactions which transfer particles beyond the adjacent group. That is:

$$\Sigma_{inel,g \rightarrow g} = \Sigma_{inel,g \rightarrow g+1} = 0 \quad (3.3.41)$$

essentially by definition. The scheme we then have is as simple as:

$$\Sigma_{inel,g' \rightarrow g} = \tilde{\Sigma}_{inel,g' \rightarrow g} + \frac{2}{2L+3} \delta(\mu - 1) \Sigma_{inel,L+1,g' \rightarrow g} \quad (3.3.42)$$

or the analogous equation for FEXS, and this gets applied to the BTE straightforwardly.

On the other hand, the ‘diagonal’ elements of the RCSDA operator are, by default, moved to the attenuation coefficients. This should generally happen whether or not the RCSDA operator’s angular dependence is treated exactly (i.e., by essentially lumping it in with $\Sigma_{inel,L+1,g' \rightarrow g}$, up to some constants), or inexactly (i.e., by expanding $\delta(\mu - 1)$ in terms of Legendre polynomials). To do this is essentially to decide to apply the ETC only to the diagonal elements of the RCSDA, but not to the off-diagonal elements, which is acceptable.

Lastly, we will once again make a modification to our ‘uncollided’ fluence in the FCS method. Since we have some $\delta(\mu - 1)$ terms, from either the inelastic ETC, the RCSDA, or both, we can say that our uncollided fluence should now refer to particles which have not *changed direction*, rather than saying that they’re particles which have not changed both energy and direction. This becomes an intuitive approximation if you write out $\hat{L}(\tilde{\psi} + \tilde{\psi}^0) = s$ with our newly defined $\delta(\mu - 1)$ terms. We therefore take:

$$\hat{\mathbf{k}} \cdot \nabla \tilde{\psi}_g^0 + \tilde{\Sigma}_{t,g} \tilde{\psi}_g^0 = s_g + \sum_{g'=1}^{g-1} C_{g'g} \tilde{\psi}_{g'}^0 \quad (3.3.43)$$

To see why this matters, write out the BTE with respect to an uncollided fluence defined as such. Essentially, this says we don’t need to use $C_{g'g}$ when constructing \tilde{s}_g . So we don’t quite take $\tilde{s}_g = \hat{K} \tilde{\psi}^0$, instead we take:

$$\tilde{s}_g \rightarrow \sum_{\ell=0}^L \sum_{m=-\ell}^{\ell} \sum_{g'=1}^g \frac{4\pi}{2\ell+1} y_{\ell}^m \tilde{\Sigma}_{\ell,g'g} \tilde{\psi}_{g'}^0 \quad (3.3.44)$$

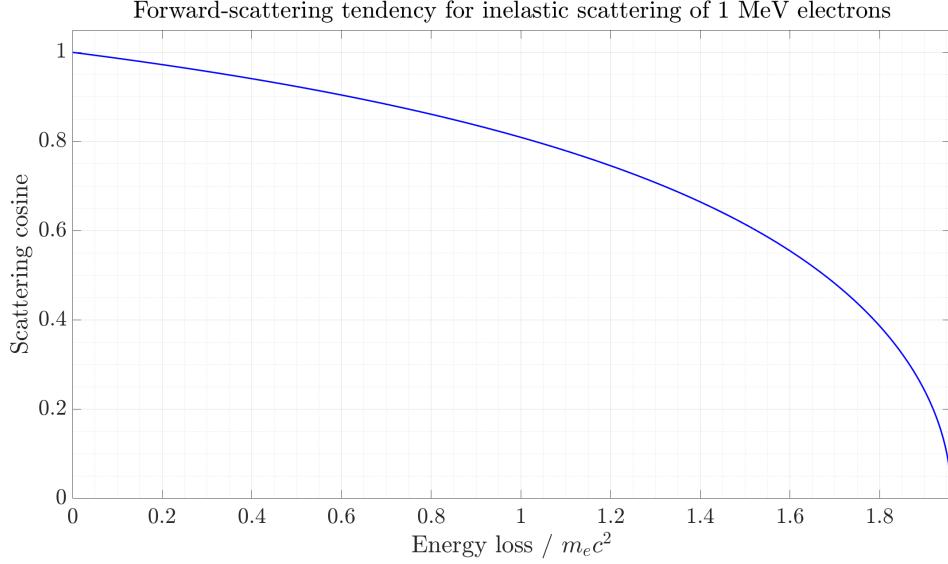


Figure 7: Inelastic scattering angle cosine vs. energy loss for electrons of 1 MeV. That this is ≈ 1 for low energy loss indicates that ‘soft’ collisions tend to forward scatter. That this is ≈ 0 for energy loss on the order of 1 MeV (or ~ 1.96 electron masses) indicates that ‘hard’ or ‘catastrophic’ collisions are not quite as forward-peaked. Nevertheless, the decrease from 1 to 0 is not very steep, so inelastic scattering will generally exhibit rather forward-peaked scattering, at least for 1 MeV incidence.

which differs from the original definition of \tilde{s}_g because it doesn’t involve the $C_{g'g}$ term which is attached to $\Sigma_{\ell,g'g}$.

We can construct $\hat{T}^{-1}s_g$ using typical methods, i.e., ray tracing [47–49]. However, one must consider the full $\tilde{\psi}_g^0$ to be obtained from $\hat{T}^{-1}s_g + \hat{T}^{-1}\sum_{g'} C_{g'g}\tilde{\psi}_{g'}^0$, where the second term *must* be obtained by sweeping $\sum_{g'} C_{g'g}\tilde{\psi}_{g'}^0$ with zero boundary condition. Since we typically obtain $\hat{T}^{-1}s_g$ via the old definition of $\tilde{\varphi}_g^0$, acknowledging that the accompanying angular distribution is singular, we may wonder what can be said to simplify (or forego) the problem of sweeping. Since our particles $\tilde{\psi}_g^0$ constitute particles which have not changed their direction, we can say with confidence that their angular distribution is singular. However, it is still most practical perform a sweep, and we do so with the angular distribution given by the discretization of $\delta^2[\hat{\mathbf{k}} - \hat{\mathbf{k}}_0(\mathbf{r})]$ in terms of spherical harmonics. At the end, we may still carry only $\tilde{\varphi}_g^0$ by integrating over angle the quantity $\tilde{\psi}_g^0$.

3.3.4 Integration of uncollided fluence

Here, we discuss the formation of \tilde{s}_k^e as described in **Section 2.2.3**.

First, we defined much earlier:

$$\tilde{s}_k^e = \sum_{k'=1}^{N_K^e} (I_1^{-1})_{kk'}^e \int_{V^e} d^3\mathbf{r} u_{k'}^e(\mathbf{r}) \tilde{s}(\mathbf{r}) \quad (3.3.45)$$

We saw that, regardless of energy and angular discretization, we can relate the above elements directly to what we will now define as:

$$\tilde{\varphi}_k^{0,e} \equiv \sum_{k'=1}^{N_K^e} (I_1^{-1})_{kk'}^e \int_{V^e} d^3\mathbf{r} u_{k'}^e(\mathbf{r}) \tilde{\varphi}^0(\mathbf{r}) \quad (3.3.46)$$

Now, the question becomes – how do we perform this integral efficiently using only values of $\tilde{\varphi}^0(\mathbf{r})$ at pre-determined points? The answer is quadrature. Specifically, within a linear tetrahedron, we can perform ‘ N –point’ quadrature to yield exact solutions to 3D polynomials of a particular order, similar to 1D Gauss-Legendre quadrature. For instance, a four-point quadrature rule yields exact integration to polynomials of order up to 3 in any dimension [20]. Let the nodes for such a quadrature rule be expressed in the linear tetrahedron as \mathbf{c}_k . The weights are w_k . We then transform our integral to the linear tetrahedron using:

$$\int_{V^e} d^3\mathbf{r} u_k^e(\mathbf{r}) \tilde{\varphi}^0(\mathbf{r}) = 6V^e \int_{\text{lin-tet}} d^3\mathbf{r}' u_k(\mathbf{r}') \tilde{\varphi}^0[\mathbf{r}^e(\mathbf{r}')] \quad (3.3.47)$$

which follows directly from **Section 2.2.4**, using notation from there as well. It is most important to restate:

$$\mathbf{r}^e(\mathbf{r}') = \sum_{k=1}^{N_K^e} u_k(\mathbf{r}') \mathbf{r}_k^e \quad (3.3.48)$$

which is, of course, valid within the tetrahedron in question e . Now, the integral in the linear tetrahedron follows from quadrature:

$$\int_{\text{lin-tet}} d^3\mathbf{r}' u_k(\mathbf{r}') \tilde{\varphi}^0[\mathbf{r}^e(\mathbf{r}')] \approx \sum_{p=1}^N w_p u_k(\mathbf{c}_p) \tilde{\varphi}^0[\mathbf{r}^e(\mathbf{c}_p)] \quad (3.3.49)$$

