NE 155
Final Project Rubric

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.

Final report: Please include these items as clearly labeled sections

1. **Introduction:** What does the code you wrote do? 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 any derivations needed to reach discretized equations as applicable (can be included an the appendix if they are lengthy).
3. **Algorithms:** Include the algorithms that you implemented in your code.
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.
6. **References:** You must have references that you cite in your paper (including the interim report).

Products: 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.

Interim report: please 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.).

Final Presentation: please include these same items as the final paper in the final presentation. If you are using equations and algorithms we learned in class, just mention you used what we did in class and spend more time on how you implemented the code and how you tested it. The presentations should be approximately 6, 9, or 12 minutes for 1, 2, or 3 people, respectively. This may vary based on the specific project; please use your best judgment. Part of the final presentation should be a code execution demonstration.

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.

I will use the following rubric for evaluating the paper:

Category	**Possible Points**	**Earned Points**
Math, discretizations, and algorithms are correct	12	
Code input, output, and tests are well-designed	12	
Work implemented correctly	10	
Appropriate logical flow of paper	3	
Complete sentences; correct grammar and spelling	8	
Sources properly documented	5	
Total	50	

and for the presentation:

Category	**Possible Points**	**Earned Points**
Explanation of implemented code (math, discretizations, algorithms) is clear	8	
Code demonstration works and is understandable	8	
Good presentation skills: eye contact, volume, clarity of slides, etc.	6	
Appropriate presentation length	3	
Total	25	