Computational Fluid Dynamics

Project 1: Elliptic PDEs

Report

Indian Institute of Technology,

Gandhinagar 382355, Gujarat

Submitted By:
Vardan Mittal (24250104)
Aaditya Nimkar (24250002)

Problem Statement

Given the equation of 2D steady-state diffusion on a square domain of unit length

$$\frac{\partial^2\phi}{\partial x^2}+\frac{\partial^2\phi}{\partial y^2}=S\phi$$

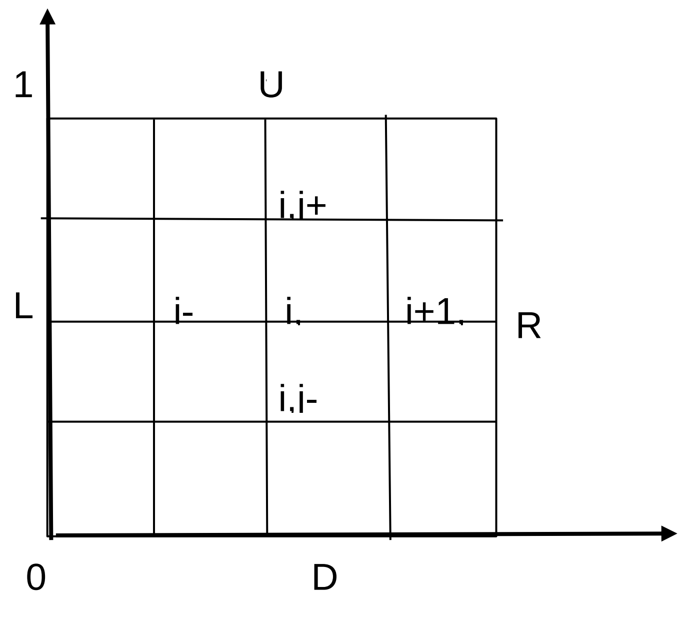


Figure 1 Discretization of surface

For the following boundary conditions

$$L=\phi(0,y)=500e^{(-50[1+y^2])}$$
$$R=\phi(1,y)=100(1-y)+500e^{(-50y^2)}$$
$$D=\phi(x,0)=100x+500e^{(-50[1-x^2])}$$
$$U=\phi(x,1)=500e^{(-50[1-x]^2+1)}$$

The following source term

$$S\phi=50000e^{(-50\{[1-x]^2+y^2\})}(100\{[1-x]^2+y^2\}-2)$$

Analytical solution

$$\phi(x,y)=500e^{(-50\{[1-x]^2+y^2\})}+100(1-y)$$

1. Gaussian elimination

Solve the 2D steady-state diffusion equation using the Gaussian elimination method for the following grids: 21, 41, and 81 grid points in each direction. Show the computed field as a contour plot for the finest grid. Plot the CPU run time vs the total number of grid points, discuss the trend, and comment on the computational efficiency of the Gauss elimination method.

We shall start by discretising the steady-state diffusion equation using the 2nd-order accurate central difference scheme.

$$\frac{\partial^2\phi}{\partial x^2} = \frac{\phi(i+1,j) - 2\phi(i,j) + \phi(i-1,j)}{\Delta x^2}$$

$$\frac{\partial^2\phi}{\partial y^2} = \frac{\phi(i,j+1) - 2\phi(i,j) + \phi(i,j-1)}{\Delta y^2}$$

Adding

$$\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} = \frac{\phi(i+1,j) - 2\phi(i,j) + \phi(i-1,j)}{\Delta x^2} + \frac{\phi(i,j+1) - 2\phi(i,j) + \phi(i,j-1)}{\Delta y^2} = Si,j$$

