**QR Iteration**

\[ A_0 = A \]

for \( k = 0, 1, 2, \ldots \)

\[ A_k = Q_k R_k \]

\[ A_{k+1} = R_k Q_k \]

end

**QR decomposition of \( A_k \)**

**Note:**

- \( A_{k+1} = R_k Q_k = Q_k^T A_k Q_k \)
  so \( A_{k+1} \) is similar to \( A_k \) (Same \( \lambda \)'s)

- **stable algorithm**, since it is based on **orthogonal similarity transforms**.

- Under certain conditions, \( A_k \) converges to **Schur form of \( A \)**: \[ A = QTQ^* \]
  \( T \) triangular
Simultaneous Iteration

\[ Q^{(0)} = I \]
\[ Z = A \cdot Q^{(k-1)} \]
\[ Z = Q^{(k)} \cdot R^{(k)} \]
\[ A^{(k)} = (Q^{(k)})^T \cdot A \cdot Q^{(k)} \]

Unshifted QR Algorithm

\[ A^{(0)} = A \]
\[ A^{(k-1)} = Q^{(k)} \cdot R^{(k)} \]
\[ A^{(k)} = R^{(k)} / Q^{(k)} \]
\[ Q^{(k)} = Q^{(1)} \cdot Q^{(2)} \cdot \ldots \cdot Q^{(k)} \]

\[ R^{(k)} = R^{(k)} \cdot R^{(k-1)} \cdot \ldots \cdot R^{(1)} \]

\[ R^{(k)}, Q^{(k)}, \text{ and } A^{(k)} \] equivalent, and

\[ A^{(k)} = Q^{(k)} \cdot R^{(k)} \]

Proof: induction in \( k \)

\[ k = 0 \]

SI: \[ A^{(0)} = Q^{(0)} \cdot R^{(0)} = I \]
\[ A^{(0)} = A \]

QR: \[ A^{(0)} = Q^{(0)} \cdot R^{(0)} = I \]
\[ A^{(0)} = A \]

\[ k \geq 1 \]

SI: \[ A^{(k)} = Q^{(k)} \cdot R^{(k)} \]
\[ A^{(k)} = A \cdot A^{(k-1)} = A \cdot Q^{(k-1)} \cdot R^{(k-1)} = Q^{(k)} \cdot R^{(k)} \]
\[ A^{(k)} = Q^{(k)} \cdot R^{(k)} \]

QR: \[ A^{(k)} = A \cdot A^{(k-1)} = A \cdot Q^{(k-1)} \cdot R^{(k-1)} = Q^{(k)} \cdot R^{(k)} \]
\[ A^{(k)} = Q^{(k)} \cdot A^{(k-1)} \cdot Q^{(k)} = Q^{(k)} \cdot A \cdot Q^{(k)} \]
Eigenvalues of tridiagonal $T$ by QR iteration

\[
\begin{align*}
T_0 &= T \\
T_0 &= Q_0 R_0 \\
T_1 &= R_0 Q_0 \\
T_2 &= R_1 Q_1 \\
T_k &= Q_k R_k \\
T_{k+1} &= R_k Q_k \\
\text{end}
\end{align*}
\]

Note:
- $T_{k+1}$ similar to $T_k$
- $T_k$'s all tridiagonal
- $T_k$'s converging to diagonal matrix $\Lambda$

Accelerated convergence: use shifts

Choose shift $S_k$

\[
T_k - S_k I = Q_k R_k \\
T_{k+1} = R_k Q_k + S_k I
\]

Shifted QR achieves cubic convergence
Note: we still have the condition.

$T_{k+1}$ is similar to $T_k$

\[ T_k - S_k I = Q_k R_k \]

\[ \Rightarrow T_{lc} = Q_{lc} R_k + S_k I \]

\[ \Rightarrow R_k = Q_k^T T_k - Q_k^T S_k \]

\[ T_{lc+1} = R_k Q_k + S_k I \]

\[ = (Q_k^T T_k - Q_k^T S_k) Q_k + S_k I \]

\[ = Q_{lc}^T T_{lc} Q_{lc} - S_k Q_{lc} Q_{lc} Q_{lc} + S_k I \]

\[ = Q_{lc}^T T_{lc} Q_{lc} \checkmark \]
Upper Hessenberg form via Householder

Since we are trying to preserve the eigenvalues, want similarity transform.