In matrix multiplication form, we can write:

$$T_{kp} \equiv w_p u_k(\mathbf{c}_p) \quad (3.3.50)$$

$$\tilde{\Phi}_p^{0,e} \equiv \tilde{\varphi}^0[\mathbf{r}^e(\mathbf{c}_p)] \quad (3.3.51)$$

then:

$$\int_{\text{lin-tet}} d^3\mathbf{r}' u_k(\mathbf{r}') \tilde{\varphi}^0[\mathbf{r}^e(\mathbf{r}')] \approx \sum_{p=1}^N T_{kp} \tilde{\Phi}_p^{0,e} \quad (3.3.52)$$

where we emphasize that the elements T_{pk} have no dependence on any specified element. Note that the quantities $\tilde{\Phi}_p^{0,e}$ are relatively easy to obtain once one decides on the entire set of points $\mathbf{r}(\mathbf{c}_p)$ at which one wishes to evaluate $\tilde{\varphi}^0$. We outlined the construction of $\tilde{\varphi}^0(\mathbf{r})$ in **Section 1.2**, and the most difficult factor in this quantity is always going to be the optical path length from the source to \mathbf{r} , $\tau(\mathbf{r}, \mathbf{r}_0)$, except when one has an internal source *and* a homogeneous medium. Otherwise, it can be obtained via ray-tracing, which is described in **wiscobolt implementation** as well as in [47]. Nevertheless, here the final vector element we obtain is:

$$\tilde{\varphi}_k^{0,e} = 6V^e \sum_{k'=1}^{N_K^e} (I_1^{-1})_{kk'}^e \sum_{p=1}^N T_{k'p} \tilde{\Phi}_p^{0,e} \quad (3.3.53)$$

Now, the above is straightforward, but can still be computationally expensive. Additionally, it is hardly necessary for every element. It is only worthwhile for shallow elements, i.e., those elements close to the surface facing the beam, over which $\tilde{\varphi}$ exponentially decays with decay constant $1/\Sigma_t$. Thus, wiscobolt uses this formulation for $\tilde{\varphi}_k^{0,e}$ only for electrons *and* when a mesh element passes a certain criterion relating to the optical path length across the entire element. The energy group that is chosen to evaluate the optical path length is the one which is most represented in the source's energy spectrum. The criterion is that the minimum optical path length in the element is less than $1/2$ while the maximum optical path length is greater than $1/2$. This selects elements which are simultaneously shallow, but also bigger than we'd like them to be if they are meant to sample $e^{-\tau}$. Note that the percent error between an integral of e^{-x} and an integral of the linearly interpolated form of e^{-x} between 0 and $1/2$ is around 2% . So, elements which will only provide an error around 2% (while not being very close to zero anyway) are accepted, but elements which would exceed this error are treated with quadrature.

A Angular inner products

We seek to express the angular inner products, $\mathbf{A}_{\ell\ell'}^{mm'}$, $\mathbf{A}_{\uparrow,\ell\ell'}^{mm'}$, and $\mathbf{A}_{\downarrow,\ell\ell'}^{mm'}$, from **Section 2.4**. This section will soon be split into a subsection wherein we describe the *analytical* solutions to the integrals with which we are dealing, as well as one wherein we describe the *numerical* solutions, i.e., via quadrature. The analytical solutions are a bit complicated but nonetheless fascinating, and one would hope, can be implemented optimally to make the construction of the relevant arrays during implementation rather fast (despite how many quantities must be constructed – a lot, so many that storage is a significant issue here). As of this version of wiscobolt, however, the quadrature solutions are much faster (and, frankly, the analytical solutions have not been validated at all, so take their derivation with a mountain of salt). They are still rather slow. So, implementation of these angular inner products is very much a work-in-progress.

Nevertheless, both the analytical and numerical solutions follow from a common, simpler analytical derivation. First, note that each integral will have an integrand involving:

$$k_x y_\ell^m y_{\ell'}^{m'} = \cos \phi \sqrt{1 - \mu^2} y_\ell^m y_{\ell'}^{m'} \quad (\text{A.0.1})$$

$$k_y y_\ell^m y_{\ell'}^{m'} = \sin \phi \sqrt{1 - \mu^2} y_\ell^m y_{\ell'}^{m'} \quad (\text{A.0.2})$$

$$k_z y_\ell^m y_{\ell'}^{m'} = \mu y_\ell^m y_{\ell'}^{m'} \quad (\text{A.0.3})$$

Our approach will be to express a given $k_i y_\ell^m$ in terms of a superposition of other y_k^n . Analytically, this is the first step to a solution, but numerically, this allows us to drastically reduce the number of integrals we are really performing. Additionally, it can be done rather simply.

Now, we will first suppose that we have found the expansion coefficients, i.e., we have found:

$$k_i y_\ell^m = \sum_{\ell'=0}^{\infty} \sum_{m'=-\ell'}^{\ell'} \kappa_{i,\ell\ell'}^{mm'} y_{\ell'}^{m'} \quad (\text{A.0.4})$$

Then, if we attempt to form the whole-sphere inner products $\mathbf{A}_{\ell\ell'}^{mm'}$ immediately by multiplying through with $y_{\ell'}^{m'}$ and integrating, we have:

$$A_{i,\ell\ell'}^{mm'} = \sum_{\ell'',m''} \kappa_{i,\ell\ell'}^{mm''} \int d\Omega y_{\ell'}^{m'} y_{\ell''}^{m''} = \kappa_{i,\ell\ell'}^{mm'} \quad (\text{A.0.5})$$

Therefore, the expansion coefficients *are* our whole-sphere inner products. That is to say:

$$\hat{\mathbf{k}}y_{\ell}^m = \sum_{\ell',m'} \mathbf{A}_{\ell\ell'}^{mm'} y_{\ell'}^{m'}$$

(A.0.6)

This expansion will be used to simplify determination of the half-sphere integrals shortly. First, we'll determine $\mathbf{A}_{\ell\ell'}^{mm'}$.

To begin, we will first return to the definition:

$$y_{\ell}^m(\mu, \phi) = C_{\ell}^m S^m(\phi) P_{\ell}^m(\mu) \quad (\text{A.0.7})$$

We recognize that we can actually express terms like $\cos \phi S^m(\phi)$ and $\sin \phi S^m(\phi)$ in terms of a superposition of $S^n(\phi)$. We can *also* express $\sqrt{1 - \mu^2} P_{\ell}^m$ and μP_{ℓ}^m in terms of superpositions of P_k^n using recursion relations. Then, we will be able to match up terms in the product of these expansions and relate them to distinct y_k^n . Following that, we can exploit orthonormality and other properties of the spherical harmonics in order to perform the necessary integrals. The associated Legendre polynomial recursion relations we will need are [25]:

$$\mu P_{\ell}^m = \frac{1}{2\ell + 1} [(\ell - m + 1) P_{\ell+1}^m + (\ell + m) P_{\ell-1}^m] \quad (\text{A.0.8})$$

$$\sqrt{1 - \mu^2} P_{\ell}^m = \frac{1}{2\ell + 1} [(\ell - m + 1)(\ell - m + 2) P_{\ell+1}^{m-1} - (\ell + m - 1)(\ell + m) P_{\ell-1}^{m-1}] \quad (\text{A.0.9})$$

$$\sqrt{1 - \mu^2} P_{\ell}^m = \frac{1}{2\ell + 1} (P_{\ell-1}^{m+1} - P_{\ell+1}^{m+1}) \quad (\text{A.0.10})$$

It goes without saying that you should only use valid pairs of (ℓ, m) , but if one puts some (ℓ, m) that invokes a polynomial with an invalid pair (ℓ, m) , that polynomial just goes to zero. Notice as well that the first relation relates polynomials of the same m and some ℓ to $\ell \pm 1$, while the second relates polynomials of some m and ℓ to those with $m - 1$ and $\ell \pm 1$, and the last relates polynomials of some m and ℓ to those with $m + 1$ and $\ell \pm 1$. Now, we must express $\cos \phi S^m(\phi)$ as a sum of $S^k(\phi)$. Let's look for:

$$\cos \phi S^m(\phi) = \sum_{k=-\infty}^{\infty} c_{mk} S^k(\phi) \quad (\text{A.0.11})$$

and similarly for $\sin \phi S^m(\phi)$. How do we find an expression for c_{mk} ? One may suggest approaching this like a Fourier expansion or other basis expansion, where if we can find a convenient function with which to integrate both sides, we can express c_{mk} easily, then sum

over $c_{mk}S^k$ for a convenient expression for $\cos\phi S^m$. And, we could do a similar thing with $\sin\phi S^m$. We'll first check if $S^n(\phi)$ itself permits a convenient expansion:

$$\int_0^{2\pi} d\phi S^n(\phi) S^k(\phi) = \pi(1 + \delta_{n0})\delta_{nk} \quad (\text{A.0.12})$$

which is convenient enough that we can then say:

$$c_{mn} = \frac{1}{\pi(1 + \delta_{n0})} \int_0^{2\pi} d\phi S^n(\phi) \cos\phi S^m(\phi) \quad (\text{A.0.13})$$

So what is the integral on the RHS? We can find:

$$\int_0^{2\pi} d\phi S^n(\phi) \cos\phi S^m(\phi) = \begin{cases} \frac{1}{2}\pi\delta_{|n-m|,1}(1 + \delta_{|n+m|,1}), & \text{sign}(n) = \text{sign}(m), \\ 0, & \text{else} \end{cases} \quad (\text{A.0.14})$$

so finally:

$$c_{mn} = \begin{cases} \frac{1}{2}\delta_{|n-m|,1}\frac{1+\delta_{|n+m|,1}}{1+\delta_{n0}}, & \text{sign}(n) = \text{sign}(m), \\ 0, & \text{else} \end{cases} \quad (\text{A.0.15})$$

