ECS231 Handout

Subspace projection methods for Solving Large-Scale Eigenvalue Problems

Part I: Review of basic theory of eigenvalue problems

1. Let $A \in \mathbb{C}^{n\times n}$.

   (a) A scalar λ is an *eigenvalue* of an $n \times n$ A and a nonzero vector $x \in \mathbb{C}^n$ is a corresponding *right eigenvector* if
   $$Ax = \lambda x.$$
   A nonzero vector y such that $y^H A = \lambda y^H$ is a *left eigenvector*.

   (b) $p_A(\lambda) \stackrel{\text{def}}{=} \det(\lambda I - A)$, a polynomial of degree n, is called *characteristic polynomial* of A.

   (c) The set $\lambda(A) \stackrel{\text{def}}{=} \{\lambda : p_A(\lambda) = 0\}$ is called the *spectrum* of A.

   (d) $\mathcal{L}_{A,\lambda} \stackrel{\text{def}}{=} \{x : Ax = \lambda x\}$ is an *eigenspace* of A corresponding to the eigenvalue λ.

2. The following is a list of basic properties straightforwardly from the definition

   (a) λ is A's eigenvalue $\Leftrightarrow \lambda I - A$ is singular $\Leftrightarrow \det(\lambda I - A) = 0 \Leftrightarrow p_A(\lambda) = 0$.

   (b) There is at least one eigenvector x associated with A's eigenvalue λ; in the other word, the dimension $\dim(\mathcal{L}_{A,\lambda}) \geq 1$.

   (c) $\mathcal{L}_{A,\lambda}$ is a subspace, i.e., it has the following two properties:
   (1) $x \in \mathcal{L}_{A,\lambda} \Rightarrow \alpha x \in \mathcal{L}_{A,\lambda}$ for all $\alpha \in \mathbb{C}$.
   (2) $x_1, x_2 \in \mathcal{L}_{A,\lambda} \Rightarrow x_1 + x_2 \in \mathcal{L}_{A,\lambda}$.

   (d) When $A \in \mathbb{R}^{n\times n}$, λ is A's eigenvalue $\Leftrightarrow$ conjugate $\bar{\lambda}$ is also A's eigenvalue.

   (e) A is singular $\Leftrightarrow 0$ is A's eigenvalue.

   (f) If A is upper (or lower) triangular, then its eigenvalues consist of its diagonal entries.

3. $A \in \mathbb{C}^{n\times n}$ is *simple* if it has n linearly independent eigenvectors; otherwise it is *defective*.

   Examples:

   (a) I and any diagonal matrices is simple. $e_1, e_2, \ldots, e_n$ *are* n linearly independent eigenvectors.

   (b) $\begin{pmatrix} 1 & 2 \\ 4 & 3 \end{pmatrix}$ is simple. It has two different eigenvalues -1 and 5. By the fact that each eigenvalue corresponds to at least one eigenvector, it must have 2 linearly independent eigenvectors.

   (c) If $A \in \mathbb{C}^{n\times n}$ has n different eigenvalues, then A is simple.

   (d) $\begin{pmatrix} 2 & 1 \\ 0 & 2 \end{pmatrix}$ is defective. It has two repeated eigenvalues 2, but only one eigenvector $e_1 = (1, 0)^T$.

4. Let $Ax_i = \lambda_i x_i$ for $i = 1, 2, \ldots k$, and $\lambda_i \neq \lambda_j$ for $i \neq j$. Then $x_1, x_2, \ldots, x_k$ are linearly independent (proof by induction)

5. Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the eigenvalues of A, and $x_1, x_2, \ldots, x_n$ be a set of corresponding eigenvectors, then

$$AX = X\Lambda$$

where $X = [x_1, x_2, \ldots, x_n]$ and $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$.

If A is *simple*, namely the eigenvectors are linearly independent, then X^{-1} exists and

$$A = X\Lambda X^{-1}.$$

This is known as the *eigenvalue decomposition* of the matrix A.

6. An *invariant subspace* of A is a subspace $\mathcal{V}$ of $\mathbb{C}^n$, with the property that $v \in \mathcal{V}$ implies that $Av \in \mathcal{V}$. We also write this as $A\mathcal{V} \subseteq \mathcal{V}$.

Examples:

(1) The simplest, one-dimensional invariant subspace is the set $\mathrm{span}(x)$ of all scalar multiples of an eigenvector x.

