Solvers for Laplace Equation

Mahendra Verma

भारतीय प्रौद्योगिकी संस्थान कानपुर

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

- Laplace eqn: $\nabla^2\phi = 0$
- Occurs in electrodynamics, hydrodynamics, magnetostatics, etc.
- $$\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} = 0 \text{ in 2D}$$
  $$\text{and } \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} + \frac{\partial^2\phi}{\partial z^2} = 0 \text{ in 3D}$$

Boundary condition

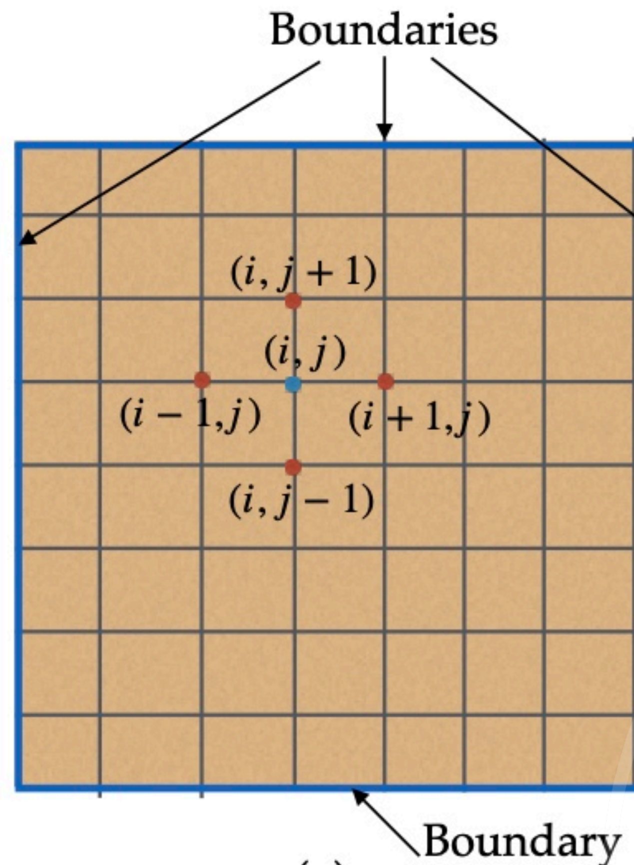
- Dirichlet boundary condition: Specify ϕ at the boundary of the domain Ω.
- Neumann boundary condition: The normal derivative of ϕ is specified at the boundary.
- Mixed boundary condition: Some regions of the boundary have Dirichlet BC, while the other regions have Neumann BC.

Properties

- Uniqueness theorem: The solution of the Laplace equation is uniquely determined given the boundary condition.

- In 3D, the average value of ϕ over a sphere is equal to its value at the center of the sphere.

- In 2D, the average value of ϕ over a disk is equal to its value at the center of the disk.



(a)

$$\frac{\partial^2\phi}{\partial x^2} = \frac{\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}}{h^2},$$

$$\frac{\partial^2\phi}{\partial y^2} = \frac{\phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1}}{h^2},$$

$$\frac{\phi_{i-1,j} - 2\phi_{i,j} + \phi_{i+1,j}}{h^2} + \frac{\phi_{i,j-1} - 2\phi_{i,j} + \phi_{i,j+1}}{h^2} = 0$$

$$\text{or, } \phi_{i,j} = \frac{1}{4}(\phi_{i-1,j} + \phi_{i+1,j} + \phi_{i,j-1} + \phi_{i,j+1}), \quad \ldots(110)$$

Direct Method

$$A \begin{bmatrix} \phi_{0,1} \\ \cdots \\ \phi_{0,m} \\ \phi_{1,0} \\ \phi_{1,1} \\ \cdots \end{bmatrix} = 0 \,,$$

