AE227: Numerical fluid flow Assignment 1

Manish Sharma

March 2, 2025

Problem Statement

Any system of linear equations can be represented as a Matrix-Vector equation. Consider such a system of linear equations expressed in the standard form:

$$A\mathbf{u} = \mathbf{b} \tag{1}$$

where A is a full-rank square matrix, $\mathbf{b}$ is the known right-hand-side (RHS) vector, and $\mathbf{u}$ is the solution vector.In this assignment, you are required to find u using different methods as described in class for given A and b using Jacobi, Gauss-Seidel (GS), Steepest Gradient Descent (SGD) and Conjugate Gradient (CG) iterative methods with the same initial guess of

$$\mathbf{u}^{(0)} = [1, 1, 1, \dots, 1]^T \tag{2}$$

Please note that the iterations should be continued until the relative change in the solution of system of linear equations (Au = b) from one iteration to another is less than 10^{-13} . More precisely, stop the iterations when

$$\frac{\|\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\|_2}{\|\mathbf{u}^{(k)}\|_2} \leq 10^{-13} \tag{3}$$

The aforementioned A and $\mathbf{b}$ are available for download at folder Assignment 1 at https://github.com/hkishnani/Numerical_Fluid_Flow_AE227_TA.git. You only need to solve for A and $\mathbf{b}$ with filenames: `A_20.txt`, `A_160.txt`, and `A_320_pentadiag.txt` and their corresponding $\mathbf{b}$'s.

Submit a short report, which contains the following deliverables for different cases of A and $\mathbf{b}$ to find $\mathbf{u}$:

1. **Perform the iterations for Jacobi, GS, SGD and CG methods until the convergence criteria (Eq. 2) is met. For a given linear system, plot the relative change in the solution (LHS of Eq. 2) versus the iteration count (k) for all four methods in one figure. Repeat the procedure for all three linear systems (three different plots). In the plot, the relative change in the solution (y-axis) should be in base-10 logarithmic scale (For example, see the command "semilogy" in Matlab) but iteration count (x-axis) must be on linear scale. Write the inferences from the plot based on your understanding.**

   Given linear systems ($Au = b$) are

   - `A_20.txt` - $20 * 20$ matrix
   - `A_160.txt` - 160 * 160 matrix
   - `A_320_pentadiag.txt` - 320 * 320 pentadiagonal matrix

   By plotting the relative change in the solution against the iteration count for Jacobi,GS,SGD and CG methods, we can visually compare their convergence rates.

