AE227: Numerical fluid flow Assignment 2

Manish Sharma

March 24, 2025

Problem: Solving 2D Poisson's problem using Jacobi, GS, SGD, and CG methods

Consider the 2D Poisson's equation in the domain $\Omega = [0,1] \times [0,1]$, the unit square:

$$-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = f \tag{1}$$

with the boundary conditions

$$u = g \quad \text{on } \partial\Omega \tag{2}$$

where f and g are given functions, and $\partial\Omega$ represents the boundary of Ω. Eq. 1 can be discretized using the centered Finite difference method (as explained in the class). After discretizing the Eq. 1. you will get a linear system of the form Au=b. Here f = 0, and g is given as,

$$g(x,y) = \begin{cases} 0 & \text{if } x = 0, \\ y & \text{if } x = 1, \\ (x-1)\sin x & \text{if } y = 0, \\ x(2-x) & \text{if } y = 1. \end{cases}$$

Consider three mesh intervals: h1 = 1/10,h2 = 1/20 and h3 = 1/40, used to discretize the Eq. 1., resulting in three different linear systems

Solve the obtained linear systems using Jacobi, Gauss-Seidel, Steepest Gradient Descent, and Conjugate Gradient methods. Take the initial guess as u(0) = 0 for all four iterative methods.

The iterations should be continued until the relative change in the solution u from one iteration to another is less than 10^{-10}. More precisely, stop the iterations when

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

Submit a short report, which has the following:

1. **For a given mesh size h, plot the relative error (LHS of 3) versus the iteration index (k) for all four iterative methods. Repeat the same for all three mesh sizes (a total of three figures). In the plot, the relative error (y-axis) should be in base-10 logarithmic scale, and the iteration count (x-axis) must be on a linear scale. Write the inferences from the plots based on your understanding**

   Given 2D poissons equation can be written, after substituting f=0 as:

$$\Delta^2 = 0 \tag{4}$$

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \tag{5}$$

   (a) As the Laplace equation is typical for diffusion phenomena, it has to be discretized with central differences, in order for the discretization to be consistent with the physics it simulates.

   Applying centered finite difference to the above eqaution with respect to x and y as:

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta y^2} + O(\Delta x^2, \Delta y^2) = f_{i,j} = 0 \tag{6}$$

   It is a well known *five point difference operator* and is of second order accuracy. for $\Delta x = \Delta y$ (i.e) uniform mesh.

$$u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j} = f_{i,j} \tag{7}$$

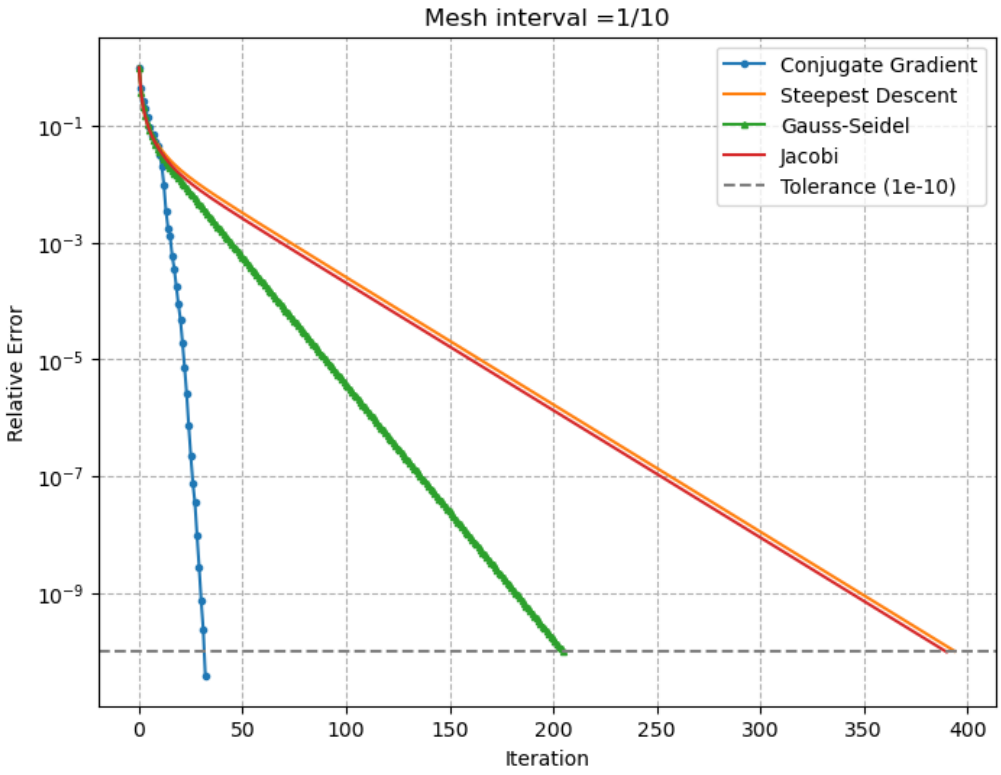

Figure 1: relative error against iteartions for Mesh size=1/10

   (b) **Discretize the Domain**

   We divid ethe domain [0,1] × [1,0] into a uniform grid with spacing h(mesh interval) in both x and y direction.