It can be demonstrated through a lot of tedious work that this is equivalent to a simpler expression:

$$c_{mn} = \frac{1}{2}(1 + \delta_{m0} - \delta_{m,-1})\delta_{n,m+1} + \frac{1}{2}(1 - \delta_{m0})\delta_{n,m-1} \quad (\text{A.0.16})$$

which I found by supposing that I could write c_{mn} in the form $a_m\delta_{n,m+1} + b_m\delta_{n,m-1}$, an intuition I gained from the presence of $\delta_{|n\pm m|,1}$. If you are not convinced, you can check this expression by constructing a matrix with these elements (I did this to $n, m \sim 100$, the elements are identical at least that far). Nevertheless, we are left with:

$$\cos\phi S^m = \frac{1}{2}(1 + \delta_{m0} - \delta_{m,-1})S^{m+1} + \frac{1}{2}(1 - \delta_{m0})S^{m-1} \quad (\text{A.0.17})$$

Now, the case of $\sin\phi S^m$ is almost as tedious, except for the ability to use integration by parts to get the integral of $S^n \sin\phi S^m$ from our result for the integral of $S^n \cos\phi S^m$. Nevertheless, the result will simply be provided:

$$\sin\phi S^m = \frac{1}{2}(1 + \delta_{m0})S^{-(m+1)} - \frac{1}{2}(1 - \delta_{m0} - \delta_{m1})S^{-(m-1)} \quad (\text{A.0.18})$$

We will define:

$$\begin{aligned} A_{c,m} &\equiv \frac{1}{2}(1 + \delta_{m0} - \delta_{m,-1}) \\ B_{c,m} &\equiv \frac{1}{2}(1 - \delta_{m0}) \\ A_{s,m} &\equiv \frac{1}{2}(1 + \delta_{m0}) \\ B_{s,m} &\equiv -\frac{1}{2}(1 - \delta_{m0} - \delta_{m1}) \end{aligned} \quad (\text{A.0.19})$$

So now let's see how to use these. We will expand each $k_i y_\ell^m$ and evaluate its whole-sphere integral immediately, then later return and discuss the half-sphere integrals. We'll start trying to work on the x -component. We first consider:

$$\cos \phi \sqrt{1 - \mu^2} y_\ell^m = C_\ell^m \sqrt{1 - \mu^2} P_\ell^m (A_{c,m} S^{m+1} + B_{c,m} S^{m-1}) \quad (\text{A.0.20})$$

For the term $\sqrt{1 - \mu^2} P_\ell^m S^{m+1}$, we use the recursion relation which involves $P_{\ell \pm 1}^{m+1}$, i.e., (A.0.10). That is to say, we take:

$$\sqrt{1 - \mu^2} P_\ell^m S^{m+1} = \frac{1}{2\ell + 1} (P_{\ell-1}^{m+1} S^{m+1} - P_{\ell+1}^{m+1} S^{m+1}) \quad (\text{A.0.21})$$

and for the term $\sqrt{1 - \mu^2} P_\ell^m S^{m-1}$, we use (A.0.9):

$$\begin{aligned} \sqrt{1 - \mu^2} P_\ell^m S^{m-1} &= \frac{1}{2\ell + 1} [(\ell - m + 1)(\ell - m + 2) P_{\ell+1}^{m-1} S^{m-1} \\ &\quad - (\ell + m - 1)(\ell + m) P_{\ell-1}^{m-1} S^{m-1}] \end{aligned} \quad (\text{A.0.22})$$

We then leverage:

$$P_k^n S^n = \frac{y_k^n}{C_k^n} \quad (\text{A.0.23})$$

so that we can finally write:

$$\begin{aligned} \cos \phi \sqrt{1 - \mu^2} y_\ell^m &= \frac{C_\ell^m}{2\ell + 1} \left[\frac{A_{c,m}}{C_{\ell-1}^{m+1}} y_{\ell-1}^{m+1} \right. \\ &\quad - \frac{A_{c,m}}{C_{\ell+1}^{m+1}} y_{\ell+1}^{m+1} \\ &\quad + \frac{B_{c,m}}{C_{\ell+1}^{m-1}} (\ell - m + 1)(\ell - m + 2) y_{\ell+1}^{m-1} \\ &\quad \left. - \frac{B_{c,m}}{C_{\ell-1}^{m-1}} (\ell + m - 1)(\ell + m) y_{\ell-1}^{m-1} \right] \end{aligned} \quad (\text{A.0.24})$$

For the integral over the whole sphere, we go to integrate this with $y_{\ell'}^{m'}$, then we utilize the orthonormality relationship for the real spherical harmonics. Then:

$$\begin{aligned} A_{\ell\ell',x}^{mm'} &= \frac{C_\ell^m}{2\ell + 1} \left[\frac{A_{c,m}}{C_{\ell-1}^{m+1}} \delta_{\ell-1,\ell'}^{m+1,m'} \right. \\ &\quad - \frac{A_{c,m}}{C_{\ell+1}^{m+1}} \delta_{\ell+1,\ell'}^{m+1,m'} \\ &\quad + \frac{B_{c,m}}{C_{\ell+1}^{m-1}} (\ell - m + 1)(\ell - m + 2) \delta_{\ell+1,\ell'}^{m-1,m'} \\ &\quad \left. - \frac{B_{c,m}}{C_{\ell-1}^{m-1}} (\ell + m - 1)(\ell + m) \delta_{\ell-1,\ell'}^{m-1,m'} \right] \end{aligned} \quad (\text{A.0.25})$$

Which is quite sparse. Now, the y -component is just a bit more difficult. We have in the integrand:

$$\sin \phi \sqrt{1 - \mu^2} y_\ell^m = C_\ell^m \sqrt{1 - \mu^2} P_\ell^m [A_{s,m} S^{-(m+1)} + B_{s,m} S^{-(m-1)}] \quad (\text{A.0.26})$$

Now, the term with, for instance, $S^{-(m-1)}$ gets what recursion relation? We will use the relation with $P_{\ell \pm 1}^{m-1}$, but, we will relate $P_{\ell \pm 1}^{m-1}$ to $P_{\ell \pm 1}^{-(m-1)}$. The relationship we need is:

$$P_\ell^m = (-1)^m \frac{(\ell + m)!}{(\ell - m)!} P_\ell^{-m} \quad (\text{A.0.27})$$

Now first take $m \rightarrow m - 1$ and $\ell \rightarrow \ell \pm 1$:

$$P_{\ell \pm 1}^{m-1} = (-1)^{m-1} \frac{(\ell \pm 1 + m - 1)!}{(\ell \pm 1 - m + 1)!} P_{\ell \pm 1}^{-(m-1)} \quad (\text{A.0.28})$$

so what we are saying is:

$$\begin{aligned} \sqrt{1 - \mu^2} P_\ell^m S^{-(m-1)} &= \frac{(-1)^{m-1}}{2\ell + 1} \left[\frac{(\ell - m + 1)(\ell - m + 2)(\ell + m)!}{(\ell - m + 2)!} P_{\ell+1}^{-(m-1)} S^{-(m-1)} \right. \\ &\quad \left. - \frac{(\ell + m - 1)(\ell + m)(\ell + m - 2)!}{(\ell - m)!} P_{\ell-1}^{-(m-1)} S^{-(m-1)} \right] \end{aligned} \quad (\text{A.0.29})$$

Then the $S^{-(m+1)}$ term needs us to know:

$$P_{\ell \pm 1}^{m+1} = (-1)^{m-1} \frac{(\ell \pm 1 + m + 1)!}{(\ell \pm 1 - m - 1)!} P_{\ell \pm 1}^{-(m+1)} \quad (\text{A.0.30})$$

And then:

$$\begin{aligned} \sqrt{1 - \mu^2} P_\ell^m S^{-(m+1)} &= \frac{(-1)^{m-1}}{2\ell + 1} \left[\frac{(\ell + m)!}{(\ell - m - 2)!} P_{\ell-1}^{-(m+1)} S^{-(m+1)} \right. \\ &\quad \left. - \frac{(\ell + m + 2)!}{(\ell - m)!} P_{\ell+1}^{-(m+1)} S^{-(m+1)} \right] \end{aligned} \quad (\text{A.0.31})$$

We could then put these together to have:

$$\begin{aligned} \sin \phi \sqrt{1 - \mu^2} y_\ell^m &= \frac{(-1)^{m-1} C_\ell^m}{2\ell + 1} \left[\frac{A_{s,m}}{C_{\ell-1}^{-(m+1)}} \frac{(\ell + m)!}{(\ell - m - 2)!} y_{\ell-1}^{-(m+1)} \right. \\ &\quad - \frac{A_{s,m}}{C_{\ell+1}^{-(m+1)}} \frac{(\ell + m + 2)!}{(\ell - m)!} y_{\ell+1}^{-(m+1)} \\ &\quad + \frac{B_{s,m}}{C_{\ell+1}^{-(m-1)}} \frac{(\ell - m + 1)(\ell - m + 2)(\ell + m)!}{(\ell - m + 2)!} y_{\ell+1}^{-(m-1)} \\ &\quad \left. - \frac{B_{s,m}}{C_{\ell-1}^{-(m-1)}} \frac{(\ell + m - 1)(\ell + m)(\ell + m - 2)!}{(\ell - m)!} y_{\ell-1}^{-(m-1)} \right] \end{aligned} \quad (\text{A.0.32})$$

which integrates to:

$$\begin{aligned}
A_{\ell\ell',y}^{mm'} = & \frac{(-1)^{m-1} C_\ell^m}{2\ell+1} \left[\frac{A_{s,m}}{C_{\ell-1}^{-(m+1)}} \frac{(\ell+m)!}{(\ell-m-2)!} \delta_{\ell-1,\ell'}^{-(m+1),m'} \right. \\
& - \frac{A_{s,m}}{C_{\ell+1}^{-(m+1)}} \frac{(\ell+m+2)!}{(\ell-m)!} \delta_{\ell+1,\ell'}^{-(m+1),m'} \\
& + \frac{B_{s,m}}{C_{\ell+1}^{-(m-1)}} \frac{(\ell-m+1)(\ell-m+2)(\ell+m)!}{(\ell-m+2)!} \delta_{\ell+1,\ell'}^{-(m-1),m'} \\
& \left. - \frac{B_{s,m}}{C_{\ell-1}^{-(m-1)}} \frac{(\ell+m-1)(\ell+m)(\ell+m-2)!}{(\ell-m)!} \delta_{\ell-1,\ell'}^{-(m-1),m'} \right]
\end{aligned} \tag{A.0.33}$$

