

# Discrete Ordinates

Edward Norris

February 22, 2017

## Abstract

The abstract text goes here.

## Contents

<b>1 CODE</b>	<b>2</b>
<b>2 Introduction</b>	<b>5</b>
<b>3 Mathematical Development</b>	<b>5</b>
3.1 The Boltzmann Equation . . . . .	5
3.2 Angular Discretization . . . . .	5
3.3 Energy Discretization . . . . .	6
3.4 Spatial Discretization . . . . .	6
3.5 Harmonic Approximation . . . . .	8
3.6 Discretization . . . . .	8
3.7 Raytracer . . . . .	9
<b>4 Implementation</b>	<b>11</b>
4.1 Design . . . . .	11
4.2 Language Selection . . . . .	11
4.3 Salome . . . . .	11
4.4 Framework Integration . . . . .	11
4.5 C++ Implementation . . . . .	11
<b>5 MPI Implementation</b>	<b>11</b>
<b>6 GPU Implementation</b>	<b>12</b>
<b>7 Conclusion</b>	<b>12</b>
<b>8 Appendix</b>	<b>12</b>
8.1 Laplace's Equation . . . . .	12
8.2 Legendre Polynomials . . . . .	13
8.3 Associated Legendre Polynomials . . . . .	13
8.4 Spherical Harmonic Function . . . . .	14
8.5 Klein-Nishina Formula . . . . .	15
8.6 Numeric Integration . . . . .	16

# 1 CODE

Consider that we have the following variables,  $\Psi$  which is a function of  $x, y, z, E$ , and  $\hat{\Omega}$ . Further, we have the in-flux variables  $\Psi_{x-in}$ ,  $\Psi_{y-in}$ , and  $\Psi_{z-in}$ .

Given:

- A list of all energies,  $\{E\}$
- A list of all directions in the quadrature,  $\{\Omega\}$
- A mapping from  $r$  to zone number,  $\mathcal{Z}(r)$
- Lists of direction cosines,  $\mu(\Omega)$ ,  $\xi(\Omega)$ , and  $\eta(\Omega)$
- A map of  $r$  to voxel volume,  $V(r)$
- A mapping of weights for the quadrature,  $\omega(\Omega)$
- A mapping of scatter cross sections in zone  $\mathcal{Z}(r)$  from energy  $E'$  to  $E$ ,  $\Sigma_s(\mathcal{Z}(r), E' \rightarrow E)$
- A list of total cross sections in zone  $\mathcal{Z}(r)$  at energy  $E$ ,  $\Sigma_T(\mathcal{Z}(r), E)$
- The surface area of all voxels,  $A_{xy}(r)$ ,  $A_{yz}(r)$ ,  $A_{xz}(r)$

```

1 function sweep()
2
3 # Knowns
4 V(r) # [cm^3] Volume of cell ijk
5 Axy(Ω, i, j), Ayz(Ω, j, k), Axz(Ω, i, k) # [cm^2] Area of each face of cell ijk
6 Sx(E, r) # [#] External source
7
8 # Unknowns
9 φ(r, E) = 0 # Scalar flux
10 Ψ(r, Ω, E) = 0 # Angular flux
11
12 for every E ∈ {E} in descending order
13   while not converged
14     φi-1(r) = φi+1(r)
15     S(r) = 0
16     for every E' >= E
17       for every cell ijk
18         S(ijk) = S(ijk) + 1/4πφ(E', ijk) * Σs(Z(ijk), E') * V(ijk)
19     for every cell ijk
20       S(ijk) = S(ijk) + Sx(ijk)
21     φi+1 = 0
22     for every Ω ∈ Q in any order
23       i0 = f(μ(Ω))
24       Δi = f'(μ(Ω))
25       j0 = f(ξ(Ω))
26       Δj = f(ξ(Ω))
27       k0 = f(η(Ω))
28       Δk = f(η(Ω))
29     for every cell ijk starting at (i0, j0, k0) in direction <Δi, Δj, Δk>
30       if i ∈ ∂
31         φx+ = 0
32       else
33         φx+ = φx-(i ± 1, j, k)
34       if j ∈ ∂
35         φy+ = 0
36       else
37         φy+ = φy-(i, j ± 1, k)
38       if k ∈ ∂
39         φz+ = 0
40       else
41         φz+ = φz-(i, j, k ± 1)
42
43       n = S(ijk) + Ayz(Ω, j, k)φx+ + Axz(Ω, i, k)φy+ + Axy(Ω, i, j)φz+
44       d = V(ijk) * Σt(Z(ijk), E) + Ayz(Ω, j, k) + Axz(Ω, i, k) + Axy(Ω, i, j)
45       Ψ0 = n/d
46       Ψ(E, Ω, ijk) = Ψ0
47       φx-(ijk) = 2Ψ0 - φx+
48       φy-(ijk) = 2Ψ0 - φy+
49       φz-(ijk) = 2Ψ0 - φz+
50       φi+1(ijk) = φi+1 + ω(Ω)Ψ(e, Ω, ijk)
51     for every cell ijk
52       φ(E, ijk) = φi+1(ijk)
53     EMIT φ(E, ijk)
54   ε = max((φi+1 - phii-1)/φi+1)
55
56   for every cell ijk
57     φ(E, ijk) = φi+1(ijk)
58 output time to completion