For a uniform mesh

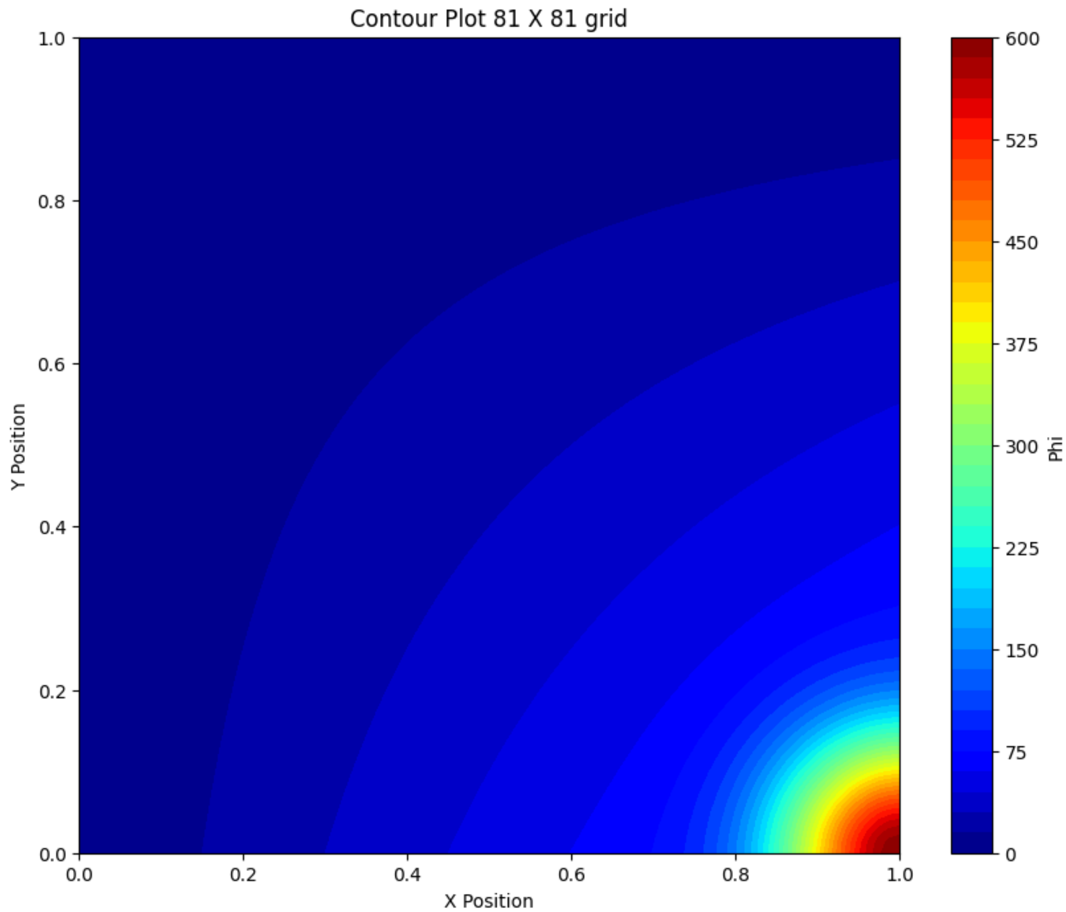
$$\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} = \frac{\phi(i+1,j) - 2\phi(i,j) + \phi(i-1,j)}{\Delta^2} + \frac{\phi(i,j+1) - 2\phi(i,j) + \phi(i,j-1)}{\Delta^2} = Si,j$$

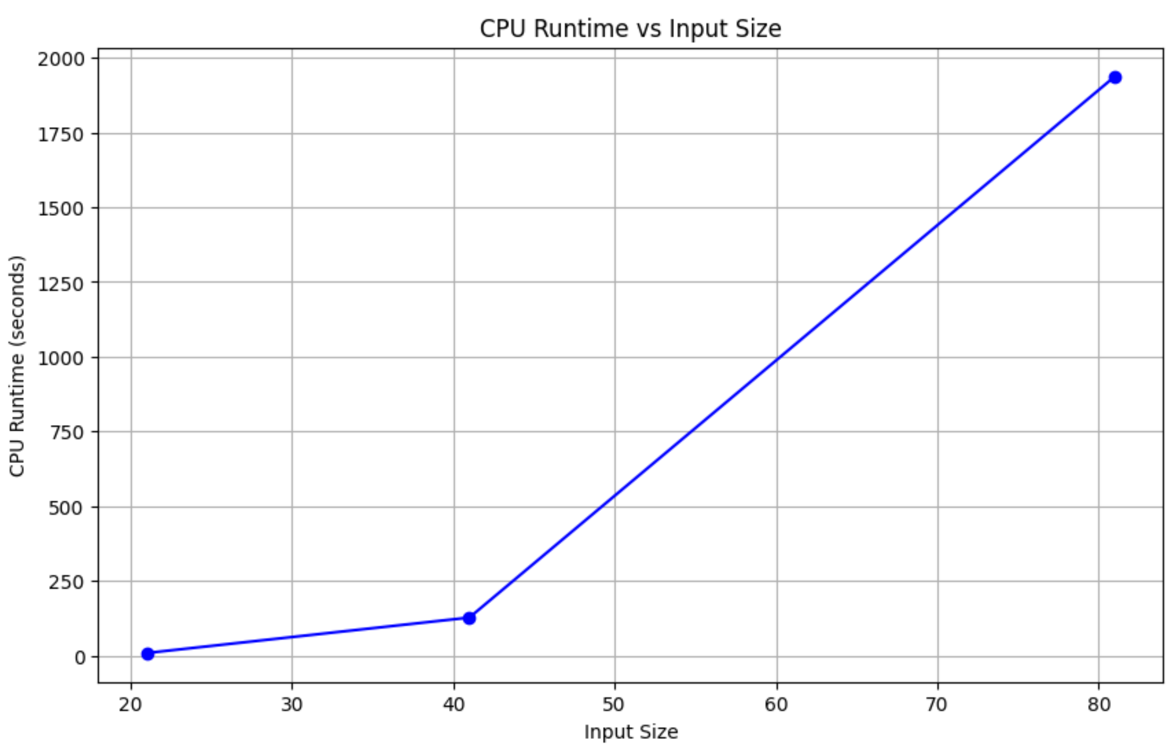
Thus we get the final discretised equation as

$$\phi(i+1,j) + \phi(i-1,j) + \phi(i,j+1) + \phi(i,j-1) - 4\phi(i,j) = Si,j * \Delta^2$$

The process is as follows:

- This equation has to be obtained for each interior node in the domain by putting the values of i,j, S and delta.
- The value of S has to be computed at every x and y and then substituted in the discretised equation.
- After all equations have been obtained for the nodes, they must be assembled in a matrix form
- $[A]\{\phi\} = \{B\}$
- After which, we can perform the forward elimination and backward substitution to obtain the values ϕ.