The easiest is done last. $A_{\ell\ell',z}^{mm'}$ does not involve shifting m values. So we just have:

$$\mu y_\ell^m = \frac{C_\ell^m}{2\ell+1} \left(\frac{\ell-m+1}{C_{\ell+1}^m} y_{\ell+1}^m + \frac{\ell+m}{C_{\ell-1}^m} y_{\ell-1}^m \right) \tag{A.0.34}$$

Then:

$$A_{\ell\ell',z}^{mm'} = \frac{C_\ell^m}{2\ell+1} \left(\frac{\ell-m+1}{C_{\ell+1}^m} \delta_{\ell+1,\ell'}^{mm'} + \frac{\ell+m}{C_{\ell-1}^m} \delta_{\ell-1,\ell'}^{mm'} \right) \tag{A.0.35}$$

So, we are done with the whole-sphere integrals.

How would we treat the half-sphere integrals? Notice that each integral has been fully reduced to:

$$\int_{\pm \hat{\mathbf{k}} \cdot \hat{\mathbf{n}} > 0} d\Omega k_i y_\ell^m y_{\ell'}^{m'} = \sum_{n,k} A_{i,\ell k}^{mn} \int_{\pm \hat{\mathbf{k}} \cdot \hat{\mathbf{n}} > 0} d\Omega y_k^n y_{\ell'}^{m'} \tag{A.0.36}$$

We are therefore only concerned with integrating $y_k^n y_\ell^m$ over the half-spheres. We are going to distinguish the set of integrals we must evaluate into four categories:

$$\mathcal{I}_{k\ell,\pm}^{nm}(\hat{\mathbf{n}}) \equiv \int_{\pm \hat{\mathbf{k}} \cdot \hat{\mathbf{n}} > 0} d\Omega y_k^n(\hat{\mathbf{k}}) y_\ell^m(\hat{\mathbf{k}}) \tag{A.0.37}$$

1. Integral over $\hat{\mathbf{k}} \cdot \hat{\mathbf{n}} > 0$, $k + \ell$ is even.
2. Integral over $\hat{\mathbf{k}} \cdot \hat{\mathbf{n}} < 0$, $k + \ell$ is even.
3. Integral over $\hat{\mathbf{k}} \cdot \hat{\mathbf{n}} > 0$, $k + \ell$ is odd.
4. Integral over $\hat{\mathbf{k}} \cdot \hat{\mathbf{n}} < 0$, $k + \ell$ is odd.

We can evaluate some of these outright utilizing the parity of these harmonics. In particular, we first note that:

$$\int_{\hat{\mathbf{k}} \cdot \hat{\mathbf{n}} > 0} d\Omega y_k^n(\hat{\mathbf{k}}) y_\ell^m(\hat{\mathbf{k}}) = \int_{-\hat{\mathbf{k}} \cdot \hat{\mathbf{n}} > 0} d\Omega y_k^n(-\hat{\mathbf{k}}) y_\ell^m(-\hat{\mathbf{k}}) \tag{A.0.38}$$

We are saying that integrating $f(\hat{\mathbf{k}})$ over some half-sphere is identical to integrating $f(-\hat{\mathbf{k}})$ over the opposite side of the sphere. However, we have a special $f(-\hat{\mathbf{k}})$, i.e., we can relate the two sides of the equation above in another manner:

$$y_k^n(\hat{\mathbf{k}})y_\ell^m(\hat{\mathbf{k}}) = (-1)^{k+\ell}y_k^n(-\hat{\mathbf{k}})y_\ell^m(-\hat{\mathbf{k}}) \quad (\text{A.0.39})$$

Thus:

$$\int_{\hat{\mathbf{k}} \cdot \hat{\mathbf{n}} > 0} d\Omega y_k^n(\hat{\mathbf{k}})y_\ell^m(\hat{\mathbf{k}}) = (-1)^{k+\ell} \int_{-\hat{\mathbf{k}} \cdot \hat{\mathbf{n}} > 0} d\Omega y_k^n(\hat{\mathbf{k}})y_\ell^m(\hat{\mathbf{k}}) \quad (\text{A.0.40})$$

Thus, the half integral for $\hat{\mathbf{k}} \cdot \hat{\mathbf{n}} > 0$ is equal to that over $\hat{\mathbf{k}} \cdot \hat{\mathbf{n}} < 0$ up to a factor of $(-1)^{k+\ell}$. This is useful when considering the orthonormality of the whole-sphere integrals of $y_k^n y_\ell^m$, written as:

$$\int_{\hat{\mathbf{k}} \cdot \hat{\mathbf{n}} > 0} d\Omega y_k^n(\hat{\mathbf{k}})y_\ell^m(\hat{\mathbf{k}}) + \int_{-\hat{\mathbf{k}} \cdot \hat{\mathbf{n}} > 0} d\Omega y_k^n(\hat{\mathbf{k}})y_\ell^m(\hat{\mathbf{k}}) = \delta_{k\ell}^{nm} \quad (\text{A.0.41})$$

Suppose that $k + \ell$ is even. Then, the two integrals are identical, and because of the orthonormality condition, they are equal to:

$$\int_{\pm \hat{\mathbf{k}} \cdot \hat{\mathbf{n}} > 0} d\Omega y_k^n(\hat{\mathbf{k}})y_\ell^m(\hat{\mathbf{k}}) = \frac{1}{2}\delta_{k\ell}^{nm}, \quad k + \ell = \text{even} \quad (\text{A.0.42})$$

Thus, we have completely evaluated categories 1 and 2. If, however, $k + \ell$ is odd, then the two integrals are of opposite sign. Since odd $k + \ell$ means $k \neq \ell$, we have $\delta_{k\ell} = 0$ according to the orthonormality condition. But this provides no new information. Thus, we *can not* immediately state the result of either integral. All we know is that they sum to zero. We can therefore only yet say:

$$\int_{\hat{\mathbf{k}} \cdot \hat{\mathbf{n}} > 0} d\Omega y_k^n(\hat{\mathbf{k}})y_\ell^m(\hat{\mathbf{k}}) = - \int_{-\hat{\mathbf{k}} \cdot \hat{\mathbf{n}} > 0} d\Omega y_k^n(\hat{\mathbf{k}})y_\ell^m(\hat{\mathbf{k}}), \quad k + \ell = \text{odd} \quad (\text{A.0.43})$$

But, knowing this, we can seek to integrate either term above given $\hat{\mathbf{n}}$. Thus, if we know category 3, we know category 4, and vice versa. None of this actually simplifies the *general* case that we will ultimately be able to know how to solve, however, it's stated because the process of actually solving these integrals is computationally expensive, and this discussion allows us to minimize the actual number of integrals we eventually calculate to only those with $\hat{\mathbf{k}} \cdot \hat{\mathbf{n}} > 0$ and $\ell + k = \text{even}$.

A.1 Analytical solution

What follows is a method that I have yet to implement, and a derivation which I have not seen elsewhere. For this reason, the derivation should be taken with a grain of salt. However, if validated and implemented, it may serve as a far more efficient method of generating angular inner products than the quadrature solution we will discuss in the following section.

Let's pick the $\hat{\mathbf{k}} \cdot \hat{\mathbf{n}} > 0$ term to solve (category 3). Our rationale will be as follows: we don't wish to perform an entire integral for every normal vector in our mesh. Instead, we will transform our integrals to run exclusively over the northern hemisphere, over which we

can analytically integrate the real spherical harmonics. Now, rotation of complex spherical harmonics is well understood. Thus, we will prefer to perform the angular inner products in terms of complex spherical harmonics, then perform a change of basis to the real spherical harmonics. That is, both the complex and real spherical harmonics span the unit sphere. They are related as:

$$y_\ell^m(\hat{\mathbf{k}}) = \sum_{m'=-m,0,m} \chi_{mm'}^\ell Y_\ell^{m'}(\hat{\mathbf{k}}) \quad (\text{A.1.1})$$

for the set of complex coefficients $\chi_{mm'}^\ell$ given by direct comparison of y_ℓ^m and Y_ℓ^m . That is:

$$\chi_{mm'} = \begin{cases} \frac{1}{\sqrt{2}} [\delta_{m',-m} + (-1)^m \delta_{m'm}], & m > 0 \\ \delta_{m'0}, & m = 0 \\ \frac{i}{\sqrt{2}} [\delta_{m'm} - (-1)^m \delta_{m',-m}], & m < 0 \end{cases} \quad (\text{A.1.2})$$