- Grid points in x-direction: $x_i = ih$, where i=0,1,....,N and h=$\frac{1}{N}$ is the mesh size.
- Grid points in y-direction: Similarly, $y_i = jh$, where i=0,1,....,N and h=$\frac{1}{N}$ is the mesh size.

This creates a grid with (N+1)x(N+1) points.

For a given 2D equation grid contains of boundary points and interior points.Interior grid points where i,j = 1,2,3...(N-1), this gives (N-1)x(N-1) interior points.

(c) Incorporate the Mesh Interval(h): **Equation 7 applies only to all interior points (i,j)**

- h=$\frac{1}{10}$
  This creates a grid of 11*11=121 grid points. 9*9=81 Interior points.
- h=$\frac{1}{20}$
  This creates a grid of 21*21=441 grid points. 19*19=361 Interior points.
- h=$\frac{1}{40}$
  This creates a grid of 41*41=1681 grid points. 39*39=1521 Interior points.

(d) Applying Bounadary Conditions:

For Interior points ,if i,j corrspond to a bounadry then (i-1 =0,j+1 =1),we substitute the kown boundary conditions frm eq.. Consider at (i,j) =(1,1) in eq.7

$$u_{2,1} + u_{0,1} + u_{1,2} + u_{1,0} - 4u_{1,1} = 0 \tag{8}$$

$u_{0,1}$ and $u_{1,0}$ are known boundary values, so we move them to the right hand side.

$$u_{2,1} + u_{1,2} - 4u_{1,1} = -u_{1,0} - u_{0,1} \tag{9}$$

For an interior point like (i,j)=(5,5),all neighbouring points are interior ,so the equation 7 remians same:

$$u_{6,5} + u_{4,5} + u_{5,6} + u_{5,4} - 4u_{5,5} = 0 \tag{10}$$

(e) Set up the Linear system: Write the finite difference equation for each interior point (i,j),where i,j=1,2,3,...(N-1). this results in (N-1)*(N-1) equations with (N-1)*(N-1) unknowns. To solve this we represent the system as a linear system:

$$Au = b \tag{11}$$

- **u**:A vector of unknowns $u_{i,j}$, (row-wise: $u_{1,1}, u_{1,2}, ..., u_{1,N-1}, u_{2,1}, ..u_{N-1,N-1}$), so 'u' is a vector of length $(N-1)^2$
- **A**:A sparse matrix representing coefficients of the finite difference equations 7
- **b**: A vector containing f(x,y) and contribuitions from boundary conditions.

Matrix structure

The matrix A is sparse because each equation only five points. For a point $u_{i,j}$, the coefficients are:

- -4 for $u_{i,j}$
- 1 for each $u_{i+1,j}, u_{i-1,j}, u_{i,j+1}, u_{i,j-1}$.
- 0 for all other points.

Solving the system: Solve the linear equation 11 by iterative methods like:

