Eigenvalue Problems

Introduction (Matrices from now on)

Occur in many areas of Science and Engineering

Standard Form: Find nonzero vector $x$ and scalar $\lambda$

such that

$$Ax = \lambda x$$

$Ax$ parallel to $x$

If $A$ is real symmetric or complex Hermitian

If $A$ is real non-symmetric or complex non-Hermitian

Definition

Spectrum $\sigma(A) = \{\lambda \in \mathbb{C} | \exists x \neq 0 : Ax = \lambda x\}$

Spectral radius $r(A) = \max \{ |\lambda| : \lambda \in \sigma(A) \}$

Remark

$x$ is a right eigenvector

$y^*A = y^*y$

$y$ is a left eigenvector.

$$y^*y = \lambda y$$

We note that $x = y$ and $A = A^*$

$\lambda = \bar{\lambda}$ (real)

$\lambda = \sigma(A)$
equation \( Ax = \lambda x \) is equivalent to
\[
(A - \lambda I)x = 0
\]
which has non-trivial solution (\( x \neq 0 \)) if and only if \( (A - \lambda I) \) is singular
\[
\Rightarrow \quad \det [A - \lambda I] = 0
\]
Characteristic polynomial of degree \( N \) (\( A \) of size \( N \))

Example \( A = \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix} \)

\[
\det [A - \lambda I] = \det \begin{bmatrix} 3 - \lambda & -1 \\ -1 & 3 - \lambda \end{bmatrix} = \lambda^2 - 6\lambda + 8 = 0
\]
\[
(3 - \lambda)^2 - (-1)(4) = \lambda^2 - 6\lambda + 8 = 0
\]
\[
\Rightarrow \lambda_1, \lambda_2 = \frac{6 \pm \sqrt{36 - 32}}{2} = 3 \pm 1
\]
\[
\Rightarrow \lambda_1 = 4, \quad \lambda_2 = 2
\]
\[
\text{eigenvalues} = \begin{bmatrix} 4, 2 \end{bmatrix}
\]

Remark: Roots of polynomial of degree \( \geq 4 \) cannot always be computed as a flash number of steps.

Computations of
- Eigenvalue problem is checkable in exacts (by nature)
- "Algebraic multiplicity of a root \( x \) is the number of times \( x \) is a root"
Characteristic polynomial is powerful theoretical tool but useless computationally (too demanding, too sensitive).

Shortcoming problem

\[ A = \begin{bmatrix} 1 & \varepsilon \\ \varepsilon & 1 \end{bmatrix} \quad \varepsilon \ll 1 \text{ Fermi} \]

eigenvalue (exact) \( 1 + \varepsilon, 1 - \varepsilon \).

polynomial \( \det (A - I) = \lambda^2 - 2\lambda + (1 - \varepsilon^2) = \lambda^2 - 2 > + 1 \)

\( \Rightarrow 1 \) (double root), although they are distinct in working precision.

**Definition**

Two matrices \( A \) and \( B \) are similar if there exists a non-singular matrix \( X \) such that

\[ A = XBX^{-1} \]

(\( \Rightarrow \) same eigenvalue).

If \( B \) is similar to \( A \) and \( C \) to \( B \), then \( C \) is similar to \( A \) \( \Rightarrow \) similarity transform can take several steps.
A is diagonalizable if it is similar to a diagonal matrix

\[ A = XDX^{-1} \Rightarrow D = X^*AX \]

\( D = \text{Diagonal matrix} \)

Theorem of AE: The following are equivalent:

1. \( A \) is nondefective
2. \( A \) is diagonalizable
3. \( A \) has \( n \) linearly independent eigenvectors (not unique)
4. There exists a non-singular matrix \( X \in \mathbb{C}^{n \times n} \) and \( D = \text{diag}(\lambda_1, \ldots, \lambda_n) \) such that \( D = X^{-1}AX \)

\[ \Rightarrow \text{Any matrix is unitarily similar to a triangular matrix} \]

\[ A = QRQ^* \quad \text{and} \quad R \text{ such that} \]

\[ Q^*AQ = R \]

