2D MOC Convergence Study

Braden Pecora
The University of Texas at Austin
ME 388F: Computational Methods in Radiation Transport

May 5, 2025

1 Solution verification with code to code comparison

To verify that my 2D Method of Characteristics code provides correct answers, I did a code-to-code comparison with my 1D S_n code for several test cases. To make my 2D MOC code behave like a 1D code, I made the domain of the second dimension very large with vaccuum boundary conditions (initial fluxes) along the borders of that dimension. Then, I plotted the flux along the midline of the second to dimension and compared it to my S_n code. A more accurate comparison would have been to use reflecting boundary conditions, but I did not implement that due to time constraints. All problems have an incident source on the left surface.

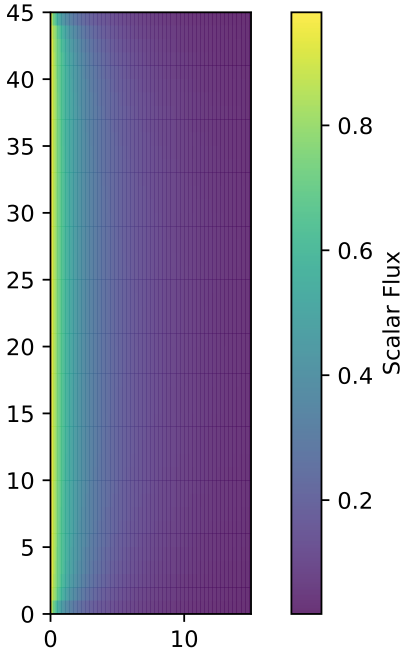
Figure 1 demonstrates that my 2D code is in proper agreement with my 1D code. The flux starts at one and decays at about the same rate. The solutions do differ a little bit.

Figure 2 compares my 2D code to a 1D for a highly scattering material. Both codes end up with very similar fluxes at the boundary, which indicates that I am doing something correctly with scattering. Again, the shape is slightly different but mostly similar.

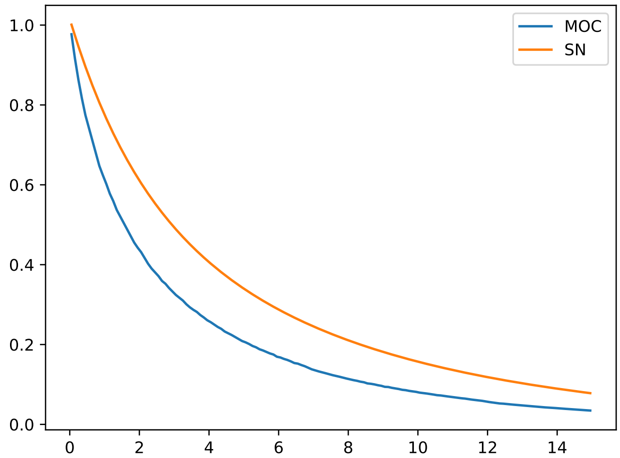
Figure 3 compares my codes when an isotropic source is used. The shape is mostly similar.

Lastly, I did a multiple-material case to verify my 2D code against my 1D code. The geometry contains air, a scatterer, and an isotropic source as shown in Figure 4. Like previous problems, there is an isotropic source flux on the left surface. The results are shown in Figure 6. The 2D results are fairly symmetric along the Y axis, which leads me to believe in the solution. Further, the shape and general magnitude of my MOC code matches my S_n code. As before, there is some discrepancy between the two codes.

I am not sure what the reason is. The first thing I would try is volume correct with my rays and the mesh. I did do a volume study to make sure my MOC rays were accurately representing the area of the source mesh cells, but there is some error that can be fixed. The error could also be related to my mesh resolution or angle discretization, but I do not think this is the case. I played with the number of angles and mesh size and it did not necessarily make the codes agree more. Perhaps I need to add reflecting boundary conditions or make the y dimension even larger for a proper comparison. Otherwise, there may be some sort of error in how I am calculating streaming along a ray that could be causing this discrepancy. My guess is that I am not properly accounting for rays that hit corners of cells or ride exactly on the interface between two rows or columns of cells.

