Chap 6+7 : Iterative Methods
Part3: Splitting Methods

Given initial guess for $Ax=b$, x_0
cheaply compute $x_1, x_2, \dots \to x$

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

Iterative Method: $Ax = Mx - Kx = b$
$\Rightarrow Mx = Kx + b \Rightarrow$ solve $Mx_{i+1} = Kx_i + b$
for $x_{i+1} \Rightarrow x_{i+1} = \underline{M^{-1}K}x_i + \underline{\underline{M^{-1}b}} = \underline{R}\,x_i + \underline{\underline{c}}$

To work well need:
1. it should converge (quickly)
2. multiplying by R (solving $Mx_{i+1} = Kx_i + b$ for x_{i+1}) and computing $c = M^{-1}b$ should be much cheaper than solving with A

Recall operator norm: $\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|}$

Lemma: Let $\|\cdot\|$ be any operator norm:
Then if $\|R\| < 1$, then $x_{i+1} = Rx_i + c$ converges

for any x_0.

Proof:
$$\begin{array}{r} x_{i+1} = Rx_i + c \\ -\,[x = Rx + c] \\ \hline x_{i+1} - x = R(x_i - x) \end{array} \qquad e_i = x_i - x$$

$$e_{i+1} = Re_i = \cdots = R^{i+1} e_0$$

$$\|e_{i+1}\| \le \|R^{i+1}\| \cdot \|e_0\| \le \|R\|^{i+1} \cdot \|e_0\|$$

$\longrightarrow 0$ if $\|R\| < 1$ QED

Def The spectral radius of R is

$$\rho(R) = \max_i |\lambda_i(R)|$$

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

For all R, $\varepsilon > 0$, $\exists$ operator norm $\|\cdot\|_*$ such that $\|R\|_* \le \rho(R) + \varepsilon$

Proof: To show $\rho(R) \le \|R\|$, let $Rx = \lambda x$ where $|\lambda| = \rho(R)$

$$\|R\| \ge \frac{\|Rx\|}{\|x\|} = \frac{\|\lambda x\|}{\|x\|} = |\lambda| = \rho(R)$$

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

Let $S^{-1} R S = J$ be Jordan form

$$J = \begin{bmatrix} \square & & \\ & \square & \\ & & \square \end{bmatrix} \qquad \square = \begin{bmatrix} \lambda & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \lambda \end{bmatrix}$$

Let $D = \mathrm{diag}(1, \varepsilon, \varepsilon^2, \ldots, \varepsilon^{n-1})$

$J_\varepsilon = D^{-1} J \cdot D$: has same diagonal as J
each superdiagonal of J ($=1$ or 0)
is multiplied by ε

$$J_\varepsilon = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \varepsilon \\ & & \lambda_n \end{bmatrix}$$

Construct $\|\cdot\|_*$ as follows:

$\|x\|_* = \|(S\cdot D)^{-1} x\|_\infty$ = max entry in $|\cdot|$

(Recall Q 1.5 : shows $\|\cdot\|_*$ is a vector norm)

$$\|R\|_* = \max_{x\neq 0} \frac{\|Rx\|_*}{\|x\|_*}$$

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

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

$$= \max_{y\neq 0} \frac{\|(D^{-1} S^{-1} R S D) y\|_\infty}{\|y\|_\infty}$$

$$= \max_{y\neq 0} \frac{\|J_\varepsilon y\|_\infty}{\|y\|_\infty}$$

$$= \|J_\varepsilon\|_\infty \leq \max|\lambda| + \varepsilon$$

$$= \rho(R) + \varepsilon \qquad \text{QED}$$

Thm: $x_{i+1} = Rx_i + c$ converges to solution of $Ax=b$ for all x_0 if and only if $\rho(R) < 1$

proof: if $\rho(R) \geq 1 \Rightarrow$ choose x_0 so $x_0 - x$ is an eigenvector for A where $|\lambda| = \rho(R) \Rightarrow x_i - x = R^i(x_0 - x)$.
$= \lambda^i (x_0 - x)$
does not converge since $|\lambda| \geq 1$

if $\rho(R) < 1$ use Lemma to build and operator norm $\|R\|_A \leq \underline{\rho(R)} + \underline{\varepsilon} < 1$
$\Rightarrow \|x_c - x\|_A \leq \underline{\|R\|_A^i} \|x_0 - x\|_A \to 0$ QED