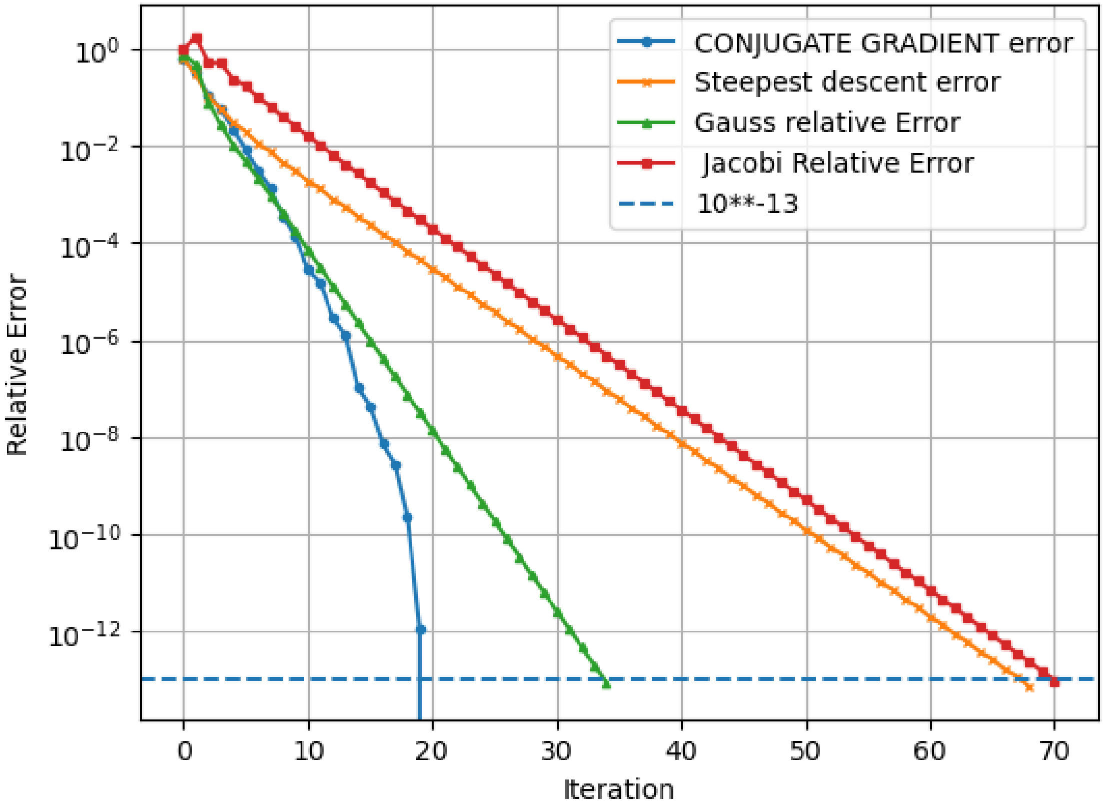

Figure 1: relative error against iteartions for `A_20.txt`

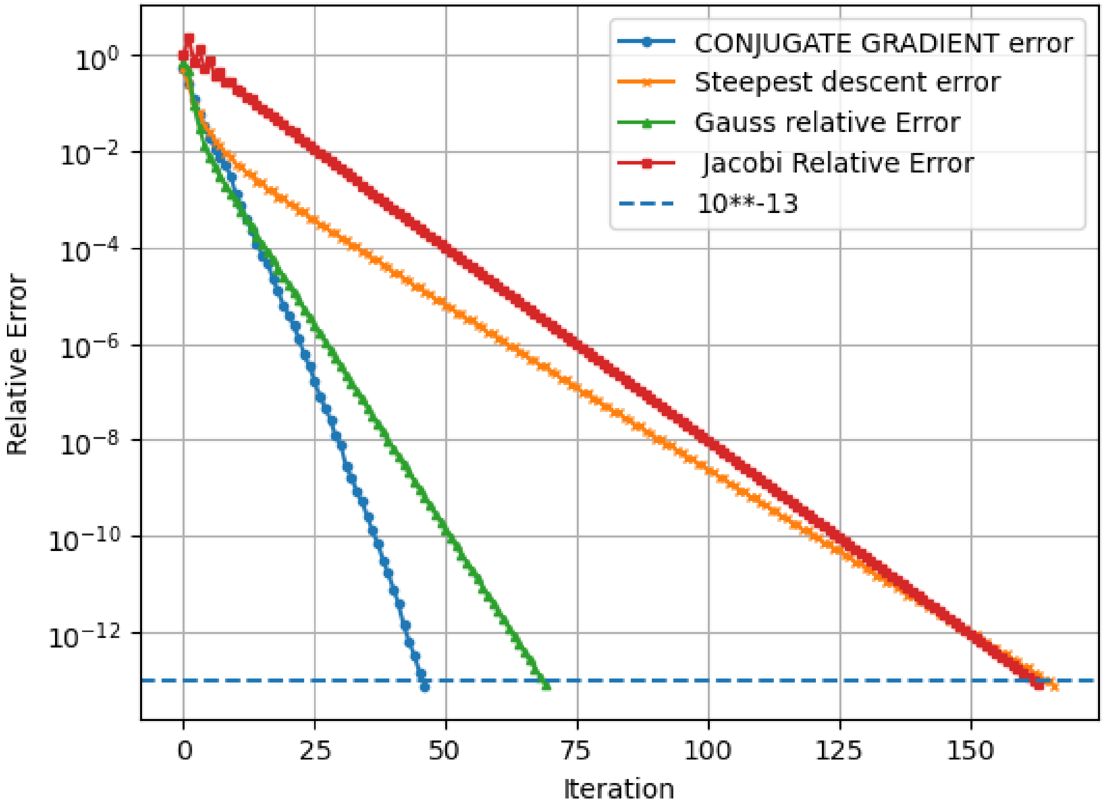

Figure 2: relative error against iteartions for `A_160.txt`

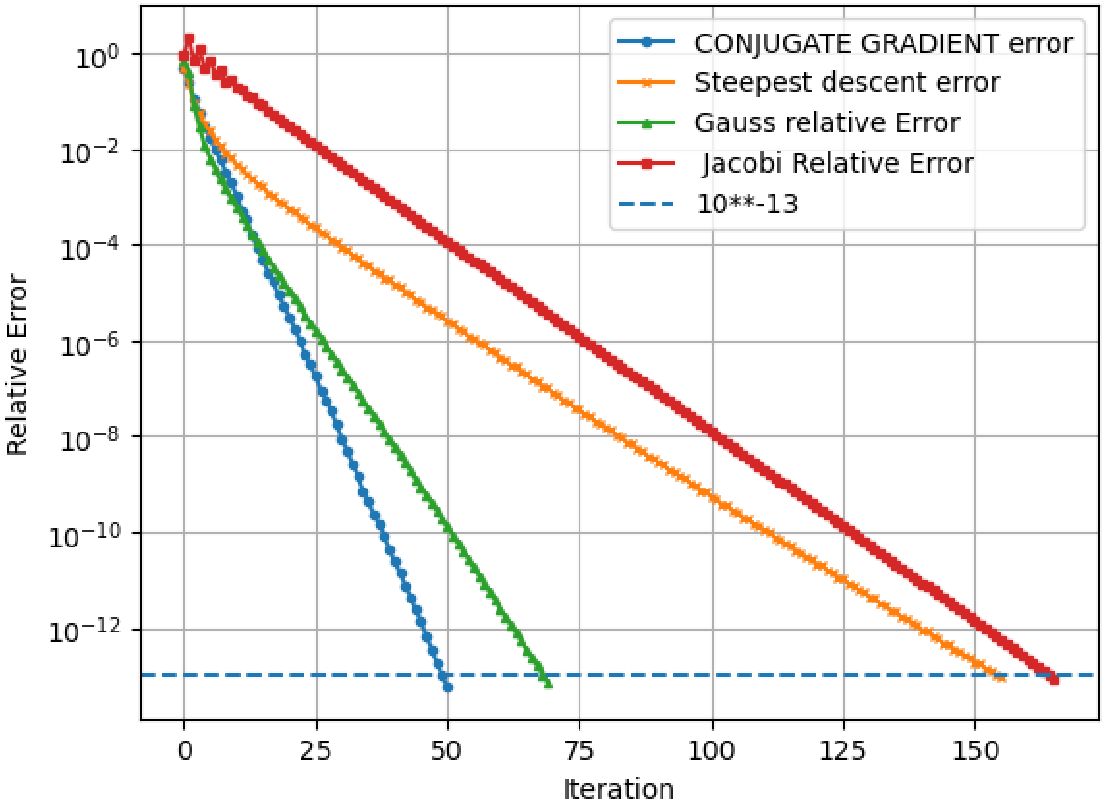

Figure 3: relative error against iteartions for `A_320_pentadiag.txt`

We can see as size of matrix increases from 20 to 160, iterations increase but for a pentadiagonal matrix of size 320 iterations are comapratively near to `A_160.txt` matrix. This is because `A_320_pentadiag.txt` is a special matrix.

A **pentadiagonal matrix** is a square matrix that has non-zero elements only on the main diagonal, the two diagonals directly above the main diagonal, and the two diagonals directly below the main diagonal. All the other elements are zero i.e it is **sparse**.

Comparing for all the three linear sysytems we can see that Jacobi converges the slowest and the conjugate gradient converges faster.

Gauss seidel also comparatively converges faster than Jacobi and Steepest descent method.

Steepest descent takes almost equal steps as Jacobi meethod.

2. **Report the optimal method for which the number of iterations required to reach the convergence criteria (Eq. 2) is minimum for each linear system.**

After comparing plots of relative error against iteartions we can see Conjugate gradient is the optimal method to reach the convergence criteria .

$$\frac{\|\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\|_2}{\|\mathbf{u}^{(k)}\|_2} \leq 10^{-13} \tag{4}$$

For all the three linear systems Conjugate gradient takes least iterations to solve the system. Iteration count is not more than size of the matrix A. This convergence is because of search direction and line search algorithm of Conjugate gradient method.

3. **In a table, report the number of iterations required corresponding to Jacobi, GS, SGD, and CG methods for all three linear systems to reach the convergence criteria (Eq. 2). Based on your understanding, explain the difference in convergence rates for different methods for a given linear system.**

All the iteration counts are from (i+1) i.e 1,2,3.....n iteartions.