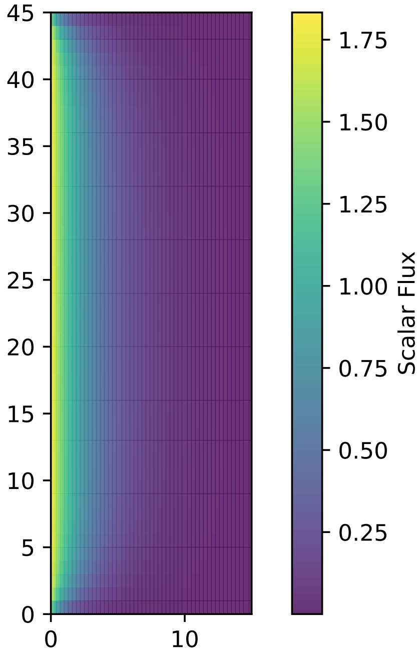


(a) 2D Flux

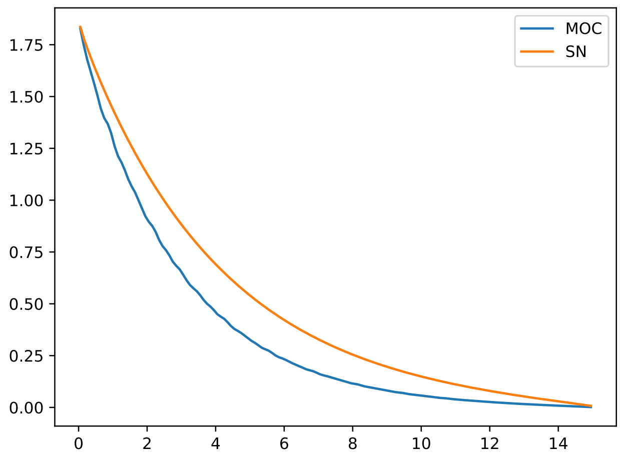


(b) Flux along centerline vs. S_n

Figure 1: Comparison of flux for air, which has low total and scattering cross sections.

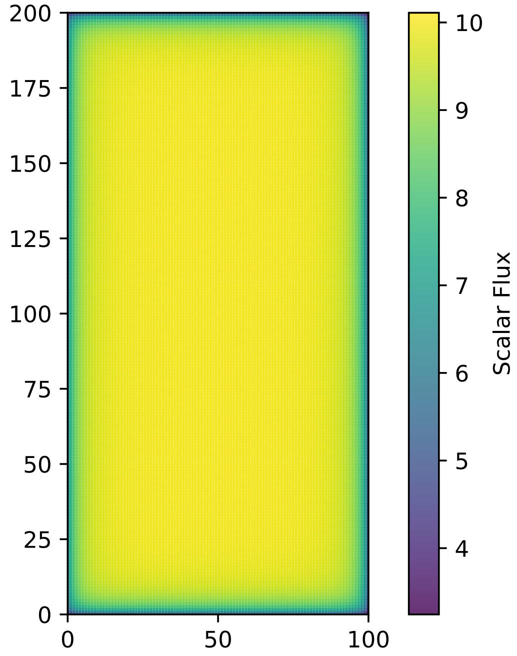


(a) 2D Flux

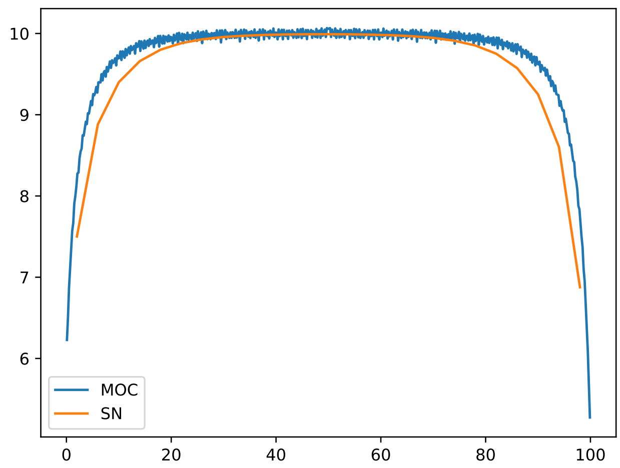


(b) Flux along centerline vs. S_n

Figure 2: Comparison of flux for a highly scattering material.