Contour Plot 81 X 81 grid
Y Position
X Position
Phi
1.0
0.8
0.6
0.4
0.2
0.0
600
525
450
375
300
225
150
75
0


CPU Runtime vs Input Size
CPU Runtime (seconds)
Input Size
2000
1750
1500
1250
1000
750
500
250
0
20
30
40
50
60
70
80

- Gauss elimination is really very time consuming and slow method required high computation time as seen from the graph
- The solution is very close to the analytical solution

2. Gauss - Seidel method

Solve the same 2D steady-state diffusion equation using the Gauss-Seidel iterative method for the following three grids: 41, 81, and 161 grid points in each direction. Show the computed field as a contour plot for the finest grid. Plot the residual vs number of iterations for the three grids in the same plot. Compare and discuss the dependence of the convergence rate on the total number of grid points. Plot the variation of CPU run time with the total number of grid points for the Gauss-Seidel iterative method and discuss the trend. How do the CPU run times of the Gauss-Seidel iterative method compare with those of the Gauss elimination method?

We start off with the same discretised equation

$$\phi(i+1,j) + \phi(i-1,j) + \phi(i,j+1) + \phi(i,j-1) - 4\phi(i,j) = Si,j * \Delta^2$$

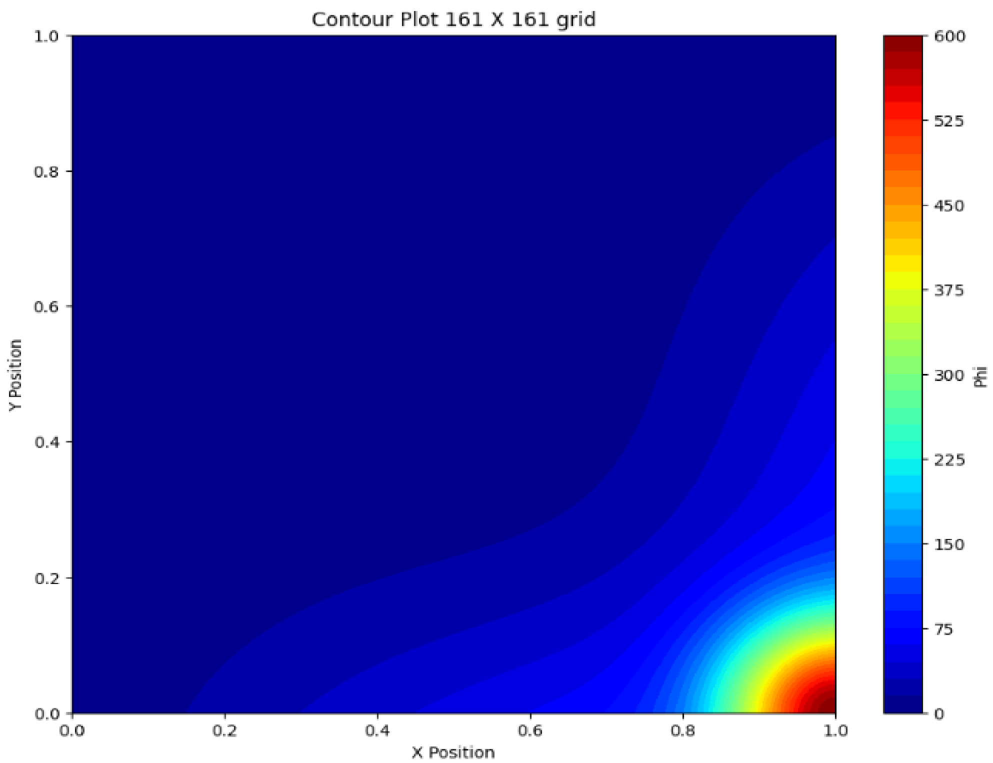
This method is an iterative method thus, we must modify the above equation to include iteration components. The methodology used is as follows:

