Iterative Methods (Heath §11.5)

- Direct methods compute exact solution subject to rounding error.
- Can be too costly in storage and work for large systems.
- Iterative methods start with guess $x_0$ and improve until desired accuracy is achieved.

Several approaches based on fixed point iteration:

$$ M^{-1}x_{k+1} = Gx_k + c $$

Matrix $G$ can be obtained by splitting $A = M - N$

$$ Mx_{k+1} = Nx_k + b $$

$$ x_{k+1} = M^{-1}Nx_k + M^{-1}b $$

$$ G(x) = M^{-1}N $$

Convergent if $\rho(M^{-1}N) < 1$

$M$ is chosen so that it is easier to solve than $A$. 
Jacobi Method

\[ M = D \]
\[ N = -(L+U) \]

\[ D X^{(k+1)} = -(L+U)X^{(k)} + b \]
\[ X^{(k+1)} = D^{-1}(b - (L+U)X^{(k)}) \]

Equation for a single component \( i \):

\[ X_i^{(k+1)} = b_i - \sum_{j \neq i} a_{ij} X_j^{(k)} \]

\[ \frac{1}{a_{ii}} \]

- Slow convergence usually
- Easy to parallelize

Gauss-Seidel

- Use updated values of \( X_j^{(k+1)} \) after they've been computed:

\[ X_i^{(k+1)} = \frac{1}{a_{ii}} \left[ b_i - \sum_{j<i} a_{ij} X_j^{(k+1)} - \sum_{j>i} a_{ij} X_j^{(k)} \right] \]

- Equivalent to

\[(D+L)X^{(k+1)} = -U X^{(k)} + b\]
Gauss-Seidel (cont.)

- faster convergence than Jacobi
- sequential evaluation of $x_i^{(k+1)}$
- in-place update of $x_i$
Preconditioning

- improve condition # of a matrix

\[ M \approx A, \text{ easier to solve } M \]

replace
\[ Ax = b \]
with
\[ M^{-1}Ax = M^{-1}b \]

Note that "M^{-1}" does not necessarily mean we compute the inverse of M.

Intuitively: stretch quadratic form to make it more spherical.
\[ M = EE^T \implies M^{-1} = E^{-T}E^{-1} \]

\[ M^{-1}Ax = M^{-1}b \]

\[ E^{-T}E^T Ax = E^{-T}E^{-1}b \]

\[ \implies (E^{-T}A E^{-T}) (E^T x) = E^{-1}b \]

\[ \hat{A} \hat{x} = \hat{b} \]

Applying CG to the transformed equation, get terms with \( E^{-1} \). These can be eliminated to get the following PCG algorithm:

\[ r_0 = b - Ax_0 \]

\[ s_0 = M^{-1}r_0 \]

for \( k = 0, 1, 2, \ldots \)

\[ \alpha_k = \frac{r_k^T M^{-1}r_k}{s_k^T A s_k} \]

\[ x_{k+1} = x_k + \alpha_k s_k \]

\[ r_{k+1} = r_k - \alpha_k A s_k \]

\[ \beta_{k+1} = \frac{r_{k+1}^T M^{-1} r_{k+1}}{r_k^T M^{-1} r_k} \]

\[ s_{k+1} = M^{-1} r_{k+1} + \beta_{k+1} s_k \]