Programming linear solver

Praveen. C
praveen@math.tifrbng.res.in

Tata Institute of Fundamental Research
Center for Applicable Mathematics
Bangalore 560065
http://math.tifrbng.res.in

March 24, 2023

Linear problem

We want to find x such that

$$Ax = b \tag{1}$$

A is symmetric positive definite. The function

$$J(y) = \frac{1}{2} y^\top A y - b^\top y$$

is strictly convex and has unique minimum x which is characterized by

$$J'(x) = Ax - b = 0$$

The solution of (1) is the unique minimizer of J.

Steepest descent method

Descent direction

$$d_n = -J'(x_n) = b - Ax_n =: r_n$$

Determine step size

$$J(x_n + \omega_n d_n) = \min_{\omega} J(x_n + \omega d_n) \quad \Longrightarrow \quad \omega_n = \frac{d_n^\top r_n}{d_n^\top A d_n}$$

Update

$$x_{n+1} = x_n + \omega_n d_n$$

Conjugate direction method

In first step, descent direction

$$d_0 = -J'(x_0) = b - Ax_0$$

Then for $n = 1, 2, \ldots$, choose next descent direction d_n to be A-orthogonal to all previous descent directions

$$d_j^\top A d_n = 0, \qquad 0 \le j \le n-1$$

Determine step size

$$J(x_n + \omega_n d_n) = \min_\omega J(x_n + \omega d_n) \quad \Longrightarrow \quad \omega_n = \frac{d_n^\top r_n}{d_n^\top A d_n}$$

Update

$$x_{n+1} = x_n + \omega_n d_n$$

CG algorithm

Assume initial guess x_0 is given. Otherwise set $x_0 = 0$
Set $n = 0$

$$r_0 = b - Ax_0, \qquad d_0 = r_0$$

While $\|r_n\| > \varepsilon \|r_0\|$ and $n < N_{max}$

- If $n > 0$

$$d_n = r_n + \frac{\|r_n\|^2}{\|r_{n-1}\|^2} d_{n-1}$$

- $\omega_n = \frac{\|r_n\|^2}{d_n^\top A d_n}$
- $x_{n+1} = x_n + \omega_n d_n$
- $r_{n+1} = r_n - \omega_n A d_n$
- $n = n + 1$

Need matrix-vector product and dot product of two vectors.

CG algorithm: Computer version

Assume initial guess x is given. Otherwise set $x = 0$
Set $n = 0$

$$r = b - Ax, \qquad d = r, \qquad \rho_0 = \|r_0\|^2$$

While $\sqrt{\rho_n} > \varepsilon\sqrt{\rho_0}$ and $n < N_{max}$

- If $n > 0$
$$d \leftarrow r + \frac{\rho_n}{\rho_{n-1}} d$$
- $v = Ad$
- $\omega = \frac{\rho_n}{d^\top v}$
- $x \leftarrow x + \omega d$
- $r \leftarrow r - \omega v$
- $\rho_{n+1} = \|r\|^2$
- $n \leftarrow n + 1$

Remark: Need storage for three vectors: r, d, v

Jacobi method

Solve i'th equation of

$$Ax = b$$

for x_i

$$x_i = \frac{1}{a_{ii}} \left[b_i - \sum_{j=1, j\neq i}^{N} a_{ij} x_j \right]$$

Jacobi method: Make an initial guess x^0 and then iterate

$$x_i^{n+1} = \frac{1}{a_{ii}} \left[b_i - \sum_{j=1, j\neq i}^{N} a_{ij} x_j^n \right]$$

Rewrite

$$x_i^{n+1} = x_i^n + \frac{1}{a_{ii}} \left[b_i - \sum_{j=1}^{N} a_{ij} x_j^n \right]$$

Jacobi method

Algorithm:

Make initial guess x, set $n = 0$

1. $r = b - Ax$, $\rho = \|r\|$
2. If $n = 0$, then $\rho_0 = \rho$
3. If $\rho < \varepsilon\rho_0$, then stop.
4. For $i = 1, 2, \ldots, N$
$$x_i \leftarrow x_i + \frac{r_i}{a_{ii}}$$
5. $n \leftarrow n + 1$, go to step (1)

Remark: Need additional storage for vector r.

Remark: Very good for parallelization

Gauss-Seidel (GS) method

Solve i'th equation of

$$Ax = b$$

for x_i

$$x_i = \frac{1}{a_{ii}} \left[b_i - \sum_{j=1, j\neq i}^{N} a_{ij} x_j \right]$$

GS method: Make an initial guess x^0 and then iterate

$$x_i^{n+1} = \frac{1}{a_{ii}} \left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{n+1} - \sum_{j=i+1}^{N} a_{ij} x_j^n \right]$$

Rewrite

$$x_i^{n+1} = x_i^n + \frac{1}{a_{ii}} \left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{n+1} - \sum_{j=i}^{N} a_{ij} x_j^n \right]$$

Gauss-Seidel (GS) method

Algorithm:

Make initial guess x, set $\rho = \rho_0 = 1$

1. If $\rho < \varepsilon \rho_0$, then stop.
2. $\rho = 0$
3. For $i = 1, 2, \ldots, N$

$$r = b_i - \sum_{j=1}^{N} a_{ij} x_j$$

$$x_i \leftarrow x_i + \frac{r}{a_{ii}}$$

$$\rho \leftarrow \rho + r * r$$

4. $\rho \leftarrow \sqrt{\rho}$
5. If $n = 0$, $\rho_0 = \rho$
6. $n \leftarrow n + 1$, go to step (1)

Remark: No additional storage is required.

Remark: Cannot be parallelized.

Successive Over Relaxation (SOR) method

A variation on GS method. Choose $\omega > 1$

For $i = 1, 2, \ldots, N$

$$
\begin{aligned}
x_i^* &= \frac{1}{a_{ii}} \left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{n+1} - \sum_{j=i}^{N} a_{ij} x_j^n \right] + x_i^n \\
x_i^{n+1} &= x_i^n + \omega (x_i^* - x_i^n)
\end{aligned}
$$

or equivalently

$$
x_i^{n+1} = x_i^n + \frac{\omega}{a_{ii}} \left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{n+1} - \sum_{j=i}^{N} a_{ij} x_j^n \right]
$$

Remark: $\omega = 1$ yields the GS method.

Successive Over Relaxation (SOR) method

Algorithm:

Make initial guess x, set $\rho = \rho_0 = 1$

1. If $\rho < \varepsilon \rho_0$, then stop.
2. $\rho = 0$
3. For $i = 1, 2, \ldots, N$

$$r = b_i - \sum_{j=1}^{N} a_{ij} x_j$$

$$x_i \leftarrow \omega \frac{r}{a_{ii}} + x_i$$

$$\rho \leftarrow \rho + r * r$$

4. $\rho \leftarrow \sqrt{\rho}$
5. If $n = 0$, $\rho_0 = \rho$
6. $n \leftarrow n + 1$, go to step (1)

Symmetric Gauss-Seidel (SGS) method

For $i = 1, 2, \ldots, N$

$$x_i^* = \frac{1}{a_{ii}} \left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j^* - \sum_{j=i}^{N} a_{ij} x_j^n \right] + x_i^n$$

For $i = N, N-1, \ldots, 1$

$$x_i^{n+1} = \frac{1}{a_{ii}} \left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j^* - \sum_{j=i}^{N} a_{ij} x_j^{n+1} \right] + x_i^*$$

Remark: Symmetric SOR (SSOR) method is obtained by doing over-relaxation in SGS method.