Lecture 7  Matrices and Eigensystems

- As we have seen, physics is full of eigenvalue equations,

\[ H \psi_n = E_n \psi_n \]

1. Special form of \( V \),

Solutions are given by special functions,

\[
\begin{align*}
\text{Hermite (}\ V(x) = \frac{1}{2}kx^2) \\
\text{Spherical Harmonics (spherical coordinates)} \\
\text{Laguerre (central force)}
\end{align*}
\]

- Solve only limited form of \( H \).

2. Numerical Solve to Diff. Eq. \( (H_u \# 10) \)

3. Matrix Solutions
**General Approach**

\[ H \Psi_\alpha = E_\alpha \Psi_\alpha \]

1. Find a complete set of suitable basis states \( \{ \Phi_n \} \)
   - such that \( \{ \Phi_n \} \) is complete (any wavefunction can be expanded in terms of \( \{ \Phi_n \} \))
   \[ \Psi_\alpha = \sum_{n=1}^{\infty} C_{\alpha n} \Phi_n \]
   - \( \{ \Phi_n \} \) is an orthonormal set
   \[ \langle \Phi_\alpha | \Phi_\beta \rangle = \delta_{\alpha \beta} \]
   - Suitable: \( \{ \Phi_n \} \) based on some knowledge of the system, such as symmetry or
   \[ H = H_0 + \Delta H \] such that \( H_0 \Phi_\alpha = E_\alpha \Phi_\alpha \)

\[ V(x) = \frac{1}{2} m \omega^2 x^2 + e \Phi \Psi \left( \frac{m \omega}{\hbar} \right)^2 x^4 \]

\[ (\text{We have already seen 1st order perturbation theory}) \]

\[ H \# 3 \text{ in getting } \Delta E \]

This is an alternative approach
$H |\Psi \rangle = E_{\Psi} |\Psi \rangle$

$\langle \phi_j | H |\Psi \rangle = E_{\Psi} \langle \phi_j |\Psi \rangle$  \hspace{1cm} (2)

In terms of $|\Psi \rangle = \sum_{i=1}^{\infty} C_i |\phi_i \rangle$, (not typically is infinite)

$\Rightarrow \quad \sum_{i=1}^{\infty} C_i H_{ji} = E_{\Psi} \sum_{i=1}^{\infty} C_i$  \hspace{1cm} (3)

$H_{ji} = \langle \phi_j | H |\phi_i \rangle$

$\Rightarrow$ This is a matrix eigenvalue equation:

$$
\begin{pmatrix}
H_{11} & H_{12} & \cdots & H_{1n} \\
H_{21} & H_{22} & \cdots & H_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
H_{n1} & H_{n2} & \cdots & H_{nn}
\end{pmatrix}
\begin{pmatrix}
C_1 \\
C_2 \\
\vdots \\
C_n
\end{pmatrix}
= E_{\Psi}
\begin{pmatrix}
C_1 \\
C_2 \\
\vdots \\
C_n
\end{pmatrix}
$$

(Note: Typically $H_{ji}$ is symmetric)

$$(H - E \mathbf{I}) \cdot \mathbf{C} = 0$$

So, $E_{\Psi}, |\Psi \rangle$ can be solved by this matrix equation!

$\Rightarrow$ The problem becomes find the eigenvalues $E$

and eigenvectors $\mathbf{C}$ of the above equation.