(2) Let $x_1, x_2, \ldots, x_m$ be any set of independent eigenvectors with eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_m$. Then $\mathcal{X} = \mathrm{span}(\{x_1, x_2, \ldots, x_m\})$ is an invariant subspace.

7. Let A be n-by-n, let $V = [v_1, v_2, \ldots, v_m]$ be any n-by-m matrix with linearly independent columns, and let $\mathcal{V} = \mathrm{span}(V)$, the m-dimensional space spanned by the columns of V. Then $\mathcal{V}$ is an invariant subspace if and only if there is an m-by-m matrix B such that

$$AV = VB.$$

In this case the m eigenvalues of B are also eigenvalues of A.

8. Similarity transformations: $n \times n$ matrices A and B are *similar* if there is an $n \times n$ nonsingular matrix P such that $B = P^{-1}AP$. We also say A is *similar* to B, and likewise B is similar to A; P is a *similarity transformation*. A is *unitarily similar* to B if P is unitary.

9. Suppose that A and B are similar: $B = P^{-1}AP$.

(a) A and B have the same eigenvalues. In fact $p_A(\lambda) \equiv p_B(\lambda)$.

(b) $Ax = \lambda x \Rightarrow B(P^{-1}x) = \lambda(P^{-1}x)$.

(c) $Bw = \lambda w \Rightarrow A(Pw) = \lambda(Pw)$.

10. Schur decomposition. Let A be of order n. Then there is an $n \times n$ unitary matrix U ($U^H U = I$) such that

$$A = UTU^H,$$

where T is upper triangular. By appropriate choice of U, the eigenvalues of A, which are the diagonal elements of T, may be made to appear in any order.

11. Real Schur Decomposition. If A is real, there is an orthogonal matrix Q such that

$$A = QTQ^T,$$

where T is block triangular with 1×1 and 2×2 blocks on its diagonal. The 1×1 blocks contain the real eigenvalues of A, and the eigenvalues of the 2×2 blocks are pairs of complex conjugate eigenvalues.

Part II. Basic algorithms

1. *The power method* is based on the following simple analysis:

   Assume that A is simple and has the eigenvalue decomposition

   $$A = X\Lambda X^{-1}$$

   with $X = [x_1, x_2, \ldots, x_n]$ and $\Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$, and assume that eigenvalues λ_j are ordered such that

   $$|\lambda_1| > |\lambda_2| \geq \ldots \geq |\lambda_n|.$$

   Let u_0 be a vector such that

   $$u_0 = \gamma_1 x_1 + \gamma_2 x_2 + \cdots + \gamma_n x_n \quad \text{and} \quad \gamma_1 \neq 0.$$

   Then by simple algebraic manipulation, we can show that

   (a) $u_j = \frac{A^j u_0}{\|A^j u_0\|} \to \pm \frac{x_1}{\|x_1\|}$ as $j \to \infty$.

   (b) $\theta_j = u_j^H A u_j \to \lambda_1$ as $j \to \infty$.

   (c) $\frac{|\lambda_2|}{|\lambda_1|}$ is the rate of convergence.

2. Pseudocode:

   Select an initial vector u_0
   for $j = 1, 2, \ldots$ until convergence
   $w = Au_{j-1}$
   $u_j = w/\|w\|_2$
   $\theta_j = u_j^H A u_j$

3. Example. Let

   $$A = \begin{bmatrix} -261 & 209 & -49 \\ -530 & 422 & -98 \\ -800 & 631 & -144 \end{bmatrix}$$

   Then $\lambda(A) = \{\lambda_1, \lambda_2, \lambda_3\} = \{10, 4, 3\}$. Let $u_0 = e_1$, by the power method, we have

i	1	2	3	$\cdots$	10
θ_i	994.49	13.0606	10.07191	$\cdots$	10.0002

4. The drawback of the power method is that if $\frac{|\lambda_2|}{|\lambda_1|}$ is close to 1, then the power method could be very slow convergent or doesn't converge at all.

5. The method of *Inverse iteration* has two purposes:

   (a) overcome the drawbacks of the power method (slow convergence).

   (b) find an eigenvalue closest to a particular given number σ, referred to as a *shift*).