By 1, \( A \) is real if \( Q^*AQ = R \) is:

\[ R_{ii} = \Re \lambda_i \quad \text{or} \quad 2 \times 2 \text{ matrix with complex conjugate eigenvalues} \]

\( \Re \lambda_i \) and \( \Im \lambda_i \)
Any Hermitian matrix is unitarily similar to a real diagonal matrix. (Schafer form is real diagonal)

Jordan form

- Not all matrices are diagonalizable, they are limited to similarity transform.
- A defective matrix is a matrix that does not have a full set of N linearly independent eigenvectors.
- Defective matrix has an eigenvalue of multiplicity R > 1 with fewer than R linearly independent corresponding eigenvectors.

Example:

\[
\begin{bmatrix}
0 & 1 \\
0 & 0
\end{bmatrix} \Rightarrow \text{eigenvalue } 0, \text{ geometric multiplicity } 1
\]

\[
\begin{bmatrix}
1 & 1 \\
0 & 1
\end{bmatrix} \Rightarrow \text{eigenvalue } 1, \text{ algebraic multiplicity } 2 \Rightarrow \text{ geometric multiplicity } 1
\]

Defective matrix
The Jordan Canonical Form of matrix $A$ is an upper bidiagonal matrix whose main diagonal contains the eigenvalues of $A$ and superdiagonal contains 0 or 1.

\[ A = X J X^{-1} \]

**Example**

\[ J = \begin{bmatrix}
1 & 1 & 0 \\
0 & 2 & 1 \\
0 & 0 & 3
\end{bmatrix} \]

- $4$ is a single eigenvalue with algebraic multiplicity 1.
- $3$ has algebraic multiplicity 2.
- $2$ has algebraic multiplicity 2, but geometric multiplicity 1 (i.e., 1 eigenvector).

**General Theorem**

If $A$ is a non-singular $X$ such that

\[ A = X J X^{-1} = X \begin{bmatrix} J_1 & & \\ & \ddots & \& \\ & & J_k \end{bmatrix} X^{-1} \]

Then $J_i$ is called a Jordan block.

$A$ has only one eigenvector for each $J_i$.

$J_i$ could be equal to $J_j$.

The size of $J_i$ is the algebraic multiplicity of $\lambda_i$. 

\[ J_i = \begin{bmatrix}
\lambda_i & 1 \\
0 & \ddots
\end{bmatrix} \]
Some more definitions

\[ d(\lambda) \# \text{of alg. mult.} \]
\[ r(\lambda) \# \text{of linearly ind. eigenvectors [geo. mult.]} \]

\[ \Rightarrow d(\lambda) \geq r(\lambda) \]

- if \( d(\lambda) = r(\lambda) = 1 \) \Rightarrow \lambda \text{ is simple eigenvalue}
- if \( d(\lambda) > r(\lambda) \) \Rightarrow \lambda \text{ is a defective eigenvalue}
- if \( d(\lambda) = 0 \) \Rightarrow \lambda \text{ is a non-defective } \square

A is non-defective if all its eigenvalues are distinct.

Eigendecomposition

If \( A \) is non-defective, \( Ax = \lambda x \Rightarrow A = XDX^{-1} \)

\[ D = XAX \]

Problem transformations

\[ \Rightarrow \text{shift} \quad (A - \sigma I)x = (\lambda - \sigma) x. \]

\[ (A - \sigma I) = X (D - \sigma I) X \]

Eigenvalues are shifted \[ D - \sigma I = \begin{bmatrix} 3 - \sigma & 0 \\ 0 & 2 - \sigma \end{bmatrix} \]
Inversion

\[ A^{-1} = \frac{1}{\lambda} x \]
\[ A^{-1} = \lambda \ D^{-1} \ x \]
\[ A^k = X \ D^k \ X^{-1} \]
\[ D^k = \int_{\lambda}^{\infty} \ d\lambda \]

Powers

\[ A^k x = \lambda^k x \]
\[ A^k = X \ D^k \ X^{-1} \]

Polynomials

if \( p(t) \) is polynomial

\[ p(A) x = p(\lambda) x \]
\[ p(A) = X \ p(D) \ X^{-1} \]

\( \square \) Computing the eigenvalue problem

Different questions:

- Are all eigenvalues/eigenvectors needed at any a few?
- How accurate the solution should be?
- Is matrix relatively small and dense, or large and sparse?
- Does matrix have any special properties?

