ME667 Course Notes

Sourabh Apte

1 Iterative Methods for Matrix Inversion

1. Basic Concept
2. Point-Jacobi, Gauss-Seidel, SOR

Inversion by iterative schemes:

① $A\vec{x} = \vec{b}$

Let $A = A_1 - A_2$

$$A_1\vec{x} = A_2\vec{x} + \vec{b}$$

One can construct an iterative method by writing

$$A_1\,\vec{x}^{(k+1)} = A_2\,\vec{x}^{(k)} + \vec{b}$$

k is the iteration index. Starting with some guess $\vec{x}^{(0)}$ this scheme can be used to obtain solution to the system $A\vec{x} = \vec{b}$

The idea is as $k \to \infty$ $\quad \vec{x}^{(k+1)} \longrightarrow \vec{x}^{(k)}$

Then we get $\quad A_1\,\vec{x}^{(k+1)} - A_2\,\vec{x}^{(k)} \approx (A_1 - A_2)\,\vec{x}^{(k)}$

$$= A\,\vec{x}^{(k)} = \vec{b}$$

Requirements:

① A_1 should be easily invertable

② Iterations should converge

$$\lim_{k\to\infty} \vec{x}^{(k)} = \vec{x}$$

Let the error at iteration k be $\epsilon^{(k)} = x - x^{(k)}$

we can write

$$A_1\left(\vec{x} - \vec{x}^{(k+1)}\right) = A_2\left(\vec{x} - \vec{x}^{(k)}\right) + \left(\vec{b} - \vec{b}\right)$$

$$\Rightarrow \quad A_1\,\epsilon^{(k+1)} = A_2\,\epsilon^{(k)}$$

$$\epsilon^{(k+1)} = A_1^{-1}A_2\,\epsilon^{(k)}$$

$$\epsilon^{(k)} = \left(A_1^{-1}A_2\right)^k \epsilon^{(0)}$$

so we need

$$\lim_{k\to\infty} \epsilon^{(k)} = 0$$

$A_1^{-1} A_2 = B$ $\quad$ B^K will tend to zero

iff $|\lambda_i|_{max} \leq 1$ $\quad$ λ_i are eigenvalues

$\rho = |\lambda_i|_{max}$ is the spectral radius

Point - Jacobi

To have easy inversion of a matrix a diagonal matrix is preferred.

So let A_1 be Diagonal matrix

$$A_1 = \begin{bmatrix} -4 & & & & \\ & -4 & & & \\ & & -4 & & \\ & & & \ddots & \\ & & & & \ddots \end{bmatrix}$$

in the Laplace problem

Then $A_2 = A_1 - A = \begin{bmatrix} 0 & & & -1 & \\ -1 & 0 & -1 & & -1 \\ & -1 & 0 & -1 & \\ & & \ddots & \ddots & \\ & & & \ddots & \ddots \end{bmatrix}$ The other elements

$$A_1^{-1} = \begin{bmatrix} -1/4 & & \\ & -1/4 & \\ & & \ddots \end{bmatrix}$$

$A_1 \phi^{(K+1)} = A_2 \phi^{(K)} + \vec{b}$ where ϕ is the unknown vector

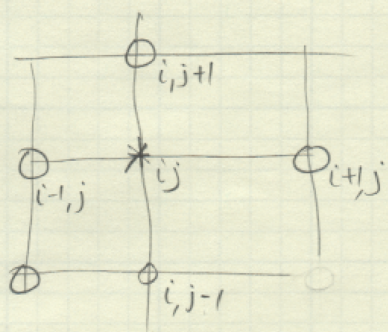
For each ϕ,

$$\boxed{\phi^{(k+1)} = -\frac{1}{4} A_2 \phi^{(k)} - \frac{1}{4}\vec{b}}$$

Using A_2 and the index notation

$$\boxed{\phi_{ij}^{(k+1)} = \frac{1}{4}\left[\phi_{i-1,j}^{(k)} + \phi_{i+1,j}^{(k)} + \phi_{i,j-1}^{(k)} + \phi_{i,j-1}^{(k)}\right] - \frac{1}{4} b_{ij}} \quad (1)$$

i, j are used in the same order as the matrix eq^n in the Laplace eq^n



ϕ_{ij} depends one values of ϕ at "immediate" neighbors. No matrix storage involve

So starting with $\phi_{ij}^{(0)}$ guess (could be $=0$) one just updates $\phi_{ij}^{(k+1)}$ using above eq^n. (1).