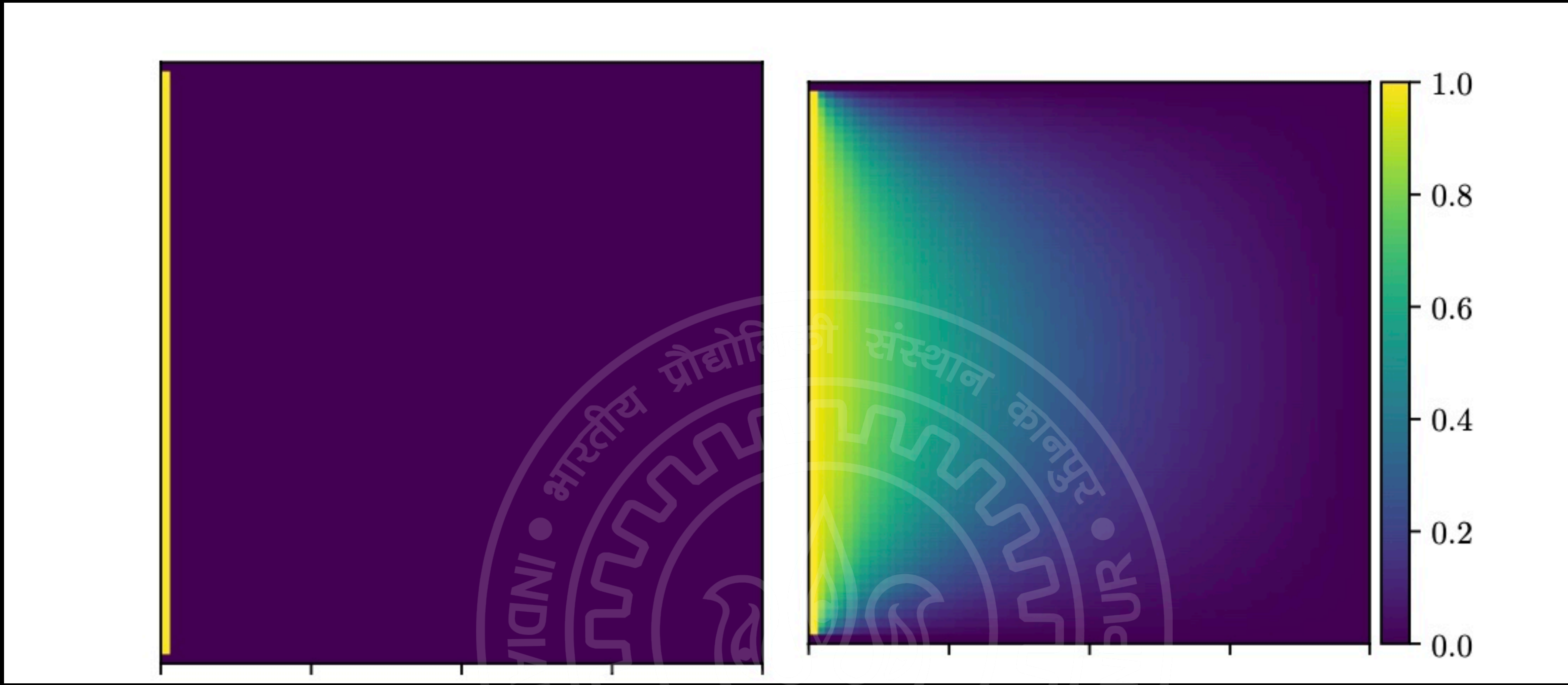
Banded matrix

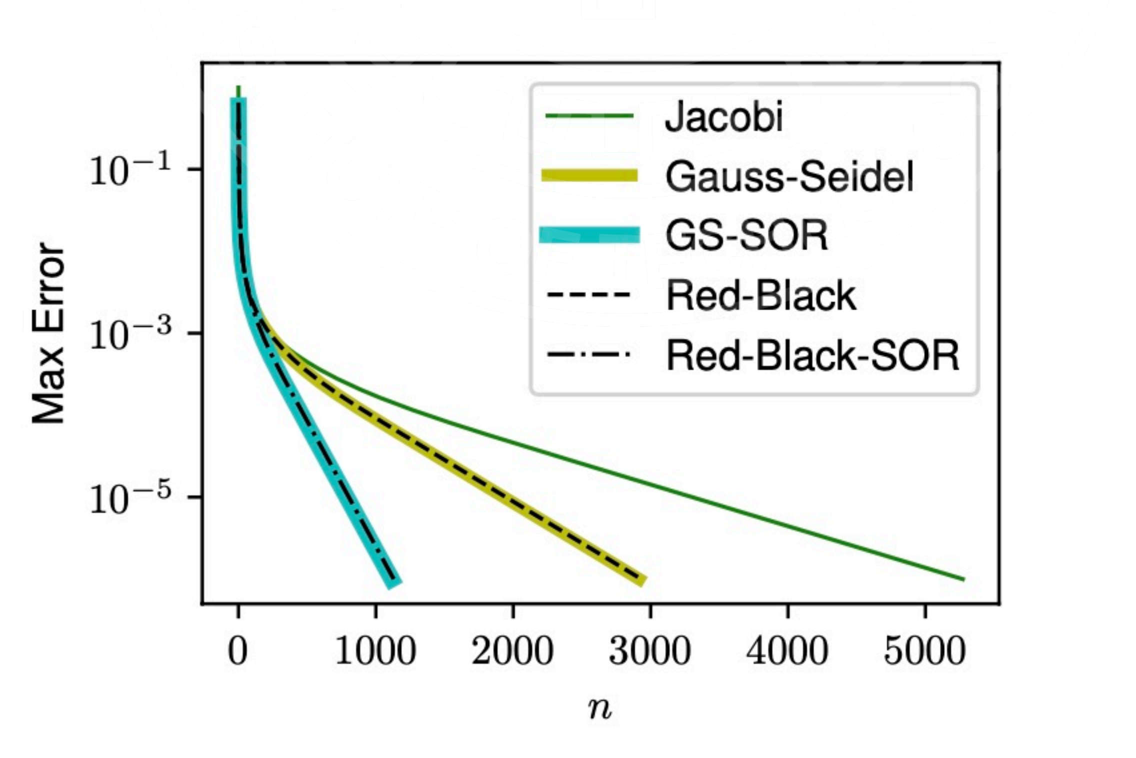
Iterative Methods: Jacobi Method

$$\bar{\phi}_{i,j} = \frac{1}{4}(\phi_{i-1,j} + \phi_{i+1,j} + \phi_{i,j-1} + \phi_{i,j+1}) .$$

$$\max(|\bar{\phi} - \phi|) < \varepsilon,$$

```
y  = np.zeros([N+2,M+2])
y[0,:]=V

while (error > eps):
    yp[1:N+1,1:M+1] = y[0:N,1:M+1] +y[2:N+2,1:M+1] \
                    + y[1:N+1,0:M] +y[1:N+1,2:M+2]
    yp[1:N+1,1:M+1] /= 4

    error = np.max(np.absolute(yp[1:N+1,1:M+1]-
y[1:N+1,1:M+1]))
    y = yp.copy()
```

