Numerical Methods I: Iterative solvers for

\[ Ax = b \]

Georg Stadler  
Courant Institute, NYU  
stadler@cims.nyu.edu

December 7, 2017
Target problems: very large \((n = 10^5, 10^6, \ldots)\), \(A\) is usually sparse and has specific properties.
To solve
\[ Ax = b \]
we construct a sequence
\[ x_1, x_2, \ldots \]
of iterates that converges as fast as possible to the solution \(x\), where \(x_{k+1}\) can be computed from \(\{x_1, \ldots, x_k\}\) with as little cost as possible (e.g., one matrix-vector multiplication).
Iterative solution of (symmetric) linear systems

Let $Q$ be invertible, then

\begin{equation}
Ax = b \iff Q^{-1}(b - Ax) = 0
\end{equation}

\begin{equation}
\iff (I - Q^{-1}A)x + Q^{-1}b = x
\end{equation}

\begin{equation}
\iff Gx + c = x
\end{equation}

Fixed point method:

\begin{equation}
x_{k+1} = Gx_k + c,
\end{equation}

$k = 0, 1, 2, \ldots$

$x_0 \in \mathbb{R}^n$ initialization
Iterative solution of (symmetric) linear systems

**Theorem:** The fixed point method $x_{k+1} = Gx_k + c$ with an invertible $G$ converges for each starting point $x_0$ if and only if

$$\rho(G) < 1,$$

where $\rho(G)$ is the largest eigenvalue of $G$ (i.e., the spectral radius).

**Proof:** $G$ spd., $\exists \Omega$ orthogonal s.t. $G = \Omega \Lambda \Omega^T$

$$QGQ^T = \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$$

Since $|\lambda_i| = \rho(G) < 1 \implies \Lambda^k \to 0$ as $k \to \infty$

$$\implies (Q\Lambda Q^T)^k = G^k = Q\Lambda^k Q^T \to 0 \text{ as } k \to \infty$$

Since $\rho(G) \leq \|G\|$ for any matrix norm induced by vector norm.

$$x_k - x^* = Gx_{k-1} + c - (Gx^* + c) = G(x^* - x^*) = \cdots G^k(x^* - x^*)$$

$$\implies \|x_k - x^*\| \leq \left\| G^k \right\| \|x_0 - x^*\| \to 0 \text{ if } \|G\| < 1.$$
Iterative solution of (symmetric) linear systems

Choices for $Q$:

- Choose $Q = I$...

  Richardson method

\[ G = I - Q^T A = I - A, \quad x_{k+1} = x_k - A x_k + b \]

As such, $\varrho(G) = \varrho(I - A) = \max \left\{ \left| 1 - \lambda_{\min}(A) \right|, \left| 1 - \lambda_{\max}(A) \right| \right\}$

\[ \Rightarrow \varrho(G) < 1 \quad \text{iff} \quad \lambda_{\max}(A) \leq 2 \]

For more choices, consider $A = L + D + U$, where $D$ is diagonal, $L$ and $U$ are lower and upper triangular with zero diagonal.
Iterative solution of (symmetric) linear systems

\[ A = L + D + U \]

- Choose \( Q = D \) ... Jacobi method

\[
G = I - Q'A = I - D'(L + D + U) = \\
\left[ \begin{array}{c} x_{k+1} = -(D'(L+U)x_k + D'b) \\
\end{array} \right] = -D'(L+U)
\]

**Theorem:** The Jacobi method converges for any starting point \( x_o \) to the solution of \( Ax = b \) if \( A \) is strictly diagonal dominant, i.e.,

\[ |a_{ii}| > \sum_{j \neq i} |a_{ij}|, \quad \text{for } i = 1, \ldots, n. \]

**Proof:** \( \varrho(G) = \varrho(-D'(L+R)) \leq \|D'(L+R)\|_\infty = \\
= \max_i \sum_{j \neq i} \frac{|a_{ij}|}{|a_{ii}|} \leq 1 \quad \text{since } A \text{ is strictly diagonal dom.} \)
Iterative solution of (symmetric) linear systems

\[ A = L + D + U \]