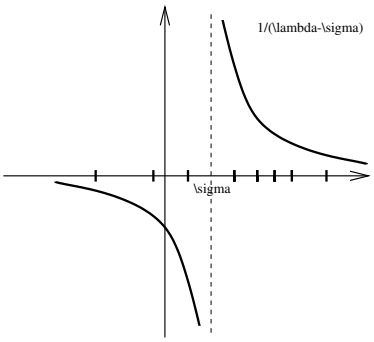
6. Spectral transformation: if λ is an eigenvalue of A, then

   (a) $\lambda - \sigma$ is an eigenvalue of $A - \sigma I$,

   (b) $\frac{1}{\lambda - \sigma}$ is an eigenvalue of $(A - \sigma I)^{-1}$.

   This is referred to as *shift-and-invert spectral transformation.*

The following plot illustrates the transformation of eigenvalues:



7. By applying the power method to the shift-and-invert eigenvalue problem

$$(A - \sigma)^{-1} x = \mu x,$$

we derive the follow algorithm, which is referred to as the *inverse iteration*:

Given an initial vector u_0 and a shift σ
for $j = 1, 2, \ldots$ until convergence
$w = (A - \sigma I)^{-1} u_{j-1}$
$u_j = \frac{w}{\|w\|_2}$ (approximate eigenvector)
$\mu_j = u_j^H A u_j$ (approximate eigenvalue)
end for
Return approximate eigenpair $(\theta_j, \sigma + \frac{1}{\mu_j})$

8. Assume λ_k is the eigenvalue cloest to the shift σ. It can be shown that

   (a) u_j converges to $x_k/\|x_k\|$, where $s_k = Se_k$ $j \to \infty$.

   (b) θ_j converges to λ_k as $j \to \infty$.

   (c) $\max_{j \neq k} \frac{|\lambda_k - \sigma|}{|\lambda_j - \sigma|}$ is the convergence rate.

9. The advantages of inverse iteration over the power method is the ability to converge to any desired eigenvalue (the one nearest to the shift σ). By choosing σ very close to a desired eigenvalue, the method converges very quickly and thus not be as limited by the proximity of nearby eigenvalues as is the power method. The method is particularly effective when we have a good approximation to an eigenvalue and want only its corresponding eigenvector.

   However, the inverse iteration is expensive in general. It requires solving $(A - \sigma I)w = j_j$ for u. One (sparse) LU factorization of $A - \sigma I$ is required, which could be very expensive in memory requirements.

Part III. Subspace projection method framework (Rayleigh-Ritz procedure)

1. *Rayleigh-Ritz procedure* is a framework of the orthogonal projection methods for solving large scale eigenvalue problems

   Let A be an $n \times n$ real matrix and $\mathcal{K}$ be an m-dimensional subspace of $\mathbb{R}^n$. An orthogonal projection technique seeks an approximate eigenpair

   $$(\tilde{\lambda}, \tilde{u}) \quad \text{with} \quad \tilde{\lambda} \in \mathbb{C} \text{ and } \tilde{u} \in \mathcal{K}.$$

   by imposing the following so-called Galerkin condition:

   $$A\tilde{u} - \tilde{\lambda}\tilde{u} \perp \mathcal{K} \ , \tag{1}$$

   or, equivalently,

   $$v^T(A\tilde{u} - \tilde{\lambda}\tilde{u}) = 0, \ \ \forall \ v \in \mathcal{K}. \tag{2}$$

2. To translate into a matrix problem, assume that an orthonormal basis $\{v_1, v_2, \ldots, v_m\}$ of $\mathcal{K}$ is available. Denote $V = [v_1, v_2, \ldots, v_m]$, and let $\tilde{u} \ = \ Vy$. Then, equation (2) becomes

   $$V^T(AVy - \tilde{\lambda}Vy) = 0$$

   Therefore, y and $\tilde{\lambda}$ must satisfy the following reduced eigenvalue problem:

   $$B_m y \ = \ \tilde{\lambda} y \tag{3}$$

   with $B_m \ = \ V^H AV$. The eigenvalues $\tilde{\lambda}_i$ of B_m are called *Ritz value* values, and the vectors Vy_i are called *Ritz vector*.

3. This procedure is known as the *Rayleigh-Ritz procedure*:

   (a) Compute an orthonormal basis $\{v_i\}_{i=1:m}$ of the subspace $\mathcal{K}$.
   (b) Compute $B_m = V^T AV$, where $V = [v_1, v_2, \ldots, v_m]$.
   (c) Compute the eigenvalues of B_m and select the k desired ones $\tilde{\lambda}_i, i = 1 : k$, where $k \leq m$.
   (d) Compute the eigenvectors y_i of B_m associated with $\tilde{\lambda}_i$.
   (e) return $(\lambda_i, \tilde{u}_i = Vy_i)$ as approximate eigenvectors of A.

