How should one optimally choose quadrature rules for a new class of radiative transfer models?

Amro Ahmed Ahmed
BACs in Mechanical Engineering and BACs in Applied Science in Computing Technology

Introduction

Radiation plays a very important role in many applications. It is used in medical imaging and cancer treatment. It is obviously fundamental to nuclear-power generation and to heat transfer. However, the accurate prediction of radiative behaviours remains challenging due the presence of the unacceptable modelling artifacts in solution from existing models. Our research group had proposed new methods for the numerical prediction of radiation transport that circumvent these artifacts. The model predicts photon movement through a collection of quadrature-based representations of the probability-density function defining photons moving in each direction. However, this method requires a large set of groups of four directions and only works well using a single, hard-coded set of quadrature that was naively chosen. The goal of this study is to compare methods to generate these groups of directions

Methodology

One way to represent a direction of motion is as a point on the unit sphere. This project is therefore about distributing points on a sphere as evenly as possible. However, the current radiation model requires points to be in groups of four that are equidistant from each other (methane molecule).

One of the well studied methods that I will be comparing my methods with is implemented by randomly distributing points around the surface of the sphere without using hard-coded set of quatrature. Although, this method proved to distribute points that cover the surface of the sphere, it can not be used in real applications as they do not produce consistent distriutions all the time.

Trial 1: The first method that I approcahed was rotating a set of tetrehedral points about two axis of rotations, Z and X, using the rotation matrix euations 1,1.

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \quad R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{(EQ 1)}$$

Each point in the tetrehedral can be defined in cartesian coordinates (x, y, z) and spherical coordinates (ϕ, θ) or (Longitude, Latitude), respectively. The program starts with an initial tetrahedral–representation of a set of four coordinates [(0.817, 0.577, 0), (-0.817, 0.577, 0), (0, -0.577, -0.817), (0, -0.577, 0.817)] that will then be rotated about the Z and X axis, respectively, and store the new set of tetrehedral that next lies on the surface of the sphere. Each rotation requires an increament to the previous angle of $\Delta\theta$.

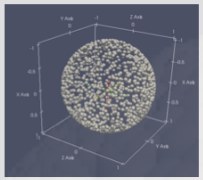
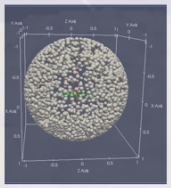
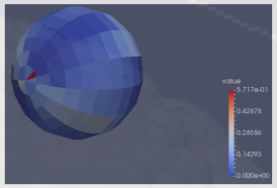
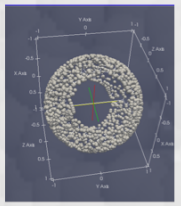
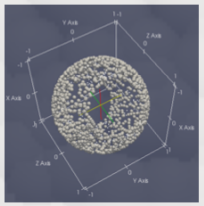
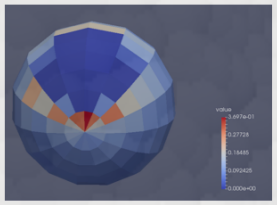
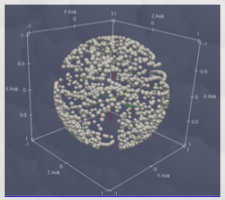
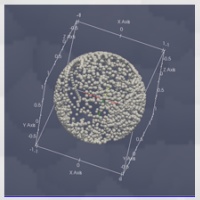
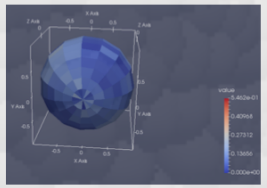
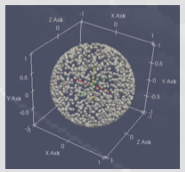
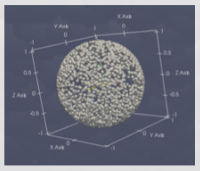
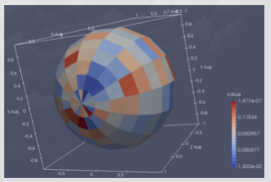
$$\Delta\theta = 2\pi/\text{interval} \quad \text{(EQ 2)}$$

The "Interval" is a predefined variable which controls the number of rotations that the tetrehedal points can go through to fill a unit sphere.

Trial 2: The second method uses the same concept as the first method but rotation about the Y-axis and then about the X-axis is uitilized to rotate the initial tetrahedral set of points. Rotation about the Y-axis uses the matrix Ry:

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \qquad \text{(EQ 3)}$$

Trial 3: The last method uses both the matrix rotations as discussed above but this time the tetrahedral is rotated about the each axis with different incremented angle $\Delta\theta$ in every loop. In other words, the method starts with the same initial tetrehedral-representation which then gets rotated about the Z-axis after the first increment, then gets rotated about the X-axis after the second increment, and finally gets rotated about the Y-axis after the third increment all under the same loop. The process cycles again until the increaments are added up to 2π. The main goal of this last trial is to distribute the points on the sphere as evenly as possible while keeping the points in the tetrahedral geometry.

	2000 Points	6000 Points	Density over 6000 points with a (15x15) mesh
Random Distribution			
Trial 1			
Trial 2			
Trial 3			

Results

Each method was tested with 2000 and 6000 set of points, which was then plotted by Paraview to produce the spherical scatter of points as shown in the left table. For comparison purposes, the desnity of the points on each segement of the unit sphere was also plotted on Paraview to deteremine the efficiency of the methods in distributing the points evenly. The density of a segement or cell can be calculated with the help of the Jacobian determinant (math.oregonstate.edu & en.wikipedia.org):

Density = (dots per cell / JD)/ Total # of dots

Jacobian determinant = JD = $\iint r^2 \sin\phi \, d\phi d\theta$

In this case, r=1 and we can assume $d\theta$ is constant

$= (\theta 1\text{-}\theta 2)(\cos\phi 2 - \cos\phi 1)$ (EQ 4)

In the third trial, the points that were distributed on the surface of the sphere showed similar depiction to the randomly distributed results. However, since the density represnation in tthe table at the left shows that the points were not evenly distrubuted on the surface of the unit sphere, work is still under progress to achieve consistent desnities over each segement of the sphere. Note, the real density of the points distrubution of the random method is different from the one presented below becuase real random distrubution must have been more even.

Conclusion

In conclusion, after carrying out trial 1 and 2, I learned that rotating a tetrahedral about two axis of rotations after every increament resulted in missing a huge set of points from the surface of the sphere, as shown in the results sections. However, after tackling the problem and rotating the tetrahedral about each axis of rotation with different incremented angle $\Delta\theta$ in every loop, proved to distrubute the points around the surface as evenly as the the random distribution method. Due to the unconsistent densities of points on the sphere for the third trial, more work is needed to try make these tetrahedrals rotate in a certain way as such the points are distributed as evenly while keeping the densities consistent and the geometry of the tetrehedral in mind.

References Acknowledgements

- **EQ 1 and EQ 3:** https://en.wikipedia.org/wiki/Rotation_matrix
- https://math.oregonstate.edu/home/programs/undergrad/CalculusQuestStudyGuides/vcalc/jacpol/jacpol.html
- **EQ 4:** https://en.wikipedia.org/wiki/Jacobian_matrix_and_determinant