(a) Jacobi Method

(b) Gauss seidel method

(c) Steepest descent

(d) Conjugate gradient method

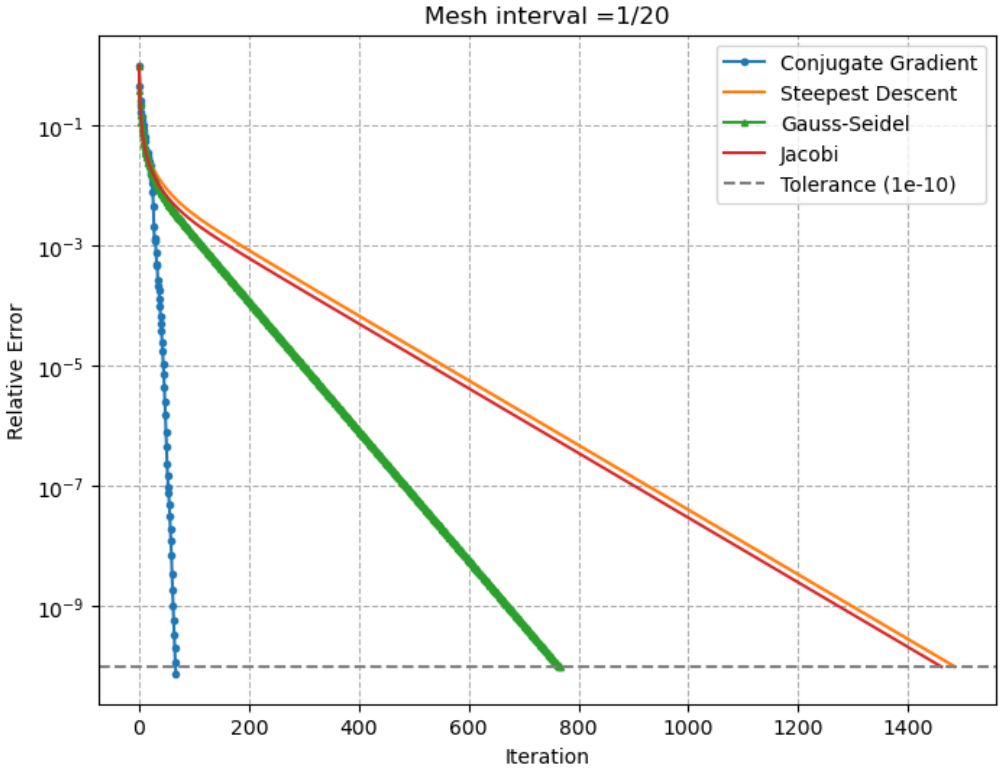

Figure 2: relative error against iteartions for Mesh size=1/20

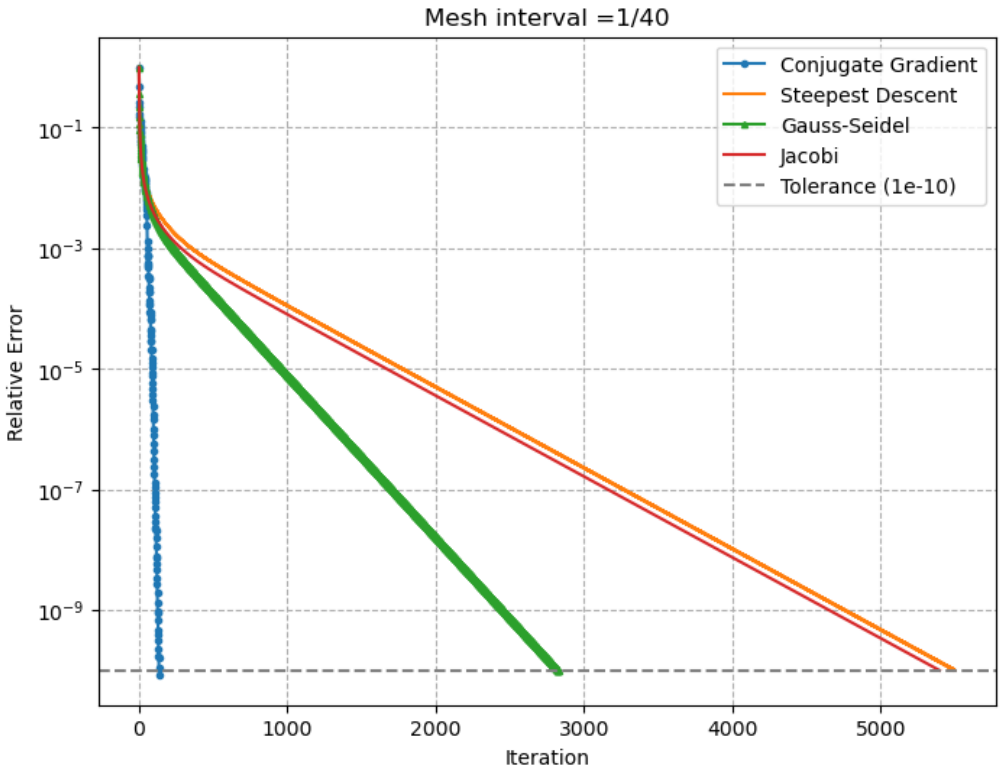

Figure 3: relative error against iteartions for Mesh size=1/40

For the given mesh sizes matrix A has dimension of interior points i.e 81x81 , 361x361 ,1521x1521 for h1,h2,h3 respectively.

1,2 and 3 are the plots for relative error against the iteration count for mesh size = h1,h2 and h3 respectively solved by different iterative methods, we can visually compare their convergence rates.