\[ H_1 A H_1^T \]

does not work

\[ H_1 A = \begin{pmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \end{pmatrix}, \quad H_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \overline{H_1} \end{pmatrix} \]

\[ (H_1 A) H_1^T = \begin{pmatrix} x & x & x \\ x & x & x \\ 0 & 0 & x \end{pmatrix} \]

\[ H_2 = \begin{pmatrix} I_2 & 0 \\ 0 & \overline{H_2} \end{pmatrix}, \quad \ldots, \quad H_{n-2} = \begin{pmatrix} I_{n-2} & 0 \\ 0 & \overline{H_{n-2}} \end{pmatrix} \]

\[ H_{n-2} H_2 H_1 A H_1^T H_2^T \ldots H_{n-2}^T = H \]

\[ Q A Q^T = H \]

\[ A = Q^T H Q \]
**Krylov Subspaces and Arnoldi Iteration**

**Krylov vectors**

\[ b, Ab, A^2 b, \ldots \]

**Krylov Subspace**

\[ K_r = \text{span} \{ b, Ab, \ldots, A^{r-1} b \} \]

first \( r \) Krylov vectors

not generally orthogonal, so use Gram-Schmidt to orthogonalize.

This is the **Arnoldi Iteration**.

After iteration \( k \), we have

\[ AQ_k = Q_{k+1} H_{k+1,k} \]

Multiply both sides by \( Q_k^T \), we get

\[ Q_k^T A Q_k = Q_k^T Q_{k+1} H_{k+1,k} \]

\[ = \begin{bmatrix} I_{k \times k} & \mathbf{0}_{k \times 1} \end{bmatrix} H_{k+1,k} \]

\[ = H_{k} \quad \text{(first \( k \) rows of \( H_{k+1,k} \))} \]

\[ H_k = Q_k^T A Q_k \]

projection of \( A \) onto \( k \text{th} \) Krylov space.
Arnoldi Iteration

\[ q_1 = b / \| b \| \]

\[ A q_1 \rightarrow q_2 \]

After iteration \( k \), \( q_1, q_2, \ldots, q_k \)

\[ v = A q_k \]

Orthogonalize w.r.t. to \( q_1, \ldots, q_k \)

\[ v \leftarrow v - (q_j^T v) q_j \quad j = 1, \ldots, k \]

Normalize

\[ q_{k+1} = v / \| v \| \]

\[ A q_k = h_{1k} q_1 + h_{2k} q_2 + \cdots + h_{kk} q_k + h_{k+1,k} q_{k+1} \]

\[ A q_k = \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} h_{1k} \\ h_{2k} \\ \vdots \\ h_{k+1,k} \end{bmatrix} \]

Af
\[ A_{q_1} = \begin{pmatrix} a_1 & a_2 \\ \hline h_2 \\ 1 \end{pmatrix} \begin{pmatrix} h_{11} \\ \hline h_{21} \end{pmatrix} \]

\[ A_{q_2} = \begin{pmatrix} a_1 & a_2 & a_3 \\ \hline h_2 & h_3 \\ 1 \end{pmatrix} \begin{pmatrix} h_{12} \\ \hline h_{22} \end{pmatrix} \]

\[ \vdots \]

\[ A_{q_k} \]

\[ \begin{pmatrix} A_{q_1} & A_{q_2} & \ldots & A_{q_k} \end{pmatrix} = \begin{pmatrix} a_1 & a_2 & \ldots & a_k \\ \hline h_2 & h_3 & \ldots & h_k \end{pmatrix} \begin{pmatrix} h_{11} & h_{12} & \ldots & h_{1k} \\ \hline h_{21} & h_{22} & \ldots & h_{2k} \end{pmatrix} \]

\[ A Q_{k+1} = Q_{k+1} H_{k+1, k} \]

\[ Q_{k+1}^T A Q_{k+1} = Q_{k+1}^T Q_{k+1} H_{k+1, k} \]

\[ = [I \ \overline{0}] H_{k+1, k} \]

\[ = [I \ \overline{0}] \begin{pmatrix} h_{11} & h_{12} & \ldots & h_{1k} \\ \hline h_{21} & h_{22} & \ldots & h_{2k} \end{pmatrix} \]

