AMS526: Numerical Analysis I
(Numerical Linear Algebra)
Lecture 24: Biorthogonalization Methods;
Other Krylov Subspace Methods; Preconditioners

Xiangmin Jiao
SUNY Stony Brook
Outline

1 Biorthogonalization Methods

2 Other Krylov Subspace Methods

3 Preconditioners
Tridiagonal Biorthogonalization

- Lanczos iteration may be viewed as *tridiagonal orthogonalization*. If it runs through $m$ iterations for $A \in \mathbb{C}^{m \times m}$, then $A = QTQ^T$
- If $A$ is not Hermitian, we have to either give up tridiagonal form as in reduction to Hessenberg form, or give up orthogonality
- Tridiagonal Biorthogonalization method constructs
  \[ A = VTV^{-1}, \]
  where $V$ is nonsingular but generally not unitary
- Its adjoint is
  \[ A^* = W^T^*W^{-1}, \]
  where $W = V^{-*} = (V^*)^{-1} = (V^{-1})^*$
- *Biorthogonalization* refers to the fact that $W^*V = V^*W = I$
Matrix Form of Tridiagonal Biorthogonalization

- Starting from *arbitrary* nonzero $v_1$ and $w_1$ with $v_1^*w_1 = 1$, it constructs

$$AV_k = V_{k+1} \tilde{T}_k$$
$$A^* W_k = W_{k+1} \tilde{S}_k$$
$$V_k^* W_k = W_k^* V_k = I$$

- Let $T_k$ and $S_k$ be $k \times k$ leading blocks of $\tilde{T}_k$ and $\tilde{S}_k$, respectively.

$$T_k = S_k^*$$

since $W_k^* AV_k = T_k$ and $V_k^* A^* W_k = S_k$

- Tridiagonal biorthogonalization is also known as *nonsymmetric* Lanczos iteration
Nonsymmetric Lanczos Iteration

Let \( \tilde{T}_k = \begin{bmatrix} \alpha_1 & \gamma_1 & & & & \beta_1 & \alpha_2 & \gamma_2 & & & \beta_2 & \alpha_3 & & & & \beta_{k-1} & \alpha_k \\ & & & & & & & & \beta_2 & \alpha_3 & \gamma_2 \\ & & & & & & & & & & \beta_{k-1} & \alpha_k \\ \end{bmatrix} \)

Note that \( \alpha_k = w_k^* A v_k \). Let \( \beta_0 = \gamma_0 = 0 \). Given \( v_k, w_k, \alpha_k, \gamma_{k-1}, \) and \( \beta_{k-1} \), we can determine \( \beta_k v_{k+1} \) and \( \bar{\gamma}_k w_{k+1} \) from

\[
Av_k = \gamma_{k-1} v_{k-1} + \alpha_k v_k + \beta_k v_{k+1} \\
A^* w_k = \bar{\beta}_{k-1} w_{k-1} + \bar{\alpha}_k w_k + \bar{\gamma}_k w_{k+1}
\]

Choose \( \beta_k \) and \( \gamma_k \) arbitrarily under constraint of \( w_{k+1}^* v_{k+1} = 1 \).
Properties of Nonsymmetric Lanczos Iteration

- Nonsymmetric Lanczos iteration enjoys three-term recurrence, instead of $k$-term recurrence of Arnoldi iteration, at twice of cost of symmetric Lanczos iteration.
- It essentially builds nonorthogonal bases \( \{v_1, v_2, \ldots, v_k\} \) and \( \{w_1, w_2, \ldots, w_k\} \) for Krylov subspaces \( \mathcal{K}_k(A, v_1) \) and \( \mathcal{K}_k(A^*, w_1) \), i.e.,
  \[
  \mathcal{K}_k(A, v_1) = \text{span}\{v_1, Av_1, \ldots, A^{k-1}v_1\} = \text{span}\{v_1, v_2, \ldots, v_k\}
  \]
  \[
  \mathcal{K}_k(A^*, w_1) = \text{span}\{w_1, A^*w_1, \ldots, (A^*)^{k-1}w_1\} = \text{span}\{w_1, w_2, \ldots, w_k\}
  \]
- Nonsymmetric Lanczos iteration may breakdown for some matrices if
  - \( v_k = 0 \) or \( w_k = 0 \), for which \( T \) is reducible, or
  - \( v_k \neq 0 \) and \( w_k \neq 0 \), but \( w_k^*v_k = 0 \) (this is more serious)
- Potential breakdowns pose robustness issues. This can be mitigated (but not fully resolved) by lookahead (QMR)
BiConjugate Gradient Methods

