Topic Title
COS 574 - Explorations in Numerical Analysis February 7, 2024

Directions: Graduate students taking the class for COS 574 credit are required to complete a project. The project will explore a numerical (linear algebra) topic in more depth. Projects should have sufficient complexity to merit graduate-level work. The project requires a computational component demonstrating proficiency in programming languages following coding best practices. The write-up should provide background concepts, motivating context, documentation of data sources, code, results, analysis, and conclusions. The report needs to be written in LaTeX. Images contained in figures should be at least 300-600 dpi. The final project, including all source code, test matrices, and LaTeX must be available on a GitHub repository.

Project Ideas

1. Implement iterative methods for solving large sparse linear systems (conjugate gradient, GMRES, BiCGSTAB) and study convergence rates for different conditions.
2. Develop parallel algorithms for matrix computations, such as LU decomposition or singular value decomposition, using technologies like OpenMP or MPI.
3. Create a program to analyze the effects of floating point precision and rounding error on linear algebra algorithms. How does precision affect stability, accuracy, and efficiency?
4. Implement and test different preconditioning techniques (Jacobi, Gauss-Seidel, incomplete LU/Cholesky) for the conjugate gradient method. Analyze the impact on convergence and error.
5. Develop a program to compute incomplete LU factorizations for use as preconditioners—test performance on sparse matrices from real applications. Test on several matrices.
6. Compare direct solvers versus preconditioned iterative solvers for certain linear systems. Implement both types of methods.
7. Implement a parallel sparse approximate inverse preconditioner using a multicore or GPU environment. Analyze speedup over the sequential version.
8. Explore combining different preconditioners, estimating optimal parameters, using variable preconditioners, and testing linear systems from applied areas like finite elements or networks.
9. Implement structured eigenvalue solvers for special matrix types like Hessenberg, tridiagonal, and banded. Compare QR, divide-and-conquer, and inverse iteration approaches.
10. Analyze mixed precision algorithms that combine single and double precision to improve the efficiency of linear solvers while maintaining accuracy.
11. Develop optimized GPU-accelerated solvers for sparse linear systems using libraries like cuSPARSE.
12. Select a project related to your interest(s). Project must include a scientific computing component and related analysis of the methods.