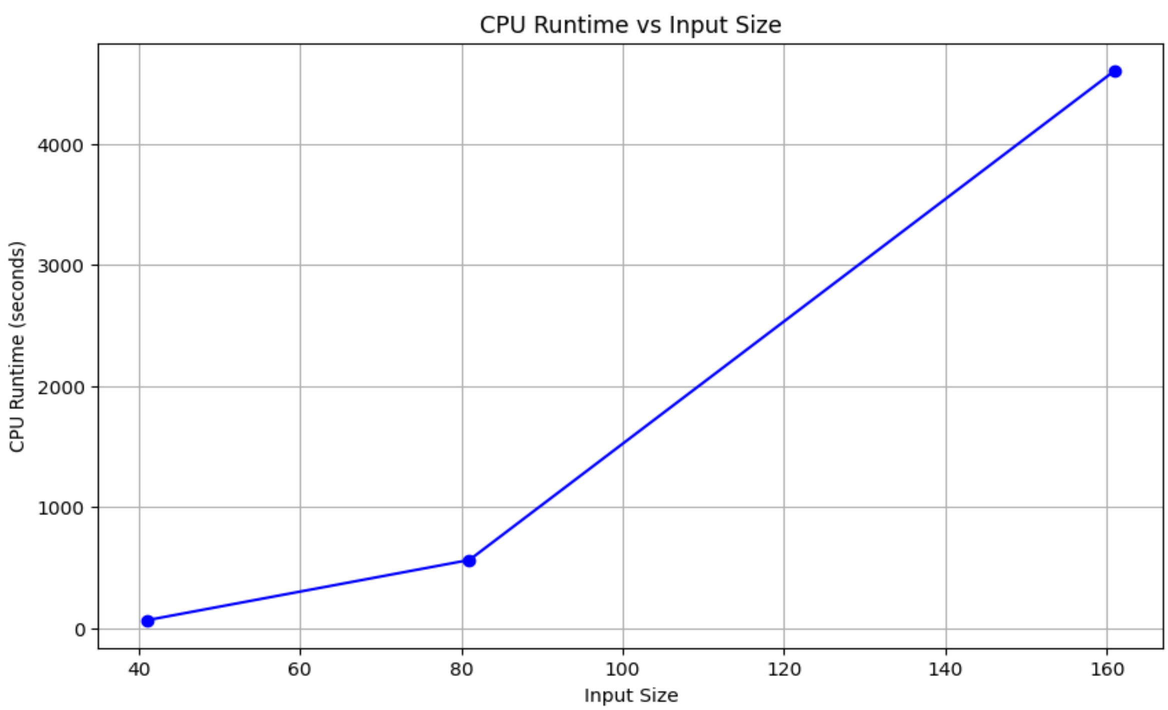
- Compute the values of phi for each node using the latest values.
- This completes one iteration of the algorithm.
- After each iteration, compute the residue and check the convergence.
- Keep iterating until the residue is equal to or within the tolerance value.

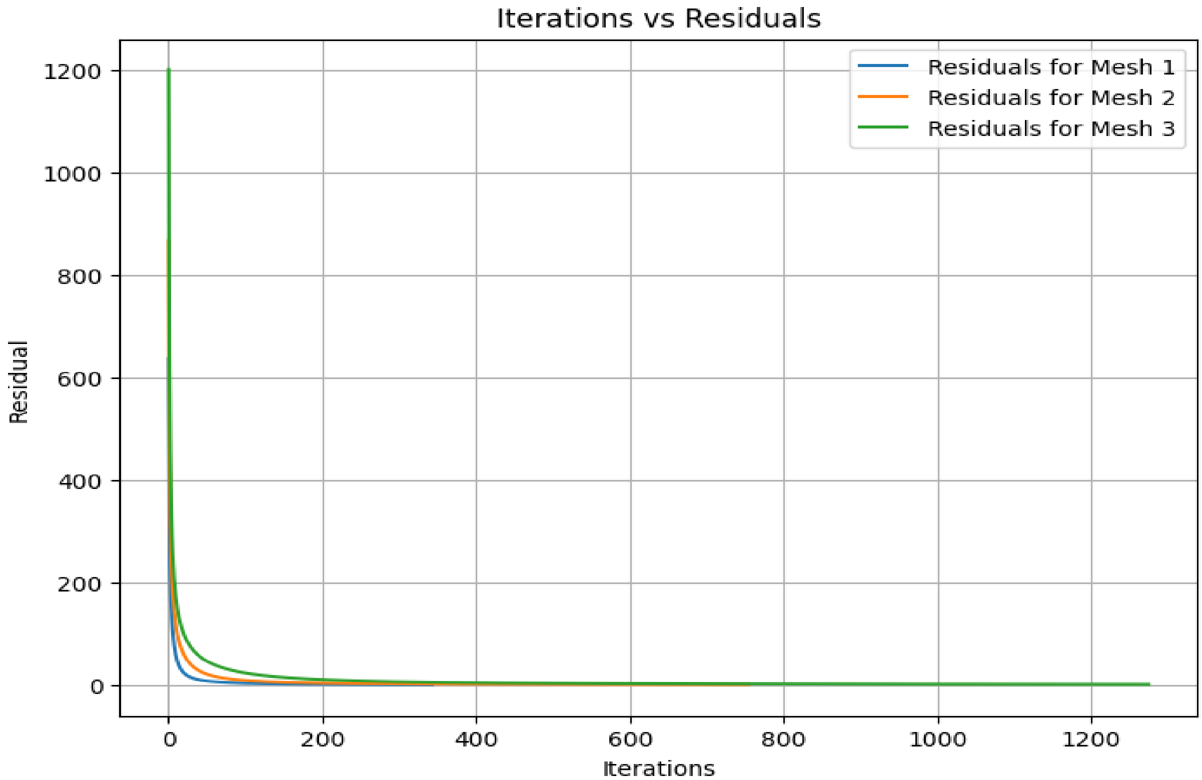
We have the modified discretised equation as follows

$$\phi^{n+1}(i+1,j) + \phi^{n+1}(i-1,j) + \phi^{n}(i,j+1) + \phi^{n}(i,j-1) - 4\phi^{n+1}(i,j) = S^{n}i,j * \Delta^2$$