(Note: $n$ is infinite for most systems but a finite truncation usually gives good results for low energy states.)
Eigenvalues & Eigenvectors of Matrices

(Review of terms)

\[ A \cdot x_r = \lambda x_r \]
\[ \uparrow \quad \uparrow \quad \text{right eigenvectors} \]
General matrix
\[ \text{eigenvector} \quad \text{Note: } \lambda \text{'s are solution to} \]
\[ \text{det} (A - \lambda I) = 0 \]

\[ x_l \cdot A = \lambda^t x_l \]
\[ \rightarrow \text{left eigenvectors} \]

\[ \lambda = \lambda^t \quad (\text{left eigenvectors = right eigenvectors}) \]
\[ x_l \cdot A \cdot x_r = x_l \lambda x_r \quad \Rightarrow \quad \lambda^t x_l \cdot x_r = \lambda x_l^t \cdot x_r \]

Symmetric Matrices
\[ \text{Def. } A = A^t \]
\[ \rightarrow x_l = x_r^T \]

To show: \[ A \cdot x_r = \lambda x_r \]
\[ A^T \cdot x_r = \lambda x_r \]
\[ x_r^T \cdot A = \lambda x_r^T \quad \Rightarrow \quad x_l = x_r^T \]
Orthogonality

\[ X_K - \text{columns of } X_K \]
\[ X_L - \text{rows of } X_L \]
\[ A - X_K \text{ diag} (\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_n) \]
\[ X_L A = \text{diag} (x_1, x_2, \ldots, x_n) X_L \]
\[ X_L A X_K = X_L X_K \text{ diag} (x_1, x_2, \ldots, x_n) \]

\[ \implies \]
\[ X_L X_K \text{ diag} (\lambda_1, \lambda_2, \ldots, \lambda_n) = \text{diag} (x_1, x_2, \ldots, x_n) X_L X_K \]
\[ \implies X_L X_K \text{ is diagonal} \]

Thus, left eigenvectors are orthogonal to right eigenvectors!

Symmetric Matrices: \[ X_L X_K = \text{diag} \]
\[ X_L X_K^T = I \]

Orthonormal
**Direct Method**

Recall eigenvalue equation:

\[(A - \lambda I) \cdot x = 0\]

The eigenvalues are determined by

\[\det (A - \lambda I) = 0\]

\[f(\lambda) = 0\]

⇒ Eigenvalues solve zeros of \(f(\lambda)\)!

- Newton's Method if \(f'(\lambda)\) can be easily calculated
- Bisection, False position, Scan
Evaluation of Characteristic Polynomials
from Tridiagonal Form

- From Householder:
\[ P_{n-2} \cdots P_1 AP_{n-1} P_n - I \lambda = (\hat{A} - I \lambda) \]

- Characteristic equation defined by
\[ 0 = \det(\hat{A} - I \lambda) \]
will be a polynomial in \( \lambda \) of order \( n \).
Eigenvalues \( \alpha_i \) are the roots of this equation.

- Because of the form of the tridiagonal form,
\[ P_n(\lambda) = \det(\hat{A} - I \lambda) \]
can be found recursively:
\[
\begin{bmatrix}
\alpha_1 - \lambda & \beta_2 \\
\gamma_2 & \alpha_2 - \lambda & \beta_3 \\
& \gamma_3 & \alpha_3 - \lambda & \beta_4 \\
& & \gamma_4 & \ddots & \ddots \\
& & & \gamma_{n-1} & \alpha_{n-1} - \lambda & \beta_n \\
& & & & \gamma_n & \alpha_n - \lambda
\end{bmatrix}
\]

\[
P_n(\lambda) = (\alpha_1 - \lambda) P_{n-1}(\lambda) - \beta_1 \delta_1 P_{n-2}(\lambda)
\]

with
\[
P_0(\lambda) = 1, \quad P_1(\lambda) = \alpha_1 - \lambda.
\]

can easily check with a small matrix.
- Similar to other recursion relations, this is the most effectively used arithmetically (i.e., for a specific value of $\lambda$).

The recursion generates a sequence of values correspond to $P_n(\lambda)'s$.