Different Algorithms possible

Basic:
- Power iteration
- Inverse iteration
- Rayleigh quotient
- QR algorithm
- Jacobi iteration

Advanced:
- Krylov subspace: Arnoldi, Lanczos,
- Jacobi Davidson, Trace-min, FEAST

Power Method

- Repeatedly multiplies matrix times a vector
- Initial vector ≠ 0
- Assuming that \( A \) has unique eigenvalue of
  largest modulus, say \( \lambda \) with eigenvector \( \mathbf{v} \)

then \( x_k = A x_{k-1} \) converges to multiple of \( \mathbf{v} \)
  corresponding to dominant eigenvalue.
Why? We can express the initial vector 

\[ x_0 = \sum_{i=1}^{N} a_i v_i \]

where \( v_i \) are eigenvectors of \( A \)

And \( Av_i = \lambda_i v_i \)

Then 

\[ x_k = A^k x_0 = \sum_{i=1}^{N} \lambda_i^k a_i v_i = x_k \left[ \lambda_1 + \sum_{i=2}^{N} \left( \frac{\lambda_i}{\lambda_1} \right)^k \right] \]

Since \( \left| \frac{\lambda_i}{\lambda_1} \right| < 1 \) for \( i > 1 \)

\[ \Rightarrow \quad x_k \rightarrow 0 \]

\[ \Rightarrow \quad \text{we get only the component corresponding to } v_1 \text{ at convergence.} \]

In practice, we should normalize along the iteration to avoid overflow/underflow if \( \lambda_1 < 1 \)

\[ \Rightarrow \quad \begin{bmatrix} y_k \\ x_k \end{bmatrix} = \begin{bmatrix} A & I \\ \frac{1}{\|y_k\|} & 0 \end{bmatrix} \begin{bmatrix} y_k \\ x_k \end{bmatrix} \Rightarrow \quad \|y_k\| \rightarrow 1 \quad \|x_k\| \rightarrow \frac{1}{\|y_k\|} \]

**Example**

\[ A = \begin{bmatrix} 1.5 & 0.5 \\ 0.5 & 1.5 \end{bmatrix} \]

\[ x_0 = \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} \]

\[ \text{Result} \]

\[ \lambda_1 = 2 \]

\[ v_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \]
* geometric interpretation

**Successive**

Initial vector $x_0$ + $x_1$ by multiplication components of $v_1$ will start to dominate dominant eigenvector (largest eigenvalue).

**Remark**

- There may be more than one eigenvector having maximum modulus $\Rightarrow$ortonormal may converge to linear combination of corresponding eigenvectors.

- Convergence rate of power iteration depends on ratio

$$\frac{x_2}{x_1}$$

Where $x_2$ is eigenvector having second largest modulus.
Invert creation

The eigenvalues of $A$ and $A^{-1}$ are identical.

But largest eigenvalues of $A^{-1}$ are reciprocal of smallest eigenvalues of $A$ (in magnitude)

Idea: Use the power method on $A^{-1}$ to compute the smallest eigenvalue of $A$.

$$y_k = A^{-1} x_{k-1} \quad \Rightarrow \quad A y_k = x_{k-1} \quad \text{[linear system]}$$

$$x_k = \frac{y_k}{\|y_k\|_\infty}$$

The factorization of $A$ need to be performed only once, only solve stage needed along the iteration.