- A relatively fast method this method depends on the iteration until convergence approach
- It reaches to the solution but never get to the exact solution due to discretization, truncation and roundoff errors with the solver


Contour Plot 161 X 161 grid
Y Position
X Position
Phi
0.0
0.2
0.4
0.6
0.8
1.0
0
75
150
225
300
375
450
525
600


CPU Runtime vs Input Size
CPU Runtime (seconds)
Input Size
0
1000
2000
3000
4000
40
60
80
100
120
140
160


Iterations vs Residuals
Residuals for Mesh 1
Residuals for Mesh 2
Residuals for Mesh 3
Residual
Iterations
0
200
400
600
800
1000
1200

3. Line-by-line (row sweep) method

Solve the same 2D steady-state diffusion equation using the line-by-line method (row sweep) for the following three grids: 41, 81, and 161 grid points in each direction. Please note that you will need to use the TDMA solver to solve the resulting system of linear equations because of the tridiagonal structure of the resulting coefficient matrices. Show the computed field as a contour plot for the finest grid. Plot residual vs number of iterations for the line-by-line method and the Gauss-Seidel method for the (161x161) grid (both in the same plot). Compare and discuss the trends. Plot the variation of CPU run time with the total number of grid points for the line-by-line method. How do the CPU run times and the obtained scaling compare with those obtained for the Gauss elimination method and point-wise iterative method?

The reasons for which row sweep is preferred as compared to Gauss-Seidel for solving elliptic PDEs are:

- **Global Influence**: The row sweep method propagates boundary and interior information more efficiently across the grid, which is critical for the global nature of elliptic PDEs.
- **Convergence Rate**: The row sweep method typically offers a faster convergence rate due to simultaneous updates across rows or columns compared to the sequential updates of Gauss-Seidel.
- **Handling Large Systems**: The row sweep method scales better and is more efficient for large systems, making it a better choice for solving complex meshes.

Starting from the previous equation

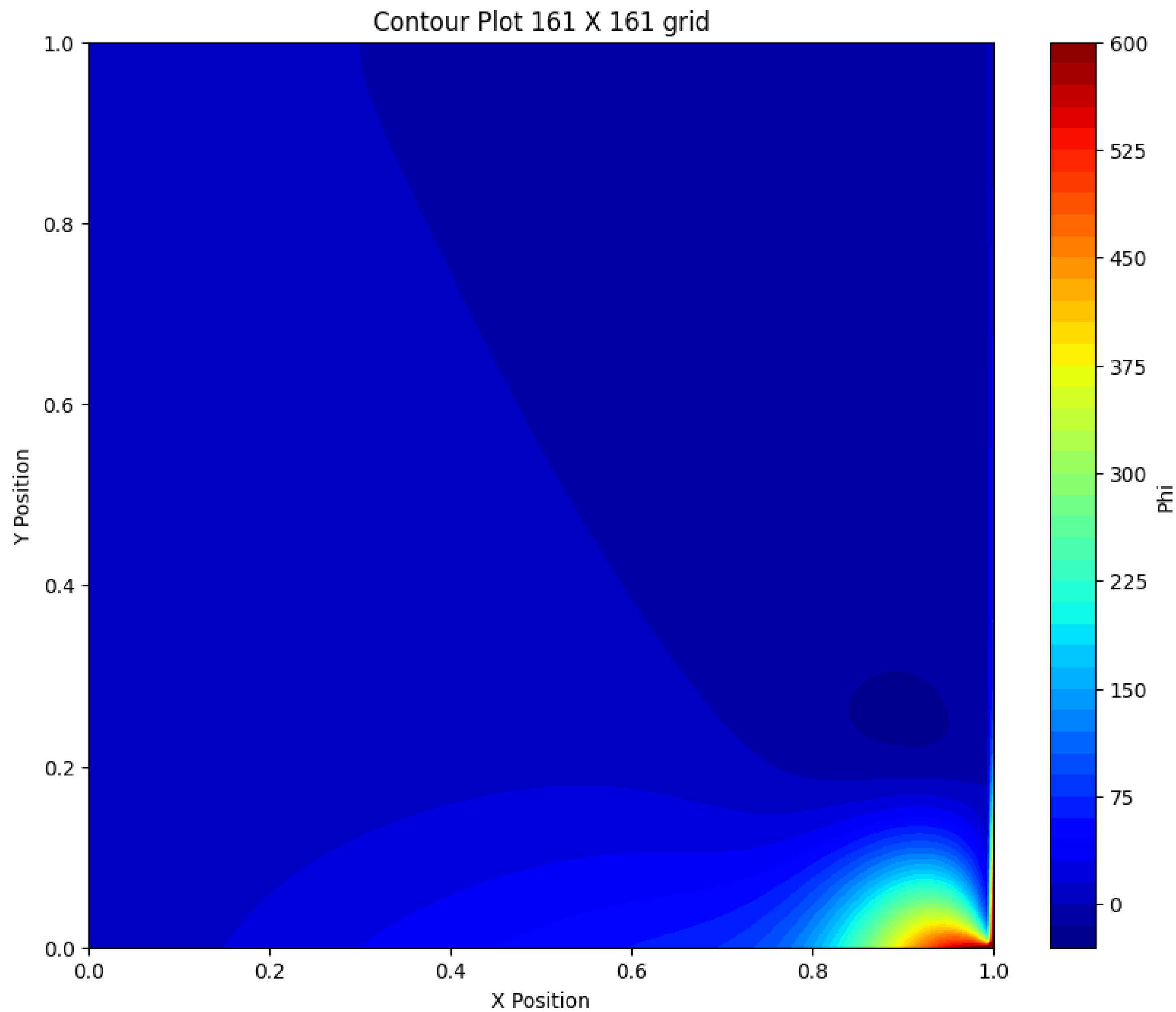
$$\phi^{n+1}(i+1,j) + \phi^{n+1}(i-1,j) + \phi^{n}(i,j+1) + \phi^{n}(i,j-1) - 4\phi^{n+1}(i,j) = S^{n}i,j * \Delta^2$$

