Fast Iterative Solvers. Project 1.

Summer Semester 2025

Douglas R.Q. Pacheco

Due: Monday 16th June, 2025, 23:59 hrs.

The project may be completed individually or in pairs. However, if it is developed in pairs, ***each participant must submit their own copy*** *of the project to Moodle, including both names on the submission.*

Overview

The main objective of this project is to identify and evaluate the variations among different Krylov subspace approaches in a given application and their behavior under specific circumstances. Its main objectives are understanding runtime, the orthogonality of the Krylov subspace, the number of iterations (i.e., Krylov vectors generated), and the effect of preconditioning on the problem.

You are asked to implement the following Krylov subspace methods:

- GMRES with preconditioner,
- Conjugate Gradient (CG) method,

to solve a linear system $\mathsf{Ax} = \mathsf{b}$, where A is a real-valued square matrix (symmetric positive definite in the case of CG), and b is a given vector.

General Instructions

Download the matrices

- `gmres_matrix_msr_1` (non-symmetric and indefinite)
- `cg_matrix_msr_1` and `cg_matrix_msr_2` (symmetric positive definite)

from the project page on Moodle. These matrices are stored in *modified compressed sparse row* (MSR) format. Some annotations to help read the files are provided along with the assignment.

For all tests, you should use the following setup:

- Prescribe the solution vector $\mathsf{x} = (1, 1, \ldots, 1)^\top$, and determine the corresponding right-hand side $\mathsf{b} = \mathsf{Ax}$.
- Use the initial guess $\mathsf{x}_0 = \mathsf{0}$.
- Use a tolerance of $\|\mathsf{r}_k\|_2/\|\mathsf{r}_0\|_2 = 10^{-8}$ to establish convergence. Whenever the relative residual drops below this value, we consider the iteration to be converged.
- **Important**: Based on the previous point, for preconditioned GMRES, the residual is also preconditioned, i.e., $\mathsf{r}_k = \mathsf{M}^{-1}(\mathsf{b} - \mathsf{Ax}_k)$. For restarted GMRES, the iteration index k represents the cumulative iteration count, that is, the total number of Krylov vectors generated.

As we determine the right-hand-side b such that it corresponds to a known solution x, we can also compute the error $\mathsf{e}_k = \mathsf{x}_k - \mathsf{x}$ at each iteration k.

Specific Instructions

GMRES Method

- The GMRES algorithm should be implemented in restarted formulation GMRES(m). Full GMRES can be tested by choosing the restart parameter large enough. (For the present project, $m = 600$ will be enough.)
- Apply **left** preconditioning to the GMRES procedure. Implement the following options:
  1. Jacobi preconditioning;
  2. Gauss–Seidel preconditioning;
  3. ILU(0) preconditioning.
- For the full GMRES method, with and without preconditioning, plot the relative residual against iteration index on a semi-log scale. *Compare and analyze the results obtained based on how many Krylov vectors were needed to converge in each case.*
- For GMRES(m), *test and compare the following restart parameters* $m = 10, 50, 200$ *with the runtime required by the full* GMRES.

  Hints: Does the runtime increase or decrease compared to full GMRES? If the runtime changes, why does this happen? What factors, other than runtime, might motivate the use of restarts instead of full GMRES?

  Optional: Recommend a restart parameter and justify your choice.
- For full GMRES (without preconditioning): *check the orthogonality of the Krylov vectors.* **Hints:** Plot the computed values of $\mathsf{v}_1 \cdot \mathsf{v}_k$ against k on a semi-log scale.

Conjugate Gradient Method

- The CG method should be implemented as discussed in class. **You will not implement preconditioning for this case.**
- Plot the error in A-norm, i.e., $\|\mathsf{e}\|_{\mathsf{A}} = \sqrt{\langle \mathsf{Ae}, \mathsf{e} \rangle}$, as well as the residual in standard 2-norm, i.e., $\|\mathsf{r}\|_2 = \sqrt{\langle \mathsf{r}, \mathsf{r} \rangle}$, against iteration index on a semi-log scale for both cases. *Compare qualitatively the difference in convergence between* $\|\mathsf{e}\|_{\mathsf{A}}$ *and* $\|\mathsf{r}\|_2$*. Comment on what you can observe.*

  Compare the differences in convergence between the two CG cases. Comment what is happening, and why.

Report

You should write a short report that addresses all the points raised in the previous section. **This report should not exceed four pages (excluding figures).**

Recommended Workflow

MSR format reader

- Write a function that reads matrices in MSR format. Download the matrices `msr_test_non_symmetric.txt` and `msr_test_symmetric.txt` for testing your function.
- This should work for both symmetric and non-symmetric matrices.
- For symmetric matrices, just one triangle is stored (lower part).

GMRES(m)

- Implement the restarted GMRES(m) method without preconditioning (This will be added later). Recommendation, use the algorithm derived in class.
- It is recommended to follow these steps for the GMRES(m) programming:
  - Implement the Gram-Schmidt procedure. (Check Orthogonality of the new basis vector.)
  - Integrate Givens rotation into the algorithm.
  - Implement a backward substitution algorithm to get the coefficient vector $\mathbf{y}^*$ (which is used to assemble the solution).
  - You can use a small matrix (e.g., a 3×3 matrix, or the one provided as example) to verify your algorithm step by step using the analytical procedure.
  - Another test you may want to do: It is relatively easy to implement the Minimal Residual (MR) method discussed in class. GMRES(1) should give the same residuals (up to machine accuracy) as MR.

Conjugate Gradients (CG)

- Implement the Conjugate gradients method **without** preconditioning. Recommendation, use the algorithm derived in class.
- You can use a small matrix (e.g., a 3×3 matrix, or the one provided as example) to verify your algorithm step by step using the analytical procedure.
- Note that for a small, well-conditioned $n \times n$ matrix the algorithm should give you the exact solution, at least up to machine precision, in n steps.
- On the other hand, you will find that for the large (sparse) matrix used for the assignment, that property is lost, due to round-off errors!

GMRES(m) with preconditioners

- Apply the preconditioner to the GMRES(m) method. (A modification is required in the algorithm, just as seen in the lecture.)
- Implement the Jacobi preconditioner.
- Implement the Gauss-Seidel preconditioner.
- Implement the ILU(0) preconditioner. Follow the algorithm shown in the lecture.