```

```

1 function octant1()
2
3   for  $x \in X$  in ascending order
4     for  $y \in Y$  in ascending order
5       for  $z \in Z$  in ascending order
6
7      $S = \text{totalScatter}(x, y, z, \Omega, E)$ 
8
9     # Get the in-flux from the previous out-flux
10     $\Psi_{x-in}(x, y, z, E, \Omega) \leftarrow \Psi_{x-out}(x - \Delta x, y, z, E, \Omega)$ 
11     $\Psi_{y-in}(x, y, z, E, \Omega) \leftarrow \Psi_{x-out}(x, y - \Delta y, z, E, \Omega)$ 
12     $\Psi_{z-in}(x, y, z, E, \Omega) \leftarrow \Psi_{x-out}(x, y, z - \Delta z, E, \Omega)$ 
13
14    # Calculate the angular flux
15     $n \leftarrow S + 2\mu(\Omega)A_{yz}(y, z)\Psi_{x-in}(x, y, z, E, \Omega) +$ 
16       $2\xi(\Omega)A_{xz}(x, z)\Psi_{y-in}(x, y, z, E, \Omega) +$ 
17       $2\eta(\Omega)A_{xy}(x, y)\Psi_{z-in}(x, y, z, E, \Omega)$ 
18     $d \leftarrow 2\mu(\Omega)A_{yz}(y, z) + 2\xi(\Omega)A_{xz}(x, z) + 2\eta(\Omega)A_{xy}(x, y) + \Sigma_T(\mathcal{Z}(r), E)$ 
19     $\Psi(x, y, z, E, \Omega) \leftarrow n/d$ 
20
21    # Calculate the out-flux
22     $\Psi_{x-out}(x, y, z, E, \Omega) \leftarrow 2\Psi(x, y, z, E, \Omega) - \Psi_{x-in}(x, y, z, E, \Omega)$ 
23     $\Psi_{y-out}(x, y, z, E, \Omega) \leftarrow 2\Psi(x, y, z, E, \Omega) - \Psi_{y-in}(x, y, z, E, \Omega)$ 
24     $\Psi_{z-out}(x, y, z, E, \Omega) \leftarrow 2\Psi(x, y, z, E, \Omega) - \Psi_{z-in}(x, y, z, E, \Omega)$ 
25
26    # Increment the scalar flux
27     $\phi(x, y, z, E) \leftarrow \phi(x, y, z, E) + \omega(\Omega)\Psi(x, y, z, E, \Omega)$ 

```

```

1 function totalScatter(r,Ω,E)
2
3 # Assume a point source at r₀ with energy E₀ and strength S₀ particles / sec
4 S_x ← 0
5 if r = r₀ and E = E₀
6   S_x ← (1/4π) × S₀
7
8 S_D ← 0
9 for E' ∈ E | E' ≥ E
10   S_D = S_D + (1/4π) × φ(r,Ω,E') × Σ_s(Ζ(r),E',E) × V(r)
11
12 S_T = S_x + S_D
13 return S_T

```

The indexing scheme is as follows: E, A, x, y, z, l, m

## 2 Introduction

This document will teach you everything you need to know about the Boltzmann Equation from its derivation to the details of its implementation in computer code.

## 3 Mathematical Development

### 3.1 The Boltzmann Equation

The time dependent gamma transport equation can be written as

$$\left[ \frac{1}{v(E)} \frac{\partial}{\partial t} + \hat{\Omega} \cdot \nabla + \Sigma_t(\mathbf{r}, E, t) \right] \psi(\mathbf{r}, E, \hat{\Omega}, t) = \int_{4\pi} \int_0^\infty \Sigma_s(\mathbf{r}, E' \rightarrow E, \hat{\Omega}' \rightarrow \hat{\Omega}, t) \psi(\mathbf{r}, E', \hat{\Omega}', t) dE' d\hat{\Omega}' + S(\mathbf{r}, E, \hat{\Omega}, t) \quad (3.1.1)$$

However, we are typically not concerned with the transient case in medical diagnostic imaging. We are more interested in the steady state case. The time independent form of Eq. 3.1.1 is written as

$$\left[ \hat{\Omega} \cdot \nabla + \Sigma_t(\mathbf{r}, E) \right] \psi(\mathbf{r}, E, \hat{\Omega}) = \int_{4\pi} \int_0^\infty \Sigma_s(\mathbf{r}, E' \rightarrow E, \hat{\Omega}' \rightarrow \hat{\Omega}) \psi(\mathbf{r}, E', \hat{\Omega}', t) dE' d\hat{\Omega}' + S(\mathbf{r}, E, \hat{\Omega}) \quad (3.1.2)$$

The following sections show how Eq. 3.1.2 is discretized.