Thus, the problem can be phrased in terms of complex spherical harmonics, using:

$$\int_{\hat{\mathbf{k}} \cdot \hat{\mathbf{n}} > 0} d\Omega y_k^n(\hat{\mathbf{k}}) y_\ell^m(\hat{\mathbf{k}}) = \sum_{m'=-m,0,m} \sum_{n'=-n,0,n} \chi_{nn'}^{k*} \chi_{mm'}^\ell \int_{\hat{\mathbf{k}} \cdot \hat{\mathbf{n}} > 0} d\Omega Y_k^{n'*}(\hat{\mathbf{k}}) Y_\ell^{m'}(\hat{\mathbf{k}}) \quad (\text{A.1.3})$$

where we've conventionally chosen to take the complex conjugate of one of the real spherical harmonics (since y_k^n is real, this is fine, but produces complex conjugated coefficients $\chi_{nn'}^k$ and complex spherical harmonics $Y_k^{n'}$). Now, the problem is phrased in terms of integrals of complex spherical harmonics. We wish to rotate these. First, we must describe conventions we are using. We decide that the matrix R will rotate the normal vector $\hat{\mathbf{n}}$ to $\hat{\mathbf{z}}$:

$$R\hat{\mathbf{n}} = \hat{\mathbf{z}} \quad (\text{A.1.4})$$

We are integrating over vectors $\hat{\mathbf{k}}$ which are pointing in a hemisphere (particularly, $\hat{\mathbf{k}} \cdot \hat{\mathbf{n}} > 0$) about $\hat{\mathbf{n}}$, but when we apply R , these vectors are all rotated to the northern hemisphere. So we will define:

$$R\hat{\mathbf{k}} \equiv \hat{\mathbf{k}}' \quad (\text{A.1.5})$$

so that our integrals transform like:

$$\int_{\hat{\mathbf{k}} \cdot \hat{\mathbf{n}} > 0} d\Omega f(\hat{\mathbf{k}}) = \int_0^{2\pi} d\phi' \int_0^1 d\mu' f(R^{-1}\hat{\mathbf{k}}') \quad (\text{A.1.6})$$

(The determinant of the Jacobian of a rotation is one). So now we briefly ask, what is $Y_\ell^m(R^{-1}\hat{\mathbf{k}}')$? It can be demonstrated [15] that these can be expanded in terms of spherical harmonics of the same order ℓ , with the so-called **Wigner D-matrix** elements as the coefficients:

$$Y_\ell^m(R^{-1}\hat{\mathbf{k}}') = \sum_{m'=-\ell}^{\ell} D_{m'm}^\ell(R) Y_\ell^{m'}(\hat{\mathbf{k}}') \quad (\text{A.1.7})$$

where $R(\hat{\mathbf{n}})$ is determined as the rotation required to bring $\hat{\mathbf{n}}$ to $\hat{\mathbf{z}}$, and $D_{m'm}^\ell(R)$ is the corresponding set of Wigner D-matrix elements. We will discuss their construction in explicit terms of $\hat{\mathbf{n}}$ later. First, recognize that our problem now comes down to:

$$\int_{\hat{\mathbf{k}} \cdot \hat{\mathbf{n}} > 0} d\Omega Y_k^{n'*}(\hat{\mathbf{k}}) Y_\ell^m(\hat{\mathbf{k}}) = \sum_{n'=-k}^k \sum_{m'=-\ell}^{\ell} D_{n'n}^{k*} D_{m'm}^\ell \int_0^{2\pi} d\phi \int_0^1 d\mu Y_k^{n'*}(\hat{\mathbf{k}}) Y_\ell^{m'}(\hat{\mathbf{k}}) \quad (\text{A.1.8})$$

so we have kicked the can down the road until finally, we need only to evaluate the integrals:

$$Q_{k\ell}^{nm} \equiv \int_0^{2\pi} d\phi \int_0^1 d\mu Y_k^{n*}(\hat{\mathbf{k}}) Y_\ell^m(\hat{\mathbf{k}}) \quad (\text{A.1.9})$$

This is a very doable task. The ϕ integral, which can be separated, simplifies the problem substantially:

$$\int_0^{2\pi} d\phi e^{-in\phi} e^{im\phi} = \int_0^{2\pi} d\phi e^{i(m-n)\phi} \quad (\text{A.1.10})$$

Recognize that if $m - n$ is zero, this is just 2π . However, if $m - n$ is anything but zero, this integral corresponds to summing together all vectors lying on a unit circle, or asking the center of ‘mass’ or a circle, which is zero. Thus:

$$\int_0^{2\pi} d\phi e^{i(m-n)\phi} = 2\pi \delta_{mn} \quad (\text{A.1.11})$$

This is great, because now we need only to evaluate the μ integral when $m = n$. Now, these integrals are known for μ spanning from -1 to 1 :

$$\int_{-1}^1 d\mu P_\ell^m(\mu) P_k^m(\mu) = \frac{2(\ell+m)!}{(2\ell+1)(\ell-m)!} \delta_{k,\ell}, \quad m \geq 0 \quad (\text{A.1.12})$$

The case for $m < 0$ is doable using the identity relating P_ℓ^{-m} to P_ℓ^m , which we wrote in (A.0.27). What can we say for this integral with our bounds? First, recognize that if $\ell + k$ is an even number, then the integrand is even over μ , and we would have the integral be half of the above:

$$\int_0^1 d\mu P_\ell^m(\mu) P_k^m(\mu) = \frac{(\ell+m)!}{(2\ell+1)(\ell-m)!} \delta_{k,\ell}, \quad \ell + k = \text{even} \quad (\text{A.1.13})$$

Note that, while we are only looking for $\mathcal{I}_{k\ell,+}^{nm}$ for $\ell + k = \text{even}$, we need the integrals above for the general case, by virtue of the many expansions we made. The case for $\ell + k = \text{odd}$ is not too difficult. Let’s again focus on $m \geq 0$, which can be generalized later.

There could be a better way to do this, but the best way I can come up with is to expand $P_\ell^m P_k^m$ as a polynomial. We know we *can* expand it as a polynomial, despite that P_ℓ^m and P_k^m are themselves not generally polynomials. Specifically, write:

$$P_\ell^m(\mu) = (-1)^m 2^\ell (1 - \mu^2)^{m/2} \sum_{\ell'=m}^{\ell} \binom{\ell}{\ell'} \binom{(\ell + \ell' - 1)/2}{\ell} \frac{\ell'!}{(\ell' - m)!} \mu^{\ell' - m} \quad (\text{A.1.14})$$

Due to the factor $(1 - \mu^2)^{m/2}$, this is not a polynomial for odd m . However, we can now write:

$$\begin{aligned} P_k^m(\mu) P_\ell^m(\mu) &= 2^\ell (1 - \mu^2)^m \sum_{\ell'=m}^{\ell} \sum_{k'=m}^k \times \\ &\quad \binom{\ell}{\ell'} \binom{(\ell + \ell' - 1)/2}{\ell} \binom{k}{k'} \binom{(k + k' - 1)/2}{k} \frac{k'!}{(k' - m)!} \mu^{\ell' - m + k' - m} \end{aligned} \quad (\text{A.1.15})$$

which, with another cumbersome expansion of $(1 - \mu^2)^m$, is always a polynomial. We use the binomial theorem:

$$(1 - \mu^2)^m = \sum_{m'=0}^m \binom{m}{m'} (-1)^{m'} \mu^{2m'} \quad (\text{A.1.16})$$

Thus, in total, we have:

$$\begin{aligned} P_k^m(\mu) P_\ell^m &= 2^\ell \sum_{m'=0}^m \sum_{\ell'=m}^{\ell} \sum_{k'=m}^k \times \\ &(-1)^{m'} \binom{m}{m'} \binom{\ell}{\ell'} \binom{(\ell + \ell' - 1)/2}{\ell} \binom{k}{k'} \binom{(k + k' - 1)/2}{k} \frac{k'!}{(k' - m)!} \mu^{\ell' + k' - 2m + 2m'} \end{aligned} \quad (\text{A.1.17})$$

which, of course, can be analytically integrated.

$$\begin{aligned} \int_0^1 d\mu P_k^m(\mu) P_\ell^m &= 2^\ell \sum_{m'=0}^m \sum_{\ell'=m}^{\ell} \sum_{k'=m}^k \times \\ &\frac{(-1)^{m'}}{\ell' + k' - 2m + 2m' + 1} \binom{m}{m'} \binom{\ell}{\ell'} \binom{(\ell + \ell' - 1)/2}{\ell} \binom{k}{k'} \binom{(k + k' - 1)/2}{k} \frac{k'!}{(k' - m)!} \end{aligned} \quad (\text{A.1.18})$$

It's a big, ugly, inefficient equation, but since this needs to be calculated only once, this is fine.