(a) 2D Flux



(b) Flux along centerline vs. S_n

Figure 3: Comparison of flux for a highly scattering material.

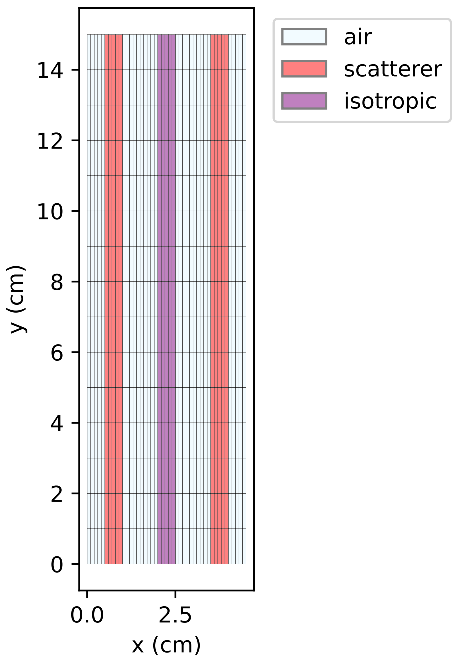
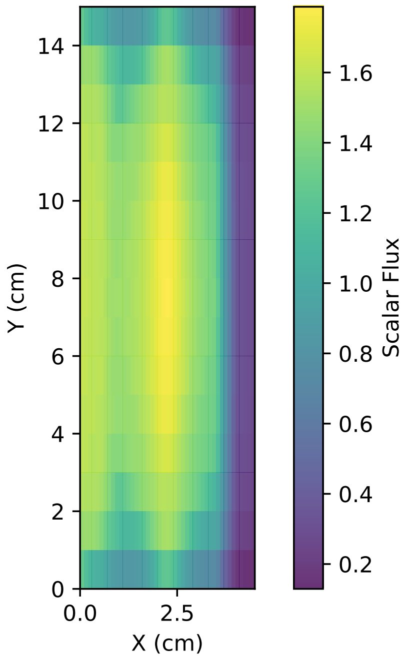
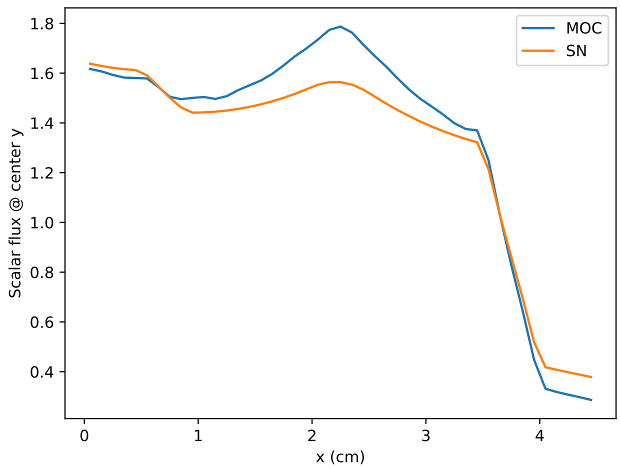


