# QR Iteration

$A_0 = A$

for $k = 0, 1, 2, \ldots$

$\quad A_k = Q_k R_k$      QR decomposition of $A_k$

$\quad A_{k+1} = R_k Q_k$

end

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 $A$: $\quad A = QTQ^*$
  $T$ triangular

## Simultaneous Iteration

$$\underline{Q}^{(0)} = I$$
$$Z = A\,\underline{Q}^{(k-1)}$$
$$Z = \underline{Q}^{(k)} R^{(k)}$$
$$A^{(k)} = (\underline{Q}^{(k)})^T A\, \underline{Q}^{(k)}$$

## Unshifted QR Algorithm

$$A^{(0)} = A$$
$$A^{(k-1)} = Q^{(k)} R^{(k)}$$
$$A^{(k)} = R^{(k)} Q^{(k)}$$
$$\underline{Q}^{(k)} = Q^{(1)} Q^{(2)} \cdots Q^{(k)}$$

$$\underline{R}^{(k)} = R^{(k)} R^{(k-1)} \cdots R^{(1)}$$

---

* $\underline{R}^{(k)}$, $\underline{Q}^{(k)}$, and $A^{(k)}$ equivalent, and
$A^{k} = \underline{Q}^{(k)} \underline{R}^{(k)}$, $A^{(k)} = \underline{Q}^{(k)T} A\, \underline{Q}^{(k)}$

---

Proof. induction in $k$

$\boxed{k=0}$

SI: $A^0 = \underline{Q}^{(0)} = \underline{R}^{(0)} = I$, $A^{(0)} = A$

QR: $A^0 = \underline{Q}^{(0)} = \underline{R}^{(0)} = I$ $A^{(0)} = A$

$\boxed{k \geq 1}$

SI: $A^{(k)} = \underline{Q}^{(k)T} A\, \underline{Q}^{(k)}$ ✓

$A^{k} = A A^{k-1} = A\, \underline{Q}^{(k-1)} \underline{R}^{(k-1)} = \underline{Q}^{(k)} R^{(k)} \underline{R}^{(k-1)}$

$\Rightarrow$ $A^{k} = \underline{Q}^{(k)} \underline{R}^{(k)}$ ✓

QR: ✓ $A^{k} = A A^{k-1} = A\, \underline{Q}^{(k-1)} \underline{R}^{(k-1)} = \underline{Q}^{(k-1)} A^{(k-1)} \underline{R}^{(k-1)} = \underline{Q}^{(k)} \underline{R}^{(k)}$

$A^{(k)} = Q^{(k)T} A^{(k-1)} Q^{(k)} = \underline{Q}^{(k)T} A\, \underline{Q}^{(k)}$ ✓

# Eigenvalues of tridiagonal T by QR iteration

1. $T = T_0$

2. $T_0 = Q_0 R_0$

   $T_1 = R_0 Q_0$

   $T_1 = Q_1 R_1$

   $T_2 = R_1 Q_1$

   $\vdots$

$$\boxed{\begin{array}{l} T_0 = T \\ \text{for } k = 0, 1, 2, \ldots \\ \quad T_k = Q_k R_k \\ \quad T_{k+1} = R_k Q_k \\ \text{end} \end{array}}$$

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

$$\boxed{\begin{array}{l} \text{Choose shift } s_k \\ T_k - s_k I = Q_k R_k \\ T_{k+1} = R_k Q_k + s_k I \end{array}}$$

shifted QR

shifted QR achieves cubic convergence

Note: we still have the condition

$T_{k+1}$ is <u>similar</u> to $T_k$

$$T_k - s_k I = Q_k R_k$$

$$\Rightarrow \quad T_k = Q_k R_k + s_k I$$

$$\Rightarrow \quad R_k = Q_k^T T_k - Q_k^T s_k$$

$$\begin{aligned} T_{k+1} &= R_k Q_k + s_k I \\ &= \left(Q_k^T T_k - Q_k^T s_k\right) Q_k + s_k I \\ &= Q_k^T T_k Q_k - s_k Q_k^T Q_k + s_k I \\ &= Q_k^T T_k Q_k \quad \checkmark . \end{aligned}$$