\[ Q_{k+1}^T A Q_{k+1} = H_k \]
Arnoldi Iteration

In each iteration

\[ k \]

\[ (q_1, q_2, \ldots, q_k \text{ are known}) \]

\[ v = A q_k \]

for \( j = 1, \ldots, k \)

\[ h_{jk} = q_j^T v \]

\[ v = v - h_{jk} q_j \]

\[ h_{k+1,k} = \| v \| \]

\[ q_{k+1} = v / h_{k+1,k} \]

\[ A q_1 = h_{11} q_1 + h_{21} q_2 \]

\[ \Rightarrow A q_k = h_{1k} q_1 + \cdots + h_{kk} q_k + h_{k+1,k} q_{k+1} \]

In matrix form, we are computing this factorization:

\[
\begin{bmatrix}
A
\end{bmatrix}
\begin{bmatrix}
q_1 & \cdots & q_k
\end{bmatrix} =
\begin{bmatrix}
q_1 & \cdots & q_k
\end{bmatrix}
\begin{bmatrix}
h_{11} & & h_{1k} \\
0 & \ddots & \vdots \\
0 & & h_{kk}
\end{bmatrix}
\]

\[ A Q_k = Q_{k+1} H_{k+1,k} \]

\[ \text{upper Hessenberg matrix} \]
\[ A g_k = h_{1k} g_1 + h_{2k} g_2 + \ldots + h_{kk} g_k + h_{k+1, k} g_{k+1} \]

\[ h_{k+1, k} g_{k+1} = A g_k - h_{1k} g_1 - h_{2k} g_2 - \ldots - h_{kk} g_k \]
The Arnoldi iteration is computing
\[ H_k = Q_k^T A Q_k \]
If we continue until \(k = \text{size of } A\), we have
\[ H = Q^T A Q \]
a Hessenberg matrix similar to \(A\).
It therefore has the same eigenvalues.

In practice, we don't continue that far, but stop for some \(m \ll k\). The eigenvalues of \(H_k\) are usually good approximations to the extreme eigenvalues of \(A\).
Symmetric Matrices

\[ A = S \]

1. Then \( H_k = Q_k^T S Q_k \) is also symm.
2. \( H_k \) is tridiagonal
   only 1 orthogonalization is needed
   in the Arnoldi iteration!

Lanczos iteration

\[ q_0 = 0, \quad q_1 = b / \| b \| \]

for \( k = 1, 2, 3, \ldots \)

\[ \nu = S q_k \]

\[ a_k = q_k^T \nu \]

\[ \nu = \nu - b_{k-1} q_{k-1} - a_k q_k \]

\[ b_k = \| \nu \| \]

\[ q_{k+1} = \nu / b_k \]

\[
\begin{pmatrix}
S & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & S
\end{pmatrix}
\begin{pmatrix}
q_1 \\
\vdots \\
q_k
\end{pmatrix}
= 
\begin{pmatrix}
b_1 \\
\vdots \\
b_k
\end{pmatrix}
\begin{pmatrix}
q_1 \\
\vdots \\
q_k
\end{pmatrix}
\begin{pmatrix}
q_{k+1} \\
\vdots \\
q_k
\end{pmatrix}
\begin{pmatrix}
A_1 \\
A_2 \\
\vdots \\
A_{k-1}
\end{pmatrix}
\begin{pmatrix}
b_1 \\
b_2 \\
\vdots \\
b_{k-1}
\end{pmatrix}
\begin{pmatrix}
q_{k+1} \\
\vdots \\
q_k
\end{pmatrix}
\begin{pmatrix}
A_1 \\
A_2 \\
\vdots \\
A_{k-1}
\end{pmatrix}
\begin{pmatrix}
b_1 \\
b_2 \\
\vdots \\
b_{k-1}
\end{pmatrix}
\]
\[ T_k = Q_k^T S Q_k \]
\[ S Q_k = Q_{k+1} T_{k+1, k} \]

Lanczos algorithm presented above is unstable numerically.

Lanczos compared with Householder tridiagonalization:
- Lanczos takes advantage of sparsity; Householder has fill-in.
- Lanczos uses \( A \) as a black box.
- Each iteration of Lanczos produces the Householder produces factor \( H_{k} \) of \( Q \).
- Householder is stable.