We can see as size of matrix increases iterations increase for all methods to converge. Comparing for all the three linear sysytems we can see that Steepest descent method converges the slowest and the conjugate gradient converges faster.

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

Effect of grid size:

- As h decreases (finer grid), the system size increases ($(N1)^2$), and the condition number of A grows($O(1/h^2)$), making convergence harder.
- Methods like Jacobi and Steepest Descent are more affected by this, leading to larger differences from the reference solution.

2. **For a given method (e.g. Jacobi), plot the relative error (LHS of Eq. 3) versus the iteration index (k) for all three mesh sizes (h1,h2, and h3). Repeat the same for all four methods (a total of four figures). In the plot, the relative error (y-axis) should be in base-10 logarithmic scale, and the iteration count (x-axis) must be on a linear scale. Write the inferences from the plots based on your understanding.**

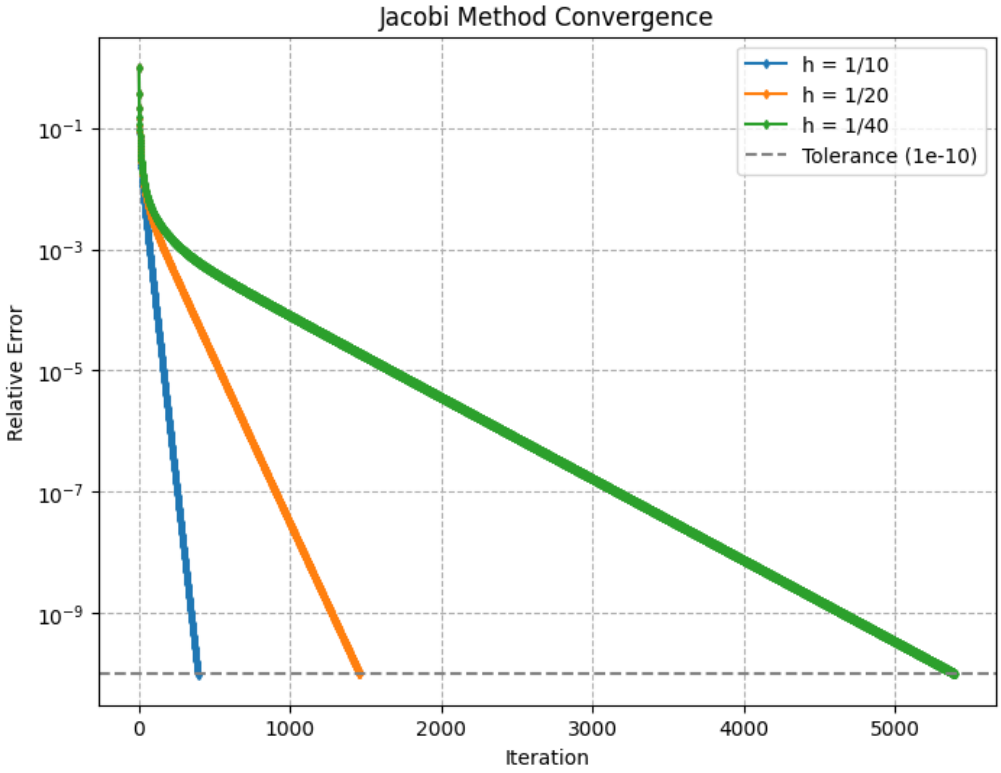

Figure 4: Jacobi Method

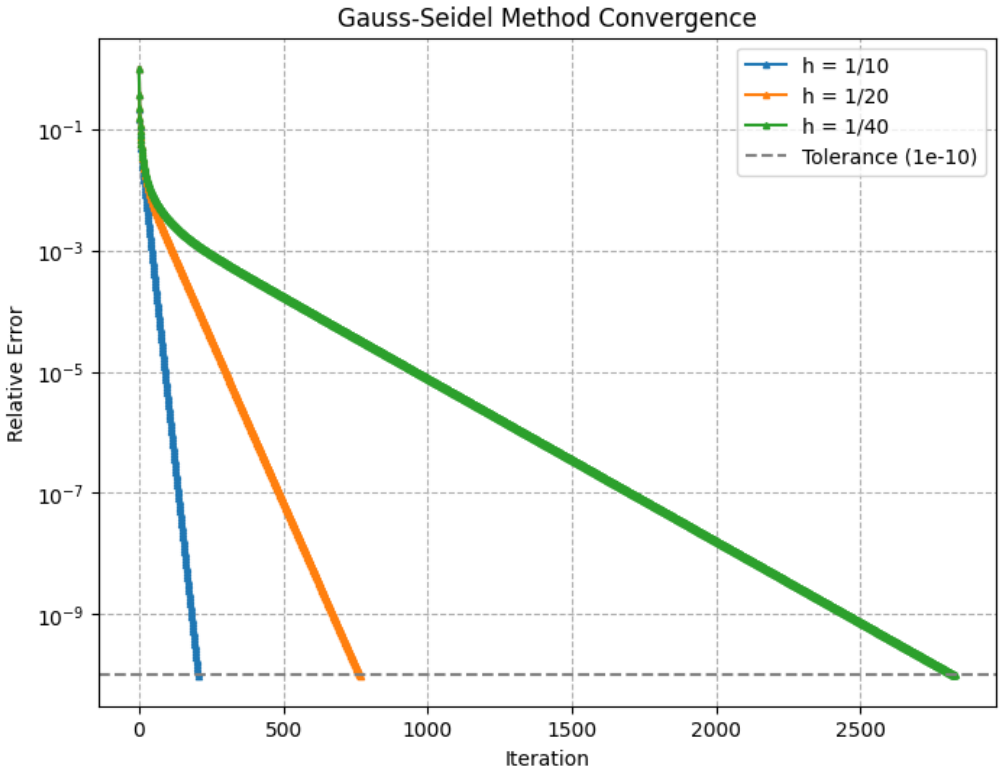