- Choose \( Q = D + L \ldots \) Gauss-Seidel method

\[ G = I - Q'A = I - (D+L)'A = I - (D+L)'(L+D+U) \]

\[ x_{k+1} = (D+L)'U x_k + (D+L)'b \]

**Theorem**: The Gauss-Seidel method converges for any starting point \( x_0 \) if \( A \) is spd.
Iterative solution of (symmetric) linear systems

Relaxation methods: Use linear combination between new and previous iterate:

$$G_\omega = \omega G + (1-\omega) I$$

$$x_{k+1} = \omega (G x_k + c) + (1 - \omega) x_k = G_\omega x_k + \omega c,$$

where $\omega \in [0, 1]$ is a damping/relaxation parameter (sometimes, $\omega > 1$ is used, leading to overrelaxation). Target is to choose $\omega$ such that $\rho(G_\omega)$ is as small as possible.
Def: A fixed point method $x_{k+1} = Gx_k + c$ with $G = G(A)$ is called *symmetrizable* if for any spd matrix $A$, $I - G$ is similar to an spd matrix. That is, \( \exists W \in \mathbb{R}^{n \times n} \) invertible such that $W(I - G)W^{-1}$ is spd.

**Examples:** Richardson: \( G = I - A \), so \( I - G = = A \) is spd \( \left[W = I \right] \)

Jacobi: \( G = I - D^{-\frac{1}{2}}DA \), \( W = D^{-\frac{1}{2}} \)

\[
D^{-\frac{1}{2}}(I - G)D^{-\frac{1}{2}} = \tilde{D}^{-\frac{1}{2}}(I - I + DA)\tilde{D}^{-\frac{1}{2}} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}} \text{ spd if } A \text{ spd.} 
\]
Iterative solution of (symmetric) linear systems

Let the fixed point method be symmetrizable, and $A$ an spd matrix. Then all eigenvalues of $G$ are real and less than 1.

Proof: method is symmetrizable $\implies I-G$ is similar to an spd matrix $\implies$ eigenvalues of $I-G$ are real and positive $\implies$ eigenvalues of $G$ are real and $< 1$. 
Iterative solution of (symmetric) linear systems

Finding the optimal damping parameter:

Symmetric method, $\lambda_{\min} \leq \lambda_{\max} < 1$ extreme eigenvalues of $G$.

Eigenvalues of $G_w = wG + (1-w)I$:

$\lambda_i(G_w) = w\lambda_i(G) + (1-w) = 1 - w(1 - \lambda_i(G)) < 1$

$g(G_w) = \max \{ |1 - w(1 - \lambda_{\min}(G))|, \ |1 - w(1 - \lambda_{\max}(G))| \}$

$w^*$ satisfies:

$-w^*(1 - \lambda_{\max}(G)) = -1 + w^*(1 - \lambda_{\min}(G))$

$\Rightarrow 2 = w^*(2 - \lambda_{\max}(G) - \lambda_{\min}(G))$
Iterative solution of (symmetric) linear systems

Krylov methods:
Idea: Build a basis for the Krylov subspace \( \{ r_0, Ar_0, A^2r_0 \ldots \} \) and reduce residual optimally in that space.

- spd matrices: Conjugate gradient (CG) method
- symmetric matrices: Minimal residual method (MINRES)
- general matrices: Generalized residual method (GMRES), BiCG, BiCGSTAB
Iterative solution of (symmetric) linear systems

Krylov methods:
Idea: Build a basis for the Krylov subspace \( \{r_0, Ar_0, A^2r_0 \ldots \} \) and reduce residual optimally in that space.