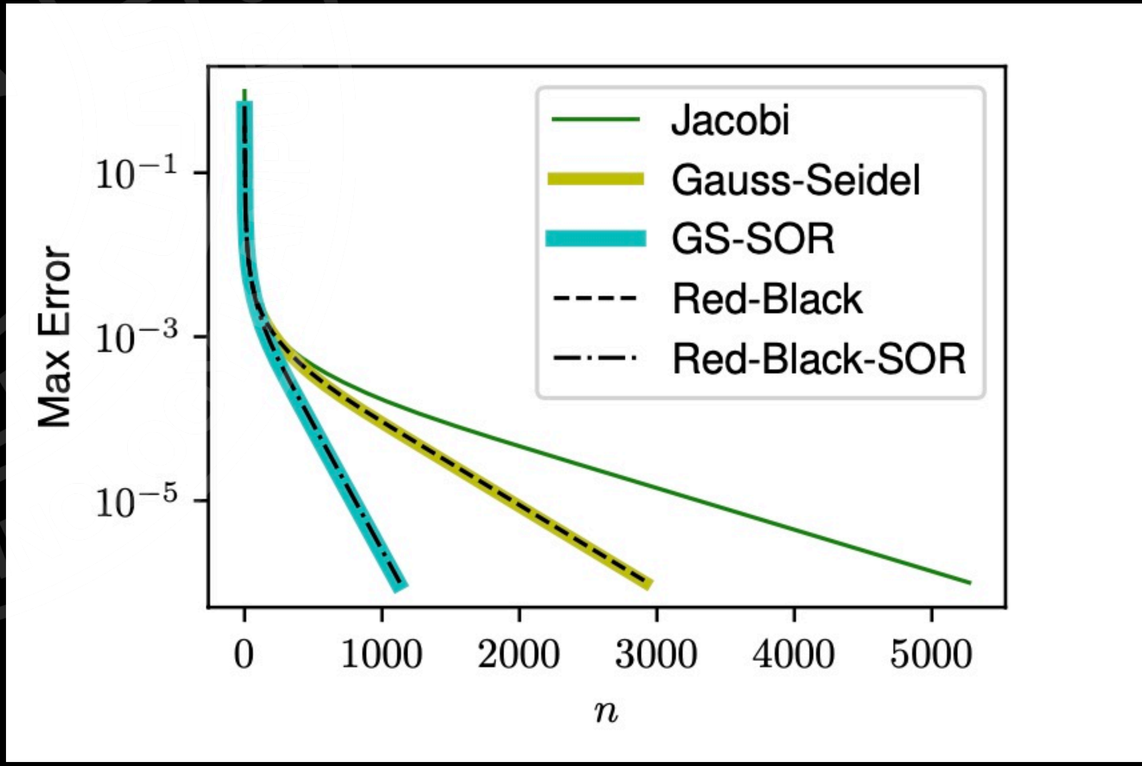

1.0
0.8
0.6
0.4
0.2
0.0


Jacobi
Gauss-Seidel
GS-SOR
Red-Black
Red-Black-SOR
Max Error
10^{-1}
10^{-3}
10^{-5}
0
1000
2000
3000
4000
5000
n

Gauss-Seidel method

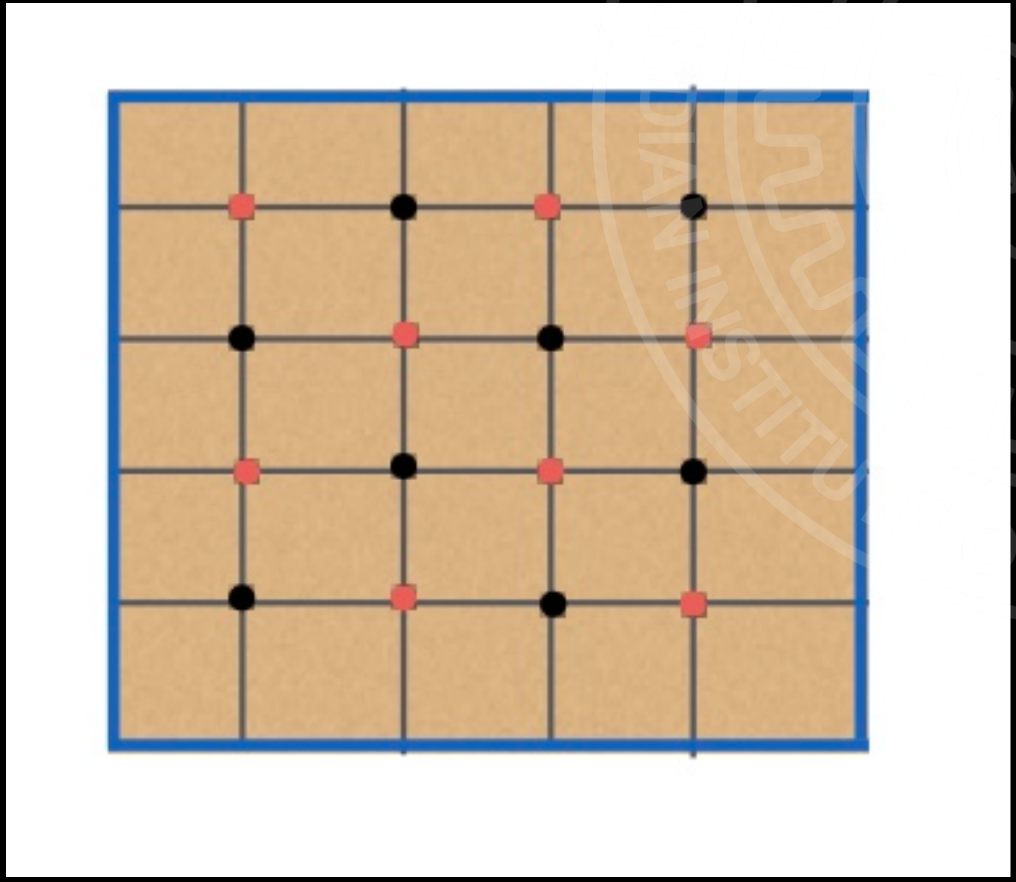
$$\bar{\phi}_{i,j} \leftarrow \frac{1}{4}(\phi_{i-1,j} + \phi_{i+1,j} + \phi_{i,j-1} + \phi_{i,j+1}) \, .$$