Figure 5: Gauss seidel method

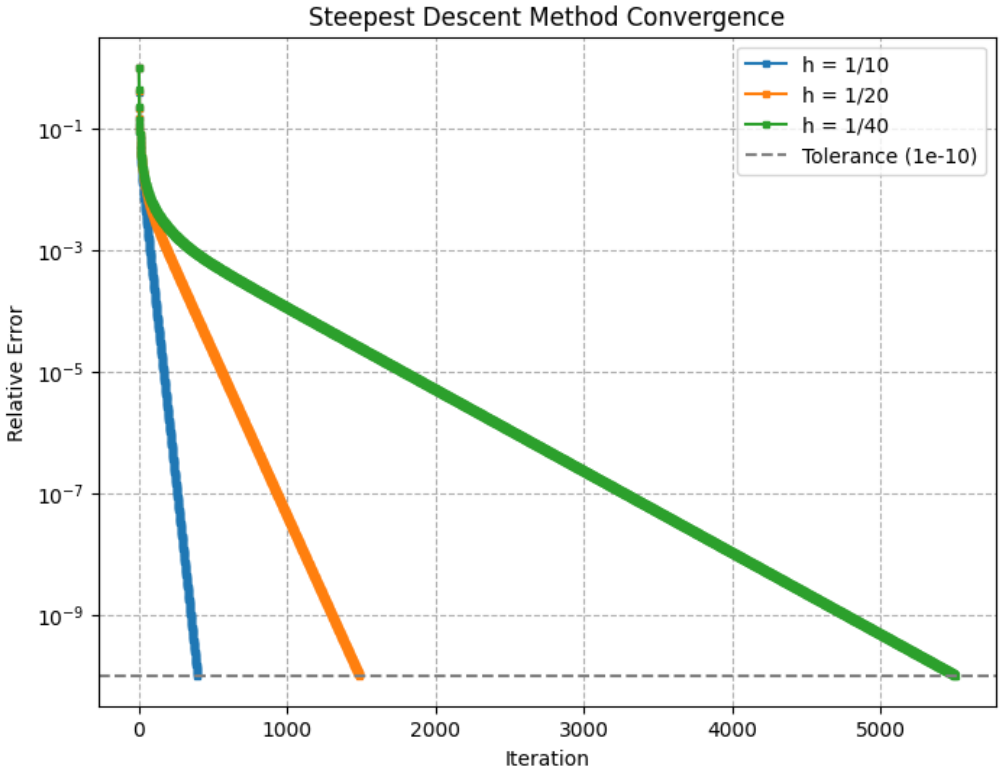

Figure 6: Steepest Descent Method

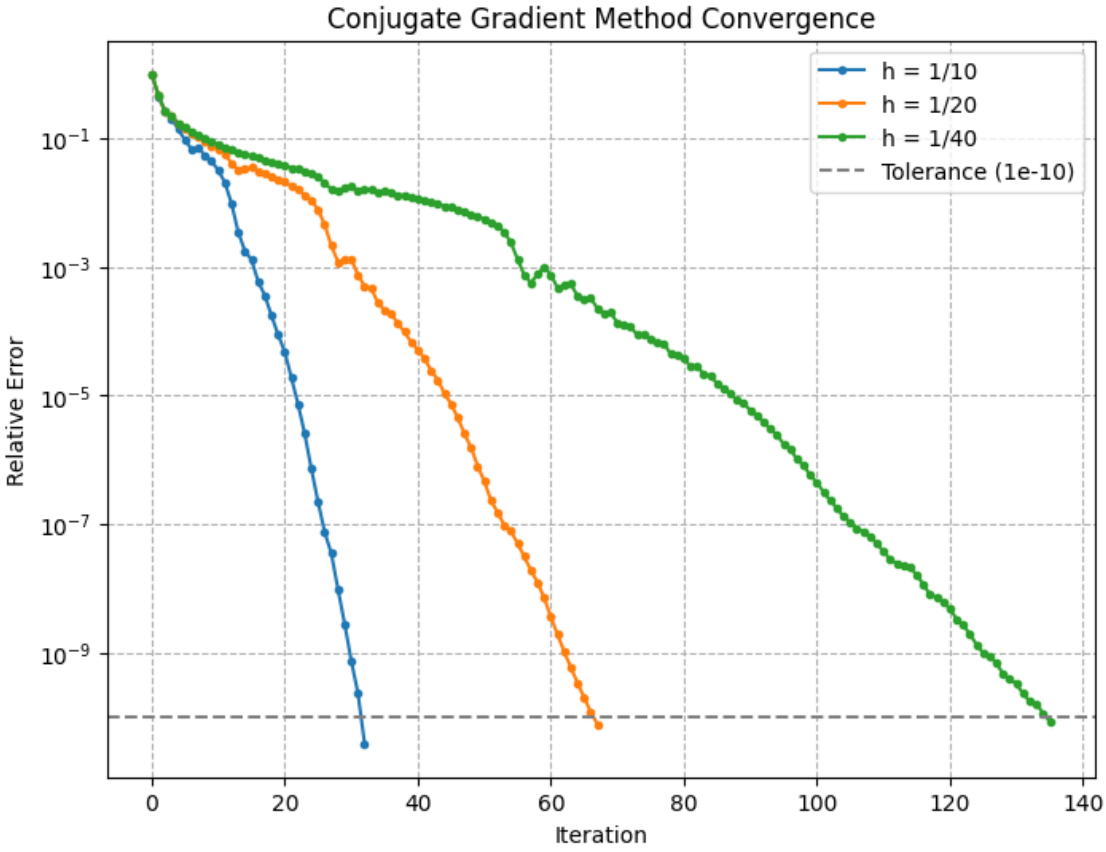

Figure 7: Conjugate Gradient Method

Figure 4,figure 5, figure 6 and figure 7 are the plots of relative error versus iteartion index for Jacobi,Gauss seidel,Steepest descent and conjugate gradiient respectively. After comparing plots of relative error against iteartions we can see Conjugate gradient is the optimal method to reach the convergence criteria .

$$\frac{\|\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\|_2}{\|\mathbf{u}^{(k+1)}\|_2} \leq 10^{-10} \tag{12}$$

For all the three linear systems with respect to mesh size ,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.

- For a given method all three mesh sizes show the same trend of plots for relative error versus iteration index.
- As mesh size decreases from 0.1 to 0.025 the iteration count increases by ≈ 3.7 times for jacobi method and steepest descent method, while conjuagte method takes increases only by twice.
- **Steepest Descen**t: Performs the worst, with the largest differences, as it struggles to converge within 6000 iterations for all h. The error is most pronounced for h=1/40.
- **Gauss-Seidel:** Converges slower than Conjugate Gradient but still reaches the tolerance within 6000 iterations for all h (617, 1234, 2338 iterations). The differences are very small, indicating high accuracy.

3. **Show a 3D surface plot of the final solution u (as a function of x and y) for all three mesh sizes.**

The plot will show a smooth surface over [0,1]x[1,0].

- At x=0,u=0, so the surface will be flat at 0 along the left edge.
- At x=1,u=y, so the surface will rise linearly from 0 to 1 as y goes from 0 to 1 along the right edge.
- At y=0, u=(x-1)sinx, which is negative (since x-1≤0) and varies sinusoidally, reaching a minimum near x≈0.5.
- At y=1, u=x(2-x), which is a parabola with u=0 at x=0 and x=2, peaking at u=1 at x=1.
- **Interior behaviour:** The surface will be smooth, transitioning between the boundary values,with higher value near (x,y)=(1,1)(where u ≈ 1) and lower values near (x,y)=(0,0)(where u=0).

Figure 8, figure 9 and figure 10 are the surface plots of the solution vector(u) from conjugate gradient method.

