LARGE SPARSE EIGENVALUE PROBLEMS

- Projection methods
- The subspace iteration
- Krylov subspace methods: Arnoldi and Lanczos
- Golub-Kahan-Lanczos bidiagonalization

General Tools for Solving Large Eigen-Problems

- Projection techniques – Arnoldi, Lanczos, Subspace Iteration;
- Preconditionings: shift-and-invert, Polynomials, ...
- Deflation and restarting techniques
- Computational codes often combine these three ingredients

A few popular solution Methods

- Subspace Iteration [Now less popular – sometimes used for validation]
- Arnoldi’s method (or Lanczos) with polynomial acceleration
- Shift-and-invert and other preconditioners. [Use Arnoldi or Lanczos for $(A - \sigma I)^{-1}$.]
- Davidson’s method and variants, Jacobi-Davidson
- Specialized method: Automatic Multilevel Substructuring (AMLS).
Projection Methods for Eigenvalue Problems

Projection method onto $K$ orthogonal to $L$

- Given: Two subspaces $K$ and $L$ of same dimension.
- Approximate eigenpairs $\tilde{\lambda}, \tilde{u}$, obtained by solving:
  
  Find: $\tilde{\lambda} \in \mathbb{C}, \tilde{u} \in K$ such that $(\tilde{\lambda}I - A)\tilde{u} \perp L$

- Two types of methods:
  
  Orthogonal projection methods: Situation when $L = K$.
  
  Oblique projection methods: When $L \neq K$.

  First situation leads to Rayleigh-Ritz procedure

Rayleigh-Ritz projection

Given: a subspace $X$ known to contain good approximations to eigenvectors of $A$.

Question: How to extract 'best' approximations to eigenvalues/eigenvectors from this subspace?

**Answer:** Orthogonal projection method

- Let $Q = [q_1, \ldots, q_m]$ = orthonormal basis of $X$
- Orthogonal projection method onto $X$ yields:
  
  $Q^H(A - \tilde{\lambda}I)\tilde{u} = 0 \rightarrow$

  $Q^H AQy = \tilde{\lambda}y$ where $\tilde{u} = Qy$

  Known as Rayleigh Ritz process

Subspace Iteration

**Original idea:** projection technique onto a subspace of the form $Y = A^kX$

Practically: $A^k$ replaced by suitable polynomial

**Advantages:**
- Easy to implement (in symmetric case);
- Easy to analyze;

**Disadvantage:** Slow.

- Often used with polynomial acceleration: $A^kX$ replaced by $C_k(A)X$. Typically $C_k =$ Chebyshev polynomial.
Algorithm: Subspace Iteration with Projection

1. Start: Choose an initial system of vectors $X = [x_0, \ldots, x_m]$ and an initial polynomial $C_k$.

2. Iterate: Until convergence do:
   (a) Compute $\hat{Z} = C_k(A)X$. [Simplest case: $\hat{Z} = AX$.]
   (b) Orthonormalize $\hat{Z}$: $[Z, R_Z] = qr(\hat{Z}, 0)$
   (c) Compute $B = Z^H A Z$
   (d) Compute the Schur factorization $B = Y R_B Y^H$ of $B$
   (e) Compute $X := ZY$.
   (f) Test for convergence. If satisfied stop. Else select a new polynomial $C'_k$ and continue.

THEOREM: Let $S_0 = \text{span}\{x_1, x_2, \ldots, x_m\}$ and assume that $S_0$ is such that the vectors $\{P x_i\}_{i=1}^m$ are linearly independent where $P$ is the spectral projector associated with $\lambda_1, \ldots, \lambda_m$. Let $P_k$ the orthogonal projector onto the subspace $S_k = \text{span}\{X_k\}$. Then for each eigenvector $u_i$ of $A$, $i = 1, \ldots, m$, there exists a unique vector $s_i$ in the subspace $S_0$ such that $Ps_i = u_i$. Moreover, the following inequality is satisfied

$$\|(I - P_k)u_i\|_2 \leq \|u_i - s_i\|_2 \left(\frac{\lambda_{m+1}}{\lambda_i} + \epsilon_k\right)^k, \quad (1)$$

where $\epsilon_k$ tends to zero as $k$ tends to infinity.

Krylov subspace methods

Principle: Projection methods on Krylov subspaces:

$$K_m(A, v_1) = \text{span}\{v_1, Av_1, \ldots, A^{m-1}v_1\}$$

- The most important class of projection methods [for linear systems and for eigenvalue problems]
- Variants depend on the subspace $L$

Let $\mu = \text{deg. of minimal polynom. of } v_1$. Then:

- $K_m = \{p(A)v_1|p = \text{polynomial of degree } \leq m - 1\}$
- $K_m = K_\mu$ for all $m \geq \mu$. Moreover, $K_\mu$ is invariant under $A$.
- $\text{dim}(K_m) = m$ iff $\mu \geq m$. 
Arnoldi’s algorithm

- Goal: to compute an orthogonal basis of $K_m$.
- Input: Initial vector $v_1$, with $\|v_1\|_2 = 1$ and $m$.

**ALGORITHM : 1. Arnoldi’s procedure**

For $j = 1, \ldots, m$ do

1. Compute $w := Av_j$
2. For $i = 1, \ldots, j$, do
   - $h_{i,j} := (w, v_i)$
   - $w := w - h_{i,j}v_i$
3. $h_{j+1,j} = \|w\|_2$;
4. $v_{j+1} = w/h_{j+1,j}$

End

- Based on Gram-Schmidt procedure

Result of Arnoldi’s algorithm

Let: $H_m = \begin{pmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{pmatrix}$, $H_m = \begin{pmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{pmatrix}$

1. $V_m = [v_1, v_2, \ldots, v_m]$ orthonormal basis of $K_m$.
2. $AV_m = V_{m+1}H_m = V_mH_m + h_{m+1,m}v_{m+1}e_T^m$
3. $V_m^TAV_m = H_m \equiv H_m - $ last row.