Figure 4: The mesh used for my multiple-material benchmark

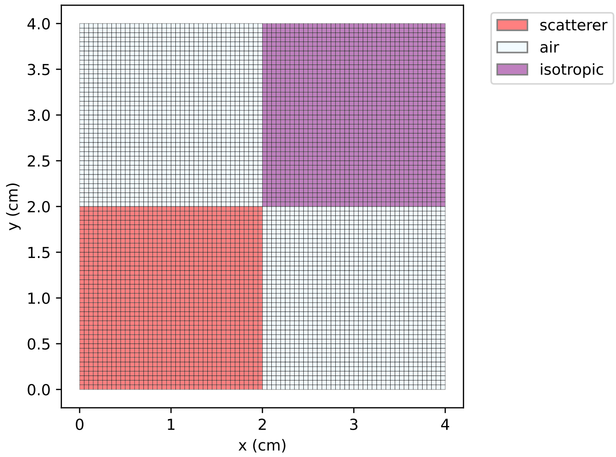


(a) 2D Flux

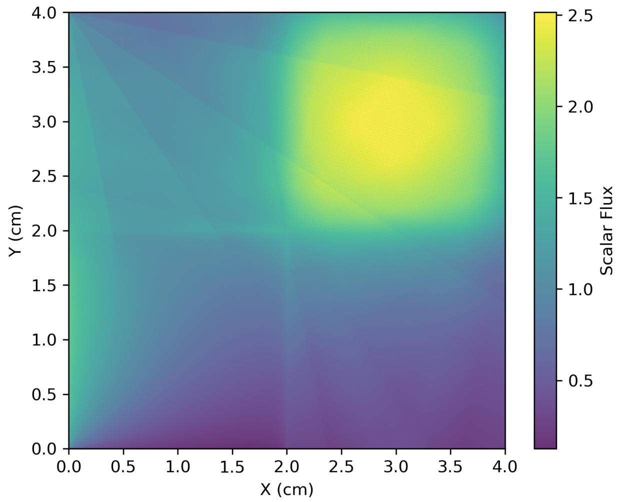


(b) Flux along centerline vs. S_n

Figure 5: Comparison of flux for a highly scattering material.



(a) The problem I used for the convergence study



(b) A high resolution mesh solution

Figure 6: The problem used in the convergence study.

2 Convergence Study

To test if my MOC code converged properly, I used the problem shown in Figure 9a. It has regions of source, scatter, and general transport. There is an incident flux of one on the left surface.

To test the convergence of my result with the spatial mesh, I used a mesh with 160,000 cells as the ground truth. All problems use regular cartesian meshes, 16 azimuthal angles and 4 polar angles. The ray width is determined as half of the length of a cell. The ground truth is shown in 9b. There are some notable ray effects, but the problem took around 25 minutes to solve, so I did not run it again. The problem does show the expected symmetry. Note that there is a source on the left side that does skew it a little bit.

I ran the same problem for smaller numbers of cells (using the same quadrature and changing the ray width to be half of the length of a cell) to calculate the order of convergence. I calculated the error by using linear interpolation to put values from the lower resolution mesh onto the higher order mesh and then calculating the L2 norm with the flux at all points. It is possible that my interpolation scheme effected the order of convergence. The error as a function of number of cells is shown in Figure 7. Based on the slope of the fitted line, the order of convergence is 2.34, or approximately 2. This seems right to me. Perhaps the dynamic ray widths or interpolation scheme are slightly impacting how the order of convergence is calculated. Further, it seems I could have a better fundamental truth to compare to.

I also did a convergence study for the number of angles in the quadrature set. Recognizing that the high resolution mesh took way too long to solve even with a small number of angles, I opted for a mesh with 64,000 cells and kept the ray spacing to be half of a cell. Admittedly, this was still way too many cells and I spent a lot of time on this problem. The ground truth had 64 polar and 64 azimuthal angles and the solution is shown in Figure 8. Notably, with more angles the ray effects seem to disappear.

To test the convergence of each, I kept one angle set fixed at 64 angles and varied the other angle. That is, I had 64 polar angles while I varied the number of azimuthal angles and vice versa. This approach made calculating the order of convergence very time consuming. The orders of convergence with number of polar and number of azimuthal angles is shown in 9. The order of convergence is below one for each

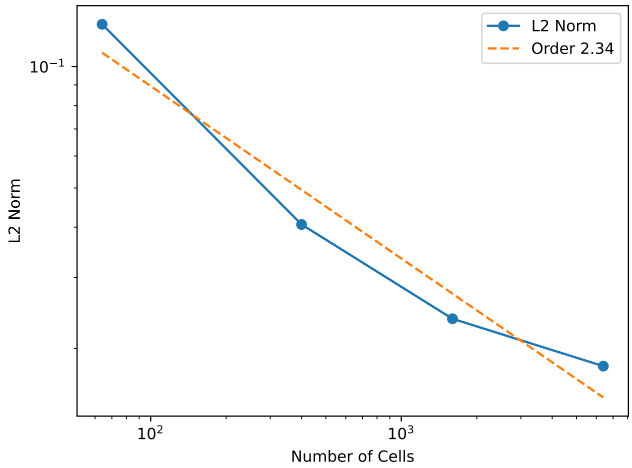


Figure 7: The order of convergence of my 2D MOC code with number of cells. The order of convergence I calculated was 2.34