- In CG, it finds $x_k \in \mathcal{K}_k(A, v_1)$ such that
  \[ r_k \perp \mathcal{K}_k(A, v_1), \]
  where $r_k = b - Ax_k$
- In BiConjugate Gradient, a.k.a. BCG or BiCG, it finds $x_k \in \mathcal{K}_k(A, v_1)$ s.t.
  \[ r_k \perp \mathcal{K}_k(A^*, w_1) \]
- Similar to CG, let $x_k = V_k y_k$. Since $AV_k = V_{k+1} \tilde{T}_k$,
  \[ r_k = b - Ax_k = b - AV_k y_k = b - V_{k+1} \tilde{T}_k y_k \]
- Let $W_k^* r_k = 0$, then, $W_k^* b - W_k^* V_{k+1} \tilde{T}_k y_k = W_k^* b - T_k y_k = 0$
- At each step, we solve $k \times k$ linear system
  \[ T_k y_k = W_k^* b \]
  and then $x_k = V_k y_k$
Convergence of BiCG

- If $A$ is HPD and $v_1 = w_1 = b/\|b\|$, then BiCG is equivalent to CG
- If $A$ is Hermitian but indefinite, then BiCG is equivalent to SYMMLQ
- If $A$ is non-Hermitian, convergence of BiCG may be erratic and unpredictable, but it is more efficient than GMRES if it converges

![Convergence Plot](image)

Figure 39.2. Comparison of GMRES and BCG for the $500 \times 500$ matrix labeled $\tau = 0.01$ in Figure 38.1, but with the signs of the entries randomized.

- BiCGSTAB smoothes convergence by resorting to GMRES sometimes (Van der Vorst 1992)
Quasi-Minimal Residual Method

- QMR minimizes residual in different norm
- Similar to GMRES, let \( x_k = V_k y_k \). Since \( AV_k = V_{k+1} \tilde{T}_k \),
  \[
  r_k = b - Ax_k = b - AV_k y_k = b - V_{k+1} \tilde{T}_k y_k
  \]
  and \( W_{k+1}^* r_k = W_{k+1}^* b - \tilde{T}_k y_k \)
- At each step, we solve \((k + 1) \times k\) least squares problem
  \[
  \tilde{T}_k y_k \approx W_{k+1}^* b,
  \]
  which minimizes \( \| W_{k+1}^* r_k \| = \| r_k \| W_{k+1} \)
- QMR is slower than BiCG, but it converges more smoothly
- QMR was developed by Freund and Nachtigal in 1991
- Transpose-Free QMR (TFQMR) was developed by Freund in 1993
- QMR/TFQMR use lookahead to mitigate (not resolve) breakdowns
Outline

1. Biorthogonalization Methods
2. Other Krylov Subspace Methods
3. Preconditioners
CG on the Normal Equations (CGN)

- Solve $A^*Ax = A^*b$ using Conjugate Gradients
- Poor convergence due to squared condition number (i.e., $\kappa(A^*A) = \kappa(A)^2$)
- One advantage is that it applies least squares problems without modification

Conjugate Gradients Squared (CGS)

- Avoids multiplication by $A^*$ in BCG, sometimes twice as fast convergence as BCG
LSQR for Least Squares Problems

- It solves $Ax = b$, or minimizes $\|Ax - b\|^2$ or minimize $\|Ax - b\|^2 + \lambda^2 \|x\|^2$
- $A$ may be square or rectangular (over-determined or under-determined), and may have any rank
- The method is based on the Golub-Kahan bidiagonalization process. It is algebraically equivalent to applying MINRES to the normal equation $(A^T A + \lambda^2 I)x = A^T b$, but has better numerical properties, especially if $A$ is ill-conditioned
- LSQR reduces $\|r\|$ monotonically (where $r = b - Ax$ if $\lambda = 0$)
- References:
Outline

1 Biorthogonalization Methods

2 Other Krylov Subspace Methods

3 Preconditioners
Preconditioning

- **Motivation:** Convergence of iterative methods heavily depends on eigenvalues or singular values of equation.