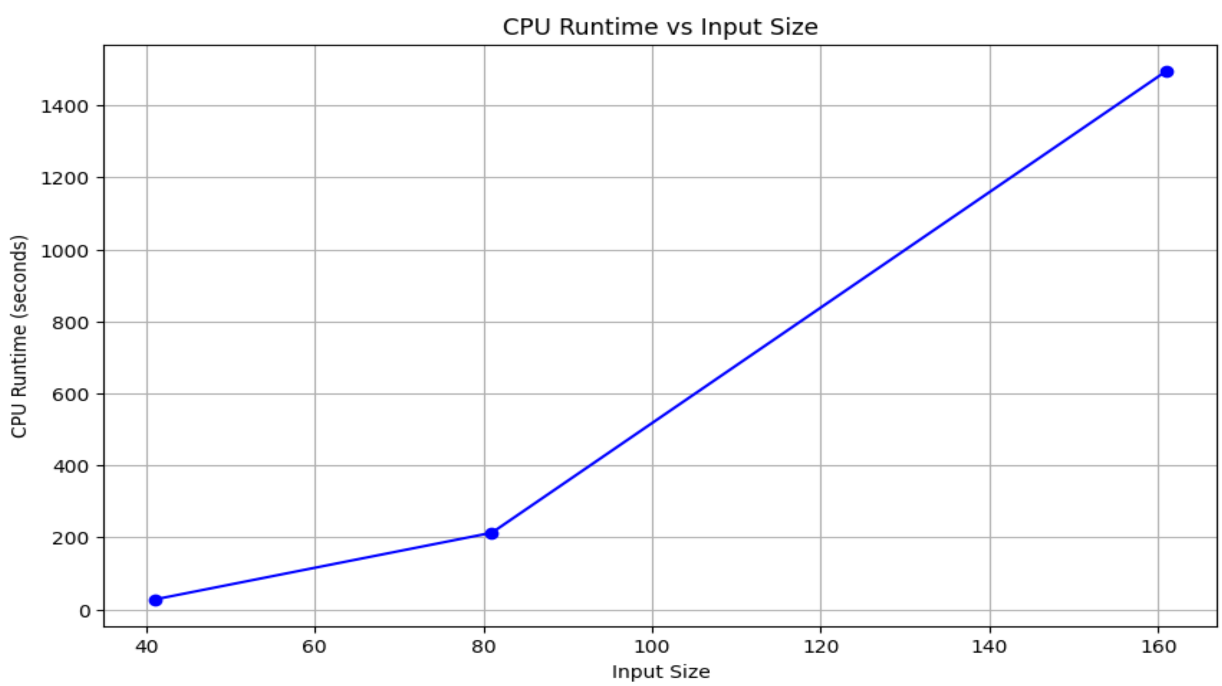
Rearranging we get

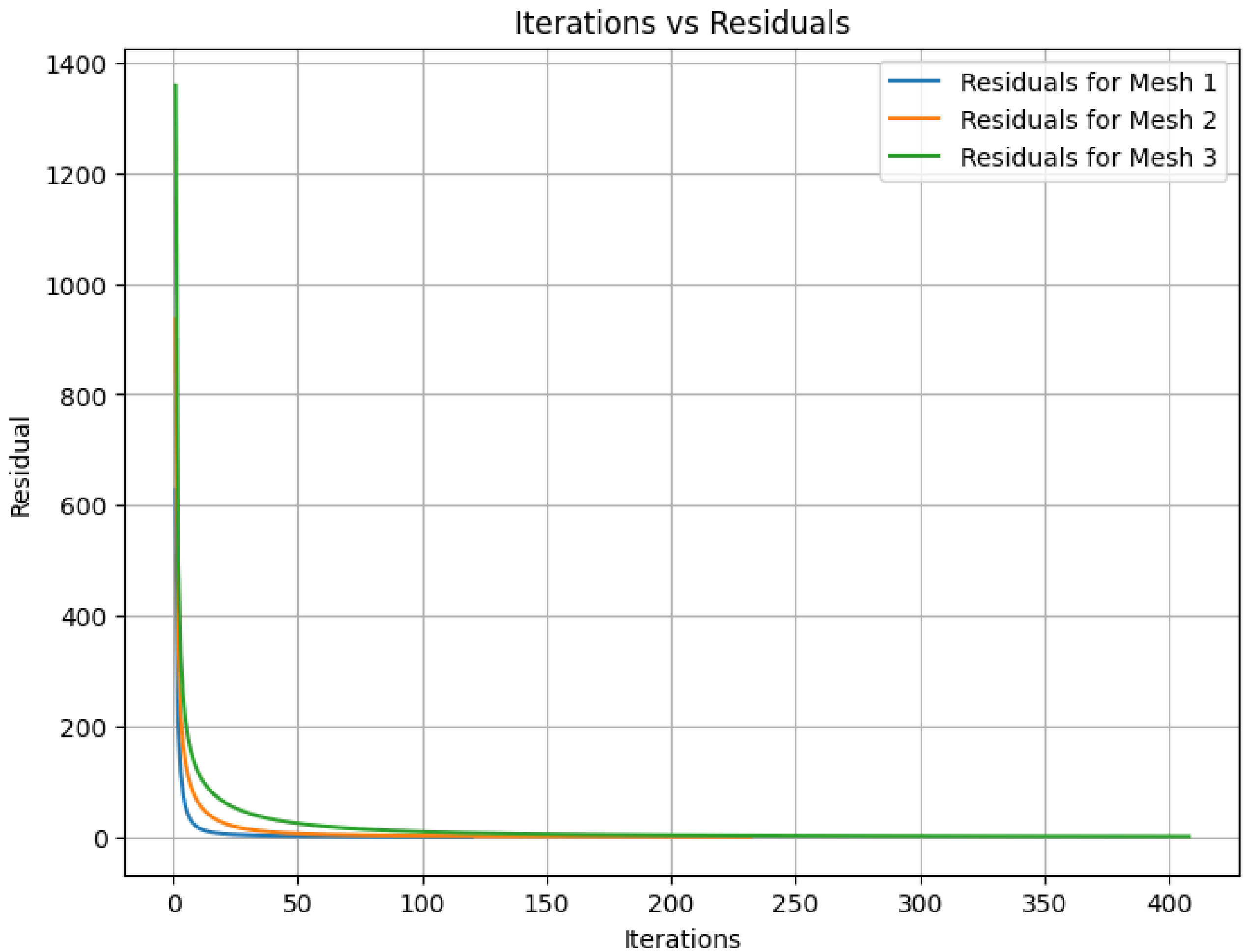
$$\phi^{n+1}(i+1,j) + \phi^{n+1}(i-1,j) - 4\phi^{n+1}(i,j) = S^{n}i,j * \Delta^2 - \phi^{n}(i,j+1) - \phi^{n}(i,j-1)$$

From the LHS, we can clearly see that to get the n+1 terms, we must solve the elements simultaneously. The process is as follows:

- Guess all the initial values of phi at the nodes.
- For each row, “i” set up the tridiagonal system using the boundary conditions and the internal points.
- Solve the tridiagonal system using the TDMA.
- Like Gauss-Seidel, check for convergence by calculating the residuals between successive intervals.
- Repeat the process for all rows until convergence.


Contour Plot 161 X 161 grid
Y Position
X Position
Phi
0.0
0.2
0.4
0.6
0.8
1.0
600
525
450
375
300
225
150
75
0


CPU Runtime vs Input Size
CPU Runtime (seconds)
Input Size
0
200
400
600
800
1000
1200
1400
40
60
80
100
120
140
160



