MATH 566
Final Project

Description

The purpose of the final project is for you to investigate a problem that interests you and where numerical linear algebra plays a central role. For example, You may be interested in exploring some particular algorithm that we did not cover in the course or that was given some cursory attention. This would involve a description of the algorithm, examples of its use, and implementation. Alternatively, you may be interested in solving some scientific problem where some aspect of numerical linear algebra is central. This would involve a thorough description of the problem, what numerical methods are needed, code that solves the problem, and numerical results. Another possibility is that you may be interested in high performance implementations of some numerical linear algebra algorithms. A list of some ideas is given at the bottom of this page

Important Dates

- Project proposal: April 1, 2021
- Final project presentation: May 4, 2021 14:30–16:30
- Final project report: May 5, 2021

Proposal

You will submit a typewritten proposal for your final project on April 1. The proposal should contain the following information:

1. A description of the problem you wish to investigate for your project;
2. The numerical methods you plan on using, and a short explanation as to why you are choosing these methods;
3. The primary references you plan to work from.

I will read through your proposals to make sure that the project is doable and interesting. If it is not, then I'll meet with you to discuss how to refine your ideas to better align with the intended purpose of the project.

Project Report

The report for your final project will be typewritten and anywhere from five to fifteen pages long, excluding any code you wish to include in the report. The report should consist of the following information

1. **Introduction:** An overview of the problem you are investigating including some background information for readers who may not be familiar with the problem. If applicable, incorporate some historical background about the problem, including where the problem arises and other people who have worked on it.
2. **Formulation:** An explanation of the details of the problem, what numerical methods you used to investigate the problem, why you chose these methods. The "why" portion should tell about the accuracy and computational cost associated with the methods, and when the method is appropriate to use. You *must* clearly cite all sources you used in the project.

3. **Results:** The results of your investigation of the problem. This should include output from your program in nice tables and/or nice figures. You should also include some convincing data that verifies your method is actually working. This can be done by comparing the output of your program for a problem with a known solution.
4. **Conclusion:** A summary of your project, including what you have learned from your investigation, what more could be learned, any limitations of the methods you used for your investigation, and possible ways improve your methods.
5. **References:** A list of all the references you used in your project.
6. **Appendix:** A listing of your programs used for the investigation. The programs should be well documented and available in electronic form if requested.

Project Presentation

You will be required to give a 15 minute oral online presentation to the class about their project on May 4. The presentation should be structured like the final report and should use slides.

Grading

The final project is worth 30% of your final grade. The breakdown for is as follows:

- Proposal: 3%
- Report: 18%
- Presentation: 9%

Possible topics (in no particular order)

1. Multigrid for solving elliptic differential equations
2. FEAST eigenvalue solver
3. Spectral graph partitioning
4. Moving least squares approximation
5. Computing Gaussian quadrature rules using companion, colleague, and comrade matrices
6. Ranking algorithms (e.g. Google's PageRank or sports rankings)
7. Classification algorithms
8. Tensor decompositions
9. Random matrix theory
10. Randomized algorithms for QR or SVD
11. Low rank approximations like CDR and how these compare to principal component analysis
12. Deep learning
13. Inverse problems - solving rank deficient linear systems
14. Pseudospectra
15. Gaussian elimination in parallel with *tournament pivoting*
16. Communication avoiding algorithms
17. Matrix completion problems
18. Netflix challenge
19. Decompositions of block mirror centrosymmetric matrices (BMC)
20. Preconditioners for Krylov subspace methods (e.g. Incomplete LU (ILU) factorizations)