# Upper Hessenberg form via Householder

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

$$H_1 A = \begin{pmatrix} x & x & x \\ 0 & x & x \\ 0 & x & x \end{pmatrix}$$

red entries changed by mult by $H_1$

$H_1 A H_1^T$ doesn't work

$$H_1 A = \begin{pmatrix} x & x & x \\ x & x & x \\ 0 & x & x \end{pmatrix}, \quad H_1 = \begin{pmatrix} 1 & \\ & \overline{H}_1 \end{pmatrix}$$

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

$$H_2 = \begin{pmatrix} I_2 & \\ & \overline{H}_2 \end{pmatrix}, \ldots, \quad H_{n-2} = \begin{pmatrix} I_{n-2} & \\ & \overline{H}_{n-2} \end{pmatrix}$$

$$\underbrace{H_{n-2} \cdots H_2 H_1}_{Q} \, A \, \underbrace{H_1^T H_2^T \cdots H_{n-2}^T}_{Q^T} = H$$

$$Q \; A \; Q^T = H$$

$$A = Q^T H Q$$

# Krylov Subspaces and Arnoldi Iteration

<u>Krylov vectors</u>

$$b,\ Ab,\ A^2 b,\ \ldots$$

<u>Krylov subspace</u>

$$K_r = \text{span}\{\underbrace{b,\ Ab,\ \ldots,\ A^{r-1}b}_{\text{first } r \text{ Krylov vectors}}\}$$

not generally orthogonal, so use <u>Gram-Schmidt</u> to <u>orthogonalize</u>.

This is the <u>Arnoldi Iteration</u>.

After iteration $k$, we have

$$A Q_k = Q_{k+1} H_{k+1,k}$$

Multiply both sides by $Q_k^T$, we get

$$\begin{aligned} Q_k^T A Q_k &= Q_k^T Q_{k+1} H_{k+1,k} \\ &= \begin{bmatrix} I_{k\times k} & \vec{0}_{k\times 1} \end{bmatrix} H_{k+1,k} = \\ &= H_k \quad \text{(first } k \text{ rows of } H_{k+1,k}) \end{aligned}$$

$$H_k = Q_k^T A Q_k$$

projection of $A$ onto $k^{th}$ Krylov space.

# Arnoldi Iteration

$$q_1 = b / \|b\|$$

$$A q_1 \rightsquigarrow q_2$$

after iteration $k$, $q_1, q_2, \ldots, q_k$

$$v = A q_k$$

<u>orthogonalize</u> w.r.t. to $q_1, \ldots, q_k$

$$v \leftarrow v - \underbrace{(q_j^T v)}_{= h_{jk}} q_j \qquad j = 1, \ldots, k$$

<u>normalize</u>

$$q_{k+1} = v / \underbrace{\|v\|}_{= h_{k+1,k}}$$

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

$$A q_k = \begin{bmatrix} | & | & & | \\ q_1 & q_2 & \cdots & q_{k+1} \\ | & | & & | \end{bmatrix} \begin{bmatrix} h_{1k} \\ h_{2k} \\ \vdots \\ h_{kk} \\ h_{k+1,k} \end{bmatrix}$$

$$\cancel{A[}$$

$$A q_1 = \begin{pmatrix} | & | \\ q_1 & q_2 \\ | & | \end{pmatrix} \begin{pmatrix} h_{11} \\ h_{21} \end{pmatrix}$$

$$A q_2 = \begin{pmatrix} | & | & | \\ q_1 & q_2 & q_3 \\ | & | & | \end{pmatrix} \begin{pmatrix} h_{12} \\ h_{22} \\ h_{32} \end{pmatrix}$$