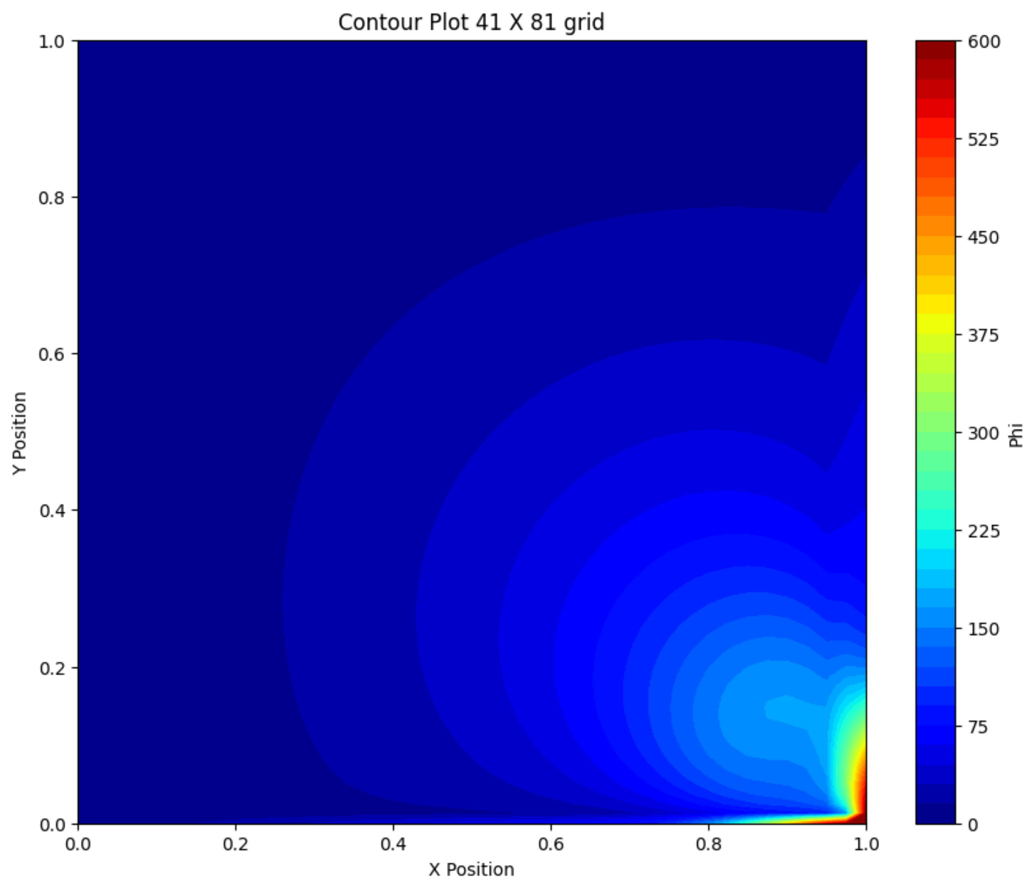
- This method takes advantage of TDMA solver which reduces the time complexity to linear and converges relatively faster than the Gauss-Seidel.
- It captures the nature of elliptic PDE better than above method

4. Alternating Direction Implicit (ADI)

Solve the same 2D steady-state diffusion equation using the Alternating Direction Implicit (ADI) method for the 41x81 grid and plot the residual vs number of iterations for row-wise sweep, column-wise sweep, and ADI method in the same plot. Discuss the trends.

ADI is one step ahead of the line-by-line approach. Instead of only doing row sweep, we also do column sweep as part of each iteration. This is a much better way to capture the nature of the elliptic PDE, and this method converges much faster than the row sweep method.
The algorithm is as follows:

- Initialise phi with initial guesses and apply boundary conditions.
- For each iteration, n:
    1. **Solve in x-Direction**: For each row j, solve the tridiagonal system using TDMA across a row for $\phi^{n+1}(i,j)$
    2. **Solve in y-Direction**: For each column i, solve the tridiagonal system using TDMA across a column for $\phi^{n+1}(i,j)$
- After each full iteration (both x and y directions), check for convergence by calculating the residuals or differences between successive iterations.



Discussion

We can set some criterions on which we are going to discuss the methods

1. Discretization method
   - **Gauss Elimination:** Creation of full matrix of coefficients of phi using Finite Difference Method
   - **Gauss-Seidel:** Iterative method with point to point sequential solving of the discretized equation for each node.
   - **Line by Line:** Still a iterative method but tries to club TDMA solver by solving either using row sweep or column sweep by sequential solving.
   - **ADI:** This method is a better version of line by line approach which switches the sweep direction between row and columns in each iteration
2. Run Time (n is the no. of points)
   - **Gauss Elimination:** Very slow as it's time complexity raises for $O(n^3)$
   - **Gauss-Seidel:** It's speed depends upon the convergence, tolerance as well as grid size of the problem, It's time complexity lies in $O(n^2)$ approximately.
   - **Line by Line:** This method is relatively faster as it uses TDMA solver whose time complexity is $O(n)$ and this method scales about $O(n^2)$.
   - **ADI:** This method converges way faster than any other method as it goes bidirectional and scales approximately $O(n^2)$.
3. Nature of PDEs
   - **Gauss Elimination:** Exact method to solve the PDEs gives exact solutions
   - **Gauss-Seidel:** This method is primarily good for diagonal dominant matrices as it is a sequential solving not simultaneous solving.
   - **Line by Line:** Generally good for a specifically structured PDE as it tries to capture the nature but unable to fully capture the elliptical nature of PDE
   - **ADI:** This method is very good for elliptical PDEs as it captures true nature of elliptical PDEs and take advantage of this to get faster convergence.
4. Accuracy
   - **Gauss Elimination:** Highly accurate as it solves simultaneous system of equations
   - **Gauss-Seidel:** Accuracy depends on the problem in hand and works well with specific systems.
   - **Line by Line:** Accuracy is better than that of Gauss-Seidel
   - **ADI:** Very accurate and efficient for large grids, often considered the most accurate for large scale problems.

Conclusion

In conclusion, the choice of method should be guided by the specific characteristics of the PDE and the computational constraints. For exact solutions in smaller systems, Gauss Elimination is appropriate. For larger grids with moderate accuracy requirements, Gauss-Seidel or Line by Line (TDMA) methods offer practical solutions. For large-scale elliptic PDEs where efficiency and accuracy are crucial, the ADI method stands out as the most effective approach.

By understanding and carefully studying computational requirement, accuracy, Nature of PDEs in hand and the discretization method can help us in selecting the appropriate method for the problem in hand.