### 3.2 Angular Discretization

The coordinate system used in DOCTORS is shown in Fig. 1. A discrete set of  $N_a$  angles ( $\Omega_a \quad a = 0 \dots N_a - 1$ ) is selected to represent continuous directional space. Particles are transported only along these discrete directions from one voxel to adjacent voxels.

The  $S_N$  quadrature is implemented in DOCTORS. There are many different quadrature sets, the  $S_N$  quadratures were selected because they are easy to implement, rotationally symmetric, and the most commonly used in production discrete ordinate solvers. This enables simple, direct comparison to other solvers. Arbitrary quadratures are permissible in a plain text file supplied by the user enabling other quadratures.

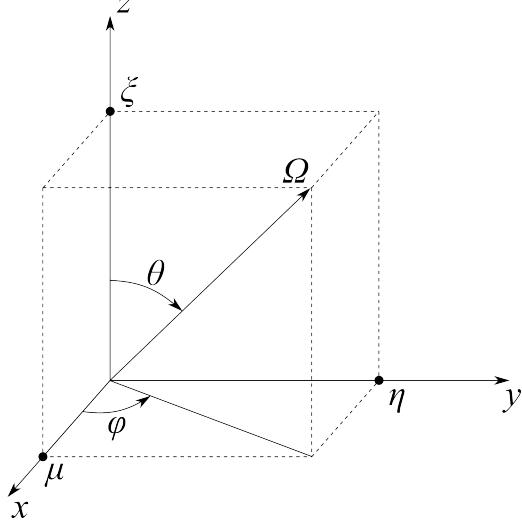


Figure 1: The coordinate system used in DOCTORS. Given an arbitrary direction,  $\Omega$ ,  $\mu$ ,  $\eta$ , and  $\xi$  are its direction cosines with respect to the  $x$ ,  $y$ , and  $z$  axes respectively.  $\varphi$  is the azimuthal angle (with respect to  $x$  and the  $d\theta$  is the polar angle (with respect to  $z$ ).

Each discrete angle is associated a weight. The weights are designed with the angles such that once a discrete quadrature is selected, integration over continuous space becomes integration in discrete space approximated by Eq. 3.2.1.

$$\int_{4\pi} f(\hat{\Omega}) d\Omega \approx \sum_{a=0}^{N_a-1} f(\hat{\Omega}_a) \omega_a \quad (3.2.1)$$

Discretizing Eq. 3.1.2 in angular space gives Eq. 3.2.2 where the angle is denoted by subscript  $a$ .

$$\begin{aligned} & \left[ \hat{\Omega}_a \cdot \nabla + \Sigma_t(\mathbf{r}, E) \right] \psi_a(\mathbf{r}, E) = \\ & \sum_{a=0}^{N_a-1} \int_0^\infty \Sigma_{s,a,a'}(\mathbf{r}, E' \rightarrow E) \psi_{a'}(\mathbf{r}, E') dE' \omega_a + S_a(\mathbf{r}, E) \end{aligned} \quad (3.2.2)$$

### 3.3 Energy Discretization

Continuous energy is discretized into  $G$  groups indexed from 0 to  $G - 1$ . Upscatter is assumed to be negligible. Therefore, Eq. 3.2.2 becomes the energy discretized Eq. 3.3.1 where the group number is indexed by superscript  $g$ . The group structure is shown in Fig. 2.

$$\left[ \hat{\Omega}_a \cdot \nabla + \Sigma_t^g(\mathbf{r}) \right] \psi_a^g(\mathbf{r}) = \sum_{a=0}^{N_a-1} \sum_{g'=g}^{G-1} \Sigma_{s,a,a'}^{g,g'}(\mathbf{r}) \psi_{a'}^{g'}(\mathbf{r}) \omega_a + S_a^g(\mathbf{r}) \quad (3.3.1)$$

### 3.4 Spatial Discretization

Finally, Eq. 3.3.1 is discretized in space to give Eq. ?? which is fully discretized. The problem domain is split into an evenly spaced Cartesian grid. The grid spacing between each of the spatial dimensions does not necessarily need to be uniform, but it often is in the case of CT voxel phantoms.