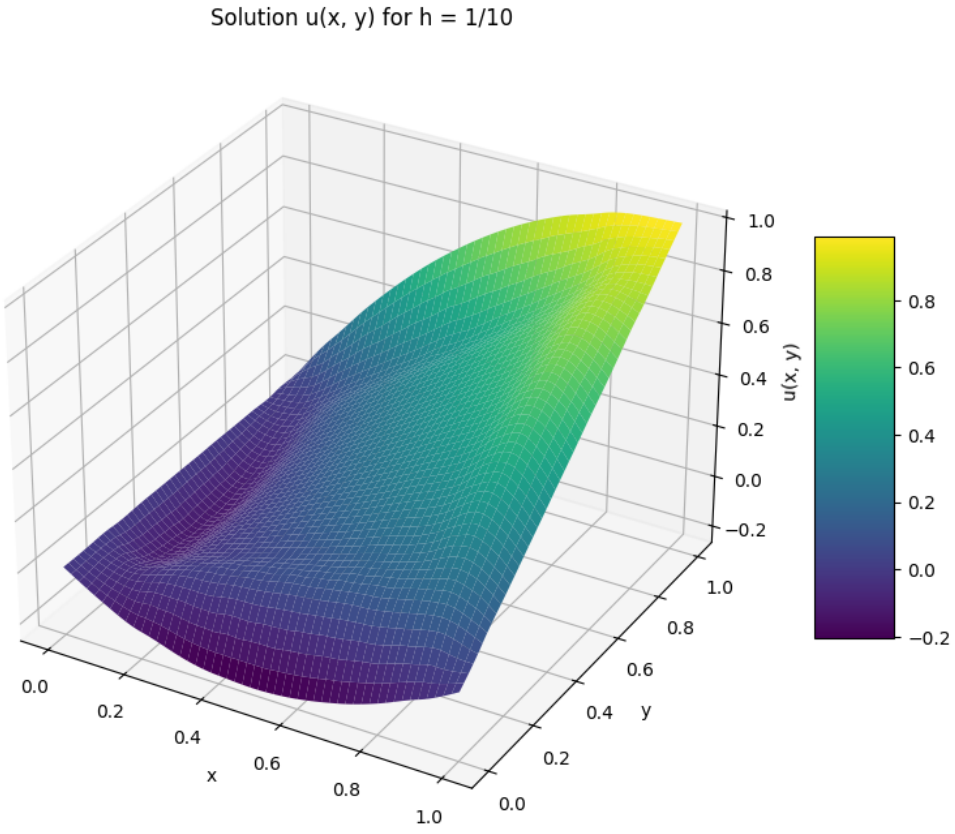

Figure 8: 3D surface plot of h=1/10

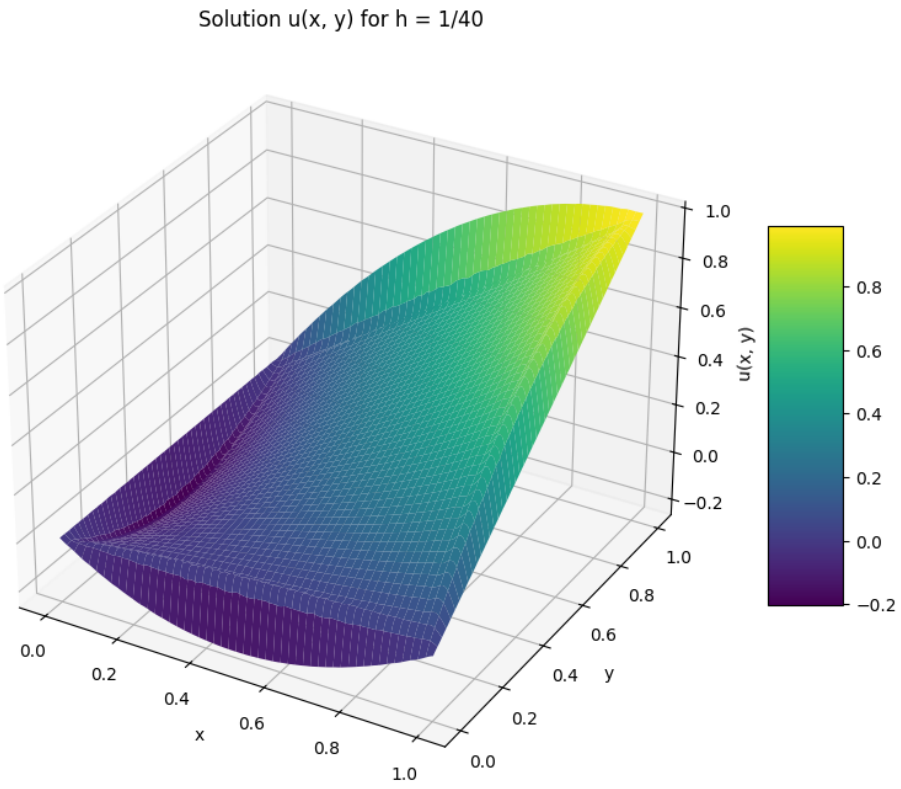

Figure 9: 3D surface plot of h=1/20

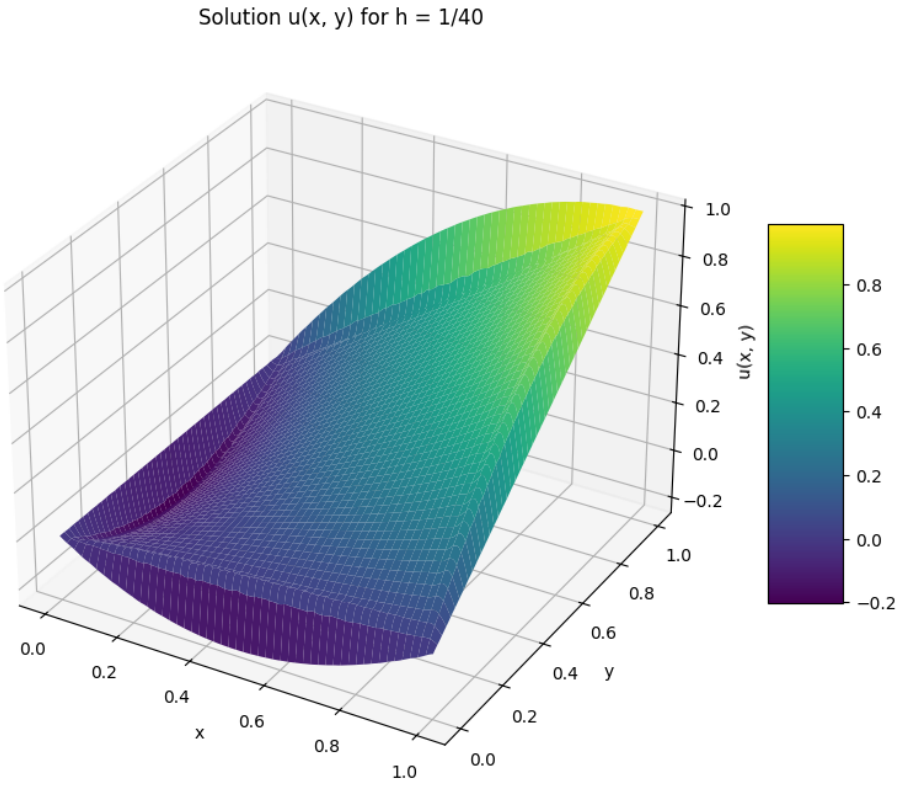

Figure 10: 3D surface plot of h=1/10

4. **In a table, report the number of iterations required to converge for Jacobi, GS, SGD, and CG methods for all three mesh sizes.**

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

Mesh Interval[h]	`[h1]1/10`	`[h2]1/20`	`[h3]1/40`
Jacobi	391	1462	5399
Gauss Seidel	206	767	2828
Steepest Descent	395	1486	5502
Conjugate Gradient	33	68	136

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 $\mathbf{A}$ is diagonally dominant.
- Symmetry and Positive Definiteness - For methods like the Conjugate Gradient, $\mathbf{A}$ 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 $\mathbf{A}$ 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 $\mathbf{A}$ 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 $\mathbf{A}$ 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

$$\rho(\mathbf{G_J})^2 = \rho(\mathbf{G_G S}) \tag{13}$$

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.

$$\alpha_k = \frac{(p^k)^T r^k}{(p^k)^T A p^k} \tag{14}$$

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

$$J(y) = \frac{1}{2}(y-x)^T A(y-x) - \frac{1}{2}(x^T Ax) \tag{15}$$

where x is the solution of Ax = b. The contours of J are ellipses.

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.

$$\mathbf{p^{k+1}} \perp \mathbf{p^{k}}; [p = -\Delta J] - \text{direction of search} \tag{16}$$

(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.

$$\beta_k = \frac{(r^{k+1})^T r^{k+1}}{(r^k)^T r^k} \tag{17}$$

$$p^{k+1} = r^{k+1} + \beta_k p^k \tag{18}$$

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.

$$p^k \perp p^{k-1} \perp p^{k-2} ..... \perp p^0 \tag{19}$$

$$(p^k)^T A p^i = 0 \tag{20}$$

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.