   The numerical solution of the $m \times m$ eigenvalue problem in steps (c) and (d) can be treated by standard algorithms for solving small dense eigenvalue problems. An important note is that in step (d) one can replace eigenvectors by Schur vectors to get approximate Schur vectors $\tilde{u}_i$ instead of approximate eigenvectors. Schur vectors y_i can be obtained in a numerically stable way and, in general, eigenvectors are more sensitive to rounding errors than are Schur vectors.

4. Further reading

   Here we provide a justification that the Ritz values and Ritz vectors are *optimal* approximations of eigenvalues and eigenvectors of a symmetrix matrix.

   Consider the case where A is real and symmetric. Let $Q = [Q_k, Q_u]$ be any n-by-n orthogonal matrix, where Q_k is n-by-k, and Q_u is n-by-$(n-k)$. In practice, the column of Q_k will be computed by the Lanczos algorithm and span a Krylov subspace (to be discussed in Part IV of this handout). But for now, we do not care where we get Q. Let

   $$T = Q^T AQ = [Q_k, Q_u]^T A[Q_k, Q_u] = \begin{bmatrix} Q_k^T AQ_k & Q_k^T AQ_u \\ Q_u^T AQ_k & Q_u^T AQ_u \end{bmatrix} \equiv \begin{bmatrix} T_k & T_{uk} \\ T_{ku} & T_u \end{bmatrix}$$

When $k = 1$, T_k is just called the Rayleigh quotient. So far $k > 1$, T_k is called a generalization of the Rayleigh quotient.

The *Rayleigh-Ritz procedure* is to approximate the eigenvalues of A by the eigenvalues of $T_k = Q_k^T A Q_k$. The Ritz values and Ritz vectors are considered *optimal* approximations to the eigenvalues and eigenvectors of A as justified by the following theorem.

Theorem. The minimum of $\|AQ_k - Q_k R\|_2$ over all k-by-k symmetric matrices R is attained by $R = T_k$, in which case, $\|AQ_k - Q_k T_k\|_2 = \|T_{ku}\|_2$.

Proof. Let $R = T_k + Z$, to proof the theorem, we just want to show that $\|AQ_k - Q_k R\|_2$ is minimized when $Z = 0$. This is shown by the following sequence of derivation:

$$\begin{aligned}
\|AQ_k - Q_k R\|_2^2 &= \lambda_{\max}\left[(AQ_k - Q_k R)^T (AQ_k - Q_k R)\right] \\
&= \lambda_{\max}\left[(AQ_k - Q_k (T_k + Z))^T (AQ_k - Q_k (T_k + Z))\right] \\
&= \lambda_{\max}\left[(AQ_k - Q_k T_k)^T (AQ_k - Q_k T_k) - ((AQ_k - Q_k T_k)^T (Q_k Z)\right. \\
&\qquad \left. -(Q_k Z)^T (AQ_k - Q_k T_k) + (Q_k Z)^T (Q_k Z)\right] \\
&= \lambda_{\max}\left[(AQ_k - Q_k T_k)^T (AQ_k - Q_k T_k) - (Q_k^T AQ_k - T_k) Z\right. \\
&\qquad \left. -Z^T (Q_k^T AQ_k - T_k) + Z^T Z\right] \\
&= \lambda_{\max}\left[(AQ_k - Q_k T_k)^T (AQ_k - Q_k T_k) + Z^T Z\right] \\
&\geq \lambda_{\max}\left[(AQ_k - Q_k T_k)^T (AQ_k - Q_k T_k)\right] \\
&= \|AQ_k - Q_k T_k\|_2^2
\end{aligned}$$

Furthermore, it is easy to compute the minimum value

$$\|AQ_k - Q_k T_k\|_2 = \|(Q_k T_k + Q_u T_{ku}) - Q_k T_k\|_2 = \|Q_u T_{ku}\|_2 = \|T_{ku}\|_2.$$

■

Corollary. Let $T_k = Y \Lambda Y^T$ be the eigendecomposition of T_k. The minimum of $\|AP_k - P_k D\|$ over all n-by-k orthogonal matrices P_k where $\text{span}(P_k) = \text{span}(Q_k)$ and over all diagonal D is also $\|T_{ku}\|_2$ and is attained by $P_k = Q_k Y$ and $D = \Lambda$.