Shift and invah

Use power method on $(A - \sigma I)^{-1} \Rightarrow$ compute eigenpairs closest to $\sigma$. 
Rayleigh (Ritz) quotient

\[ \lambda = \frac{x^T A x}{x^T x} \]

It can accelerate convergence of iterative methods → gives better approx to eigenvalue at iteration b

\[
\begin{array}{cccccc}
1 & 2 & 3 & 4 & 5 & 6 \\
1 & 1.5 & 1.67 & 1.80 & 1.91 & 1.570 \\
x^T A x & 1.5 & 1.8 & 1.941 & 1.951 & 1.599 \\
\end{array}
\]

Remark: given an approximate eigenvector \( x \) for real matrix \( A \), determine the best approximate for \( \lambda \) → similar to a least-squares problem \( n \times 1 \) size

\[ x^T A x \]

Normal equation

\[ (x^T x) \lambda = x^T A x \implies \lambda = \frac{x^T A x}{x^T x} \]

Rayleigh quotient has many useful properties

\[ \lambda_1 = \max_{x \neq 0} \frac{x^T A x}{x^T x} \quad \lambda_n = \min_{x \neq 0} \frac{x^T A x}{x^T x} \]
Rayleigh quotient iteration

\( \text{cond}\text{. inverse iteration} + \text{Rayleigh quotient} \)

\( \lambda \text{ rapidly to} \)

eigenvalue \( \approx \) approx.

eigenvalue used as shift

\( \beta_k = \frac{\mathbf{x}^T \mathbf{A} \mathbf{x}_k}{\mathbf{x}_k^T \mathbf{x}_k} \)

\( (\mathbf{A} - \sigma \mathbf{I}) \mathbf{y}_{k+1} = \mathbf{x}_k \quad \Rightarrow \quad \mathbf{x}_{k+1} = \frac{\mathbf{y}_{k+1}}{\|\mathbf{y}_{k+1}\|} \)

\( \tau \text{ needs to reorthogonalize at each iteration} \)

\( \Rightarrow \text{works for}\ \text{Hermitian matrix as well} \)

**Example**

\[
\begin{array}{ccc}
\mathbf{x}_0 & \mathbf{x}_1 & \mathbf{x}_2 \\
0.867 & 0.323 & 1 \\
0.323 & 1 & 1 \\
\Delta_k & 1.876 & 1.998 & 2
\end{array}
\]
Deflation

After \( x_1 \) have been computed, additional eigenvectors can be computed by deflation that remove known eigenvalues.

Subspace Iteration

- simplest method for computing many eigenvectors
- generalization to eigensubspace \( X_{n \times p} \)

\[
X_{k+1} = AX_k \quad \text{starting with } X_0
\]

\( \Rightarrow \) span \( [X_k] \) converges to invariant subspace determined by \( p \) largest eigenvalue of \( A \).

\( \Rightarrow \) one can generalize subspace iteration way shift-rotate as well.

Remark: * normalization is needed
* each column of \( X_k \) converge to column vec tor
\( \Rightarrow X_k \) become increasingly ill-conditioned.
Both issues can be addressed using QR factorization at each iteration (i.e. orthogonal subspace iteration).

\[
\begin{align*}
Q_k R_k &= X_{k-1} \\
X_k &= A Q_k \\
R_k &= \text{reduced QR factorization of } X_{k-1} \\
\text{orthonormalization}
\end{align*}
\]

\[
R_k = \text{span } \{Q_k\} = \text{span } \{X_{k-1}\}
\]

Fast common method \(\Rightarrow\) QR algorithm \([QR \text{ iterations}]

For \(p = AV, X_0 = I\)

we note that \(A_k = Q_k^H A Q_k\)

generated by orthogonal iterations converge to triangular or block triangular form \(\Rightarrow\) yielding all eigenvalues of \(A\).

QR iteration = compute successive matrices \(A_k\).
idea: start with \( A_0 = A \)

\[
\Rightarrow \text{at iteration } k \text{ compute } Q_k R_k = A_{k-1}
\]

and form reverse product \( A_k = R_k Q_k \)

By successive matrix, \( A_k \) are unitarily similar to each other:

\[
A_k = R_k Q_k = Q_k^H A_{k-1} Q_k
\]

Diagonal entries of \( A_k \) converge to eigenvalue of \( A \).