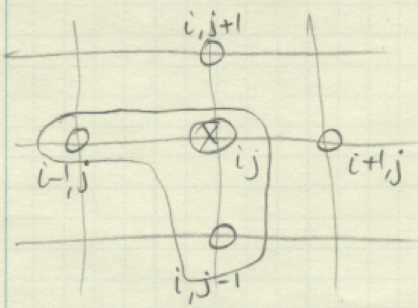
eigenvalues of $A_1^{-1} A_2 = -\frac{1}{4} A_2$ should be such that $|\lambda_i|_{max} < 1$

From Linear Algebra for (uniform grids) eigen values are

$$\lambda_{mn} = \frac{1}{2}\left[\cos\frac{\pi m}{M} + \cos\frac{\pi n}{N}\right]$$

$M = 1, 2, 3, \ldots M-1$ & $N = 1, 2, 3, \ldots N-1$ are grid points in x & y

Gauss-Seidel



In point-Jacobi iterative scheme notice that we solve ϕ_{ij} in an orderly fashion. In other words $\phi_{i-1,j}$ & $\phi_{i,j-1}$ will be solved before ϕ_{ij}.

So we will have $\phi_{i-1,j}^{(k+1)}$, $\phi_{i,j-1}^{(k+1)}$ available before we try to obtain $\phi_{ij}^{(k+1)}$. Gauss-Seidel makes use of "latest" or most updated values of ϕ_{ij}

Then

$$\phi_{ij}^{(k+1)} = \frac{1}{4}\left[\phi_{i-1,j}^{(k+1)} + \phi_{i+1,j}^{(k)} + \phi_{ij-1}^{(k+1)} + \phi_{ij+1}^{(k)}\right] - \frac{1}{4} b_{ij}$$

In the matrix form $A = A_1 - A_2$

Here $A_1 = D - L$, $A_2 = U$

$$A_1 = \begin{bmatrix} -4 & & & & \\ -1 & -4 & & & \\ & -1 & -4 & & \\ & & \ddots & \ddots & \\ -1 & & & \ddots & \ddots \end{bmatrix} \qquad A_2 = \begin{bmatrix} 0 & -1 & & & -1 & \\ & 0 & -1 & & & -1 \\ & & 0 & -1 & & \\ & & & \ddots & \ddots & \\ & & & & & 0 \end{bmatrix}$$

lower triangular
easy to invert

for Gauss-Seidel

$$\lambda_{mn} = \frac{1}{4}\left[\cos\frac{\pi m}{M} + \cos\frac{n\pi}{N}\right]^2$$

Square of the eigenvalues of point-Jacobi

Twice as fast

SOR (Successive over relaxation)

Note $\quad A_1 \vec{x}^{(k+1)} = A_2 \vec{x}^{(k)} + \vec{b}$

$\phi^{(k+1)} = \phi^{(k)} + \vec{d}$ or change in ϕ at each iteration is $\phi^{(k+1)} - \phi^{(k)} = \vec{d}$.

The idea of SOR is to "accelerate" this change at each iteration in order to reduce # of iterations.

This is done by saying

$$\phi^{(k+1)} - \phi^{(k)} = \omega \vec{d}$$

ω is the relaxation factor

$\omega > 1$ for accelerated convergence. Hence over-relaxation

$\omega = 1$ Gauss-Seidel

$\omega < 1$ slower convergence

We first use Gauss-Seidel, which can be written as

$$D\tilde{\phi}^{(k+1)} = L\phi^{(k+1)} + U\phi^{(k)} + \vec{b}$$

$\tilde{\phi}^{(k+1)}$ is not considered as the "accepted" value at $k+1$.

$\phi^{(k+1)} = \phi^{(k)} + \omega\left(\tilde{\phi}^{(k+1)} - \phi^{(k)}\right)$ is used for weighted convergence

$$\phi^{(k+1)} = (1-\omega)\,\phi^{(k)} + \omega\,\tilde{\phi}^{(k+1)}$$

$\omega = 0 \Rightarrow$ Gauss-seidel (GS)

$\omega > 1$ over-relaxation.

So we weight the predicted value for GS more than the old iteration value

Combining the two eqns

$$\tilde{\phi}^{(k+1)} = D^{-1}L\phi^{(k+1)} + D^{-1}U\,\phi^{(k)} + D^{-1}b$$

$$\phi^{(k+1)} = (1-\omega)I\phi^{(k)} + \omega D^{-1}U\,\phi^{(k)} + \omega D^{-1}b + \omega D^{-1}L\,\phi^{(k+1)}$$

$$\Rightarrow \left(I - \omega D^{-1}L\right)\phi^{(k+1)} = \left[(1-\omega)I + \omega D^{-1}U\right]\phi^{(k)} + \omega D^{-1}b$$

$$\phi^{(k+1)} = \underbrace{\left[I - \omega D^{-1}L\right]^{-1}\left[(1-\omega)I + \omega D^{-1}U\right]}_{G_{SOR}}\phi^{(k)} + \left[I - \omega D^{-1}L\right]^{-1}\omega D^{-1}\vec{b}$$

To optimize convergence means selecting ω to minimize the largest eigenvalue

For rectangular regions with uniform mesh, ω_{opt} can be obtained analytically

$$\omega_{opt} = 2\left[\frac{1-\sqrt{1-\mu_{max}^2}}{\mu_{max}^2}\right],$$

μ_{max} is max. eigen value of point-Jacobi