Proof. If we replace Q_k with $Q_k U$ in the above proof, where U is another orthogonal matrix, then the columns of Q_k and $Q_k U$ span the same space, and

$$\|AQ_k - Q_k R\|_2 = \|AQ_k U - Q_k R U\|_2 = \|A(Q_k U) - (Q_k U)(U^T R U)\|_2.$$

These quantities are still minimized when $R = T_k$, and by choosing $U = Y$ so that $U^T T_k U$ is diagonal. ■

Part IV. Symmetric Lanczos algorithm

1. *The symmetric Lanczos algorithm* combines the Lanczos process for building a Krylov subspace with the Raleigh-Ritz procedure for finding a few eigenpairs of a symmetric matrix A.

   Let us recall that the Lanczos process will generate an orthonormal basis of a Krylov subspace:

   $$\mathcal{K}_k(A, v) \stackrel{\text{def}}{=} \text{span}\{v, Av, \ldots, A^{k-1}v\} = \text{span}\{q_1, q_2, \ldots, q_k\},$$

   and yield a fundamental relation

   $$AQ_k = Q_k T_k + \beta_k q_{k+1} e_k^T, \tag{4}$$

   where

   $$T_k = Q_k^T A Q_k = \text{tridiag}(\beta_j, \alpha_j, \beta_{j+1}).$$

   Let μ be an eigenvalue of T_k and y be a corresponding eigenvector y, i.e.,

   $$T_k y = \mu y, \quad \|y\|_2 = 1.$$

   Apply y to the right of (4) to get

   $$A(Q_k y) = Q_k T_k y + f_k(e_k^T y) = \mu(Q_k y) + \beta_k q_{k+1}(e_k^T y).$$

   The scalars $\{\mu\}$ are *Ritz values*, and $\{Q_k y\}$ are *Ritz vectors*.

2. Convergence

   - If $\beta_k q_{k+1}(e_k^T y) = 0$ for some k, then the associated Ritz value μ is an eigenvalue of A with the corresponding eigenvector $Q_k y$.
   - In general, $\beta_k q_{k+1}(e_k^T y) \neq 0$, but we hope that the residual norm $\|\beta_k q_{k+1}(e_k^T y)\|_2$ may be small; and when this happens we expect that μ is going to be a good approximate to A's eigenvalue. Indeed, we have

     Lemma 1. *Let H be real and symmetric, and $Hz - \mu z = r$ and $z \neq 0$. Then*

     $$\min_{\lambda \in \lambda(H)} |\lambda - \mu| \leq \|r\|_2 / \|z\|_2.$$

     Proof. Let $H = U \Lambda U^T$ be the eigenvalue decomposition of H. Then $Hz - \mu z = r$ yields

     $$(H - \mu I)z = r \quad \Rightarrow \quad U(\Lambda - \mu I)U^T z = r \quad \Rightarrow \quad (\Lambda - \mu I)(U^T z) = U^T r.$$

     Notice that $\Lambda - \mu I$ is diagonal. Thus

     $$\|r\|_2 = \|U^T r\|_2 = \|(\Lambda - \mu I)(U^T z)\|_2 \geq \min_{\lambda \in \lambda(H)} |\lambda - \mu| \, \|U^T z\|_2 = \min_{\lambda \in \lambda(H)} |\lambda - \mu| \|z\|_2,$$

     as expected. ∎

     The following corollary is a consequence of above Lemma 1.

     Corollary 1. *There is an eigenvalue λ of A such that*

     $$|\lambda - \mu| \leq \|\beta_k q_{k+1}(e_k^T y)\|_2 = |\beta_k| \cdot |e_k^T y|.$$

3. In summary, we have the following Lanczos algorithm in the simplest form for finding eigenvalues and eigenvectors of a symmetric matrix A:

