1 Introduction

1.1 Coordinate System

We use a downward-facing right-handed Cartesian coordinate system. For spherical coordinates, we denote polar declenation from the positive z axis by ϕ and azimuthal angle from the positive x-axis towards the positive y-axis by θ.

1.2 Radiative Transfer Equation

$$\vec{\omega} \cdot \nabla L(\vec{r}, \vec{\omega}) = -(a(\vec{r}) + b(\vec{r}) + b \int_{4\pi} \beta(\vec{\omega} \cdot \vec{\omega'}) L(\vec{r}, \vec{\omega'}) \, d\vec{\omega'}) \tag{1}$$

1.3 Boundary conditions

Downwelling light:

$$L(x, y, 0, \vec{\omega}) = f(\vec{\omega}) \tag{2}$$

Upwelling light:

$$L(x, y, M, \vec{\omega}) = 0 \tag{3}$$

1.4 All angles are coupled by scattering

A major disadvantage of this formulation is that at each point in space, radiances in all angles are coupled. By asymptotic expansion, the solution can be broken up into independent scattering events, in which angles are decoupled by replacing the scattering integral with an integral over the radiance from the previous scattering event, which is already known.

2 Nondimensionalization

We nondimensionalize following Chandrasekhar.

2.1 Assumptions

We assume that the scattering coefficient is constant over space; that the primary difference in the optical effects of kelp and water is absorption, not scattering.

2.2 Scattering constant (same for kelp/water)

2.3 Table of variables

2.4 Rescale space, time

3 Limitations of Discrete Ordinates

3.1 Memory

GMRES is very memory-intensive.

3.2 CPU

It's very CPU-intensive as well.

3.3 GMRES unreliable

It also might never converge!

3.4 Also, need an initial guess for GMRES

Especially if we start from zero.

4 Asymptotics

4.1 Solve each angular problem independently

4.2 Relatively computationally cheap

4.3 Much lower memory cost

4.4 Known number of operations

4.5 Low and high accuracy available

5 Mathematical Procedure

5.1 Substitute asymptotic series

$$L(\vec{r},\vec{\omega}) = L_0(\vec{r},\vec{\omega}) + bL_1(\vec{r},\vec{\omega}) + b^2 L_2(\vec{r},\vec{\omega}) + \cdots \tag{4}$$

$$\vec{\omega}\cdot\nabla\left[L_0(\vec{r},\vec{\omega}) + bL_1(\vec{r},\vec{\omega}) + b^2 L_2(\vec{r},\vec{\omega}) + \cdots\right] = -(a(\vec{r})+b(\vec{r}))+b\int_{4\pi}\beta(\vec{\omega}\cdot\vec{\omega'})\left[L_0(\vec{r},\vec{\omega'}) + bL_1(\vec{r},\vec{\omega'}) + b^2 L_2(\vec{r},\vec{\omega'}) + \cdots\right]d\vec{\omega'} \tag{5}$$

5.2 Group like powers of b

5.3 Boundary conditions

5.4 Rewrite as ODE along ray path

5.5 Solve ODE as 1st order linear via I.F.

6 Numerical Implementation

6.1 Discrete grid

We choose a uniform rectangular spatial grid dividing each dimension into n_x, n_y, and n_z grid points respectively. We also use a uniformly spaced angular grid.

6.2 Numerical integration

We use the trapezoid rule for numerical integration as it allows for even or odd numbers of points, and places no restriction on the symmetry of the grid.

6.3 Storing pole values

We store pole values in the $(1, 1)$ and $(1, n_\phi)$ positions.

6.4 Loop rolling

6.5 Scattering integral

We do not include the current direction in the scattering integral.