$\Rightarrow$ With $\det (\hat{A} - \lambda I) = P_n(\lambda) = 0$ found, then $\lambda's$ can be found by bisection, Newton’s method, etc.
Power Method

- Let consider an arbitrary vector

\[ x = \sum a_j u_j \]

where \[ A \cdot u_j = \lambda_j u_j \]

- Also, for convenience, we order our \( \lambda \)'s so that

\[ |\lambda_1| > |\lambda_2| > \ldots > |\lambda_n| \]

Now, let \( A \) operates on \( x \)

\[ A \cdot x = A \sum a_j u_j \]

\[ = \sum a_j A u_j \]

\[ = \sum a_j \lambda_j u_j \]
\( A \cdot x \Rightarrow \sum_{j=1}^{n} a_{y_j} v_j \)

\( (x_i) \rightarrow x \)

For repeated application of \( A \),

\( A^m \cdot x = \sum_{j=1}^{n} a_{y_j} v_j \)

\( \lim_{m \rightarrow \infty} |A^m \cdot x| = |x| \lim_{m \rightarrow \infty} (a_{y_j} v_j) \)

\( = |x| \lim_{m \rightarrow \infty} (a_{y_j} v_j) \)

\( \Rightarrow \lim_{m \rightarrow \infty} |A^m \cdot x| = 0 \)

For large \( m \), \( A^m \cdot x \approx \lambda^m \cdot a_{y_j} v_j \)

(\( \text{useful later for Lyapunov arguments} \))
Smallest eigenvalue $\lambda_N$:

- Do the same with $A^{-1}x$!

since $\lambda_N$ will be the largest eigenvalue of $A^{-1}$.

A particular implementation

- get both $\lambda_1$ & $\mathbf{u}_1$ -

- Consider an arbitrary unit vector $\mathbf{z}_0$ such that $|\mathbf{z}_0| = 1$.

- Then form the follow sequences.

$\mathbf{z}_1 = A \cdot \mathbf{z}_0 / \delta_0$ where $\delta_0 = |A \cdot \mathbf{z}_0|$ 

(normalization constant)

and $\mathbf{z}_2 = A \cdot \mathbf{z}_1 / \delta_1$, $\delta_1 = |A \cdot \mathbf{z}_1|$
Similar argument as before,

$\Rightarrow \theta_m \to \lambda_1$ as $m \to \infty$

$\mathbf{z}_m \to \mathbf{y}_1$ as $m \to \infty$.

A: Extension to $[A_m]$ with varying $A_m$

$\Rightarrow$ basic numerical method to
get Lyapunov exponents.
To get eigenvector first if we know a good estimate of \( \lambda \).

\[ A \mathbf{y}_j = \lambda \mathbf{y}_j \]

- let say we have \( \mathbf{y} \) being a good estimate of \( \lambda \)

then consider,

\[ (A - \frac{\lambda}{2} I) \mathbf{y}_j = (\lambda_j - \frac{\lambda}{2}) \mathbf{y}_j \]

\( \bigstar \) this is eigenvalue equation again:

\( \mathbf{y}_j \) is an eigenvector of \( A - \frac{\lambda}{2} I \) with eigenvalue \( \lambda_j - \frac{\lambda}{2} \).

\( \bigstar \) - Now, since \( \lambda_j \geq \mathbf{y} \), \( \lambda_j - \frac{\lambda}{2} \geq 0 \).

\( \bigstar \) - if we apply inverse Power Method on \( A - \frac{\lambda}{2} I \):
Note: if \((A - \lambda I) u_j = (\lambda_j - \lambda) u_j\),
multiply \(\frac{1}{\lambda_j - \lambda} (A - \lambda I)^t\) to both sides,
\[
\left( \frac{1}{\lambda_j - \lambda} \right) u_j = (A - \lambda I)^t u_j
\]

1. \(\frac{1}{\lambda_j - \lambda}\) will be very large.