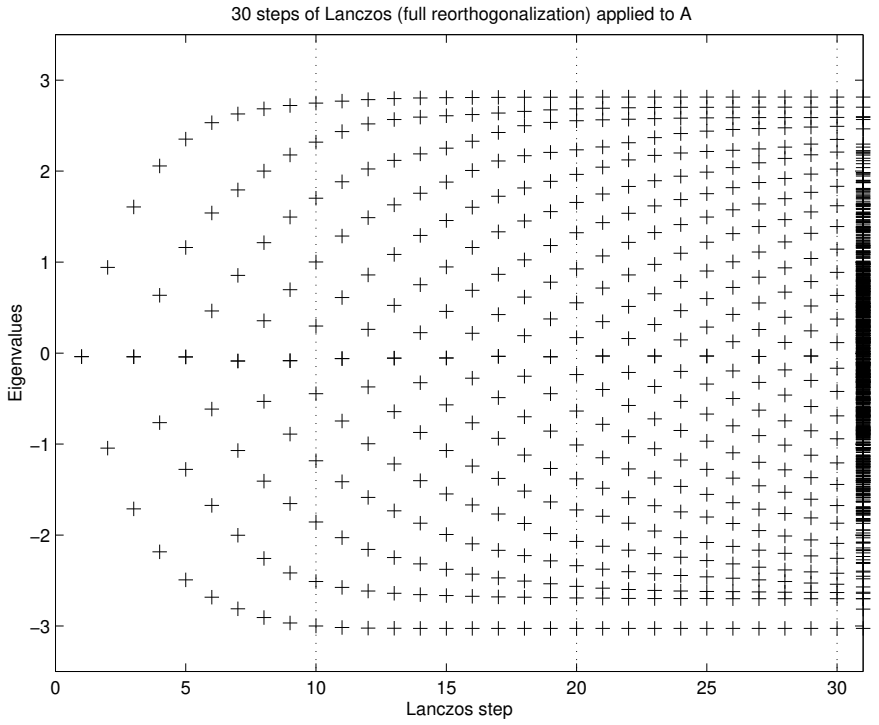
   Lanczos Algorithm

   1. $q_1 = v/\|v\|_2$, $\beta_0 = 0$; $q_0 = 0$;
   2. for $j = 1$ to k, do
   3. $\quad w = Aq_j$;
   4. $\quad \alpha_j = q_j^T w$;
   5. $\quad w = w - \alpha_j q_j - \beta_{j-1} q_{j-1}$;
   6. $\quad \beta_j = \|w\|_2$;
   7. $\quad$ if $\beta_j = 0$, quit;
   8. $\quad q_{j+1} = w/\beta_j$;
   9. $\quad$ compute eigenvalues and eigenvectors of T_j
   10. $\quad$ test for convergence
   11. endfor

4. We illustrate the Lanczos algorithm by a running an example, a 1000-by-1000 diagonal matrix A, most of whose eigenvalues were chosen randomly from a normal Gaussian distribution. To make the plot easy to understand, we have also sorted the diagonal entries of A from largest to smallest, so $\lambda_i(A) = a_{ii}$ with the corresponding eigenvector e_i. There are a few extreme eigenvalues, and the rest cluster near the center of the spectrum. The starting Lanczos vector v has all equal entries.

   There is no loss in generality in experimenting with a diagonal matrix, since running the Lanczos algorithm on A with starting vector $q_1 = v/\|v\|_2$ is equivalent to running the Lanczos algorithm on $Q^T A Q$ with starting vector $Q^T q_1$.

   The following figure illustrates convergence of the Lanczos algorithm for computing the eigenvalues of A. In this figure, the eigenvalues of each T_k are shown plotted in column k, for $k = 1, 2, 3, \ldots, 30$, with the eigenvalues of A plotted in an extra column at the rightmost column. The column k has k "+"s, one marking each eigenvalues of T_k.



We observe that:

- Extreme eigenvalues, i.e., the largest and smallest ones, converge first, and the interior eigenvalues converge last.
- Convergence is monotonic, with the ith largest (smallest) eigenvalues of T_k increasing (decreasing) to the ith laregst (smallest) eigenvalue of A, provided that the Lanczos algorithm does not stop prematurely with some $\beta_k = 0$.

5. All the discussion in this lecture is under the assumption of exact arithmetic. In the presence of finite precision arithmetic, the numerical behaviors of the Lanczos algorithm could be significantly different. For example, in finite precision arithmetic, the orthogonality of the computed Lanczos vectors $\{q_j\}$ is lost when j is as small as 10 or 20. The simplest remedy (and also the most expensive one) is to implement the the full reorthogonalization, namely after the step 5, do

$$w = w - \sum_{i=1}^{j-1} (w^T q_i) q_i.$$

   This is called the Lanczos algorithm with full reorthogonalization. (Sometimes, it may be needed to execute *twice*). A more elaborate scheme, necessary when convergence is slow and several eigenvalues are sought, is to use the selective orthogonalization.

