Welcome to Ma221! (Apr 12)

Splitting Methods for $Ax=b$

Goal: Given an initial guess x_0 for solution of $Ax=b$, cheaply compute sequence $x_i \to A^{-1}b$

Def: Splitting of $A = M - K$, M nonsingular

$$Ax=b \Rightarrow Mx = Kx + b$$

compute x_{i+1} from x_i by solving

$$M x_{i+1} = K x_i + b$$

or $x_{i+1} = \underbrace{M^{-1}K}\, x_i + M^{-1}b$

or (*) $x_{i+1} = R x_i + c$

for this to work well, need

(1) x_i should converge to $A^{-1}b$

(2) Solving $Mx_{i+1} = Kx_i + b$ for x_{i+1} should be much cheaper than solving with A

Lemma: Let $\|\cdot\|$ be any operator norm then if $\|R\| < 1$, (*) converges to $A^{-1}b$ for any x_0

Proof: Subtract $x = Rx + c$ from $(*)$

to get $x_{i+1} - x = R(x_i - x)$

$$\dots = R^{i+1}(x_0 - x)$$

$$\|x_{i+1} - x\| \le \|R^{i+1}\| \cdot \|x_0 - x\|$$

$$\le \|R\|^{i+1} \cdot \|x_0 - x\|$$

$$\to 0 \text{ if } \|R\| < 1$$

Def: The spectral radius of R is

$$\rho(R) = \max_{\lambda \text{ eval of } R} |\lambda|$$

Lemma: For all operator norms $\rho(R) \le \|R\|$

For all R and $\varepsilon > 0$, there exists an operator norm $\|\cdot\|^*$ such that

$$\|R\|^* \le \rho(R) + \varepsilon$$

Proof: To show $\rho(R) \le \|R\| = \max_{x \ne 0} \frac{\|Rx\|}{\|x\|}$

choose $x =$ evec for λ, $|\lambda| = \rho(R)$

so $\|R\| \ge \frac{\|Rx\|}{\|x\|} = \frac{\|\lambda x\|}{\|x\|} = |\lambda|$

To construct $\|\cdot\|^*$, use Jordan Form of R

$$S^{-1}RS = J = \begin{bmatrix} \square & & & \\ & \square & & \\ & & \square & \\ & & & \ddots \end{bmatrix} \qquad \square = \begin{bmatrix} \lambda & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \lambda \end{bmatrix}$$

$$D = \begin{bmatrix} 1 & & & & \\ & \varepsilon & & & \\ & & \varepsilon^2 & & \\ & & & \ddots & \\ & & & & \varepsilon^{n-1} \end{bmatrix}$$

$D^{-1} J D$ multiplies superdiagonal entries by ε

$$J_\varepsilon = D^{-1} S^{-1} R S D = \begin{bmatrix} \lambda_1 & \varepsilon & & \\ & \ddots & \ddots & \\ & & \ddots & \varepsilon \\ & & & \lambda_n \end{bmatrix}$$

new vector norm $\|x\|^* = \|(SD)^{-1} x\|_\infty$

$$\|R\|^* = \max_{x \neq 0} \frac{\|Rx\|^*}{\|x\|^*} = \max_{x \neq 0} \frac{\|(SD)^{-1} R x\|_\infty}{\|\underbrace{(SD)^{-1} x}_{y}\|_\infty}$$

$$= \max_{y \neq 0} \frac{\|(SD)^{-1} R (SD) y\|_\infty}{\|y\|_\infty} = \max_{y \neq 0} \frac{\|J_\varepsilon y\|_\infty}{\|y\|_\infty}$$

$$= \|J_\varepsilon\|_\infty \leq \rho(R) + \varepsilon$$

Thm: $x_{i+1} = R x_i + c$ converges to $A^{-1} b$ for all x_0 if and only if $\rho(R) < 1$

proof: If $\rho(R) \geq 1$, choose x_0 so that $x_0 - x$ = evec of R for largest eval λ

$$x_i - x = R^i (x_0 - x) = \lambda^i (x_0 - x)$$

since $|\lambda| \geq 1$, no convergence

If $\rho(R) < 1$, use last lemma to construct an operator norm $\|R\|^*$ such that $\|R\|^* \le \underbrace{\rho(R) + \varepsilon}_{<1}$, choose ε small enough so $\|R\|^* < 1$

$\Rightarrow$ convergence for all x_0 by earlier lemma

Goal: $\rho(R)$ to be as small as possible, but still cheap to solve $M x_{i+1} = K x_i + b$

Ex: $M = I$, $K = I - A$ makes solving for x_{i+1} as cheap as possible, but no guarantees on $\rho(R)$

$K = 0 \Rightarrow R = 0 \Rightarrow$ converge in one step but need $c = A^{-1} b$, so no savings