2. \(u_j\) is an eigenvector of \((A - \lambda I)\) with \(\lambda_j - \lambda\) and it is also an eigenvector of \((A - \lambda I)^t\) with \(\frac{1}{\lambda_j - \lambda}\)

\[\Rightarrow\] with inverse power method on 2
\(u_j\) can be obtained quickly!
Recall that the iterative procedure for power method looks like,
- Start with $x_0 = \text{normalized random vector}$
- $(A - \frac{1}{\lambda} I) \cdot x_m \quad \text{(apply } (A - \frac{1}{\lambda} I)^{-1})$
- Form $k_m = |(A - \frac{1}{\lambda} I)^{-1} x_m|

$x_{m+1} = (A - \frac{1}{\lambda} I)^{-1} x_m / k_m$

**Alternative without explicitly calculating** $(A - \frac{1}{\lambda} I)^{-1}$

**Note:** $(A - \frac{1}{\lambda} I)^{-1} x_m = (k_m x_{m+1})$

$\Rightarrow (A - \frac{1}{\lambda} I) x = x_m$

\[\text{given need to solve known}\]

- Linear $Gx$ for $x$ can be solve with $LU$ (with pivoting)
- Normalized $x \rightarrow x_{m+1} = \frac{x}{\|x\|_1}$
- Repeat $LU$ backward!

*(LU decom is already done)*

*Get should be reused since it involve RHS at each application*
General Tactics in Solving Eigenvalue Problems

- A few more definitions:

  **normal / defective matrices**: def. \([A, A^H] = AA^H - A^H A = 0\) H-conjugate transpose

  - A matrix \(A\) **normal**, one can always find a **complete set** orthogonal eigenvectors that span the \(N\)-dim vector space.

  - A matrix \(A\) is **defective** if its set eigenvectors is not complete.

**Symmetric / Hermitian** (they are normal):

- Symmetric: \(A = A^T\) (real matrices)

- Hermitian: \(A = A^H\) (complex matrices)

  \(\Rightarrow\) Hermitian conjugate

- If \(A^H A = A A^H\), \(A\) is normal.

  (Commute)

\(*\quad \text{Eigenvalues are real for Sym / Hermitian matrices!}\

**the complete set of**

- Nonsymmetric / non-Hermitian matrices might have **complex eigenvalues**.
- Starting with our eigenvalue problem definition,

\[ A \cdot \mathbf{x}_R = \mathbf{x}_R \cdot \text{diag} (\lambda_1, ..., \lambda_N) \]

If \(A\) is non-defective, \( \mathbf{x}_R \) exists, and

\[ \mathbf{x}_R^T \cdot A \cdot \mathbf{x}_R = \text{diag} (\lambda_1, ..., \lambda_N) \]

The left hand side is a similarity transformation

\[
\begin{align*}
\text{def.} \quad \Xi \cdot \Xi^{-1} \cdot A \cdot \Xi &= A' \\
\end{align*}
\]

where \( \Xi \) is a nonsingular matrix.

- This similarity transformation is special such that the particular choice of \( \Xi = \mathbf{x}_R \) diagonalizes \( A \).

Basic idea in most transformation-based eigenvalue solvers:

- to find the correct \( \Xi \) such that

\[ \Xi^T \cdot A \cdot \Xi = \text{diag} (\lambda_1, ..., \lambda_N) \]
**Note:** In the previous discussion, $A$ can be symmetric or nonsymmetric.

- For nonsymmetric $A$,
  
  the correct $Z$ that diagonalizes $A$,
  
  i.e., $Z^{-1} A Z = \text{diag} (\lambda_1, \ldots, \lambda_n)$

  might be complex!

- If $Z$ is restricted to real matrices (in most numerical applications),

  $Z^{-1} A Z \rightarrow \text{Jordan form at best}$

  - block diagonal with

  \[
  \begin{pmatrix}
  \lambda_1 & 0 & 0 & 0 \\
  0 & (a & b & 0) \\
  0 & (c & d & 0) \\
  & & & \lambda_2
  \end{pmatrix}
  \]

  \[
  \rightarrow \text{block diagonal with all real entries}
  \]

