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 $x_i \rightarrow A^{-1}b$

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

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

compute $x_{i+1}$ from $x_i$ by solving

$Mx_{i+1} = Kx_i + b$

or $x_{i+1} = M^{-1}Kx_i + M^{-1}b$

or (*): $x_{i+1} = Rx_i + c$

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 $x = R \cdot x + c$ from (4)
to get $\hat{x} = R \cdot (x - x) = R \cdot (x - x)$

\[ \| x - x \| = \| R \cdot (x - x) \| \leq \| R \| \cdot \| (x - x) \| \]
\[ \leq \| R \| \cdot \| x - x \| \]

$\rightarrow 0$ if $\| R \| < 1$

Def: The spectral radius of $R$ is

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

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

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

\[ \| R \|^* \leq \rho(R) + \varepsilon \]

Proof: To show $\rho(R) \leq \| R \| = \max_{x \neq 0} \frac{\| R \cdot x \|}{\| x \|}$

choose $x = \text{evect for } \lambda$, $|\lambda| = \rho(R)$

so $\| R \| = \| R \cdot x \| \| x \|^{-1} = \| \lambda \| \| x \|^{-1} = |\lambda|$.

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

\[ S^* R S = J = \begin{bmatrix} \lambda & 1 \\ & \ddots \\ & & \lambda \end{bmatrix} \quad 0 = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \]
\[ D = \begin{bmatrix} \varepsilon & \varepsilon^2 & \cdots & \varepsilon^n \\ \end{bmatrix} \]

\[ D^{-1} J D \text{ multiplies superdiagonal entries by } \varepsilon \]

\[ J_\varepsilon = D^{-1} S^{-1} R S D = \begin{bmatrix} \frac{\varepsilon_1}{\varepsilon_2} & \frac{\varepsilon_2}{\varepsilon_3} & \cdots & \frac{\varepsilon_n}{\varepsilon_1} \\ \end{bmatrix} \]

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

\[ \| R \|_* = \max_{x \neq 0} \frac{\| R x \|_\infty}{\| x \|_*} = \max_{x \neq 0} \frac{\| (SD)^{-1} R x \|_\infty}{\| (SD)^{-1} x \|_\infty} \]

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

\[ \Rightarrow \| J_\varepsilon \|_\infty \leq \rho(R) + \varepsilon \]

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 \( x_0 - x = \mathbf{e}_i \mathbf{v}_c \) of \( R \) for largest \( c \) and

\[ x_i - x = R^i (x_0 - x) = \mathbf{e}_i (x_0 - x) \]

since \( \| \mathbf{e}_i \| \geq 1 \), no convergence
If $p(R) < 1$, use last lemma to construct an operator norm $\|R\|_*$ such that $\|R\|_* < p(R) + \epsilon$, choose $\epsilon$ small enough so $\|R\|_* < 1$

$\Rightarrow$ convergence for all $x_0$

by earlier lemma

Goal: $p(R)$ to be as small as possible, but still cheap to solve $Mx_{i+1} = k_i + b$

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

$K = 0 \Rightarrow R = 0 \Rightarrow$ converge in one step

but need $c = A^-b$, so no savings

Describe

Jacobi

Gauss-Seidel (GS)

Successive Overrelaxation (SOR)

$$A = \begin{bmatrix} D & -u' \\ -l' & B \end{bmatrix} = D - L' - U'$$

$$= D(I - L - U)$$
Jacobi: In words:
for $j = 1$ to $n$, pick $x_{i+1}(j)$ to exactly solve equation $j$

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

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

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:
$$T_N V + V T_N = h^2 F$$

$$T_N \text{ is } 2 \text{D Poisson}$$

To get from $V_i$ to $V_{i+1}$
for $j = 1:n$, for $k = 1:n$

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

= "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 $j = 1:n$

\[
x_{i+1}(j) = (b_j - \sum_{k \neq j} A(j, k) \cdot x_{i+1}(k)) / A(j, j)
\]

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 G-S 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

⇒ 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)

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

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

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

updated Red data

**SOR:**

In words: Depends on parameter $w$

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

$$x_{w,i+1}(j) = (1-w)x_i(j) + w x_{GS,i+1}(j)$$

$w=1$ ⇒ 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 $j = 1 \rightarrow n$

$$x_{i+1}(j) = (1-w)x_i(j) + w \left( b_j - \sum_{k \neq j} A(j,k) x_{i+1}(j) - \sum_{k \neq j} A(j,k) x_i(j) \right)/A(j,j)$$
As a Splitting! Multiply inner loop by $A(j,j)$

$$(D-wL')x_{i+1} = ((1-w)Dx_{i} + wUx_{i}) + wb$$

Divide by $w$

$$A = (D/w - L') - (D/w - D + U')$$

$$= M - K$$

or $R_{so}r(w) = (D/w - L')^{-1} (D/w - D + U')$

$$= (I - wL)^{-1} ((1-w)I + wU)$$

For 2D Poisson:

