Numerical Linear Algebra Project Description

This project is designed to deepen your understanding of numerical linear algebra through problem formulation, algorithm design, and implementation. In addition, it provides an opportunity to practice presentation skills through a short oral talk at the end of this semester. The primary objectives are:

1. Develop a comprehensive understanding of an algorithm discussed in class, including its mathematical foundation, assumptions, advantages, and limitations.
2. Design and implement an algorithm from scratch for a nontrivial problem in numerical linear algebra. Understand and justify every step of the algorithm.
3. Clearly explain your problem, methodology, and results in both a written report and a short oral presentation.

Each student should select **one** of the following two project directions (do *not* attempt both).

Option 1 (solving linear system). Design and implement a numerical algorithm to solve a linear system

$$\mathbf{A}\boldsymbol{x} = \boldsymbol{b},$$

where the matrix $\mathbf{A}$ has size at least several hundreds. Larger systems are welcome if computationally feasible.

Option 2 (compute eigenvalue and eigenvector). Design and implement a numerical algorithm to compute eigenvalues and eigenvectors of a matrix $\mathbf{A}$ of size at least 50×50.

The relevant numerical algorithms you may consider include, but are not limited to, direct methods such as LU decomposition with pivoting and Cholesky decomposition (when applicable), as well as iterative methods such as the Jacobi method, Gauss–Seidel method, or the Conjugate Gradient method for solving linear systems. For eigenvalue and eigenvector computations, appropriate choices include the power method or the QR algorithm, etc. In general, any numerical algorithm for linear systems or eigenvalue problems that has been discussed in this course is acceptable, provided it is implemented and justified appropriately.

Programming requirements. For implementing your own numerical algorithm, you may use MATLAB, Python, or any programming language of your choice. Basic operations such as matrix–vector multiplication, vector norms, plotting, and timing functions are allowed.

You must write your own solver. Calling the built-in solvers or high-level numerical package for solving linear systems or eigenvalue problems (e.g., `A\b`, `eig`, `numpy.linalg.solve`, `scipy.linalg.eig`) is **not allowed**.

After obtaining results from your own solver, you may compare them with those produced by built-in routines **solely for validation purposes**. Your code should be clearly structured, well-documented, and reproducible.

1 Matrix design and project proposal – due March 6, 2026

The choice of a suitable question to solve is a critical factor in the success of the final project. Students are strongly encouraged to design the matrix $\mathbf{A}$ themselves. By March 6 (Friday), each student must email a project plan to the instructor. Completing this step earlier is highly encouraged. The project plan should clearly address the following aspects:

1. **Problem description**: specify whether you are solving a linear system (option 1) or an eigenvalue problem (option 2).
2. **Motivation and interest**: explain why the selected problem and matrix are interesting. The problem should be neither trivial nor excessively challenging.
3. **Matrix properties**: summarize the key properties of the matrix, such as whether it is symmetric or Hermitian, positive definite or indefinite, sparse (contains a lot of zeros) or dense, and any other relevant features.
4. **Feasibility**: clarify the size of your matrix. If solving a linear system, justify why the matrix $\mathbf{A}$ is invertible. Try your best to explain why the algorithm (plan to choose) is suitable for your question.

If a student is unable to design a suitable matrix or misses the deadline, the instructor will provide a matrix. In this case, the student must use the assigned matrix for the project.

2 Project presentation – before the end of this semester

Near the end of the semester, each student will give an oral presentation of approximately 6 minutes, followed by 1–2 minutes of questions and discussion. Slides are the preferred medium for communicating information clearly and effectively. The following information is recommended to be included in the slides.

- Clearly explain the problem and the design of the matrix
- Describe the numerical algorithm and justify its advantages
- Highlight your key numerical results, observations, and validations
- Summarize to demonstrate understanding of both the theory and implementation

A poorly organized or unclear presentation may result in a reduction of the final project score.

3 Final report – due May 1, 2026

The final report should provide a clear and comprehensive account of the project, beginning with a description of the problem and the design of the chosen matrix, including its key properties such as symmetry, positive definiteness, sparsity, or other feathers. Students should describe their algorithm clearly, justify its advantages, and present implementation details along with verification of correctness. Performance evaluation, including timing and other observations, should be included, followed by a brief summary of findings and reflections on both theoretical and computational aspects. Reports are expected to be more than one page, typically around two pages, and should not exceed four pages.

Submission deadline: The final report must be emailed to the instructor **before 5:00pm on May 1, 2026**. A poorly organized report may result in a reduction of the final project score.