---

Describe Jacobi
Gauss-Seidel (GS)
Successive Overrelaxation (SOR)

$$A = \begin{bmatrix} \ddots & & -U' \\ & D & \\ -L' & & \ddots \end{bmatrix} = D - L' - U'$$
$$= D(I - L - U)$$

Jacobi: In words:

for $j=1$ to n, pick $x_{i+1}(j)$ to exactly solve equation j

As a loop: for $j=1:n$

$$x_{i+1}(j) = \Big(b_j - \sum_{k\neq j} A_{jk} x_i(k)\Big)/A_{jj}$$

As a splitting: $D x_{i+1} = (L'+U') x_i + b$

$$x_{i+1} = D^{-1}(L'+U') x_i + D^{-1} b$$

$$A = M - K = D - (L'+U')$$

$$R_J = M^{-1}K = D^{-1}(L'+U') = L+U$$

For 2D Poisson:

$$T_N V + V T_N = h^2 F \qquad V \;\; N\times N$$

T_N is 1D Poisson

To get from V_i to V_{i+1}

for $j=1:n$, for $k=1:n$

$$V_{i+1}(j,k) = \big(V_i(j-1,k) + V_i(j+1,k) + V_i(j,k-1) + V_i(j,k+1) + h^2 F(j,k)\big)/4$$

= "average" of 4 nearest neighbors and right hand side

Gauss-Seidel:

In words: improve on Jacobi by using most recently updated values of x

As a loop

for $j=1:n$

$$x_{i+1}(j) = \Big(b_j - \underbrace{\sum_{k<j} A(j,k)\cdot x_{i+1}(k)}_{\text{updated } x_{i+1}} - \underbrace{\sum_{k>j} A(j,k)\cdot x_i(k)}_{\text{older } x_i}\Big)/A(j,j)$$

As a splitting:

$$A = (D-L') - U' = M - K$$

$$R_{GS} = M^{-1}K = (D-L')^{-1}U'$$
$$= (D(I-L))^{-1}U'$$
$$= (I-L)^{-1}U$$

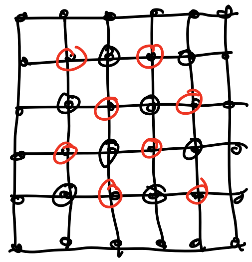
each step of GS triangular solve

In contrast to Jacobi, order in which $x_{i+1}(j)$ updated matters

For 2D Poisson

Natural Order (rowwise or columnwise updating of $V(j,k)$)

Red-Black Ordering:



4x4 with boundaries
Red nodes ($j+k$ even)
Black nodes ($j+k$ odd)

Number all Red nodes before Black nodes

Red/Black nodes only have Black/Red neighbors

$\Rightarrow$ When updating Red nodes, can update them in any order including parallel, since all Black nodes have old data

When updating Black nodes, again in any order, all Red neighbors have updated data

for all Red (j,k) ($j+k$ even)

$$V_{i+1}(j,k) = \big(\underbrace{V_i(j-1,k) + V_i(j+1,k) + V_i(j,k-1) + V_i(j,k+1)}_{\text{old (Black) data}} + h^2 F(j,k)\big)/4$$

for all Black (j,k) ($j+k$ odd)

$$V_{i+1}(j,k) = \underbrace{\big(V_{i+1}(j-1,k) + V_{i+1}(j+1,k) + V_{i+1}(j,k-1) + V_{i+1}(j+1,k)}_{\text{updated (Red) data}} + h^2 F(j,k)\big)/4$$

SOR:

In words: Depends on parameter w

Result of SOR = weighted combination of old x and result of GS

$$x^{SOR}_{w\ i+1}(j) = (1-w)\,x_i(j) + w\,x^{GS}_{i+1}(j)$$

$w=1 \Rightarrow$ same as GS

$w<1$ = "under relaxation", not useful

$w>1$ = "over relaxation"

Motivation: go farther in same direction that GS would have gone

Later: choose w optimally for Poisson

As a loop:

for $j = 1:n$

$$x_{i+1}(j) = (1-w)\,x_i(j) + w\Big(b_j - \sum_{k<j} A(j,k)\,x_{i+1}(j) - \sum_{k>j} A(j,k)\,x_i(j)\Big)/A(j,j)$$

As a Splitting: Multiply inner loop by $A(j,j)$

$$(D-wL')x_{i+1} = ((1-w)Dx_i + wU'x_i) + wb$$

Divide by w

$$A = (D/w - L') - (D/w - D + U')$$
$$= \quad M \quad - \quad K$$

or
$$R_{SOR}(w) = (D/w - L')^{-1}(D/w - D + U')$$
$$= (I - wL)^{-1}((1-w)I + wU)$$