6. An excellent reference to study the observation in theory is the book by B. N. Parlett, "The Symmetric Eigenvalue Problem", reprinted by SIAM, 1998.

Part V. Arnoldi algorithm

1. The power method is the simplest algorithm suitable for computing just the largest eigenvalue in absolute value, along with its eigenvector. Starting with a given x_0, k iterations of the power method produce a sequence of vectors $x_0, x_1, x_2, \ldots$. It is easy to see that these vectors span a *Krylov Subspace*:

$$\text{span}\{x_0, x_1, x_2, \ldots, x_m\} = \mathcal{K}_{m+1}(A, x_0) = \text{span}\{x_0, Ax_0, A^2x_0, \ldots, A^m x_0\}.$$

Now, rather than taking x_m as out approximate eigenvector, it is natural to ask for the "best" approximate eigenvector in $\mathcal{K}_{m+1}(A, x_0)$ using the *Rayleigh-Ritz procedure*. We will see that the eigenvector (and eigenvalue) approximations from $\mathcal{K}_{m+1}(A, x_0)$ are much better than x_m alone.

The *Arnoldi algorithm* for finding a few eigenpairs of a large scale matrix A combines the Arnoldi process for building a Krylov subspace with the Raleigh-Ritz procedure.

2. Let us recall that the following Arnoldi process generates an orthonormal basis of a Krylov subspace $\mathcal{K}_m(A, v)$:

$[V_{m+1}, \widehat{H}_m] = \text{arnoldi}(A, v, m)$
1. $v_1 = v/\|v\|_2$
2. for $j = 1, 2, \ldots, m$
3. compute $w = Av_j$
4. for $i = 1, 2, \ldots, j$
5. $h_{ij} = v_i^T w$
6. $w := w - h_{ij} v_i$
7. end for
8. $h_{j+1,j} = \|w\|_2$
9. If $h_{j+1,j} = 0$, *stop*
10. $v_{j+1} = w/h_{j+1,j}$
11. endfor

The Arnoldi process yields the fundamental relation, referred to as an Arnoldi decomposition of length k:

$$AV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T = V_{m+1} \widehat{H}_m, \tag{5}$$

where $V_m^H V_m = I$, $V_m^H v_{m+1} = 0$, H_m is Hessenberg:

$$H_m = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1,m-1} & h_{1m} \\ h_{21} & h_{22} & \cdots & h_{2,m-1} & h_{2m} \\ & h_{32} & \ddots & h_{3,m-1} & h_{3m} \\ & & \ddots & \vdots & \vdots \\ & & & h_{m,m-1} & h_{m,m} \end{bmatrix},$$

and $V_{m+1} = [V_m \; v_{m+1}]$ and $\widehat{H}_m = \begin{bmatrix} H_m \\ h_{m+1,m} e_m^T \end{bmatrix}$.

If $\widehat{H}_m$ is unreduced, i.e., $h_{i+1,i} \neq 0$ for $i = 1, 2, \ldots, m$, the decomposition is uniquely determined by the starting vector v (This is commonly called implicit Q-Theorem).

3. Since $V_m^H v_{m+1} = 0$, we have

$$H_m = V_m^T A V_m.$$

Let μ be an eigenvalue of H_m and y be a corresponding eigenvector y, i.e.,

$$H_m y = \mu y.$$

Then the corresponding Ritz pair is $(\mu, V_m y)$. Applying y to the right of (5), the residual vector of $(\mu, V_m y)$ is given by