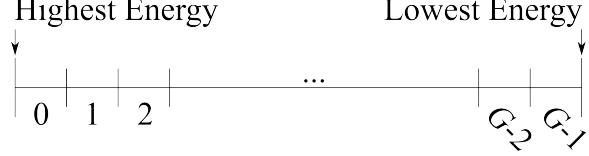


Figure 2: The energy grid structure used in DOCTORS. The highest energy group is group 0 and the lowest energy is group  $G - 1$ .

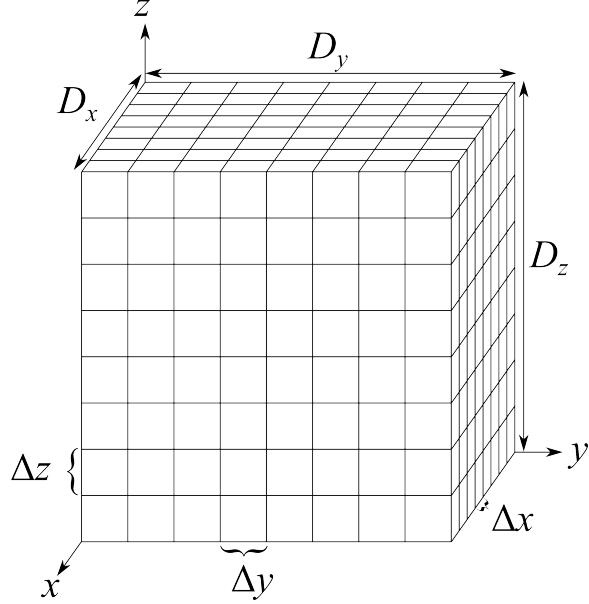


Figure 3: The spatial mesh imposed on the problem domain.

The problem domain must be a regular rectangular parallelepiped of size  $D_x$ ,  $D_y$ , and  $D_z$  in the  $x$ ,  $y$ , and  $z$  directions respectively. The mesh is partitioned into  $N_x$ ,  $N_y$ , and  $N_z$  evenly spaced bins along each direction. The total number of voxels,  $N_V$  is easily computed as the multiple of each of the component directions given in Eq. 3.4.1.

$$N_V = N_x N_y N_z \quad (3.4.1)$$

Voxel indices are flattened according to the rule given in Eq. 3.4.2. Note that all indeces always start at zero.

$$i = i_x(N_z + N_y) + i_y N_z + i_z \quad (3.4.2)$$

The spatial mesh is shown in Fig. 3. The size of each voxel in each dimension is computed by dividing the size of the problem domain along in that dimension by the number of mesh elements along the same direction. Eq. 3.4.3 shows this computation for the  $x$  direction. Analogous equations apply in the  $y$  and  $z$  directions as well.

$$\Delta x = \frac{D_x}{N_x} \quad (3.4.3)$$

### 3.5 Harmonic Approximation

Equation ?? has a number of terms that cannot be solved directly. Instead, some numerical approximation must be used. The macroscopic scattering cross section,  $\Sigma_s$  is typically expanded with a Legendre polynomial (for more information of Legendre polynomials, refer to Section 8.2). The expansion is as follows:

$$\Sigma_s(\mathbf{r}, E' \rightarrow E, \hat{\Omega}' \rightarrow \hat{\Omega}) \approx \sum_{l=0}^L \frac{2l+1}{4\pi} \Sigma_{s,l}(\mathbf{r}, E' \rightarrow E) P_l(\hat{\Omega}' \rightarrow \hat{\Omega}) \quad (3.5.1)$$

where  $\Sigma_{s,l}$  is the expansion coefficients termed the "scattering moments." The Legendre polynomials  $P_l(\hat{\Omega}' \rightarrow \hat{\Omega})$  are defined as

$$P_l(\hat{\Omega}' \rightarrow \hat{\Omega}) = \frac{1}{2l+1} \sum_{m=-l}^l Y_{l,m}^*(\hat{\Omega}') Y_{l,m}(\hat{\Omega}) \quad (3.5.2)$$

The angular fluence ( $\phi$ ) is also expanded as

$$\phi(\mathbf{r}, E', \hat{\Omega}') \approx \sum_{l=0}^L \sum_{m=-l}^l \phi_{l,m}(\mathbf{r}, E') Y_{l,m}(\hat{\Omega}') \quad (3.5.3)$$

A source term suitable for numeric integration is then

$$\begin{aligned} \int_0^\infty dE' \int_{4\pi} d\hat{\Omega}' \Sigma_s(\mathbf{r}, E' \rightarrow E, \hat{\Omega}' \rightarrow \hat{\Omega}) \phi(\mathbf{r}, E', \hat{\Omega}') \approx \\ \sum_{l=0}^L \frac{2l+1}{4\pi} \sum_{m=-l}^l \Sigma_{s,l}^{gg'} \phi_{i,j,k,lm}^{g'} Y_{l,m}(\hat{\Omega}_n) \end{aligned} \quad (3.5.4)$$