- spd matrices: Conjugate gradient (CG) method
- symmetric matrices: Minimal residual method (MINRES)
- general matrices: Generalized residual method (GMRES), BiCG, BiCGSTAB

Properties:
Do not require eigenvalue estimates; require usually one matrix-vector multiplication per iteration; convergence depends on eigenvalue structure of matrix (clustering of eigenvalues aids convergence). Availability of a good preconditioner is often important. Some methods require storage of iteration vectors.
The conjugate gradient method (CG, pcg in Matlab)

Solve \( Ax = b \), \( A \) spd, \( x, b \in \mathbb{R}^n \),
\( A \in \mathbb{R}^{n \times n} \) unique solution.

Solving \( Ax = b \) \( \iff \) \( \min_{x \in \mathbb{R}^n} \frac{1}{2} x^T Ax - b^T x \)

A-weighted norm
\[ \|y\|_A = \sqrt{(y^T A y)} \] is a norm

\[ x = A^{-1} b \] Search approximations \( x_k \in \mathbb{R}^n \) of \( x \)
in any subspace \( V_k = x_k + U_k \),
\( U_k \subset \mathbb{R}^n \) subspace

\[ x_k = \arg \min_{y \in V_k} \|y - x\|_A \]

\[ (x - x_k, u)_A = (A(x - x_k), u) = 0 \]

Why choose A-weighted norm? \( \Rightarrow \)

\( r_k = b - Ax_k \) residual
\[ (x - x_k, u)_A = (A(x - x_k), u) = (r_k, u) \]

i.e: residual \( L u \) in Euclidean inner product

\[ \text{does not include } x, \text{ which is not known.} \]
And cause $x - x_k \perp u$ in $A$-weighted inner product.

Let $P_1, \ldots, P_k$ $A$-orthogonal basis in $V_k$, i.e.

$$(P_i, P_j)_A = \delta_{ij} \quad A$\text{-orthogonal}\quad \text{also called } A\text{-conjugate}$$

$$x_k = P_k x = x_0 + \sum_{j=1}^{k} \frac{(P_j, x - x_0)_A}{(P_j, P_j)_A} P_j =$$

$$= x_0 + \sum_{j=1}^{k} \frac{(P_j, A(x - x_0))}{(P_j, P_j)_A} P_j = x_0 + \sum_{j=1}^{k} \frac{P_j (r_0)}{(P_j, P_j)_A} P_j \quad \text{indep. of } x$$

$$X_k = X_{k-1} + \alpha_k P_k$$

$$r_k = A(x - x_k) = A(x - x_{k-1} - \alpha_k P_k) = r_{k-1} - A\alpha_k P_k$$

Specific choices:

$V_k = x_0 + U_k$

$U_k = \text{span} \{ r_0, A r_0, A^2 r_0, \ldots, A^{k-1} r_0 \}$

Krylov Space

$$P_{k+1} = r_k - \frac{(r_k, P_k)}{(P_k, P_k)_A} P_k$$
We have:

- Method to choose \( v_k \), and compute \( A^T b - \text{rhs} \),
- Method to compute \( x_k \) that does not need \( x \),
- Recurrence for \( r_k \)

**Algorithm:**

A spd / \( x_0 \) starting value \( \mathbf{p}_i = \mathbf{r}_0 = b - A x_0 \)

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

\[
\alpha_k = \frac{(r_{k-1}, r_{k-1})}{(P_k, A P_k)}
\]

\( X_k := X_{k-1} + \alpha_k P_k \)

if converged stop

\( r_k := r_{k-1} - \alpha_k A P_k \)

\( \beta_{k+1} := \frac{(r_k r_k)}{(r_{k-1}, r_{k-1})} \)

\( P_{k+1} := r_k + \beta_k P_k \)

end

- Converges after at most \( m \) iterations because

\[
X_k = \arg \min_{y \in V_k} \| x - y \|_A \quad \forall x \in V_k
\]

\( V_k = \mathbb{R}^n \) after \( m \) iterations.
Converge much faster depending on the eigenvalues of $A$, in particular if eigenvalue of $A$ are clustered, i.e.