|WII - PI|

|M 504|
F18

Computation of All Eigenvalues

Similarity Transformation

Let $S \in \mathbb{C}^{n \times n}$ be invertible.

$$A \longmapsto SAS^{-1}$$

$$\begin{aligned}
\det(SAS^{-1} - \lambda I) &= \det(S(A-\lambda I)S^{-1}) \\
&= \det(S)\det(A-\lambda I)\det(S^{-1}) \\
&= \det(SS^{-1})\det(A-\lambda I) \\
&= \det(A-\lambda I)
\end{aligned}$$

A and SAS^{-1}

(i) have the same characteristic polynomial,

(ii) have the same set of eigenvalues.

Unitary Similarity Transformation

Let $Q \in \mathbb{C}^{n \times n}$ be unitary.

$$A \longmapsto QAQ^{*}$$

Computing all eigenvalues, Overview

① Reduction into Hessenberg form

Form unitary $Q \in \mathbb{C}^{n \times n}$ s.t.

$$Q^* A Q = H$$

is Hessenberg. (Midterm Q4, based on Householder reflectors # of flops $\sim 10n^3/3$

② The QR Algorithm

Form unitary $Q_1, \ldots, Q_k \in \mathbb{C}^{n \times n}$ s.t.

$$H_k := Q_k^* \cdots Q_1^* H Q_1 \cdots Q_k$$

satisfies

$$\lim_{k \to \infty} H_k = T$$

for some upper triangular matrix $T \in \mathbb{C}^{n \times n}$.

Observations

(i) $A, H, H_k \; \forall k, T$ have same eigenvalues.

(ii) T is upper triangular; its eigenvalues are on its diagonal.

The QR Algorithm

Generate a sequence $\{H_k\}$ s.t.

$H_0 = H$,

H_k and H_{k+1} are related as follows:

(i) Compute a QR factorization

$$H_k - \sigma_k I = Q_{k+1} R_{k+1}$$

for some $\sigma_k \in \mathbb{C}$.