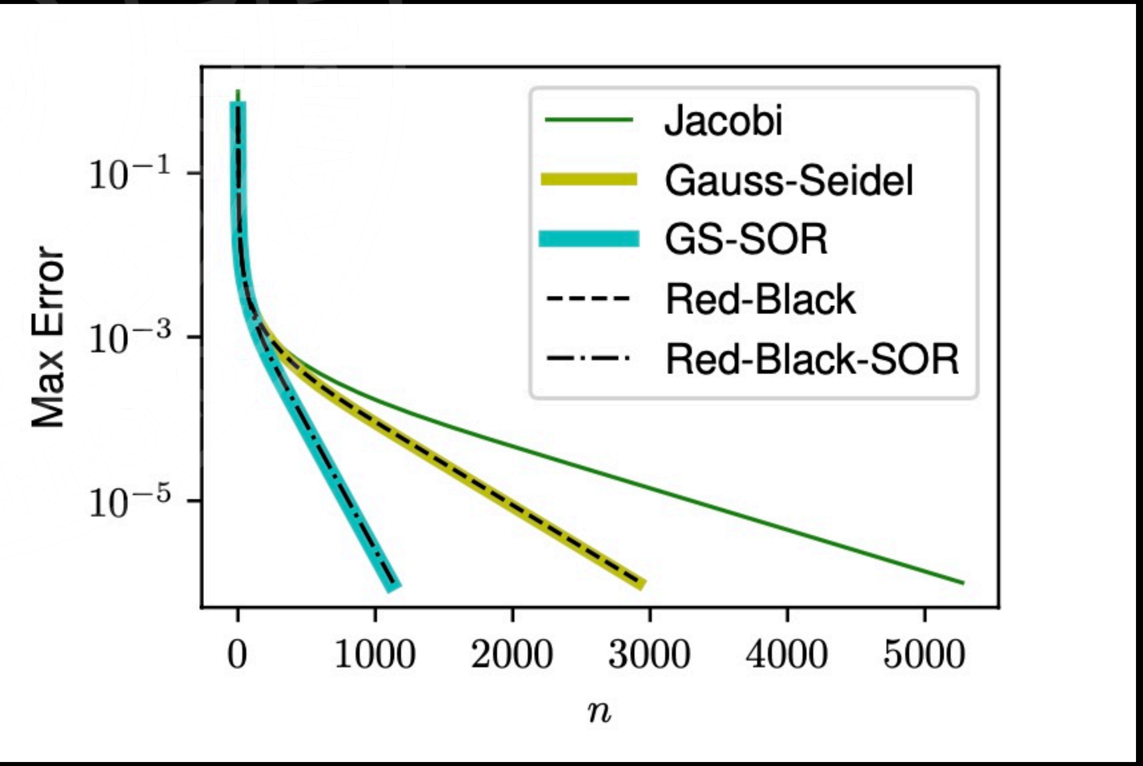
```
while (error > eps):
    yp = y.copy()  # prev step
    for i in range(1,N+1):
        for j in range(1,M+1):
            y[i,j] = (y[i-1,j] +y[i+1,j] + y[i,j-1] \
                     +y[i,j+1])/4

    error = np.max(np.absolute(yp[1:N+1,1:M+1] \
                - y[1:N+1,1:M+1]))
```



Red-Black Method

$$\bar{\phi}_{i,j} \leftarrow \frac{1}{4}(\phi_{i-1,j} + \phi_{i+1,j} + \phi_{i,j-1} + \phi_{i,j+1}) \,.$$