Finally, substituting Eqs. 3.5.4 and 3.5.3 into 3.5.1, we arrive at

$$\begin{aligned} \hat{\Omega}_n \cdot \nabla \psi_{i,j,k,n}^g + \sigma_{i,j,k}^g \psi_{i,j,k,n}^g = \\ \sum_{g'=0}^G \sum_{l=0}^L \frac{2l+1}{4\pi} \sum_{m=-l}^l \sigma_{s,l}^{g,g'} \psi_{i,j,k,l,m}^{g'} Y_{l,m}(\hat{\Omega}_n) + \frac{1}{4\pi} q_{i,j,k}^g \end{aligned} \quad (3.5.5)$$

We can calculate the scalar flux as

$$\phi_{i,j,k}^g = \sum_{n=1}^{|\Omega|} w_n \psi_{i,j,k,n}^g \quad (3.5.6)$$

### 3.6 Discretization

The continuous Boltzmann Equation Approximation has to be discretized in all dimensions to run on a computer.

The gradient of  $\psi$  is calculated as

$$\nabla \psi_{i,j,k,n}^g \approx \left\langle \frac{\psi_{i+1,j,k}^g - \psi_{i-1,j,k}^g}{\Delta x}, \frac{\psi_{i,j+1,k}^g - \psi_{i,j-1,k}^g}{\Delta y}, \frac{\psi_{i,j,k+1}^g - \psi_{i,j,k-1}^g}{\Delta z} \right\rangle \quad (3.6.1)$$

There are six faces on each parallelepiped with normals  $\langle \pm 1, 0, 0 \rangle$ ,  $\langle 0, \pm 1, 0 \rangle$ , and  $\langle 0, 0, \pm 1 \rangle$ . Taking the dot product of these normals and the evaluated gradient gives



Figure 4: Simulation Results

$$\hat{\Omega} \cdot \nabla \psi_{i,j,k}^g \approx 2 \left( \frac{\psi_{i+1,j,k}^g - \psi_{i-1,j,k}^g}{\Delta x} + \frac{\psi_{i,j+1,k}^g - \psi_{i,j-1,k}^g}{\Delta y} + \frac{\psi_{i,j,k+1}^g - \psi_{i,j,k-1}^g}{\Delta z} \right) \quad (3.6.2)$$

### 3.7 Raytracer

Discrete ordinate solutions greatly benefit from the presence of a raytracing algorithm to provide the uncollided flux to the solver.

The raytracer was implemented as...

When the solver is changed from isotropic to anisotropic, the raytracer must be able to not only compute the uncollided flux but compute the direction of flux in each voxel and map that onto the discrete quadrature.

To map an arbitrary direction onto a discrete quadrature set, the direction is split amongst the three nearest directions in the quadrature set. Each quadrature direction is weighted. The closeness of each direction computed by calculating the cosine of the angle between a direction  $\hat{\Omega}$  and the quadrature direction  $\hat{\Omega}_a$ . The cosine is computed

$$\cos(\theta) = \hat{\Omega} \cdot \hat{\Omega}_a \quad (3.7.1)$$

The three nearest points are then taken to form a triangle with the  $\hat{\Omega}$  direction between them as shown in Fig. XYZ.

The area of  $\triangle \mathbf{AB}\Omega$  is

$$A = |\mathbf{A} - \mathbf{B}| |\mathbf{A} - \mathbf{C}| \sqrt{1 - [(\mathbf{A} - \mathbf{B}) \cdot (\mathbf{A} - \mathbf{C})]} \quad (3.7.2)$$

Remove what is below this!

$$Y_l^m(\theta, \phi) = \begin{cases} \sqrt{2} K_l^m \cos(m\phi) P_l^m(\cos(\theta)), & m > 0 \\ K_l^0 P_l^0(\cos(\theta)), & m = 0 \\ \sqrt{2} K_l^m \sin(-m\phi) P_l^{-m}(\cos(\theta)), & m < 0 \end{cases} \quad (3.7.3)$$

where  $K_l^m$  is defined as

$$K_l^m = \sqrt{\frac{2l+1}{4\pi} \frac{(l-|m|)!}{(l+|m|)!}} \quad (3.7.4)$$

and the associated Legendre functions are computed by the recursion

$$\begin{aligned} P_0^0(x) &= 1 \\ P_m^m(x) &= (2m-1)!!(1-x^2)^{m/2} \\ P_{m+1}^m(x) &= x(2m+1)P_m^m(x) \\ P_l^m(x) &= \frac{x(2l-1)}{l-m}P_{l-1}^m(x) - \frac{l+m-1}{l-m}P_{l-2}^m(x) \end{aligned} \quad (3.7.5)$$