$$\vdots$$

$$A q_k$$

$$\begin{pmatrix} | & | & & | \\ Aq_1 & Aq_2 & \cdots & Aq_k \\ | & | & & | \end{pmatrix} = \begin{pmatrix} | & | & & | \\ q_1 & q_2 & \cdots & q_{k+1} \\ | & | & & | \end{pmatrix} \begin{pmatrix} h_{11} & h_{12} & \cdots & h_{1k} \\ h_{21} & h_{22} & & h_{2k} \\ & h_{32} & \ddots & \\ & & \ddots & h_{kk} \\ & & & h_{k+1,k} \end{pmatrix}$$

$$A Q_k = Q_{k+1} H_{k+1,k}$$

$$Q_k^T A Q_k = \underset{k \times k}{Q_k^T} \underset{k \times (k+1)}{Q_{k+1}} \underset{(k+1) \times k}{H_{k+1,k}}$$

$$= \underset{k \times (k+1)}{\begin{bmatrix} I & \vec{0} \end{bmatrix}} \underset{(k+1) \times k}{H_{k+1,k}}$$

$$\boxed{Q_k^T A Q_k = H_k}$$

$$q_1 = \frac{b}{\|b\|}$$

$$v = Aq_1$$

$$h_{11} = q_1^T v$$

$$v \leftarrow v - h_{11} q_1$$

$$h_{21} = \|v\|$$

$$q_2 = v / h_{21}$$

## Arnoldi Iteration

in each iteration $k$:

($q_1 = \frac{b}{\|b\|}, q_2, \cdots, q_k$ are known)

$v = A q_k$

for $j = 1, \ldots, k$

$\quad h_{jk} = q_j^T v$

$\quad v \leftarrow v - h_{jk} q_j$

$h_{k+1,k} = \|v\|$

$q_{k+1} = v / h_{k+1,k}$

$$\Rightarrow A q_1 = h_{11} q_1 + h_{21} q_2$$

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

In matrix form, we are computing this factorization:

$$\begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix} \begin{bmatrix} | & & | \\ q_1 & \cdots & q_k \\ | & & | \end{bmatrix} = \begin{bmatrix} | & & | \\ q_1 & \cdots & q_{k+1} \\ | & & | \end{bmatrix} \begin{bmatrix} h_{11} & \cdots & h_{1k} \\ h_{21} & \cdots & \vdots \\ & \ddots & \vdots \\ & & h_{k+1,k} \end{bmatrix}$$

upper Hessenberg matrix

$$A Q_k = Q_{k+1} H_{k+1,k}$$

$$A q_k = h_{1k} q_1 + h_{2k} q_2 + \dots + h_{kk} q_k + h_{k+1\,k} q_{k+1}$$

$$h_{k+1,k} q_{k+1} = A q_k - h_{1k} q_1 - h_{2k} q_2 - \dots - h_{kk} q_k$$

# Eigenvalues from Arnoldi

The Arnoldi iteration is computing

$$H_k = Q_k^T A Q_k$$

If we continue until $k =$ 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 $k$. 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 $k = 1, 2, 3, \dots$

$v = S q_k$

$a_k = q_k^T v$

$v = v - b_{k-1} q_{k-1} - a_k q_k$

$b_k = \|v\|$

$q_{k+1} = v / b_k$

$$\begin{pmatrix} & & \\ & S & \\ & & \end{pmatrix} \begin{pmatrix} | & & | \\ q_1 & \cdots & q_k \\ | & & | \end{pmatrix} = \begin{pmatrix} | & & | \\ q_1 & \cdots & q_{k+1} \\ | & & | \end{pmatrix} \begin{pmatrix} a_1 & \boxed{b_1} & & \\ b_1 & \boxed{a_2} & b_2 & \\ & b_2 & \ddots & b_{k-1} \\ & & \ddots & a_k \\ & & & b_k \end{pmatrix}$$

$$T_k = Q_k^T S Q_k$$

$$S Q_k = Q_{k+1} T_{k+1,k}$$

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 $q_k$
  Housholder produces factor $H_k$ of $Q$

- Householder is stable