Goals: make $\rho(R)$ as small as possible
multiplication by R easy

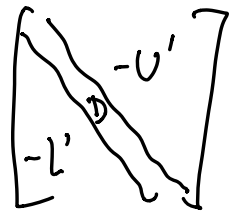
One choice: $M = I$ and $K = I - A$
$\Rightarrow R = M^{-1}K = I - A$, but $\rho(R)$ may not be small

Good news: another choice of diagonal M that works in many cases = Jacobi's method

Describe 3 basic Splitting Methods:
Jacobi, Gauss-Seidel (GS)
Successive Overrelaxation (SOR)

Notation: $A = D - L' - U'$

$= D(I - L - U)$



Jacobi:

In words: for $j = 1$ to n, pick $x_{i+1,j}$ to solve equation j

As a loop:

for $j = 1$ to n, $x_{i+1,j} = \dfrac{b_j - \sum_{k \neq j} A_{jk} x_{i,k}}{A_{jj}}$

As a splitting: $A = D - (L' + U')$

$= M - K$

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

For 2D Poisson $T_N V + V T_N = h^2 F$

□

to get from V_i to V_{i+1}:

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

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

= "average" of 4 nearest neighbors and $h^2 F$

Gauss-Seidel:

In words: Improve on Jacobi: by using most recently updated solution values

As a loop:

for $j = 1$ to $x_{i+1,j} = (b_j - \sum_{k<j} A_{jk} x_{i+1,k} - \sum_{k>j} A_{jk} x_{i,k}) / A_{jj}$

(updated x_{i+1}; older x_i)

As a splitting:

$$A = \underbrace{(D - L')}_{M} - \underbrace{U'}_{K} \text{ so}$$

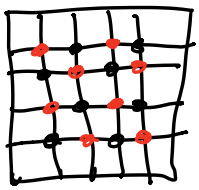
$$R_{GS} = (D - L')^{-1} U' = (D(I - D^{-1}L'))^{-1} U'$$

$$= (I - L)^{-1} U$$

Difference from Jacobi: answer depends on order of equations

For 2D Poisson: 2 orders of equations to consider:

"Natural order": rowwise or columnwise update $V(i,j)$



Red-Black Ordering:

Color the entries in 2D mesh like a checkerboard

Property: Red nodes only have Black neighbors, and vice versa

$\Rightarrow$ When updating Red nodes, I can update them in any order, Given all data at Black nodes (and vice versa)

for all (j,k) nodes that are Red $(j+k$ even$)$

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

— use old data (step i)

all be done in parallel

for all (j,k) nodes that are Black $(j+k$ odd$)$

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

— use new data (step $i+1$)

SOR:

In words: depends on parameter ω:

Let the result x_{i+1} be a weighted combination of x_i and what GS would have computed!

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

If $\omega = 1$, same as GS

If $\omega < 1$, would call this "underrelaxation"

If $\omega > 1$, overrelaxation.

Intuition: if moving x_i toward x^{GS}_{i+1} was a good idea, just go farther in same direction

- Question for later: how much farther, i.e. how big ω?

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

$$x_{i+1,j} = (1-\omega)x_{j,j} + \omega\cdot\Big(b_j - \sum_{k<j} A_{j,k}\, x_{i+1,k} - \sum_{k>j} A_{j,k}\, x_{i,k}\Big)/A_{jj}$$

As a Splitting: Multiply by A_{jj}

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

Divide by w to get

$$A = \left(\frac{1}{w}D - L'\right) - \left(\frac{1}{w}D - D + U'\right) = M - K$$

$$R_{SOR}(w) = M^{-1}K = \left(\frac{1}{w}D - L'\right)^{-1}\left(\frac{1}{w}D - D + U'\right)$$
$$= (I - wL)^{-1}((1-w)\cdot I + wU)$$

For 2D Poisson with Red Black Ordering:

for all Red (j,k) nodes

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

for all Black (j,k) nodes

$$V_{i+1}(j,k) = (1-w)V_i(j,k)$$
$$+w(V_{i+1}(j-1,k) + V_{i+1}(j+1,k)$$
$$+V_{i+1}(j,k-1) + V_{i+1}(j,k+1)$$
$$+h^2F(j,k))/4$$
$$\ldots \text{ new (Red) data}$$