Thus, we have, for $m \geq 0$, and $\ell + k = \text{even}$:

$$\int_0^1 d\mu P_\ell^m(\mu) P_k^m(\mu) = \frac{(\ell + m)!}{(2\ell + 1)(\ell - m)!} \delta_{k,\ell} \quad (\text{A.1.19})$$

and for $m \geq 0$, $\ell + k = \text{odd}$:

$$\begin{aligned} \int_0^1 d\mu P_k^m(\mu) P_\ell^m &= 2^\ell \sum_{m'=0}^m \sum_{\ell'=m}^{\ell} \sum_{k'=m}^k \times \\ &\frac{(-1)^{m'}}{\ell' + k' - 2m + 2m' + 1} \binom{m}{m'} \binom{\ell}{\ell'} \binom{(\ell + \ell' - 1)/2}{\ell} \binom{k}{k'} \binom{(k + k' - 1)/2}{k} \frac{k'!}{(k' - m)!} \end{aligned} \quad (\text{A.1.20})$$

As for $m < 0$, we use (A.0.27) with $m \rightarrow |m|$. Since $P_\ell^{|m|}$ satisfies $|m| > 0$, we can use the previous results. Thus, if $m < 0$, we perform the integral with $|m|$ and apply a factor:

$$F_{k\ell}^m \equiv \frac{(\ell - |m|)! (k - |m|)!}{(\ell + |m|)! (k + |m|)!} \quad (\text{A.1.21})$$

Now that the integrals $Q_{k\ell}^{nm}$ are determined, we just need to discuss the Wigner D-matrix elements. Written exactly like (A.1.7), we can multiply both sides by $Y_\ell^{m'*}(\hat{\mathbf{k}}')$ and integrate, yielding:

$$D_{m'm}^\ell(R) = \int d\Omega' Y_\ell^{m'*}(\hat{\mathbf{k}}') Y_\ell^m(R^{-1}\hat{\mathbf{k}}') \quad (\text{A.1.22})$$

We will provide $D_{m'm}^\ell$ without derivation, given some rotation, after we describe the rotation R in terms of the normal vector we are evaluating. Let's define an operator $\mathcal{D}(R)$ which rotates functions in the manner we desire. We will find this easiest to understand by looking at a simpler problem similar to ours. Say we have some polar plot $f(\theta)$, which we want to integrate over the semicircle whose north pole is θ_0 . If the unit vector describing the north pole of the semicircle is $(\cos \theta_0, \sin \theta_0) \equiv \hat{\mathbf{n}}$, then we want a rotation by angle α such that:

$$R(\alpha)\hat{\mathbf{n}} = \hat{\mathbf{y}} \quad (\text{A.1.23})$$

So then, if one evaluates $f(\mathbf{r})$, they will find it identical to $f(R^{-1}\mathbf{r}')$, and in the new coordinate system, one can integrate over $[0, \pi]$. We can easily see that, since $\hat{\mathbf{r}} = (\cos \theta, \sin \theta)$, the desired rotation angle leads us to perform:

$$R(\alpha)(\cos \theta, \sin \theta) = (\cos(\theta + \alpha), \sin(\theta + \alpha)) \equiv (\cos \theta', \sin \theta') \quad (\text{A.1.24})$$

This means that the relationship between the new system θ' and the old system θ is $\theta = \theta' - \alpha$. But, in this approach, we are looking at a rotation as a coordinate transformation. If we want to *actively* rotate $f(\theta)$ and let it sit in the same coordinate system, then we define a rotation *operator*, which does not leave an equality like the above, but instead:

$$\mathcal{D}[R(\alpha)]f(\theta) = f(\theta - \alpha) \quad (\text{A.1.25})$$

where we recognize that taking $\theta - \alpha$ is like applying $R^{-1}\hat{\mathbf{r}}$, since $R\hat{\mathbf{r}}$ moves θ to $\theta + \alpha$. We generalize now:

$$\mathcal{D}(R)f(\hat{\mathbf{k}}) \equiv f(R^{-1}\hat{\mathbf{k}}) \quad (\text{A.1.26})$$

Thus:

$$D_{m'm}^\ell(R) = \int d\Omega' Y_\ell^{m'*}(\hat{\mathbf{k}}') \mathcal{D}(R) Y_\ell^m(\hat{\mathbf{k}}') \quad (\text{A.1.27})$$

So, the Wigner D-matrix is the matrix formed by the matrix elements of the rotation operator. The purpose of this discussion is so that we can have the exact same convention for \mathcal{D} and R that is frequently used in the literature to define the Wigner D-matrix elements $D_{m'm}^\ell$. It is too involved for us to describe the form of the rotation operator $\mathcal{D}(R)$, but it must be noted that R is typically written with respect to **Euler angles** (α, β, γ) in the *xyz*-convention. This is useful because the effect of a polar rotation, such as that implemented by some $\mathcal{D}(\alpha, 0, 0)$ or $\mathcal{D}(0, 0, \gamma)$, on a spherical harmonic is well-known to only affect the phase $e^{im\phi}$. If these angles are successfully determined by that rotation which maps $\hat{\mathbf{n}}$ to $\hat{\mathbf{z}}$, then one can write the Wigner D-matrices as:

$$D_{m'm}^\ell(R) = e^{-im'\alpha} d_{m'm}^\ell(\beta) e^{-im\gamma} \quad (\text{A.1.28})$$

where $d_{m'm}^\ell(\beta)$ are the elements of the **Wigner d-matrix**, which can be given by:

$$\begin{aligned} d_{m'm}^\ell(\beta) &= \sqrt{(\ell + m')!(\ell - m')!(\ell + m)!(\ell - m)!} \times \\ &\sum_{s=s_{\min}}^{s_{\max}} \frac{(-1)^{m' - m + s} \cos(\beta/2)^{2\ell + m - m' - 2s} \sin(\beta/2)^{m' - m + 2s}}{(\ell + m - s)! (m' - m + s)! (\ell - m' - s)!} \end{aligned} \quad (\text{A.1.29})$$

where $s_{\min} = \max[0, m - m']$ and $s_{\max} = \min[\ell + m, \ell - m']$.

How do we determine zyz Euler angles for the rotation $R\hat{\mathbf{n}} = \hat{\mathbf{z}}$? This can be tricky to think through, so we will first outline the operation(s) we mean to perform, then discuss a geometrically simple way to perform this. Our rotation matrix is of the form:

$$R(\alpha, \beta, \gamma) = R_z(\alpha)R_y(\beta)R_z(\gamma) \quad (\text{A.1.30})$$

Where:

$$R_z(x) = \begin{pmatrix} \cos x & -\sin x & 0 \\ \sin x & \cos x & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{A.1.31})$$

$$R_y(x) = \begin{pmatrix} \cos x & 0 & \sin x \\ 0 & 1 & 0 \\ -\sin x & 0 & \cos x \end{pmatrix} \quad (\text{A.1.32})$$

It is not trivial that these are Euler rotations, because they appear to be a somewhat mundane sequence of rotations. However, a deep discussion on Euler angles is not warranted here, and we will work with these. These matrices perform counterclockwise rotations about the given axis of the vector to which they are applied, with ‘counterclockwise’ defined via the right hand rule. Now, how do we determine α , β , and γ ? We can consider $R_z(\gamma)$ to bring the vector $\hat{\mathbf{n}}$ to, say, the xz -plane, i.e., to have $\phi = 0$. Thus, it is just:

$$\gamma(\hat{\mathbf{n}}) \equiv -\phi_0(\hat{\mathbf{n}}) = -\text{atan2}(n_y, n_x) \quad (\text{A.1.33})$$

where $\text{atan2}(y, x)$ is a special function given in certain programming languages, that returns $\phi_0 \in [-\pi, \pi]$ given the y -component and x -component of a vector. It is fine to use this range of ϕ_0 because, to transform it to our customary range $\phi \in [0, 2\pi)$, we would simply add 2π to any angles that are negative. Doing so leaves $\sin \phi$, $\cos \phi$, and $e^{i\phi}$, the only arguments with which we are right now concerned, completely unchanged. We can then consider $R_y(\beta)$ to bring the resulting vector to $\hat{\mathbf{z}}$. Thus, it is just:

$$\beta(\hat{\mathbf{n}}) \equiv -\theta_0(\hat{\mathbf{n}}) = -\text{acos}(n_z) \quad (\text{A.1.34})$$

and we then take $\alpha = 0$ as a matter of convenience.

A.1.1 Summary

We now summarize the results of this section. What we ultimately have is a sequence of matrix multiplications, with only one set of matrices being dependent on the mesh normal vectors. We have broken the problem down as follows:

$$\mathcal{I}_{k\ell,+}^{nm} = \sum_{m'=-m,0,m} \sum_{n'=-n,0,n} \chi_{nn'}^{k*} \chi_{mm'}^{\ell} I_{k\ell}^{n'm'} \quad (\text{A.1.35})$$

where:

$$\chi_{mm'} = \begin{cases} \frac{1}{\sqrt{2}} [\delta_{m',-m} + (-1)^m \delta_{m'm}], & m > 0 \\ \delta_{m'0}, & m = 0 \\ \frac{i}{\sqrt{2}} [\delta_{m'm} - (-1)^m \delta_{m',-m}], & m < 0 \end{cases} \quad (\text{A.1.36})$$

and the integrals $I_{k\ell}^{nm}$ are given by:

$$I_{k\ell}^{nm} = \sum_{m'=-\ell}^{\ell} \sum_{n'=-k}^k D_{n'n}^{k*}(\hat{\mathbf{n}}) D_{m'm}^{\ell}(\hat{\mathbf{n}}) Q_{k\ell}^{n'm'} \quad (\text{A.1.37})$$

where:

$$D_{m'm}^{\ell}(\hat{\mathbf{n}}) = e^{-im'\alpha} d_{m'm}^{\ell}(\beta) e^{-im\gamma} \quad (\text{A.1.38})$$