Application to eigenvalue problems

- Write approximate eigenvector as $\tilde{u} = V_my$
- Galerkin condition:
  
  $$(A - \tilde{\lambda}I)V_my \perp K_m \rightarrow V_m^H(A - \tilde{\lambda}I)V_my = 0$$

- Approximate eigenvalues are eigenvalues of $H_m$
  
  $H_my_j = \tilde{\lambda}_jy_j$

- Associated approximate eigenvectors are
  
  $\tilde{u}_j = V_my_j$

- Typically a few of the outermost eigenvalues will converge first.

Hermitian case: The Lanczos Algorithm

- The Hessenberg matrix becomes tridiagonal:
  
  $A = A^H$ and $V_m^HAV_m = H_m \rightarrow H_m = H_m^H$

- Denote $H_m$ by $T_m$ and $\bar{H}_m$ by $\bar{T}_m$. We can write
  
  $$T_m = \begin{pmatrix} \alpha_1 & \beta_2 \\ \beta_2 & \alpha_2 & \beta_3 \\ \beta_3 & \alpha_3 & \beta_4 \\ \vdots & \vdots & \ddots \end{pmatrix}$$

- Relation $AV_m = V_{m+1}\bar{T}_m$
Consequence: three term recurrence

\[ \beta_{j+1} v_{j+1} = A v_j - \alpha_j v_j - \beta_j v_{j-1} \]

**ALGORITHM : 2. Lanczos**

1. Choose an initial \( v_1 \) with \( \| v_1 \|_2 = 1 \); Set \( \beta_1 \equiv 0, v_0 \equiv 0 \)
2. For \( j = 1, 2, \ldots, m \) Do:
   3. \( w_j := A v_j - \beta_j v_{j-1} \)
   4. \( \alpha_j := (w_j, v_j) \)
   5. \( v_j := w_j - \alpha_j v_j \)
   6. \( \beta_{j+1} := \| w_j \|_2 \). If \( \beta_{j+1} = 0 \) then Stop
   7. \( v_{j+1} := w_j / \beta_{j+1} \)
8. EndDo

Hermitian matrix + Arnoldi → Hermitian Lanczos

**Reorthogonalization**

- Full reorthogonalization – reorthogonalize \( v_{j+1} \) against all previous \( v_j \)'s every time.
- Partial reorthogonalization – reorthogonalize \( v_{j+1} \) against all previous \( v_j \)'s only when needed [Parlett & Simon]
- Selective reorthogonalization – reorthogonalize \( v_{j+1} \) against computed eigenvectors [Parlett & Scott]
- No reorthogonalization – Do not reorthogonalize - but take measures to deal with 'spurious' eigenvalues. [Cullum & Willoughby]

**Lanczos Bidiagonalization**

We now deal with rectangular matrices. Let \( A \in \mathbb{R}^{m \times n} \).

**ALGORITHM : 3. Golub-Kahan-Lanczos**

1. Choose an initial \( v_1 \) with \( \| v_1 \|_2 = 1 \); Set \( \beta_0 \equiv 0, u_0 \equiv 0 \)
2. For \( k = 1, \ldots, p \) Do:
   3. \( \tilde{u} := A v_k - \beta_{k-1} u_{k-1} \)
   4. \( \alpha_k = \| \tilde{u} \|_2 \); \( u_k = \tilde{u} / \alpha_k \)
   5. \( \tilde{v} = A^T u_k - \alpha_k v_k \)
   6. \( \beta_k = \| \tilde{v} \|_2 \); \( v_{k+1} := \tilde{v} / \beta_k \)
7. EndDo

Let:

\[ V_{p+1} = [v_1, v_2, \ldots, v_{p+1}] \in \mathbb{R}^{n \times (p+1)} \]
\[ U_p = [u_1, u_2, \ldots, u_p] \in \mathbb{R}^{m \times p} \]
Let:

\[
B_p = \begin{bmatrix}
\alpha_1 & \beta_1 \\
\alpha_2 & \beta_2 \\
\vdots & \vdots \\
\alpha_p & \beta_p \\
\end{bmatrix};
\]

\[
\hat{B}_p = B_p(:,1 : p)
\]

\[
V_p = [v_1, v_2, \cdots , v_p] \in \mathbb{R}^{n \times p}
\]

\[
V_{p+1}^T V_{p+1} = I
\]

\[
U_p^T U_p = I
\]

\[
AV_p = U_p \hat{B}_p
\]

\[
A^T U_p = V_{p+1} B_p^T
\]

Result:

\[
\text{Observe that : } A^T(AV_p) = A^T(U_p \hat{B}_p)
\]

\[
= V_{p+1} B_p^T \hat{B}_p
\]

\[
B_p^T \hat{B}_p \text{ is a (symmetric) tridiagonal matrix of size } (p + 1) \times p
\]

\[
\text{Call this matrix } T_k. \text{ Then: } (A^T A) V_p = V_{p+1} T_p
\]

\[
\text{Standard Lanczos relation!}
\]

\[
\text{Algorithm is equivalent to standard Lanczos applied to } A^T A.
\]

\[
\text{Similar result for the } u_i \text{'s [involves } AA^T]\]

\[
\text{Work out the details: What are the entries of } \bar{T}_p \text{ relative to those of } B_p?
\]