  - For symmetric $A$, with real $Z$

  $Z^{-1} A Z \rightarrow \text{diag} (\lambda_1, \ldots, \lambda_n)$

  - $\lambda_i$ real

  - $Z$ orthogonal & real.
Important Point about Similarity Transformation:

\[
\det ( z^{-1} \cdot \mathbf{A} \cdot z - \lambda I) \\
= \det ( z^{-1} \cdot ( \mathbf{A} - \lambda I) \cdot z) \\
= \det ( z^{-1} \det ( \mathbf{A} - \lambda I) \det z) \\
= \det ( \mathbf{A} - \lambda I) \\
\Rightarrow \text{So, } \mathbf{A}' = z^{-1} \cdot \mathbf{A} \cdot z \text{ will have the same eigenvalues as } \mathbf{A}! 
\]
Basic Numerical Strategy

Attempt to transform $A$ into diagonal form by a sequence of similarity transforms

$$A \rightarrow P_i^{-1} A P_i \rightarrow P_2^{-1} P_i^{-1} A P_i P_2 \rightarrow \cdots \rightarrow P_n^{-1} P_2^{-1} P_i^{-1} A P_i P_2 \cdots \rightarrow \text{diag}(\lambda_1, \ldots, \lambda_n)$$

where $I_K = P_i P_2 P_3 \cdots$ gives the eigenvectors

If only eigenvalues are of interest, then we don't need to reduce $A$ all the way to diagonal $\rightarrow$ triangular (by similar transforms)

- less work
- $\lambda$ will be diagonal elements

A finite sequence of these transforms usually can't take $A$ to actual diagonal or triangular form, e.g., Jacobi or $Q R / Q L$
Pre-process

- Speed of convergence of these transformations
  algorithm can be significantly improved if
  \( A \) is first put in special forms by a smaller
  # of pre-process steps.

  Standard shapes: tridiagonal \( TD \) (sym)
  \[
  \begin{pmatrix}
  a & b \\
  0 & c
  \end{pmatrix}
  \]
  Hessenberg \( H \) (non-sym)
  \[
  \begin{pmatrix}
  a & b & c \\
  0 & d & e \\
  0 & 0 & f
  \end{pmatrix}
  \]

**Householder Transform** \( (A \rightarrow TD \text{ or } H) \)

\( R \) can reduce a non matrix \( A \) to

\( TD \) \( H \) \( H \) in \((n-2)\) steps!

(If annihilates the required part of the
cols & the corresponding row] in one step.)

(for symmetric)

**Construction:**

Householder matrix \( P = I - 2 uu^T \), \( u \) real vector

\( (u^T)^2 = 1 \)
**Note:** \( W W^T \) is a matrix

\[ (W)(W^T) = ( ) \]

\( W^T \cdot W \) is a scalar

\[ (W^T)(W) = 1 \]

---

**Note:**

\[ P = (1 - 2W W^T)(1 - 2W W^T) \]

\[ = 1 - 4W W^T + 4W \cdot (W^T \cdot W) W^T \]

\[ = 1 \]

\( \Rightarrow \) \[ P^{-1} = P \]

\( \Rightarrow \) Householder transformer is orthogonal.

And, since \( P = P^T \) (by construction)

\[ P^T = P^{-1} \] (orthogonal)

---

Revisiting \( P \) as:

\[ P = 1 - \frac{U \cdot U^T}{H} \]

where \( H = \frac{1}{2} |U|^2 \)

In this case, \( U \) can be any vector.
Suppose \( \mathbf{x} \) is the first column of \( \mathbf{A} \),

choose \( \mathbf{u} = \mathbf{x} + 1 \times \mathbf{e}_1 \)

where \( \mathbf{e}_1 = (1 \ 0 \ 0 \ 0) \) unit vector.