$$\alpha(\hat{\mathbf{n}}) = 0 \quad (\text{A.1.39})$$

$$\beta(\hat{\mathbf{n}}) = -\cos(n_z) \quad (\text{A.1.39})$$

$$\gamma(\hat{\mathbf{n}}) = -\text{atan2}(n_y, n_x)$$

$$d_{m'm}^{\ell}(\beta) = \sqrt{(\ell+m')!(\ell-m')!(\ell+m)!(\ell-m)!} \times \sum_{s=s_{\min}}^{s_{\max}} \frac{(-1)^{m'-m+s} \cos(\beta/2)^{2\ell+m-m'-2s} \sin(\beta/2)^{m'-m+2s}}{(\ell+m-s)!s!(m'-m+s)!(\ell-m'-s)!} \quad (\text{A.1.40})$$

where $s_{\min} = \max[0, m - m']$ and $s_{\max} = \min[\ell + m, \ell - m']$. Then, the integrals $Q_{k\ell}^{nm}$ are given by:

$$Q_{k\ell}^{nm} = 2\pi\delta_{mn} \begin{cases} \mathcal{Q}_{k\ell}^m, & m \geq 0 \\ F_{k\ell}^m \mathcal{Q}_{k\ell}^{|m|}, & m < 0 \end{cases} \quad (\text{A.1.41})$$

where:

$$F_{k\ell}^m = \frac{(\ell - |m|)!}{(\ell + |m|)!} \frac{(k - |m|)!}{(k + |m|)!} \quad (\text{A.1.42})$$

and:

$$\mathcal{Q}_{k\ell}^m = \frac{(\ell + m)!}{(2\ell + 1)(\ell - m)!} \delta_{k,\ell} \quad (\text{A.1.43})$$

if $\ell + k$ is even and:

$$\begin{aligned} \mathcal{Q}_{k\ell}^m &= 2^{\ell} \sum_{m'=0}^m \sum_{\ell'=m}^{\ell} \sum_{k'=m}^k \times \\ &\quad \frac{(-1)^{m'}}{\ell' + k' - 2m + 2m' + 1} \binom{m}{m'} \binom{\ell}{\ell'} \binom{(\ell + \ell' - 1)/2}{\ell} \binom{k}{k'} \binom{(k + k' - 1)/2}{k} \frac{k'!}{(k' - m)!} \end{aligned} \quad (\text{A.1.44})$$

if $\ell + k$ is odd. The relationship of the integrals $\mathcal{I}_{k\ell,+}^{nm}$ to the integrals $\mathbf{A}_{\uparrow,\ell\ell'}^{mm'}$ and $\mathbf{A}_{\downarrow,\ell\ell'}^{mm'}$ was described in the previous section, of course. The integrals $\mathcal{I}_{k\ell,+}^{nm}$ need only be evaluated this way when $\ell + k$ is odd. Otherwise, they are simply $\delta_{k\ell}^{nm}/2$. Note also that we can make use of more identities, for instance that \mathcal{I} is symmetric when one swaps (k, n) with (m, ℓ) , etc.

A.2 Quadrature solution

An alternative is to formulate the integral problems $\mathcal{I}_{k\ell,+}^{nm}(\hat{\mathbf{n}})$ by precisely expressing the domain given by $\hat{\mathbf{k}} \cdot \hat{\mathbf{n}} > 0$. We know that this is a hemisphere whose pole is $\hat{\mathbf{n}}$. So, how

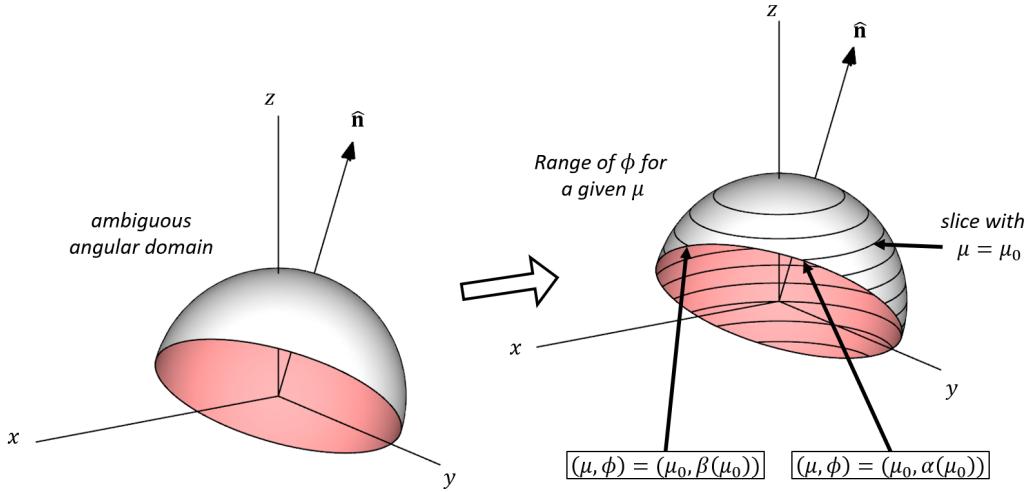


Figure 8: One method to treat integrals over an angular domain such as $\hat{\mathbf{k}} \cdot \hat{\mathbf{n}} > 0$. Slice the region at an arbitrary value of μ , and consider ϕ to range from $\alpha(\mu)$ to $\beta(\mu)$. Then, integrate over these ϕ , and finally integrate over μ .

do we specify the μ and ϕ ranges that allow us to integrate over this hemisphere? Consider that, at each polar angle θ , the hemisphere can be travelled along ϕ with a certain range, corresponding to a slice at the z -value given by θ . This is demonstrated in **Figure 8**. For a given μ and $\hat{\mathbf{n}}$, we will call the minimal ϕ value $\alpha(\mu, \hat{\mathbf{n}})$, and the maximal ϕ value $\beta(\mu, \hat{\mathbf{n}})$. Surprisingly simple trigonometry can lead to the following complicated expressions:

$$\alpha(\mu, \hat{\mathbf{n}}) = \begin{cases} \phi_0(\hat{\mathbf{n}}) - \frac{\pi}{2} + \text{sign}(n_z)\frac{\pi}{2}, & -1 \leq \mu \leq -\sqrt{1 - n_z^2} \\ \phi_0(\hat{\mathbf{n}}) - \frac{\pi}{2} - \arcsin\left[\frac{\mu n_z}{\sqrt{(1-\mu^2)(1-n_z^2)}}\right], & -\sqrt{1 - n_z^2} < \mu < \sqrt{1 - n_z^2} \\ \phi_0(\hat{\mathbf{n}}) - \frac{\pi}{2} - \text{sign}(n_z)\frac{\pi}{2}, & \sqrt{1 - n_z^2} \leq \mu \leq 1 \end{cases} \quad (\text{A.2.1})$$

$$\beta(\mu, \hat{\mathbf{n}}) = \begin{cases} \phi_0(\hat{\mathbf{n}}) + \frac{\pi}{2} - \text{sign}(n_z)\frac{\pi}{2}, & -1 \leq \mu \leq -\sqrt{1 - n_z^2} \\ \phi_0(\hat{\mathbf{n}}) + \frac{\pi}{2} + \arcsin\left[\frac{\mu n_z}{\sqrt{(1-\mu^2)(1-n_z^2)}}\right], & -\sqrt{1 - n_z^2} \leq \mu \leq \sqrt{1 - n_z^2} \\ \phi_0(\hat{\mathbf{n}}) + \frac{\pi}{2} + \text{sign}(n_z)\frac{\pi}{2}, & \sqrt{1 - n_z^2} \leq \mu \leq 1 \end{cases}$$

which you can verify graphically if you wish. With these in hand, we can write our integrals as:

$$\mathcal{I}_{k\ell+}^{nm}(\hat{\mathbf{n}}) = \int_{-1}^1 d\mu \int_{\alpha(\mu, \hat{\mathbf{n}})}^{\beta(\mu, \hat{\mathbf{n}})} d\phi y_k^n(\hat{\mathbf{k}}) y_\ell^m(\hat{\mathbf{k}}) \quad (\text{A.2.2})$$

Here, quadrature could be performed somewhat conveniently. The scheme is:

$$\phi(\mu, v, \hat{\mathbf{n}}) \equiv \frac{1}{2} [\beta(\mu, \hat{\mathbf{n}}) - \alpha(\mu, \hat{\mathbf{n}})] v + \frac{1}{2} [\beta(\mu, \hat{\mathbf{n}}) + \alpha(\mu, \hat{\mathbf{n}})] \quad (\text{A.2.3})$$

Therefore:

$$\mathcal{I}_{k\ell+}^{nm}(\hat{\mathbf{n}}) = \frac{1}{2} C_k^n C_\ell^m \times \int_{-1}^1 d\mu P_k^n(\mu) P_\ell^m(\mu) [\beta(\mu, \hat{\mathbf{n}}) - \alpha(\mu, \hat{\mathbf{n}})] \int_{-1}^1 dv S^n[\phi(\mu, v, \hat{\mathbf{n}})] S^m[\phi(\mu, v, \hat{\mathbf{n}})]$$

(A.2.4)

While this approach may be conceptually more simple, quadrature is only approximate, and rather slow. One may suspect that the exact formulation can be implemented to be dramatically faster. That said, storage of the necessary quantities $\hat{\mathbf{n}} \cdot \mathbf{A}_{\uparrow,qq'}$ and $\hat{\mathbf{n}} \cdot \mathbf{A}_{\downarrow,qq'}$ represents a bit of a problem. It is described in the **wiscobolt implementation** document how wiscobolt creates and stores these arrays in order to lessen the burden of storage. Briefly, it is recommended that they are stored as constructed for ‘global’ faces. That is, every face in the interior of the mesh actually corresponds to two faces; one owned by each of the two elements that share the face. The normal vectors of these faces have opposite sign. Thus, one can say that \mathbf{A}_\uparrow of the face referenced in element e is identical to \mathbf{A}_\downarrow of the face referenced in $e'(e, f)$. If the ‘right’ sign can be stored separately, then one can get by with only storing either \mathbf{A}_\uparrow or \mathbf{A}_\downarrow , and only for the global faces. This lessens the burden of storage by almost a factor of 4 (I find it’s closer to 3, it depends on how many boundary faces there are). A factor of two is due to only carrying either \mathbf{A}_\uparrow or \mathbf{A}_\downarrow , a factor of almost 2 is due to only carrying one element per global face. Finally, wiscobolt carries $\hat{\mathbf{n}} \cdot \mathbf{A}_\uparrow$ instead of just \mathbf{A}_\uparrow , for a few reasons. First, that is how it is needed everywhere in the problem. Second, the scalar product will reduce the number of entries required by a factor of three. Third, it will prevent the need to form the scalar product during iteration, which is very wasteful. At the same time, $\hat{\mathbf{n}}$ is indexed by face, so the quantity $\hat{\mathbf{n}} \cdot \mathbf{A}_\uparrow$ does not pickup any more indices in comparison to \mathbf{A}_\uparrow .