Matrix A	`A_20.txt`	`A_160.txt`	`A_320_pentadiag.txt`
Jacobi	166	164	71
Gauss Seidel	70	70	35
Steepest Descent	156	167	69
Conjugate Gradient	51	47	19

Table 1: Numnber of Iterations required coresponding to Jacobi,GS,SGD and CG methods

Convergence can be influenced by several factors

- Diagonal Dominance - Methods like Jacobi and Gauss-Seidel converge faster if the matrix **A** is diagonally dominant.
- Symmetry and Positive Definiteness - For methods like the Conjugate Gradient, **A** being symmetric and positive definite ensures faster convergence.
- Initial Guess: The choice of the initial guess $\mathbf{u}^{(0)}$ can influence the number of iterations needed for convergence. A guess closer to the actual solution can result in faster convergence.
- Condition Number - The condition number of the matrix **A** affects convergence. A lower condition number generally leads to faster convergence.
- Sparsity - Sparse matrices, where most elements are zero, can improve the efficiency and sometimes the convergence rate of iterative methods.

(a) **Convergence of Jacobi:** Jacobi Method is the slowest method to solve a given linear system.

- The Jacobi method converges faster if the matrix **A** is diagonally dominant. This means that for each row i,

$$|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|,$$

where a_{ii} is the diagonal element and a_{ij} are the off-diagonal elements in row i. Diagonal dominance ensures that the influence of off-diagonal elements is small compared to the diagonal element.

- The spectral radius $\rho(\mathbf{D}^{-1}(\mathbf{L}+\mathbf{U}))$ of the iteration matrix $\mathbf{D}^{-1}(\mathbf{L}+\mathbf{U})$ must be less than 1 for the method to converge, where $\mathbf{D}$ is the diagonal part of $\mathbf{A}$, and $\mathbf{L}$ and $\mathbf{U}$ are the strictly lower and upper triangular parts of $\mathbf{A}$, respectively.

(b) **Convergence of Gauss Seidel:** It is updated method of Jacobi method. Unlike Jacobi method,Gauss seidel uses new values i.e most updated value for an unknown x_j for next iterations.

$$|x_i^{k+1}| = \frac{1}{a_{ii}}(b_i - \sum_{j=1}^{i-1} |a_{ij} * x_j^{k+1}| - \sum_{j=i+1}^{n} |a_{ij} * x_j^{k}|,$$

- The Gauss-Seidel method converges faster if the matrix **A** is diagonally dominant. This means that for each row i,

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|,$$

where a_{ii} is the diagonal element and a_{ij} are the off-diagonal elements in row i. Diagonal dominance ensures that the influence of off-diagonal elements is small compared to the diagonal element.

- The spectral radius $\rho((\mathbf{D}+\mathbf{L})^{-1}(\mathbf{U}))$ of the iteration matrix $(\mathbf{D}+\mathbf{L})^{-1}(\mathbf{U})$ must be less than 1 for the method to converge, where $\mathbf{D}$ is the diagonal part of $\mathbf{A}$, and $\mathbf{L}$ and $\mathbf{U}$ are the strictly lower and upper triangular parts of $\mathbf{A}$, respectively. Gauss seidel Method converges faster than Jacobi : Consider $\rho(\mathbf{G_J})$ and $\rho(\mathbf{G_G S})$ be spectral radius of jacobi and gauss seidel then we have relation

$$\rho(\mathbf{G_J})^2 = \rho(\mathbf{G_G S}) \tag{5}$$

This means that one Gauss-Seidel iteration is about as good as two Jacobi iterations; Gauss-Seidel converges twice as fast as Jacobi.

(c) Convergence of Steepest descent:

- **Condition Number**: The condition number of the matrix $\mathbf{A}$ affects the convergence rate. A lower condition number generally leads to faster convergence. The condition number is given by

$$\kappa(\mathbf{A}) = \|\mathbf{A}\| \|\mathbf{A}^{-1}\|,$$

where $\|\mathbf{A}\|$ is a matrix norm.

- **Symmetry and Positive Definiteness**: The Steepest Gradient Descent method converges faster if the matrix $\mathbf{A}$ is symmetric and positive definite. These properties ensure that the objective function is convex.
- **Gradient Magnitude**: The magnitude and direction of the gradient influence the convergence rate. Smaller gradients may lead to slower convergence, especially near the minimum.

The step from $x^{(k)}$ to $\mathrm{x}^{(k+1)}$ has two ingredients:

i. Choice of a search direction.

ii. A line search in the chosen direction.