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

$$v_{i+1}(j,k) = (1-w) v_{i}(j,k) + w \cdot \left( v_{i}(j-1,k) + v_{i}(j+1,k) + v_{i}(j,k-1) + v_{i}(j,k+1) + h^2 f_{i}(j,k) \right)/4$$

old data

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

$$v_{i+1}(j,k) = (1-w) v_{i}(j,k) + w \cdot \left( v_{i+1}(j-1,k) + v_{i+1}(j+1,k) + v_{i+1}(j,k-1) + v_{i+1}(j,k+1) + h^2 f_{i}(j,k) \right)/4$$

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

Jacobi for 2D Poisson
\[ T_{n \times n} = M - K = 4I - (4I - T_{n \times n}) \]
\[ \Rightarrow R = M^{-1} K = I - \frac{1}{4} T_{n \times n} \]
\[ \Rightarrow \text{evals of } R \text{ are } 1 - (\lambda_1 + \lambda_2)/4 \]
\[ \lambda_i \text{ are evals of } T_n \]
\[ \lambda_i = 2 \left(1 - \cos \frac{i \pi}{n+1}\right) \]

\[ \Rightarrow p(R) = 1 - \frac{\lambda_{\text{min}}}{2} = 1 - \left(1 - \cos \frac{\pi}{n+1}\right) = \cos \frac{\pi}{n+1} \]
\[ \approx 1 - \frac{n^2}{2(n+1)^2} \text{ when } n \gg 1 \]

\[ p(R) \text{ gets closer to 1 as } n \text{ grows} \]
\[ \Rightarrow \text{slower convergence} \]

Error after \( m \) steps multiplied by \((p(R))^m\)
\[ p(R) = 1 - x \]
\[ \Rightarrow (p(R))^m = (1 - x)^m = (1 - x)^{\frac{1}{x} n x} \]
\[ \approx e^{-n x} \text{ for } x \ll 1 \]

\[ \text{for } e^{-n x} = e^1 \Rightarrow n = \frac{1}{x} \]
for Jacobi \( \frac{1}{\lambda} = \frac{2(n+1)^2}{n^2} \) for \( n \gg 1 \)

\[ = O(n^2) \]

\[ = O(N) \quad N = \text{dimension of 2D Poisson} \]

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

\[ \text{cost} = \#\text{Iteration} \cdot \#\text{Flops per iteration} \]

\[ = O(N) \cdot O(N) \]

\[ = O(N^2) \]

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

**GS:** Assuming variables updated in Red-Black order, \( p(R_{\text{res}}) = (p(R_0))^2 \)

\[ \Rightarrow 1 \text{ step of GS same as } 2 \text{ steps of Jacobi} \]

only constant factor faster

**SOR(\( w \)):** Again, with Red-Black ordering, and optimal \( w \), much faster

\[ p(R_{\text{res}}(w)) \approx 1 - \frac{2\pi}{n} \]

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

\[ \Rightarrow \text{cost} = O(N^{3/2}) \text{ to converge} \]
Next: Present (and prove some of) the general theory of convergence for Jacobi, GS, SOR (ω). More details for 2D Poisson

Thm 1: If A is 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. 

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

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

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

$\|R\|_\infty = \max_i \| R_{(j,i)} \|_\infty \leq \| R_{(j,i)} \|_\infty$  

$= \| 1 - \frac{A_{(j,j)}}{A_{(j,j)}} \|_\infty + \sum_{i \neq j} \| \frac{|A_{(j,j)}|}{A_{(j,j)}} \|_\infty \leq \| A_{(j,j)} \|_\infty \leq 1$ by strict diag dominance  

$\Rightarrow \|R\|_\infty < 1 \Rightarrow$ convergence
2D Poisson: Not strictly row diagonally because many rows are \([-1, -1, 4, -1, -1]\).

**Def.** If \(1|A(j,j)| \leq \sum_{i \neq j}|A(j,i)|\) for all \(j\) with strict inequality at least once,

A weakly row diagonally dominant.

Not enough by itself for Jacobi to converge:

\[
A = \begin{bmatrix}
1 & -1 & 0 \\
0 & 5 & 0 \\
0 & 0 & 1
\end{bmatrix}
\Rightarrow R^n = \begin{bmatrix}
1 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix}
\Rightarrow R^5 = R \Rightarrow R^1\) does not converge.

Need one more property of \(A\) related to its sparsity structure.