The spherical harmonic expansion is used to approximate a function  $f$  by

$$f(\theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^l y_l^m(\theta, \phi) f_l^m \quad (3.7.6)$$

which is exact when the expansion continues to infinity. However, it is truncated at  $L$  yielding

$$f(\theta, \phi) \approx \sum_{l=0}^L \sum_{m=-l}^l y_l^m(\theta, \phi) f_l^m \quad (3.7.7)$$

However, the value of each  $f_l^m$  must be computed before the approximation can be used. These values are computed by

$$f_l^m = \int_{4\pi} y_l^m(\theta, \phi) f(\theta, \phi) d\hat{\Omega} \quad (3.7.8)$$

In the case of the raytracer solution, since *only* the uncollided source is considered, the direction at any point is always precisely known, the particles are traveling away from the source point. Therefore, the function to be approximated is a Dirac delta function in the two dimensional surface of the unit sphere. Equation 3.7.8 can be reduced to

$$f_l^m = \int_{4\pi} y_l^m(\theta, \phi) f(\theta_0, \phi_0) d\hat{\Omega} \quad (3.7.9)$$

The property of the Dirac delta function

$$\int_{-\infty}^{\infty} \delta(x_0) f(x) dx = f(x_0) \quad (3.7.10)$$

can now be used which reduces Equation 3.7.8 to

$$f_l^m = y_l^m(\theta_0, \phi_0) \quad (3.7.11)$$

The moments are related to the angular flux by

$$\phi_l^m(\vec{r}, E) = \int Y_l^m(\hat{\Omega}') \psi(\vec{r}, \hat{\Omega}', E) d\hat{\Omega}' \quad (3.7.12)$$

which, for the uncollided flux, becomes

$$\phi_l^m(\vec{r}, E) = \Psi Y_l^m(\hat{\Omega}_0) \quad (3.7.13)$$

where  $\Psi$  is equal to the magnitude of  $\psi_{unc}$  in the  $\hat{\Omega}_0$  direction.

## 4 Implementation

At this point, we have successfully developed an algorithm that can be implemented on a computer. However, there are many challenges in actually implementing this algorithm.

### 4.1 Design

Yeah, it was designed.

### 4.2 Language Selection

Yeah, it was selected.

### 4.3 Salome

We chose to use the Salome framework for numerical pre/post-processing. Salome includes a built in geometry system and meshing capabilities.

### 4.4 Framework Integration

Yeah, it was integrated

### 4.5 C++ Implementation

Yeah, it was implemented.

```
1 #include <stdio.h>
2 #define N 10
3 /* Block
4  * comment */
5
6 int main()
7 {
8     int i;
9
10    // Line comment.
11    puts("Hello world!");
12
13    for (i = 0; i < N; i++)
14    {
15        puts("LaTeX is also great for programmers!");
16    }
17
18    return 0;
19 }
```

## 5 MPI Implementation

MPI stands for Message Passing Interface and is the *de Facto* standard for implementing parallel algorithms across multiple physical machines.

Table 1: A list of solutions to Laplace's equation in different geometrical systems

Cartesian	Exponential, Circular (a class of trig function), hyperbolic
Cylindrical	Bessel, Exponential, Circular
Spherical	Legendre polynomial, Power, Circular

## 6 GPU Implementation

GPU stands for Graphical Processing Unit and is a collection of SIMD (single instruction multiple data) processors. Each processor operates on its own piece of data but performs the same operation each other processor in its warp is executing.

## 7 Conclusion

Hooray! It works!

## 8 Appendix

### 8.1 Laplace's Equation

Many physical phenomena can be described by Poisson's Equation given below.

$$\nabla^2 f = \psi \quad (8.1.1)$$

Laplace's Equation is a special form of Poisson's Equation and is given by Eq. 8.1.2.

$$\nabla^2 f = 0 \quad (8.1.2)$$

Laplace's equation in cartesian coordinates expands to

$$\nabla^2 f(x, y, z) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2} = 0 \quad (8.1.3)$$

In cylindrical coordinates

$$\nabla^2 f(r, \phi, z) = \frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial f}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2 f}{\partial \phi^2} + \frac{\partial^2 f}{\partial z^2} \quad (8.1.4)$$

In spherical coordinates

$$\nabla^2 f(\rho, \theta, \phi) = \frac{1}{\rho^2} \frac{\partial}{\partial \rho} \left( \rho^2 \frac{\partial f}{\partial \rho} \right) + \frac{1}{\rho^2 \sin \theta} \frac{\partial}{\partial \theta} \left( \sin \theta \frac{\partial f}{\partial \theta} \right) + \frac{1}{\rho^2 \sin^2 \theta} \frac{\partial^2 f}{\partial \phi^2} \quad (8.1.5)$$

Any solution to Laplace's equation is known as a harmonic function. Further, if any two functions are a solution to Laplace's equation, then by the superposition principle, the summation of the two functions is also a solution to Laplace's equation.