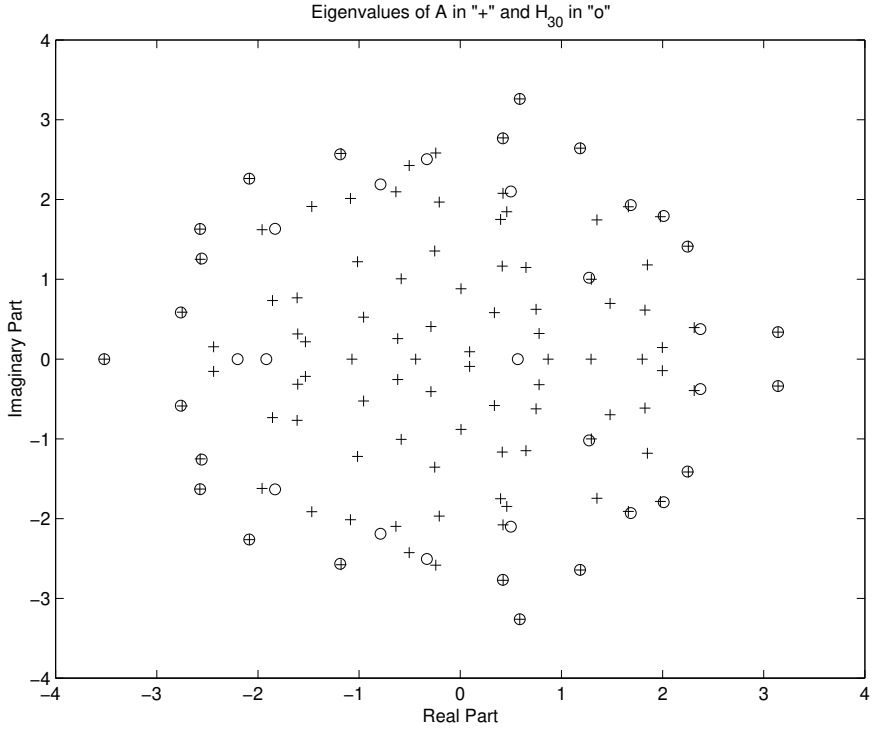
$$A(V_k y) - \mu(V_k y) = h_{m+1,m} v_{m+1}(e_m^T y).$$

Using the backward error interpretation, we know that $(\mu, V_m y)$ is an exact eigenpair of $A+E$, where $\|E\|_2 = |h_{m+1,m}| \cdot |e_m^T y|$.

4. This gives us a criterion for accepting the Ritz pair $(\mu, V_k y)$ as approximate eigenpair[1] of A.

Arnoldi's Algorithm
1. Choose a starting vector v
2. Generate the Arnoldi decomposition of length m by the Arnoldi process
3. Compute the Ritz pairs and decide which are acceptable
4. If necessary, increase m and repeat

5. We illustrate the above Arnoldi algorithm by a running a 100-by-100 random sparse matrix A with approximately 1000 normally distributed nonzero entries, $A =$ `sprandn(100,100,0.1)`. All entries of the starting vector v are 1. The following figure illustrates typical convergence behavior of the Arnoldi algorithm for computing the eigenvalues. In the figure, "+" are the eigenvalues of matrix A (computed by `eig(full(A))`). and the "$\circ$" are the eigenvalues of upper Hessenberg matrix H_{30} (computed by `eig(`H_{30}`)`).



[1]Note that because of non-symmetry of A, we generally do not have the nice forward error estimation as discussed in the Lanczos algorithm for symmetric eigenproblem. But a similar error bound involving the condition number of the corresponding eigenvalue exists.

We observe that *exterior eigenvalues converge first*. This is the typical convergence phenomenon of the Arnoldi algorithm (in fact, all Krylov subspace based methods). There is a general theory for the convergence analysis of the Arnoldi algorithm.

6. The Arnoldi algorithm has two nice aspects:

   (a) The matrix H_m is already in Hessenberg form, so that we can immediately apply the QR algorithm to find its eigenvalues.

   (b) After we increase m, say $m + p$, we only have to orthogonalize p vectors to compute the $(m + p)$th Arnoldi decomposition. The work we have done previously is not thrown away.

   Unfortunately, the algorithm has its drawbacks:

   - If A is large we cannot increase m indefinitely, since V_m requires $n \times m$ memory locations to store.
   - We have little control over which eigenpairs the algorithm finds. In a given application, we will be interested in a certain set of eigenpairs. For example, eigenvalues lying near the imaginary axis. There is nothing in the algorithm to force desired eigenvectors into the subspace or the discard undesired ones.

7. These issues have been successfully addressed to some extent by a so-called *implicitly restart* Arnoldi method, see

   - D. Sorensen, Implicit application of polynomial filters in a k-step Arnoldi method, SIAM J. Matrix Anal. Appl., Vol. 13, pp.357–385, 1992.
   - Z. Bai, J. Demmel, J. Dongarra, A. Ruhe and H. van der Vorst, editors, Templates for the solution of Algebraic Eigenvalue Problems: A Practical Guide. SIAM, Philadelphia, 2000 Available at `http://www.cs.ucdavis.edu/~bai/ET/contents.html`

   The Matlab's function `eigs` is an implementation of the implicitly restart Arnoldi method.