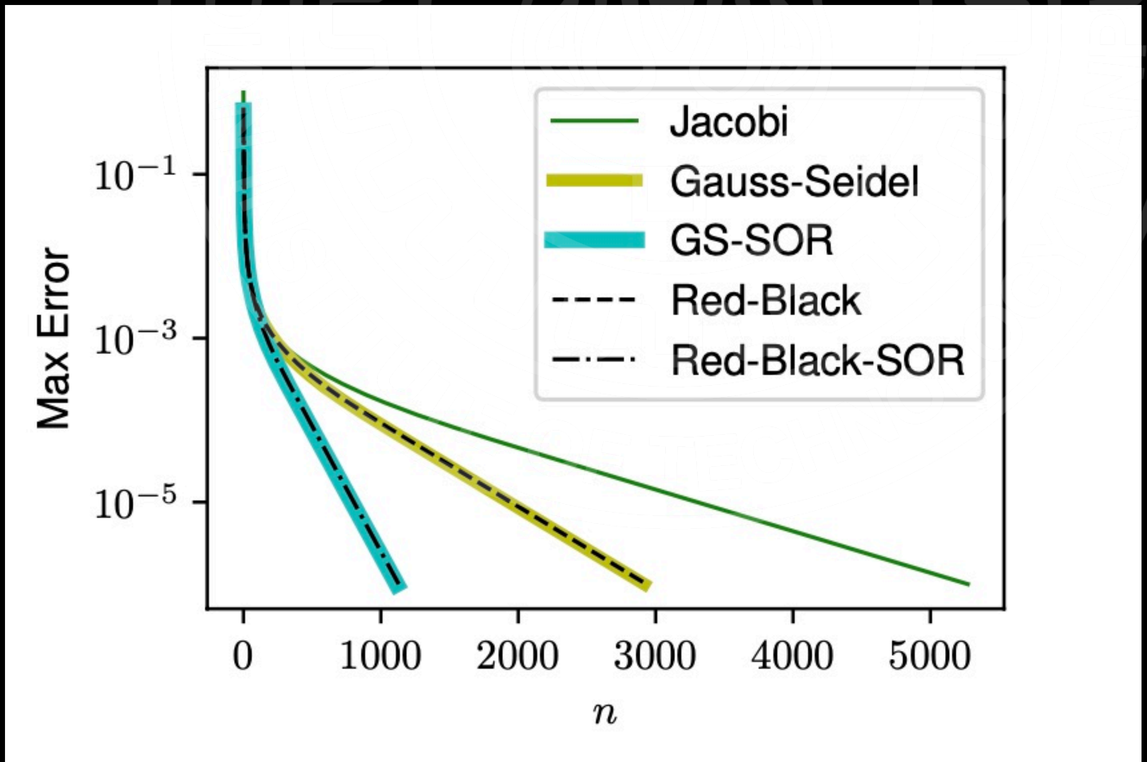




Successive Over-relaxation (SOR)

$$\bar{\phi}_{i,j} \leftarrow \frac{\omega}{4}(\phi_{i-1,j} + \phi_{i+1,j} + \phi_{i,j-1} + \phi_{i,j+1}) + (1-\omega)\phi_{i,j}.$$

ω in [1,2]



Timings

Scheme	Iterations	Time
Jacobi	2573	2.5
Gauss-Seidel	2918	15
Red-Black	2934	0.20
Gauss-Seidel-SOR	1116	7.6
Red-Black-SOR	1131	0.10

Thank you!