We want see the effect the action of \( \mathbf{P} \) on \( \mathbf{x} \):

\[
\mathbf{P} \cdot \mathbf{x} = \mathbf{x} - \frac{\mathbf{u}}{\mathbf{H}} \cdot (\mathbf{x} + 1 \times \mathbf{e}_1)^T \cdot \mathbf{x}
\]

\[
\mathbf{H} = \frac{1}{2} \mathbf{u} \mathbf{u}^T = \frac{1}{2} \mathbf{u}^T \mathbf{u}
\]

\[
= \frac{1}{2} (\mathbf{x} - 1 \times \mathbf{e}_1)^T (\mathbf{x} - 1 \times \mathbf{e}_1)
\]

\[
= \frac{1}{2} (2 \mathbf{x}^T - 2 \mathbf{x} \cdot \mathbf{e}_1)
\]

\[
= 1 \mathbf{x}^2 - 1 \mathbf{x} \cdot \mathbf{e}_1
\]

So, \( \mathbf{P} \cdot \mathbf{x} = \mathbf{x} - \mathbf{u} \frac{1 \mathbf{x}^2 - 1 \mathbf{x} \cdot \mathbf{e}_1}{1 \mathbf{x}^2 + 1 \mathbf{x} \cdot \mathbf{e}_1} \)

\[
= \pm 1 \times \mathbf{e}_1
\]

So, the effect of \( \mathbf{P} \) on \( \mathbf{x} \) is to reduce it to one with only one nonzero element.
To reduce a symmetric matrix to tridiagonal.

We choose $\mathbf{x}$ for the first householder matrix $\mathbf{P}_1$, to be the lower $(n-1)$ elements of the 1st column.

Thus,

$$
\mathbf{P}_1 \cdot \mathbf{A} = \begin{pmatrix}
1 & 0 \\
0 & (n-1)\mathbf{P}_1
\end{pmatrix}
\begin{pmatrix}
a_{11} & a_{12} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{n1} & a_{n2} & \cdots & a_{nn}
\end{pmatrix}

\begin{pmatrix}
x_1 \\
-x_1 \\
\vdots \\
-x_1
\end{pmatrix}
$$

unchanged by $\mathbf{P}_1$

$$
= \begin{pmatrix}
a_{11} & a_{12} & \cdots & a_{1n} \\
k & 0 & \cdots & 0 \\
0 & \cdots & \cdots & 0 \\
0 & 0 & \cdots & 0
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_1 \\
x_1 \\
x_1
\end{pmatrix}
$$

Where $k = 1 (a_{11}, \ldots, a_{1n})$

$\mathbf{P}_1 - (n-1)\times(n-1)$ householder matrix

with $k = (a_{11}, \ldots, a_{1n})^T$

$$
\mathbf{P} = \frac{I - \mathbf{uu}^T}{\|\mathbf{u}\|} \\
\mathbf{u} = \mathbf{x} - \|\mathbf{x}\| \mathbf{x}/\|\mathbf{x}\| \\
\mathbf{u}^T = \frac{I - \mathbf{x}\mathbf{x}^T}{\|\mathbf{x}\|}
$$
The complete orthogonal transformation is then

\[ A' = P \cdot A \cdot P^T = \begin{pmatrix} a_{11} & k & 0 \\ k & k & 0 \\ 0 & 0 & X \end{pmatrix} \]

Note: Householder is a similarity transform also.

Note: \( P^T \) with the same \( X \) reduces the first row because \( A \) is symmetric!

---

2nd step:

Choose

\[ P_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & (n-2)_{p_2} \end{pmatrix} \]

where \((n-2)_{p_2}\) is a \((n-2) \times (n-2)\) householder matrix with 

\[ x = (a_{22} - d_{12}^2) \]

being the \((n-2)\) elements of the 2nd column of \(A'\).

\[ \rightarrow \text{this reduces } A' \text{ to } A'' = \begin{pmatrix} a_{11} & k & 0 & 0 \\ k & a_{22} & k' & 0 \\ 0 & k' & 0 \\ 0 & 0 & x \end{pmatrix} \]
Thus, for a total of \((n-2)\) household transforms \(P_{n-2} \cdot P \cdot A \cdot P = P_{n-2}\) reduces \(A\) to a tridiagonal form!