Choosing a search direction amounts to choosing a vector $\mathbf{p}$ that indicates the direction in which we will travel to get from $\mathbf{x}^{(k)}$ to $\mathbf{x}^{(k+1)}$. Once a search direction has been chosen, $\mathbf{x}^{(k+1)}$ will be chosen to be a point on the line $\{\mathbf{x}^{(k)} + \alpha \mathbf{p} \mid \alpha \in \mathbb{R}\}$. Thus we will have

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)}$$

for some real α_k. The process of choosing α_k from among all $\alpha \in \mathbb{R}$ is the **line search**. We want to choose α_k in such a way that $J(\mathbf{x}^{(k+1)}) < J(\mathbf{x}^{(k)})$. One way to ensure this is to choose α_k so that

$$J(\mathbf{x}^{(k+1)}) = \min_{\alpha \in \mathbb{R}} J(\mathbf{x}^{(k)} + \alpha \mathbf{p}^{(k)}).$$

If α_k is chosen in this way, we say that the line search is exact.

The method of steepest descent takes $\mathbf{p} = \mathbf{r}$ and performs exact line searches. Since $\mathbf{r} = -\nabla J(\mathbf{x})$, the search direction is the direction of steepest descent of J from the point $\mathbf{x}$. The correct value of α can be obtained from a formula.

$$\alpha_k = \frac{(p^k)^T r^k}{(p^k)^T A p^k} \tag{6}$$

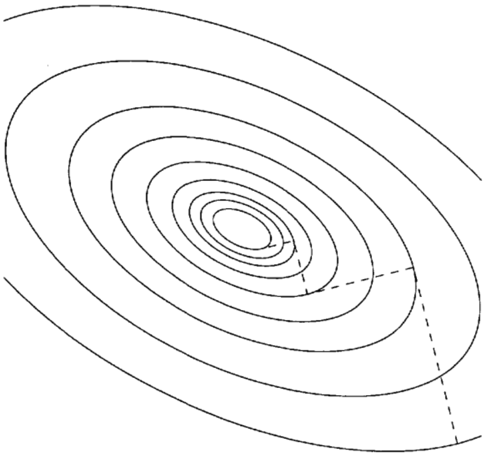

Fig. 7.4 Steepest descent in the 2 × 2 case

Geometric interpretation of steepest descent: The objective of a descent method is to minimize the function J(y) . From (7.4.3) we know that J has the form

$$J(y) = \frac{1}{2}(y-x)^T A(y-x) - \frac{1}{2}(x^T Ax) \tag{7}$$

where x is the solution of Ax = b. The contours of J are ellipses.
The dotted lines in Figure 7.4 represent four steps of the steepest descent algorithm. From a given point, the search proceeds in the direction of steepest descent, which is orthogonal to the contour line (the direction of no descent). The exact line search follows the search line to the point at which J is minimized. J decreases as long as the search line cuts through the contours. The minimum occurs at the point at which the search line is tangent to a contour. (After that, J begins to increase.) Since the next search direction will be orthogonal to the contour at that point, **we see that each search direction is orthogonal to the previous one**. Thus the search bounces back and forth in the canyon formed by the function J(y) and proceeds steadily toward the minimum.

$$\mathbf{p^{k+1}} \perp \mathbf{p^k}; [p = -\Delta J] - \text{direction of search} \tag{8}$$

(d) **Convergence of Conjugate gradient:**. The conjugate-gradient (CG) method is a simple variation on steepest descent that performs better because it has a memory. The computation of a is organized a bit differently, but this difference is cosmetic. The line searches are still exact; the CG algorithm is an instance ofAlgorithm 7.4.14. **Initially $p \leftarrow r$, so the first step is steepest descent**. On subsequent steps there is a difference. Instead of $p \leftarrow r$, we have $p \leftarrow r + \beta p$. The residual or steepest descent direction still plays an important role in determining the new search direction, but now the old search direction also matters. This is the one point at which memory of past iterations is used. This slight change makes a huge difference.

$$\beta_k = \frac{(r^{k+1})^T r^{k+1}}{(r^k)^T r^k} \tag{9}$$

$$p^{k+1} = r^{k+1} + \beta_k p^k \tag{10}$$

We see that the CG algorithm is far superior to steepest descent.

The CG method generates search directions that are A -orthogonal (i.e., conjugate). This orthogonality ensures that each search direction is independent of the previous ones, leading to more efficient progress towards the solution.

$$p^k \perp p^{k-1} \perp p^{k-2} ..... \perp p^0 \tag{11}$$

$$(p^k)^T A p^i = 0 \tag{12}$$

This is called A- orthogonality, p^k is A- conjugate to all previous directions.

The CG algorithm, applied to an n x n positive definite system Ax = b, arrives at the exact solution in n or fewer steps.