Solvers for Poisson Equation

Mahendra Verma

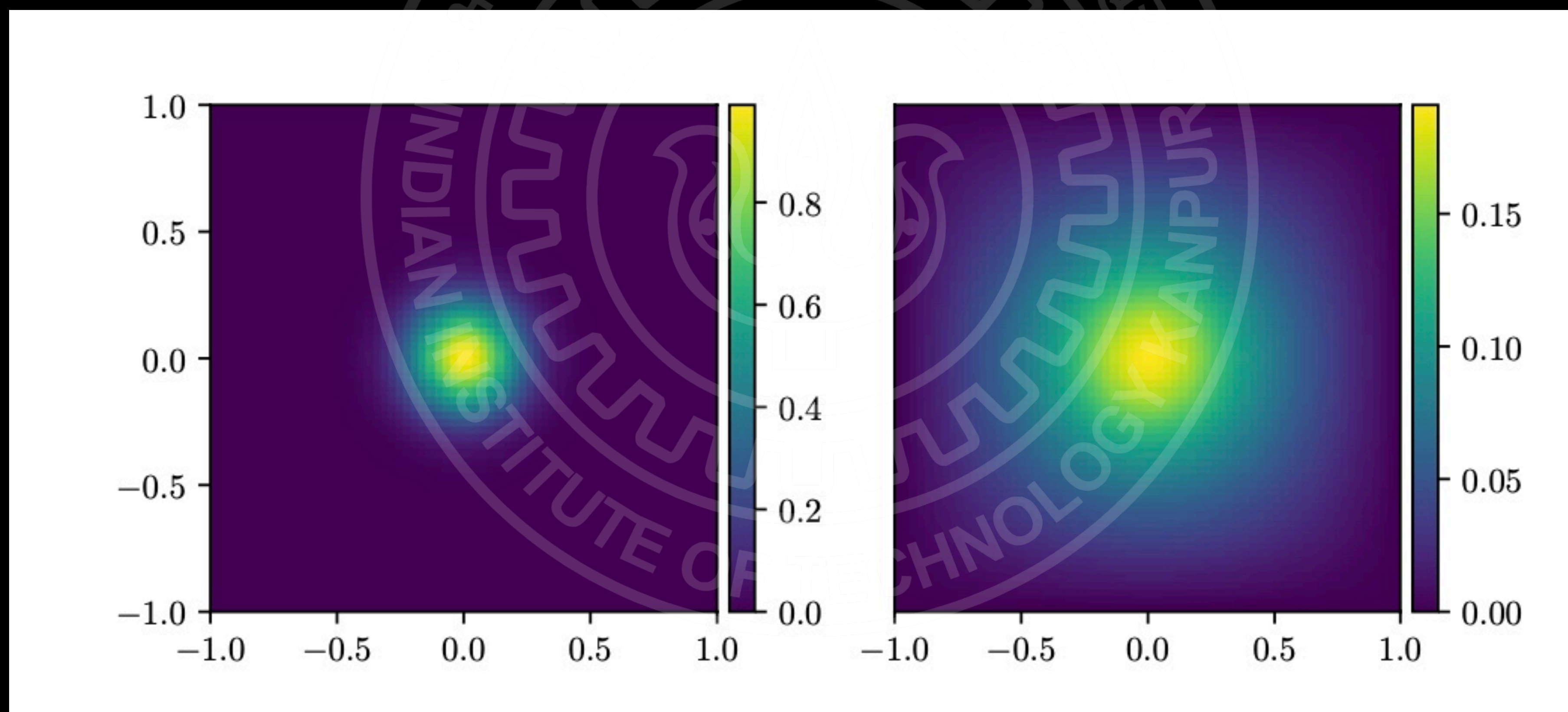
$$\nabla^2\phi = f.$$

$$\frac{1}{h^2}(-4\phi_{i,j} + \phi_{i-1,j} + \phi_{i+1,j} + \phi_{i,j-1} + \phi_{i,j+1}) = f_{i,j},$$

$$\mathrm{A}\begin{bmatrix} \phi_{0,1} \\ \cdots \\ \phi_{0,m} \\ \phi_{1,0} \\ \phi_{1,1} \\ \cdots \end{bmatrix} = \begin{bmatrix} f_{0,1} \\ \cdots \\ f_{0,m} \\ f_{1,0} \\ f_{1,1} \\ \cdots \end{bmatrix},$$

Jacobi Method

$$\bar{\phi}_{i,j} = \frac{1}{4}(\phi_{i-1,j} + \phi_{i+1,j} + \phi_{i,j-1} + \phi_{i,j+1}) - \frac{h^2}{4} f_{i,j} \, .$$



The source term $\exp(-20r^2)$.

ϕ

- We can apply other methods like Gauss-Seidel, Red-black, etc.

Thank you!