Note: Since the cancelations of large numbers tends to reduce accuracy, it is preferred to arrange the matrix so that the largest off diagonal elements will be take care of last. (by row or column rotation)

Note 2: Instead of matrix multiplications:

\[ P \cdot A \cdot P \]

it is more efficient to compute

\[ R = \frac{A \cdot u}{H} \] (a vector)

Then

\[ A' = A \cdot \left( 1 - \frac{u \cdot u^T}{H} \right) = A - R \cdot u^T \]

\[ A' = P \cdot A \cdot P = A - R \cdot u^T - K \cdot R^T + 2 K \cdot u \cdot u^T \]
$$\mathbf{z} = \mathbf{r} - \mathbf{K}\mathbf{u}$$

the  \[ \mathbf{A}' = \mathbf{A} - \mathbf{z}\mathbf{u}^T - \mathbf{u}\mathbf{z}^T \]

**Steps:**

1. \( \rightarrow \quad \mathbf{u} \)
2. \( \rightarrow \quad \mathbf{H} = \frac{1}{2} \mathbf{u}\mathbf{u}^T \)
3. \( \rightarrow \quad \mathbf{z} = \frac{\mathbf{A}\mathbf{u}}{\mathbf{H}} \)
4. \( \rightarrow \quad \mathbf{k} = \frac{\mathbf{u}^T \mathbf{z}}{2\mathbf{H}} \)
5. \( \rightarrow \quad \mathbf{z} = \mathbf{z} - \mathbf{k}\mathbf{u} \)
6. \( \rightarrow \quad \mathbf{A}' = \mathbf{A}' - \mathbf{z}\mathbf{u}^T - \mathbf{u}\mathbf{z}^T \)

---

Now, we look at two representative transformation methods that take \( \mathbf{A} \) to diagonal form or triangular form. (For complex or degenerate \( \lambda \) 's, other considerations need to be taken, please.)

**Note:** Different from the preprocess transform, the reduction to diagonal or triangular form can't be done in finite steps!
1. Jacobi transform \( (\text{for symmetric matrices}) \) 

- individual Pi \( ac \) rotations to zero two specific off-diagonal elements.

- based on the idea that

\[
R^{-1} \begin{pmatrix} a & b \\ b & d \end{pmatrix} R \rightarrow \begin{pmatrix} a' & 0 \\ 0 & d' \end{pmatrix}
\]

\[
\begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix}
\]

- problem: already zeroed elements can be charged back to non-zero by subsequent \( P \)'s.

\[
\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x & \times \times \times \times \\ \times \times \times \times & \times \times \times \times \\ \times \times \times \times & \times \times \times \times \times \times \times \times \times \times \\ \times \times \times \times & \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \time
QR (QL)

\[ A = Q \cdot R \]

Now, consider:

\[ A' = R \cdot Q \]

Since \( Q \) is orthogonal, \( R = Q^T \cdot A \) (from (i))

\( (Q \cdot Q^T = I) \)

\[ \Rightarrow A' = Q^T \cdot A \cdot Q \]

\( (A' \) is an orthogonal similarity transform of \( A \) \)

With \( A' \), we can decompose it again:

\[ A' = Q' \cdot R' \]

Then \( A^2 = Q'^T \cdot A' \cdot Q' \)

\[ = Q'^T \cdot Q^T \cdot A \cdot Q \cdot Q' \]
Thus, we can generate a sequence of orthogonal similarity transform on $A$

Again, it can be shown that

1. If $A$ has $\lambda$s of different absolute values then $A^n \rightarrow D$ (if $A$ is symmetric)

2. If $\lambda$s are degenerate then $A^n \rightarrow$ D with blocks (if $A$ is symmetric)

$A^n \rightarrow T$ with blocks (if $A$ is nonsymmetric)

---

Convergence is fast for tridiagonal matrix $O(n)$! (D with super or sub diagonal)
and $O(n^2)$ for Hessenberg matrix with shifting.