Common solutions to Laplace's equation include

## 8.2 Legendre Polynomials

Legendre polynomials are solutions to

$$P_n(x) = \frac{1}{2\pi i} \oint \frac{1}{\sqrt{1 - 2\xi x + \xi^2}} \frac{1}{\xi^{n+1}} d\xi \quad (8.2.1)$$

The first few such solutions are

$$\begin{aligned} P_0(x) &= 1 \\ P_1(x) &= x \\ P_2(x) &= \frac{1}{2}(3x^2 - 1) \\ P_3(x) &= \frac{1}{2}(5x^3 - 3x) \\ P_4(x) &= \frac{1}{8}(35x^4 - 30x^2 + 3) \\ P_5(x) &= \frac{1}{8}(63x^5 - 70x^3 + 15x) \\ P_6(x) &= \frac{1}{16}(231x^6 - 315x^4 + 105x^2 - 5) \end{aligned}$$

Legendre polynomials are useful for approximating functions truncated at some point. Legendre polynomials are mutually orthogonal which gives them the property

$$f(x) = \sum_{n=0}^{\infty} P_n(x) \quad (8.2.2)$$

where  $f(x)$  is any arbitrary function. Often, like a Fourier series which uses sin and cos instead of Legendre functions, the series is cut off at some threshold as an approximation. As an example of this, consider the function  $1.75x^3 + 0.6x^2 + .21x - .06$  over the domain  $[2, 5]$  using the first four Legendre polynomials.

The coefficients are computed

$$k_0 = \int_2^5 P_0(x)f(x)dx = \int_2^5 f(x)dx \quad (8.2.3)$$

$$k_1 = \int_2^5 P_1(x)f(x)dx = \int_2^5 xf(x)dx \quad (8.2.4)$$

$$k_2 = \int_2^5 P_2(x)f(x)dx = \int_2^5 \frac{1}{2}(3x^2 - 1)f(x)dx \quad (8.2.5)$$

$$k_3 = \int_2^5 P_3(x)f(x)dx = \int_2^5 \frac{1}{2}(5x^3 - 3x)f(x)dx \quad (8.2.6)$$

$$(8.2.7)$$

These values are then used to reconstruct the original function by

$$f(x) \approx k_0 P_0(x) + k_1 P_1(x) + k_2 P_2(x) + k_3 P_3(x) \quad (8.2.8)$$

Associated Legendre polynomials are a generalization of the Legendre polynomial and solve

## 8.3 Associated Legendre Polynomials

The associated Legendre polynomials are solutions to the equation

$$P_l^m(x) = (-1)^m (1 - x^2)^{m/2} \frac{d^m P_l(x)}{dx^m} \quad (8.3.1)$$

The associated Legendre polynomials can be computed recursively by

$$P_m^m(x) = (-1)^m (2m-1)!! (1-x^2)^{(m/2)} \quad (8.3.2)$$

$$P_{m+1}^m(x) = x(2m+1)P_m^m(x) \quad (8.3.3)$$

$$(l-m)P_l^m(x) = x(2l-1)P_{l-1}^m(x) - (l+m-1)P_{l-2}^m(x) \quad (8.3.4)$$

where  $x!!$  is the double factorial function defined by:

$$x!! = \prod_{k=0}^{\lceil x/2 \rceil - 1} x - 2k \quad (8.3.5)$$

For example,  $5!! = 5 \times 3 \times 1 = 15$  and  $6!! = 6 \times 4 \times 2 = 48$ .

Some references omit the Condon-Shortley phase  $(-1)^m$  while most others include it. These two are distinguished by the nomenclature

$$P_{lm}(x) = (-1)^m P_l^m(x) \quad (8.3.6)$$

Only the more commonly found  $P_l^m$  form will be used in this work.