- **Main idea of preconditioning:** To introduce a nonsingular matrix $M$ such that $M^{-1}A$ has better properties than $A$. Thereafter, solve

$$M^{-1}Ax = M^{-1}b,$$

which has the same solution as $Ax = b$.

- **Criteria of $M$**
  - “Good” approximation of $A$, depending on iterative solvers.
  - Ease of inversion.

- **Typically,** a precondition $M$ is *good* if $M^{-1}A$ is not too far from normal and its eigenvalues are clustered.
Left, Right, and Hermitian Preconditioners

- Left preconditioner: Left multiply $M^{-1}$ and solve $M^{-1}Ax = M^{-1}b$
- Right preconditioner: Right multiply $M^{-1}$ and solve $AM^{-1}y = b$ with $x = M^{-1}y$
- However, if $A$ is Hermitian, $M^{-1}A$ or $AM^{-1}$ breaks symmetry
- How to resolve this problem?
Left, Right, and Hermitian Preconditioners

- Left preconditioner: Left multiply $M^{-1}$ and solve $M^{-1}Ax = M^{-1}b$
- Right preconditioner: Right multiply $M^{-1}$ and solve $AM^{-1}y = b$ with $x = M^{-1}y$
- However, if $A$ is Hermitian, $M^{-1}A$ or $AM^{-1}$ breaks symmetry
- How to resolve this problem?

Suppose $M$ is Hermitian positive definite, with $M = CC^*$ for some $C$, then $Ax = b$ is equivalent to

$$\left[ C^{-1}AC^{-*} \right] (C^*x) = C^{-1}b,$$

where $C^{-1}AC^{-*}$ is Hermitian positive definite, and it is similar to $C^{-*}C^{-1}A = M^{-1}A$ and has same eigenvalues as $M^{-1}A$

- Example of $M = CC^*$ is Cholesky factorization $M = RR^*$, where $R$ is upper triangular
Preconditioned Conjugate Gradient

- When preconditioning a symmetric matrix, use SPD matrix $M$, and $M = RR^T$
- In practice, algorithm can be organized so that only $M^{-1}$ (instead of $R^{-1}$) appears

Algorithm: Preconditioned Conjugate Gradient Method

$$x_0 = 0, \ r_0 = b, \ p_0 = M^{-1}r_0, \ z_0 = p_0$$

for $n = 1, 2, 3, \ldots$

$$\alpha_n = (r_{n-1}^Tz_{n-1})/(p_{n-1}^TAp_{n-1})$$

$$x_n = x_{n-1} + \alpha_n p_{n-1}$$

$$r_n = r_{n-1} - \alpha_n Ap_{n-1}$$

$$z_n = M^{-1}r_n$$

$$\beta_n = (r_n^Tz_n)/(r_{n-1}^Tz_{n-1})$$

$$p_n = z_n + \beta_n p_{n-1}$$
Effective Preconditioners for CG

● SSOR Preconditioner
  ▶ Simpler form: use matrix splitting of form $A = L + D + L^T$ and take
  \[
  M = (D + L)D^{-1}(D + L)^T
  \]
  ▶ More generally, introduce SSOR relaxation parameter $\omega$, and take
  \[
  M = \frac{1}{2 - \omega}(\frac{1}{\omega}D + L)\left(\frac{1}{\omega}D\right)^{-1}(\frac{1}{\omega}D + L)^T.
  \]
  With optimal $\omega$, $\text{cond}(M^{-1}A) = O(\sqrt{\text{cond}(A)})$, but determining optimal $\omega$ is impractical

● Incomplete factorization
  ▶ If $A = LL^T$ were used as preconditioner, then $\text{cond}(M^{-1}A) = 1$, but impractical
  ▶ Instead, compute approximate factorization $A \approx \tilde{L}\tilde{L}^T$, which omit all fills or omit small fills and use $M = \tilde{L}\tilde{L}^T$ as preconditioner
Other Commonly Used Preconditioners

- **Jacobi preconditioning**: $M = \text{diag}(A)$. Very simple and cheap, might improve certain problems but usually insufficient.

- **Block-Jacobi preconditioning**: Let $M$ be composed of block-diagonal instead of diagonal.

- **Multigrid (coarse-grid approximations)**: For a PDE discretized on a grid, a preconditioner can be formed by transferring the solution to a coarser grid, solving a smaller problem, then transferring back. This is sometimes the most efficient approach if applicable.