(T with sub diagonal)
The QR decomposition

Recall the householder transform can reduce

\[ x = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} \rightarrow \begin{pmatrix} 1x_1 \\ \vdots \\ 0 \end{pmatrix} = P \cdot x \]

\[ P = \frac{1}{x_1} - \frac{uu^T}{H} \quad u = x - 1x_1e_1 \]

\[ H = \frac{1}{2} \|u\|^2 = \frac{1}{2} u^T u \]

This is orthogonal!

Just like the reduction to tridiagonal form, a sequence of \((n-1)\) \(P\)'s can reduce \(A\) to upper triangular form, such that

\[ P_{n-1} \ldots P_1 A = R \]
Note: \( x = 1^{st} \text{ column of } A \)

then \( P_1 A = \begin{pmatrix} k \\ 0 \\ 0 \\ \vdots \end{pmatrix} \)

\[
P_2 = \begin{pmatrix} 1 & 0 & \ldots & 0 \\ 0 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{pmatrix} \rightarrow \text{dim household with lower (n-1) elements of the 2nd column of } P_1 A.
\]

then \( P_2 P_1 A \approx \begin{pmatrix} k & x & \cdots & x \\ 0 & k' & x & \cdots \end{pmatrix} \)

Now, compare \( P_{m-1} \ldots P_1 A = R \) with \( A = QR \)

\[\Rightarrow \quad Q = (P_{m-1} \ldots P_1)^{-1} \]

Q is orthogonal since \( P_i \)'s are orthogonal.

\[\Rightarrow \quad Q = P_i \ldots P_{m-1} \]

Note: \( P = P^T = P^{-1} \) for household.
It can be shown that

\[ A^n = D = \text{diag}(\lambda_1, \ldots, \lambda_n) \]

if \( A \) is symmetric.

And,

\[ \lambda_1 < \lambda_2 < \cdots < \lambda_n \]

\[ A = QR \]

where

\[ Q = P_1 P_2 \cdots P_n \]

\[ R = P'_1 \]

So,

\[ P_i \] are orthogonal matrices.

Finally, we have

\[ \text{rank}(A) = \frac{n}{2} \]

A maximal if \( \frac{n}{2} \) should
Shifting & rate of convergence

It can be shown that for an off-diagonal element $a_{ij}$

$$\text{rate } \{a_{ij} \to 0\} \sim \left(\frac{\lambda_j}{\lambda_i}\right)^n \quad i < j$$

- While we know that $\lambda_j > \lambda_i$.
- If $\lambda_j \leq \lambda_i$, then the convergence is slow!

To prevent this, we can shift all diagonal values by $k_n$:

$$A' = A - k_n I$$

1. $\det(A') = \det(A - k_n I) = \lambda - k_n$
   - Eigenvalues shifted

2. The resulting recursive similarity transformation:
   - $A_{m+1} = Q_n^T A_{m} Q_n$
   - The same $Q_n$ transform
     - $A_{m} \to A_m^{(i)}$
     - $A_m \to A_{m+1}$
so that the convergent characteristic is the same but with a different rate:

\[ \text{rate} \{ a_{ij}^{n} \to 0 \} \sim \frac{|\lambda_i - k_n|}{|\lambda_j - k_n|} \]

so that we can adjust the ratio at each step \( n \) such that

\[ \lambda_i - k_n \neq 0 \]

to maximize the rate of convergence!

To see \( A^{'n+1} = Q_n^T A^n Q_n \)

\[ A'_n = Q_n R_n \]
\[ A'_n = R_n Q_n = Q_n^T A'_n Q_n \]

\[ A_{n+1} - k_n I = Q_n^T A_n Q_n - k_n Q_n^T Q_n \]
\[ = Q_n^T A_n Q_n - k_n I \]
\[
\Rightarrow \quad A_{n+1} = Q_n^T A_n Q_n
\]