Product of orthogonal matrix \( Q_k \) converge to eigenvector space.

If \( A \) is symmetric, symmetry is preserved by QR iteration, \( A_k \) both triangular and symmetric.

\[
\text{Example: } A_0 = \begin{bmatrix} 7 & 2 \\ 2 & 4 \end{bmatrix}, \quad A_0 = Q_1 R_1 = \begin{bmatrix} 0.562 & -0.745 \\ 0.745 & 0.562 \end{bmatrix} \begin{bmatrix} 7.28 & 3.02 \\ 3.02 & 3.32 \end{bmatrix}
\]

\[
A_1 = R_1 Q_1 = \begin{bmatrix} 7.83 & 0.96 \\ 0.96 & 3.17 \end{bmatrix}
\]
off diagonal entries become smaller and diagonal entries closer to eigenvalues 2 and 3.

- Process continues until convergence.

The basic QR algo is not used in practice (too slow), and it is usually accelerated by:

1. Use of shift
2. Transforming \( A \) first to Hessenberg form

**Shift**

**Remark** \( A \) (i) converge to real eigenvalue.

Real

**Symmetric Cap**

Until row 'ain 1 \( i \leq N \) converge to zero

Obtain next shift \( \mu = \text{any} \).

\[
A - \mu I = QR
\]

Set \( A = RQ + \mu I \)

Next step = deflation, apply algo to \((n-1) \times (n-1)\) upper triangular matrix.
Preliminary reduction

- Efficiency of QR algo can be enhanced by first transforming the matrix as close as possible to triangular form.
- Hessenberg matrix is triangular except for one additional nonzero subdiagonal.
- Any matrix can be reduced to Hessenberg form in finite number of steps by orthogonal similarity transform or Householder transformation.
- Symmetric Hessenberg matrix is Tridiagonal.

So, QR algo is usually implemented in 2 stages:

\[ A \leadsto \text{Symmetric} \leadsto \text{Tridiagonal} \leadsto \text{Diagonal} \]

or

\[ A \leadsto \text{non-Symmetric} \leadsto \text{Hessenberg} \leadsto \text{Triangular} \]

Reduction obtained in definite Kronecker stage. Continue until convergence in practice only a small number of iterations needed.

Do much of the work in preliminary reduction.

\[ Q \text{ or } R \]
Complements

Generalized eigenvalue problem \( Ax = \lambda B x \)

\( A, B \) \( N \times N \) matrices.

- Generalized problem can be converted to standard one.

\[
(B^{-1}A)x = \lambda x \quad (A^{-1}B)y = \frac{1}{\lambda} y
\]

- If \( B \) is not spec\( \lambda \), \( B = LL^T \)

\[
A = L^T^{-1} A (L^T)^{-1} \quad LL^T x = \lambda L^T x
\]

- Other alternative QZ algorithm

Reduce \( A \) and \( B \) simultaneously to upper triangular form.

Eigenvalues given by \( \lambda_i = \frac{a_{ii}}{b_{ii}} \) for \( b_{ii} \neq 0 \).
Generalized problems would add difficulty for more advanced alg like FEAST.

Solving \( Ax \rightarrow Bx \)

* If \( A, B \) symmetric, \( B = B^T \)

\[ x^T B x = I \]

\[ x = \{ x_1, \ldots, x_M \} \]

* If \( A, B \) are symmetric

\[ A x = x B x \]
\[ A^T y = B^n y \]

\( B \) - biorthogonal basis

other forms of eigenvalue problems

\( A [x] x \rightarrow B x \Rightarrow \) non-linear eigenvalue

\( e.g. \) polynomial eigenvalue problem

\[ \begin{bmatrix} A_2 + \lambda A_1 + A_0 \end{bmatrix} x = 0 \]

\( (\text{quadratic}) \)

\( A [\{x\}] x \rightarrow B x \Rightarrow \) non-linear

eigenvector problem

\( e.g. \) electronic structure calculation