

NE 255 - Numerical Simulations in Radiation Transport
Final Project
Due December 14, 2016

Below is a list of possible projects (there are Monte Carlo and deterministic options). If you intend to share a project among a team of students (maximum three students per team), check to ensure that the project has sufficient scope. The project is 30% of your grade and is due on Dec. 14 at the time of the final presentation. The following schedule will be imposed:

Oct. 25: Decide which project to work on; turn in list of team members (if applicable) and a one- or two-page *abstract*, including:

1. What you plan to do
2. Major steps to execute the project
3. Deadlines associated with each step
4. What you need to do to accomplish each step (laying out a path to success)
5. If in a team, the division of work

Nov. 15: Submit a written *interim* report (4, 6, or 8 pages maximum for 1, 2, or 3 people, respectively) explaining your project. See the project guidelines below for details of what to include.

Dec. 14: Presentations (time TBD, depending on projects and team sizes). See the project guidelines below for details of what to include.

Dec. 14: Final written reports (about 6-7 pages/team member as a rule of thumb) are due. See the rubric for details of what to include.

Potential project topics

1. Write a 2D transport solver that has vacuum boundaries on the bottom and left faces and reflecting boundaries on the top and right faces. I have more detailed specifications and some helpful tasks to facilitate completion if you choose this project.
2. Propose your own code project; this can include writing a code or studying and characterizing the way a code performs. Please use this as an opportunity to expand on or compliment work you are already doing.
3. Propose your own analysis project. Use a piece of transport software to study a problem. This project needs to focus more on what methods you use, why, and how well they work for the problem rather than on the analysis itself. Please use this as an opportunity to expand on or compliment work you are already doing.

PLEASE use version control for your project.

If you are comfortable writing your project in Python, consider using PyNE (<http://pyne.io>) to facilitate your project. Depending on what you do, we might be able to contribute your project back to the PyNE code base.

Project Guidelines

The final paper should be ~6-7 pages with 1.5 spacing per team member. This may vary based on the specific project; please use your best judgment (think of this as a technical report where you're laying out what you did so it can be reproduced and why what you did matters). Please include these items as clearly labeled sections in the FINAL report:

1. **Introduction:** What does the code you wrote do (or what is the problem you solved)? Provide an opening with high-level motivation and purpose. Also preview what you are going to talk about.
2. **Mathematics:** Write the continuous and discretized equations that you are solving, defining all terms. Include overviews of derivations needed to reach discretized equations as applicable. If this is very lengthy or a lot of information is needed, please use an appendix.
3. **Algorithms:** Include the algorithms that you implemented in your code. That is, how does your code do what it does? Include reasoning behind any important choices made (e.g. it handles multigroups this way because of X).
4. **Code Use:** Describe how to use your code, including inputs needed and output expected.
5. **Test Problems and Results:** Describe any testing you did to demonstrate your code is correct and present any results from test problems. This is verification and validation.
6. **Summary/Conclusion:** Wrap up: tell us what you told us and why that matters.
7. **References:** You must have references that you cite in your paper (including the interim report).

You must submit your code and at least one example input and corresponding output (I strongly encourage you to version control your code and submit access to the repository). Part of the project grade will be based on whether the code executes properly.

In your INTERIM report, *replace the “Test Problems and Results” section (and possibly also “Code Use” depending on how far you’ve gotten)* with **Plans for Completion** and keep in mind that these plans should include plans for what tests/inputs you will give to your code. The first four sections don't have to be completely polished, but they should at least be very solid drafts. I will provide feedback, so the better they are when I read them the more useful the feedback will be. This should be a maximum of 4, 6, or 8 pages with 1.5 spacing for 1, 2, or 3 people, respectively (varying depending on compactness of mathematics, algorithms, etc.).

Notes for writing papers properly:

- If you include figures, use a Figure number and caption; refer to the figure from within the text as Fig. # or Figure #.

- You may need to number equations and refer to them in the text.
- *Use section headings for the requested sections.*
- In the introduction, discuss what is coming up in the paper.
- In the conclusions, discuss what you told us in the paper.
- If you talk about a code (that you didn't write yourself), you need to include a reference for that code.
- For the final report, it's a good idea to include enough information for the work to be reproducible. To avoid making the report filled with mundane details you can put some items in an appendix or repository that you reference.
- Common grammar errors: that vs. which, use vs. utilize, due to vs. because of.