# Final Project
## Math/CSCI 166, Fall 2022

**Due Date:** Friday, December 9, 2022

### Overview

The final project for Math/CSCI 166 is an opportunity for you to learn about a numerical algorithm/method that we did not cover in class. You can also take a deeper dive into a topic we covered in class.

The project requires you to choose a topic related to numerical analysis. You will research your topic and learn theory and some practical aspects. You will then write a "homework assignment" related to your topic, as if you were to assign it to one of your peers. You will also turn in solutions to your homework assignment, as well as a written narrative.

### Format

You will be turning in a document for your final project. I envision this being a written document, but I am open to other forms (consult with me first for approval). Your document does not necessarily need to be typed, but it should be neat, clearly organized, and easily readable. A portion of your grade on this final project will be a reflection on the organization and readability of your work. The description and requirements for your final project are below. Your final project should include each of the following clearly labeled sections:

(1) **Introduction** – A paragraph introducing the topic of your project. You should briefly explain your project and what you have done. This should include why your topic is interesting (are there applications?), and if there are direct connections to something we learned in class.

(2) **Theory** – This is the "research" portion of the project. Just like we learned theory for the methods we used in class, you should see what theory exists for your topic. The textbook for the class may be helpful, or other textbooks/internet might be more useful. In this section, you should explain the method, its math notation, and summarize any important theorems/results/proofs that are pertinent to understanding your topic. You should also comment on any practical considerations based on the theory. You can learn the theory for your project from other sources, but the write-up should be in your own words.

(3) **Write a Homework Assignment** – This is the "teaching" portion of the project. You have chosen a topic/method to learn about – now you will write a sample homework assignment. You should think to yourself, "I have just learned about a great numerical analysis topic! If I were to teach this topic to someone else, what homework would I assign that reinforced their learning?" Your homework assignment should consist of:
   (a) Two problems.
   (b) At least one of the problems should have multiple parts that are tied together in some way.
   (c) At least one of the problems should have a substantial programming component.

   You should put some thought in what an appropriate homework assignment would be. Your homework assignment should not be completely routine and too easy, nor should it be incredibly difficult to solve. Your homework problems should reinforce what you learned in your theory section. If it's helpful, you can use the homework assignments from class (or another class) as a model. You can look at other sources for inspiration for homework problems, but the problems you write should be your own.

(4) **Solutions** – You should turn in neat and organized solutions to the homework problems you wrote. This should contain code you wrote, including any tables, plots, or figures from the output of your code. Your solutions should come with full explanations, not just the answers. It is OK if you used available code for some portion of the project, but you should have a substantial contribution of code that you wrote yourself.

(5) **Homework Narrative Reflection** – A paragraph or two reflecting on and justifying your choice of homework problems. You should answer the following questions: Why did you choose the two problems you wrote? Did they serve a specific purpose in teaching someone about the theory or application (or both)? Do you think they were at a reasonable level? Would it help one of your peers reinforce their learning if they completed your homework problems?

(6) **Conclusion** – A brief paragraph summarizing your project. Were you surprised by anything you learned? Does this make you interested in a new question/topic/algorithm?

(7) **Bibliography** – List all references you used for your project. If you were inspired by any materials for your chosen homework problems, you should make specific note of it in this section. Your bibliography does not have to be in any specific format.

## Grading Rubric

The general grading rubric is as follows (a more specific one will be given at a later time):

- Presentation Quality (including organization, readability, introduction, conclusion, and bibliography) – 20%
- Theory (including quality of mathematical content and explanation of practical aspects) –35%
- Quality/Thoughtfulness of Homework Problems (including your homework narrative reflection)– 25%
- Solutions – 20%

## Potential Topics

There are many potential topics for this project. Your project can be an extension of something we learned in class, something related to what we learned in class, or it can be unrelated to any of the course topics (but still must be related to numerical analysis). If you are looking for topics, picking up the textbook and looking at chapters/sections we did not cover is a good place to start. Also, if you superficially learned about a method in another class and want to learn more, this is a great opportunity. I am also more than happy to help you come up with a topic. Potential topics include:

- Accelerating convergence of rootfinding algorithms.
- Algorithms for finding roots of polynomials.
- Other decompositions for matrices besides LU (Cholesky, QR, SVD...)
- Iterative solvers for linear systems (Jacobi method, Gauss-Seidel, Conjugate gradient and variants...)

- Algorithms for solving nonlinear systems of equations.
- Preconditioning for faster iterative linear solvers.
- Regularization of linear systems (handling noise).
- Numerically finding eigenvalues/eigenvectors.
- Polynomial interpolation (besides what we will cover in class).
- Numerical integration topics (adaptive quadrature, improper integrals, 2D integration,...)
- Solving ODEs numerically (besides Euler's method)
- Solving PDEs numerically

Almost all of these topics have at least some coverage in the course textbook. We will most likely touch on all of these topics in class except eigenvalues/eigenvectors, and solving ODE/PDEs.