For 2D Poisson:

for all Red (j,k) $\quad$ ($j+k$ even)

$$V_{i+1}(j,k) = (1-w)V_i(j,k) + w \cdot \underbrace{(V_i(j-1,k) + V_i(j+1,k) + V_i(j,k-1) + V_i(j,k+1) + h^2F(j,k))/4}_{\text{old data}}$$

for all Black (j,k) $\quad$ ($j+k$ odd)

$$V_{i+1}(j,k) = (1-w)V_i(j,k) + w \cdot \underbrace{(V_{i+1}(j-1,k) + V_{i+1}(j+1,k) + V_{i+1}(j,k-1) + V_{i+1}(j,k+1) + h^2F(j,k))/4}_{\text{updated data}}$$

Convergence of Splitting Methods
In general, and for 2D Poisson

Jacobi for 2D Poisson

$$T_{n\times n} = M - K = 4I - (4I - T_{n\times n})$$

$$\Rightarrow R = M^{-1}K = I - \frac{1}{4} T_{n\times n}$$

$\Rightarrow$ evals of R are $1 - (\lambda_i + \lambda_j)/4$

λ_i are evals of T_n

$$\lambda_i = 2\left(1 - \cos\frac{i\pi}{n+1}\right)$$

$$\Rightarrow \rho(R) = 1 - \frac{\lambda_{min}}{2}$$

$$= 1 - \left(1 - \cos\frac{\pi}{n+1}\right) = \cos\frac{\pi}{n+1}$$

$$\cong 1 - \frac{\pi^2}{2(n+1)^2} \quad \text{when } n \gg 1$$

$\rho(R)$ gets closer to 1 as n grows

$\Rightarrow$ slower convergence

Error after m steps multiplied by $(\rho(R))^m$

$$\rho(R) = 1 - x$$

$$\Rightarrow \rho(R)^m = (1-x)^m = (1-x)^{\frac{1}{x}mx}$$

$$\approx e^{-mx} \quad \text{for } x \ll 1$$

for $e^{-mx} = e^{-1} \Rightarrow m = \frac{1}{x}$

for Jacobi $\frac{1}{X} = \frac{2(n+1)^2}{\pi^2}$ for $n >> 1$

$= O(n^2)$

$= O(N)$ N = dimension of 2D Poisson

Cost to reduce error by any constant also proportional to $N \Rightarrow$

cost = #iteration · #flops per iteration

$= O(N) \cdot O(N)$

$= O(N^2)$

Typical behavior: slower convergence for larger problems (not multigrid!)

GS: Assuming variables updated in Red-Black order, $\rho(R_{GS}) = (\rho(R_J))^2$

$\Rightarrow$ 1 step of GS same as 2 steps of Jacobi

only constant factor faster

SOR(ω): Again, with Red-Black ordering, and optimal ω, much faster

$\rho(R_{SOR(\omega)}) \cong 1 - \frac{2\pi}{n}$

$\Rightarrow O(n) = O(N^{1/2})$ steps to converge

$\Rightarrow$ cost $= O(N^{3/2})$ to converge

Next: Present (and prove some of) general theory of convergence for Jacobi, GS, SOR(ω) more details for 2D Poisson

Thm 1: If A strictly row diagonally dominant

$$|A_{ii}| > \sum_{j \neq i} |A_{ij}|$$

then both Jacobi and GS converge, GS at least as fast as Jacobi

$$\|R_{GS}\|_\infty \le \|R_J\|_\infty < 1$$

Proof, just for Jacobi (see Thm 6.2 in book for full proof)

Split: $A = D - (D-A)$

$$R = D^{-1}(D-A) = I - D^{-1}A$$

$$\|R\|_\infty = \max_j \sum_i |R(j,i)|$$

$$= \left|1 - \frac{A(j,j)}{A(j,j)}\right| + \sum_{i \neq j} \frac{|A(j,i)|}{|A(j,j)|} \quad \text{for some } j$$

$$= 0 \quad + \underbrace{\frac{1}{|A(j,j)|} \sum_{i \neq j} |A(j,i)|}_{\substack{<1 \text{ by} \\ \text{strict diag dominance}}}$$

$\Rightarrow \|R\|_\infty < 1 \Rightarrow$ convergence

2D Poisson: Not strictly row diag. dom because many rows are $[-1 \quad -1 \quad 4 \quad -1 \quad -1]$

Def: If $|A(j,j)| \geq \sum_{i \neq j} |A(j,i)|$ for all j with strict inequality at least once, A weakly row diagonally dominant

Not enough by itself for Jacobi to converge:

$$A = \left[\begin{array}{cc|c} 1 & -1 & 0 \\ 1 & 1 & 0 \\ \hline 0 & 0 & 1 \end{array}\right] \Rightarrow R \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$\Rightarrow R^5 = R \Rightarrow R^i$ does not converge

Need one more property of A related to its sparsity structure