Future updates of wiscobolt will hopefully involve better strategies.

References

- [1] E. B. Podgorsak, *Radiation physics for medical physicists* (Springer, 2010).
- [2] N. Carron, *An introduction to the passage of energetic particles through matter* (CRC Press, 2007).
- [3] J. Battista, *Introduction to megavoltage x-ray dose computation algorithms* (CRC Press, 2019).
- [4] O. Vassiliev, *Monte carlo methods for radiation transport* (Springer, 2017).
- [5] F. Salvat, in PENELOPE-2018: a code system for monte carlo simulation of electron and photon transport (2019).
- [6] Geant4 Collaboration, *Physics reference manual, release 10.4*, tech. rep. (2017).
- [7] S. A. Zein, M.-C. Bordage, Z. Francis, G. Macetti, A. Genoni, C. D. Cappello, W.-G. Shin, and S. Incerti, “Electron transport in DNA bases: An extension of the Geant4-DNA Monte Carlo toolkit”, Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms (2021).
- [8] C. Lanczos, *The variational principles of mechanics* (Dover Publications, 1986).
- [9] C. Demazière, *Modelling of nuclear reactor multi-physics* (Academic Press, 2020).
- [10] H. Fehske, R. Schneider, and A. Weiße, *Computational many-particle physics* (Springer, 2007).
- [11] L. Weih, A. Gabbana, D. Simeoni, L. Rezzolla, S. Succi, and R. Tripiccione, “Beyond moments: relativistic lattice Boltzmann methods for radiative transport in computational astrophysics”, Monthly Notices of the Royal Astronomical Society (2020).
- [12] J. W. Wilson, S. Y. Chun, F. F. Badavi, L. W. Townsend, and S. L. Lamkin, *HZETRN: a heavy ion/nucleon transport code for space radiations*, tech. rep. (NASA, 1991).
- [13] E. M. Purcell and D. J. Morin, *Electricity and magnetism* (Cambridge, 2013).
- [14] L. Landau and E. Lifshitz, *Mechanics* (Pergamon Press, 1969).
- [15] J. Sakurai and J. Napolitano, *Modern quantum mechanics* (Cambridge University Press, 2021).
- [16] L. Lorence, J. Morel, and G. Valdez, *Physics guide to CEPXS: a multigroup coupled electron-photon cross-section generating code*, tech. rep. (Sandia National Labs, 1989).
- [17] J. L. Bedford, “Calculation of absorbed dose in radiotherapy by solution of the linear Boltzmann transport equations”, Physics in Medicine and Biology (2019).
- [18] C. Drumm, *Spherical harmonics (PN) methods in the SCEPTRE radiation transport code*, 2015.
- [19] L. Bourhrara, “A new numerical method for solving the Boltzmann transport equation using the PN method and the discontinuous finite elements on unstructured and curved meshes”, Journal of Computational Physics (2019).
- [20] T. Wareing, J. Morel, and D. Parsons, *A first collision source method for ATTILA, an unstructured tetrahedral mesh discrete ordinates code*, 1998.

- [21] C. Drumm and W. Fan, “Multi-level acceleration of scattering-source iterations with application to electron transport”, Nuclear Engineering and Technology (2017).
- [22] J. Morel and J. McGhee, “A self-adjoint angular flux equation”, Nuclear Science and Engineering (1999).
- [23] D. J. Griffiths, *Introduction to electrodynamics* (Pearson Education, 2013).
- [24] S. Perkins, D. Cullen, M. Chen, J. Hubbell, J. Rathkopf, and J. Scofield, *Tables and graphs of atomic subshell and relaxation data derived from the LLNL Evaluated Atomic Data Library (EADL), Z = 1-100*, tech. rep. (Lawrence Livermore National Laboratory, 1991).
- [25] M. L. Boas, *Mathematical methods in the physical sciences* (Wiley, 2006).
- [26] M. A. Blanco, M. Florez, and M. Bermejo, “Evaluation of the rotation matrices in the basis of real spherical harmonics”, Journal of Molecular Structure: THEOCHEM (1997).
- [27] O. Zienkiewicz, R. Taylor, and J. Zhu, *The finite element method: its basis and fundamentals* (Elsevier, 2005).
- [28] T. A. Wareing, J. M. McGhee, J. E. Morel, and S. D. Pautz, “Discontinuous finite element SN methods on three-dimensional unstructured grids”, Nuclear Science and Engineering (2017).
- [29] T. Haut, P. Maginot, V. Tomov, B. Southworth, T. Brunner, and T. Bailey, “An efficient sweep-based solver for the SN equations on high-order meshes”, Nuclear Science and Engineering (2019).
- [30] P. Zygmanski and E. Sajo, “Nanoscale radiation transport and clinical beam modeling for gold nanoparticle dose enhanced radiotherapy (GNPT) using X-rays”, British Journal of Radiology (2016).
- [31] J. Morel, “A hybrid collocation-Galerkin-SN method for solving the Boltzmann transport equation”, Nuclear Science and Engineering (1989).
- [32] M. Abramowitz and I. A. Stegun, *Handbook of mathematical functions with formulas, graphs, and mathematical tables* (Dover, 1965).
- [33] N. Jeevanjee, *An introduction to tensors and group theory for physicists* (Birkhäuser, 2015).
- [34] C. Geuzaine and J. Remacle, “Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities.”, International Journal for Numerical Methods in Engineering (2009).
- [35] J. S. Warsa, T. A. Wareing, and J. E. Morel, in On the degraded effectiveness of diffusion synthetic acceleration for multidimensional SN calculations in the presence of material discontinuities (2003).
- [36] E. W. Larsen, “Diffusion-synthetic acceleration methods for discrete-ordinates problems”, Transport Theory and Statistical Physics (1984).

- [37] J. Patel, L. Moraes, R. Vasques, and R. Barros, “Transport synthetic acceleration for the solution of one-speed nonclassical spectral SN equations in slab geometry”, *Journal of Computational and Applied Mathematics* (2022).
- [38] C. Drumm and R. P. Kensek, *Scattering cross sections for electron transport using energy-finite-element weighting*, 2017.
- [39] D. E. Bruss, W. C. Fan, K. Bossler, C. Drumm, S. D. Pautz, and A. Felice, *Sandia computational engine for particle-transport cross sections v. 0.0*, tech. rep. (Sandia National Labs, 2017).
- [40] O. N. Vassiliev, T. A. Wareing, J. McGhee, G. Failla, M. R. Salehpour, and F. Mourtada, “Validation of a new grid-based Boltzmann equation solver for dose calculation in radiotherapy with photon beams”, *Physics in Medicine and Biology* (2010).
- [41] M. Berger, J. Hubbell, S. Seltzer, J. Chang, J. Coursey, R. Sukumar, D. Zucker, and K. Olsen, *XCOM: Photon cross sections database*, tech. rep. (National Institute of Standards and Technology, 2010).
- [42] J. Morel, “On the validity of the extended transport cross-section correction for low-energy electron transport”, *Nuclear Science and Engineering* (1979).
- [43] C. R. Drumm, W. C. Fan, L. Lorence, and J. Liscum-Powell, “An analysis of the extended-transport correction with application to electron beam transport”, *Nuclear Science and Engineering* (2007).
- [44] S. Goudsmit and J. Saunderson, “Multiple scattering of electrons”, *Physical Review* (1939).
- [45] Y. Saad and M. H. Schultz, “GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems”, *SIAM Journal on Scientific Computing* (1986).
- [46] L. N. Trefethen and D. Bau, *Numerical linear algebra* (SIAM, 1997).
- [47] T. Möller and B. Trumbore, “Fast, minumum storage ray/triangle intersection”, *Journal of Graphics Tools* (1997).
- [48] K. A. Gifford, J. L. H. Jr., T. A. Wareing, G. Failla, and F. Mourtada, “Comparison of a finite-element multigroup discrete-ordinates code with Monte Carlo for radiotherapy calculations”, *Physics in Medicine and Biology* (2006).
- [49] O. N. Vassiliev, T. A. Wareing, I. M. Davis, J. McGhee, D. Barnett, J. L. Horton, K. Gifford, G. Failla, U. Titt, and F. Mourtada, “Feasibility of a multigroup deterministic solution method for three-dimensional radiotherapy dose calculations”, *International Journal of Radiation Oncology Biology Physics* (2008).