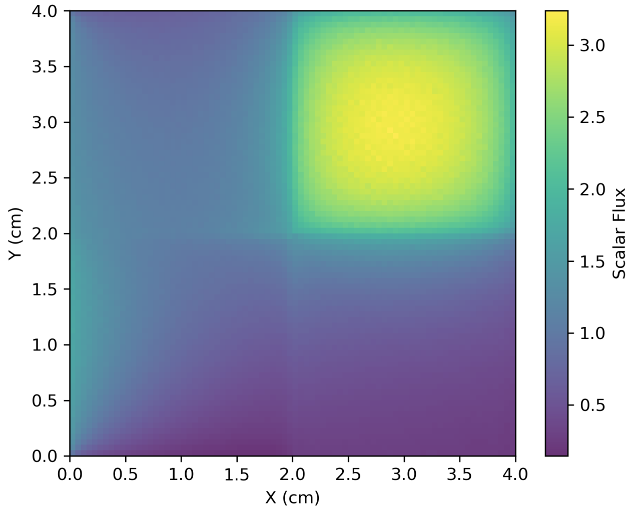
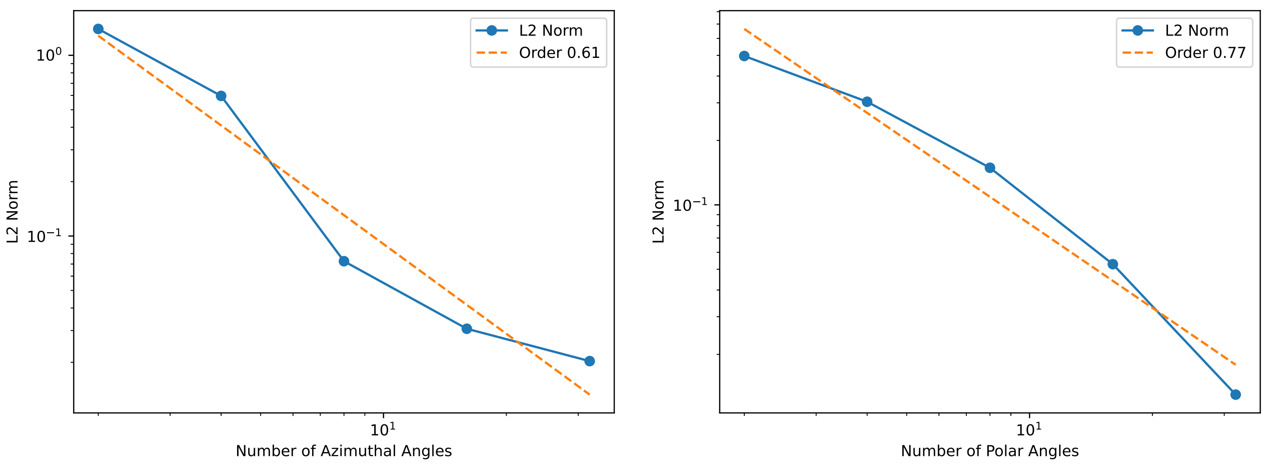


Figure 8: The high resolution solution used for angular order of convergence

angle, indicating that it is more worthwhile to increase the mesh resolution rather than increase the number of angles. I was surprised to find that polar angles increased convergence faster than azimuthal angles. I have a high resolution mesh for all problems, so the low number of azimuthal angles used in this study would lead to substantial ray effects. To get a better order of convergence, I would need to dynamically adjust the mesh resolution to avoid ray effects impacting the order of convergence.



(a) Azimuthal

(b) Polar

Figure 9: The error and order of convergence with number of angles in a product quadrature set. The order of convergence for both is approximately one.