Elliptic Equation

Typical equation: Poisson Equation $\Delta\phi = \rho$

$$\frac{d^2\phi}{dx^2} = \rho \qquad \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} = \rho \qquad \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} + \frac{\partial^2\phi}{\partial z^2} = \rho$$

Very important equation in science and engineering

- Maxwell's Equation for Scalar Potential
- Incompressible Navier-Stokes Equation
- Steady State of Diffusion (Thermal Conduction) Equation

Poisson Equation

1-dimensional case:

$$\frac{\phi_{j+1} - 2\phi_j + \phi_{j-1}}{\Delta x^2} = \rho_j$$

2-dimensional case:

$$\frac{\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}}{\Delta x^2} + \frac{\phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1}}{\Delta y^2} = \rho_{i,j}$$

Poisson Equation

N x N linear coupling equations

$$\begin{aligned}
&\phi_1 && = 0 \\
&\phi_1 - 2\phi_2 + \phi_3 && = \rho_2 \Delta x^2 \\
&\phi_2 - 2\phi_3 + \phi_4 && = \rho_3 \Delta x^2 \\
&\phi_3 - 2\phi_4 + \phi_5 && = \rho_4 \Delta x^2 \\
&\cdots && = \cdots \\
&\phi_{j-1} - 2\phi_j + \phi_{j+1} && = \rho_j \Delta x^2 \\
&\cdots && = \cdots \\
&\phi_{N-2} - 2\phi_{N-1} + \phi_N && = \rho_{N-1} \Delta x^2 \\
&\phi_N && = 0
\end{aligned}$$

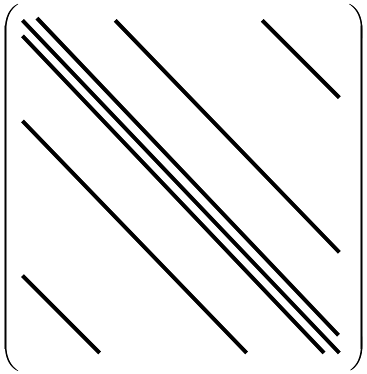
Poisson Equation

$$\begin{pmatrix}
1 & & & & & & \\
1 & -2 & 1 & & & & \\
 & \cdot & \cdot & \cdot & & & \\
 & & 1 & -2 & 1 & & \\
 & & & \cdot & \cdot & \cdot & \\
 & & & & 1 & -2 & 1 \\
 & & & & & & 1
\end{pmatrix}
\begin{pmatrix} \phi_1 \\ \phi_2 \\ \\ \phi_j \\ \\ \phi_{N-1} \\ \phi_N \end{pmatrix}
= \Delta x^2
\begin{pmatrix} \rho_1 \\ \rho_2 \\ \\ \rho_j \\ \\ \rho_{N-1} \\ \rho_N \end{pmatrix}$$

Gauss Elimination Method is not available, because the matrix size is too large.

Sparse Matrix:

3-D Poisson Equation



$$\begin{pmatrix} \phi_{1,1,1} \\ \phi_{2,1,1} \\ \vdots \\ \phi_{i,j,k} \\ \vdots \\ \phi_{N-1,N,N} \\ \phi_{N,N,N} \end{pmatrix} = \Delta x^2 \begin{pmatrix} \rho_{1,1,1} \\ \rho_{2,1,1} \\ \vdots \\ \rho_{i,j,k} \\ \vdots \\ \rho_{N-1,N,N} \\ \rho_{N,N,N} \end{pmatrix}$$

For example, N_x x N_y x N_z = 100x100x100
Double precision 8 byte x $(1000000)^2 = 8\times10^{12}$ = 8 TB

c.f. TSUBAME : 20TB

Matrix Solver

Relaxation Method:

- Point Jacobi
- Gauss-Seidel
- SOR
- ICCG (Incomplete Conjugate Gradient)
- ILUCR (Incomplete LU Conjugate Residual)
- BiCGStab (Bi-conjugate Conjugate Gradient Stabilize)

■ Advantage : Memory and CPU time
■ Disadvantage : No guarantee to be solved
limited types of matrix

Point Jacobi Method

Expecting iterative convergence:

$$\frac{\phi^n_{j+1} - 2\phi^{n+1}_j + \phi^n_{j-1}}{\Delta x^2} = \rho_j$$

The n+1-th value is calculated by the n-th value.

$$\phi^{n+1}_j = \frac{1}{2}\left(\phi^n_{j+1} + \phi^n_{j-1} - \Delta x^2 \rho_j\right)$$

If $\phi^{n+1}_j \approx \phi^n_j$, ϕ^n_j is the solution of Poisson equation.

Point Jacobi Method

Starting from the initial value ϕ^0_j,

$$\phi^1_j = \frac{1}{2}\left(\phi^0_{j+1} + \phi^0_{j-1} - \Delta x^2 \rho_j\right)$$

Iterative calculations give $\phi^1_j, \phi^2_j, \phi^3_j, \ldots \phi^n_j, \phi^{n+1}_j$

If $\left|\phi^{n+1}_j \approx \phi^n_j\right| < \varepsilon$ is satidfied, the iteration has been converged.

Point Jacobi Method

von Neumann's stability analysis
for the iterative process:

Assuming the perturbation $\phi_j^n = \delta\phi^n \, e^{ik \cdot j\Delta x}$

$$\delta\phi^{n+1} / \delta\phi^n = \frac{1}{2}\left(e^{ik\Delta x} + e^{-ik\Delta x}\right) = \cos k\Delta x$$

The iteration process is stable, but slow.
Actual stability depends on the source term.

Point Jacobi Method

Introduction of the relaxation factor : ω

$$\phi_j^{n+1} = (1-\omega)\phi_j^n + \omega \frac{1}{2}\left(\phi_{j+1}^n + \phi_{j-1}^n - \Delta x^2 \rho_j\right)$$

Stability analysis $\delta\phi^{n+1} / \delta\phi^n = (1-\omega) + \omega \cos k\Delta x \ < \ 1$

$(0 \le \omega \le 1)$

Small $\delta\phi^{n+1} / \delta\phi^n$ means rapid decrease of the error.

$\omega = 1$ is found to be the fastest convergence of Jacobi Iteration method.

Source Code

SOR Method

Any faster convergence technique ?

When we calculate ϕ_j^{n+1} , if ϕ_{j-1}^{n+1} has been calculated,
it is better convergence to use ϕ_{j-1}^{n+1} .

$$\phi_j^{n+1} = \frac{1}{2}\left(\phi_{j+1}^n + \phi_{j-1}^{n+1} - \Delta x^2 \rho_j\right)$$

This iteration method is called "Gauss-Seidel" method.

Source Code

SOR Method

Introduction of the relaxation factor : ω

$$\phi_j^{n+1} = (1-\omega)\phi_j^n + \omega \frac{1}{2}\left(\phi_{j+1}^n + \phi_{j-1}^{n+1} - \Delta x^2 \rho_j\right)$$

The stability analysis shows that the iteration process is stable for $0 < \omega < 2$.

Acceleration of the iteration : $1 < \omega < 2$.

SOR (Successive Over-relaxation) Method

The fastest convergence is achieved for $\omega \approx 1.82$.