## 8.4 Spherical Harmonic Function

Combining the associated Legendre polynomials and the sin/cos series, the spherical harmonics equations can be derived. They are orthogonal in both directions.

$$Y_l^m(\theta, \phi) = N_l^{|m|} P_l^{|m|}(\cos \theta) e^{im\phi} \quad (8.4.1)$$

Using Euler's Equation (Eq. 8.4.2) to compute the imaginary exponent, the spherical harmonic can be expanded to Eq. 8.4.3.

$$e^{i\theta} = \cos(\theta) + i \sin(\theta) \quad (8.4.2)$$

$$Y_l^m(\theta, \varphi) = N_l^{|m|} P_l^{|m|}(\cos \theta) [\cos(m\phi) + i \sin(m\phi)] \quad (8.4.3)$$

The normalization constant  $N_l^m$  is defined as

$$N_l^m = \sqrt{(2 - \delta_{m,0}) \frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!}} \quad (8.4.4)$$

where  $\delta$  is the Kronecker delta defined by

$$\delta_{x,y} = \begin{cases} 1, & \text{if } x = y \\ 0, & \text{otherwise} \end{cases} \quad (8.4.5)$$

In many cases, only the real part ( $y_l^m$ ) of the complex spherical harmonics ( $Y_l^m$ ) are necessary.

$$y_l^m = \begin{cases} N_l^m P_l^m(\cos(\theta)) \cos(m\phi) & m > 0 \\ N_l^0 P_l^0(\cos(\theta)) & m = 0 \\ N_l^{|m|} P_l^{|m|}(\cos(\theta)) \sin(|m|\phi) & m < 0 \end{cases} \quad (8.4.6)$$

$$\Sigma_s(\Omega \cdot \Omega') \approx \sum_{l=0}^L \sigma_l \sum_{m=-l}^l Y_l^m(\Omega) \bar{Y}_l^m(\Omega') \quad (8.4.7)$$

where the over-bar denotes the complex conjugate defined by Eq. 8.4.8.

$$\bar{Y}_l^m(\theta, \varphi) = (-1)^l \sqrt{\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!}} P_l^m(\cos \theta) [\cos(m\varphi) - i \sin(m\varphi)] \quad (8.4.8)$$

Expanding the inner summation of 8.4.7 with Eq. 8.4.3 and 8.4.8 yields:

$$\begin{aligned} Y_l^m(\theta, \varphi) \bar{Y}_l^m(\theta', \varphi') &= (-1)^{2l} \frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!} P_l^m(\cos \theta) P_l^m(\cos \theta') \times \\ &\quad [\cos(m\varphi) + i \sin(m\varphi)] [\cos(m\varphi') - i \sin(m\varphi')] \\ &= \frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!} P_l^m(\cos \theta) P_l^m(\cos \theta') \times \\ &\quad [\cos(m\varphi) \cos(m\varphi') + i \sin(m\varphi) \cos(m\varphi') - i \sin(m\varphi') \cos(m\varphi) - i^2 \sin(m\varphi) \sin(m\varphi')] \end{aligned} \quad (8.4.9)$$

The real part of the  
whose real part can be expanded to

$$Y_l^0 = \sqrt{\frac{2l+1}{4\pi}} P_l^0(1) \quad (8.4.10)$$

$$Y_l^{m,even} = (-1)^m \sqrt{\frac{2l+1}{2\pi} \frac{(l-m)!}{(l+m)!}} P_l^m(\cos(m\phi)) \quad (8.4.11)$$

$$Y_l^{m,odd} = (-1)^m \sqrt{\frac{2l+1}{2\pi} \frac{(l-m)!}{(l+m)!}} P_l^m(\sin(m\phi)) \quad (8.4.12)$$

Some useful recurrence relations used to calculate the Associated Legendre Polynomials are:

$$P_l^l(x) = (-1)^l (2l-1)!! (1-x^2)^{(l/2)} \quad (8.4.13)$$

$$P_{l+1}^l(x) = x(2l+1) P_l^l(x) \quad (8.4.14)$$

$$P_l^l(x) = (-1)^l (2l-1)!! (1-x^2)^{(l/2)} \quad (8.4.15)$$

where  $x!!$  is the double factorial function defined by:

$$x!! = \prod_{k=0}^{\lceil x/2 \rceil - 1} x - 2k \quad (8.4.16)$$

For example,  $5!! = 5 \times 3 \times 1 = 15$ .

## 8.5 Klein-Nishina Formula

The Klein-Nishina formula gives the differential scattering cross section of photon incident on a single free electron.

$$\frac{\partial \sigma}{\partial \Omega} = \frac{1}{2} \alpha^2 r_c^2 P(E_\gamma, \theta)^2 (P(E_\gamma, \theta) + P(E_\gamma, \theta)^{-1} - 1 + \cos^2(\theta)) \quad (8.5.1)$$

$$P(E_\gamma, \theta) = \frac{1}{1 + (\frac{E_\gamma}{m_e c^2})(1 - \cos(\theta))} \quad (8.5.2)$$

where  $\alpha$  is the fine structure constant ( $\alpha \approx 7.2971E-3$ ),  $r_c$  is the reduced Compton wavelength ( $r_c = \hbar$ )

## 8.6 Numeric Integration

$$\tilde{H}$$

(8.6.1)