(ii) $H_{k+1} := R_{k+1} Q_{k+1} + \sigma_k I$

Observations

① H_k $\forall k$ are similar

$$H_k - \sigma_k I = Q_{k+1} R_{k+1}$$

$$\Longrightarrow R_{k+1} = Q_{k+1}^* (H_k - \sigma_k I)$$

$$\Longrightarrow H_{k+1} = Q_{k+1}^* (H_k - \sigma_k I) Q_{k+1} + \sigma_k I$$

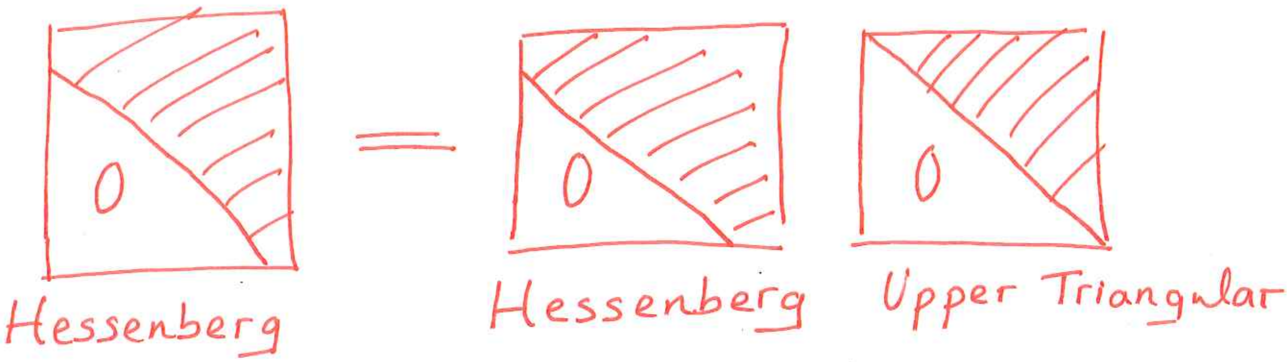
$$= Q_{k+1}^* H_k Q_{k+1}$$

$$\Longrightarrow H_{k+1} = Q_{k+1}^* \cdots Q_1^* H Q_1 \cdots Q_{k+1}$$

② $H_k \; \forall k$ are Hessenberg

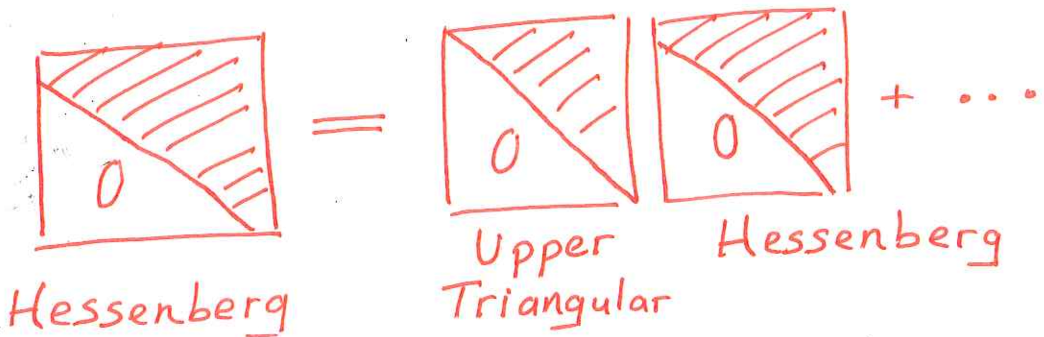
Suppose H_k is Hessenberg

$$Q_{k+1} = (H_k - \sigma_k I)\, R_{k+1}^{-1}$$



Q_{k+1} is Hessenberg

$$H_{k+1} = R_{k+1} Q_{k+1} + \sigma_k I$$



* The QR factorization

$$\underbrace{(H_k - \sigma_k I)}_{\text{Hessenberg}} = Q_{k+1} R_{k+1}$$

can be computed by performing $O(n^2)$ flops.

* The product

$$\underbrace{R_{k+1}}_{\text{upper triangular}} \underbrace{Q_{k+1}}_{\text{Hessenberg}}$$

can be computed by performing $O(n^2)$ flops.

Specification of the overall algorithm

Compute a Hessenberg $H \in \mathbb{C}^{n\times n}$ s.t.

$$H = Q^* A Q$$

for some unitary $Q \in \mathbb{C}^{n\times n}$.

if H is 1×1 or 2×2

$\Lambda \leftarrow$ compute eigenvalues of H using algebraic formulas

Return Λ

else

Repeat

(+) Choose a shift σ

Compute a QR factorization

$$H - \sigma I = QR$$

$H \leftarrow RQ + \sigma I$

If H is of the form $H = \begin{bmatrix} H_1 & H_{12} \\ 0 & H_2 \end{bmatrix}$ for some $H_1 \in \mathbb{C}^{k\times k}$, $H_2 \in \mathbb{C}^{(n-k)\times(n-k)}$ with $k \in [1, n-1]$

$\Lambda_1 \leftarrow$ Apply (*) on H_1

$\Lambda_2 \leftarrow$ Apply (*) on H_2

$\Lambda \leftarrow \begin{bmatrix} \Lambda_1 \\ \Lambda_2 \end{bmatrix}$

Return Λ

end (if)

end (repeat)

end

(*)

Deflation

Choice of shift (in (+))

Rayleigh shift

$$\sigma = H(n,n)$$

Wilkinson shift

$$\sigma = \text{eigenvalue of } H(n-1:n,\ n-1:n) \text{ closest to } H(n,n)$$

Why the QR Algorithm Works

Only the unshifted version, $\sigma = 0$.

power iteration

$q^{(0)}$ — unit vector

$$q^{(k+1)} = \frac{H q^{(k)}}{\| H q^{(k)} \|_2}$$

$$\lambda_k = [q^{(k)}]^* H q^{(k)}$$

simultaneous power iteration

$\tilde{Q}_0$ — matrix with orthonormal columns

$$\underbrace{\tilde{Q}_{k+1} \tilde{R}_{k+1}}_{\text{Reduced QR factor.}} = H \tilde{Q}_k$$

$$H_k = \tilde{Q}_k^* H \tilde{Q}_k$$

Assume $\tilde{Q}_k$ is square

Relate H_k and H_{k+1}

$$H_k = \tilde{Q}_k^* H \tilde{Q}_k = \underbrace{\tilde{Q}_k^* \tilde{Q}_{k+1}}_{\hat{Q}_{k+1} \text{ - unitary}} \tilde{R}_{k+1}$$

$$H_{k+1} = \tilde{Q}_{k+1}^* H \tilde{Q}_{k+1} = \tilde{R}_{k+1} \underbrace{\tilde{Q}_k^* \tilde{Q}_{k+1}}_{\hat{Q}_{k+1} \text{ - unitary}}$$

$$\left(\hat{Q}_{k+1} = \tilde{Q}_k^* \tilde{Q}_{k+1} \implies \tilde{Q}_{k+1} = \tilde{Q}_k \hat{Q}_{k+1} \right)$$

Simultaneous power iteration with $\tilde{Q}_0 = I_n$ is equivalent to the QR algorithm (without shifts)

In particular,

(i) both generate the same sequence $\{H_k\}$

(ii) $\tilde{Q}_k = \underbrace{\hat{Q}_1 \hat{Q}_2 \cdots \hat{Q}_k}$

by simultaneous power iteration

by QR algorithm ($H_k = \hat{Q}_{k+1} R_{k+1}$ $H_{k+1} = R_{k+1}\hat{Q}_{k+1}$)

Convergence of the simultaneous power iteration

① $\tilde{Q}_k$ generated by the simultaneous power iteration satisfies

$$(+) \quad H^k = \tilde{Q}_k \hat{R}_k$$

for some upper triangular $\hat{R}_k$ (assuming $\tilde{Q}_0 = I_n$)

<u>Proof</u> (by induction on k)

k=1 (base case)

$$H = H\tilde{Q}_0 = \tilde{Q}_1 \tilde{R}_1$$

inductive case

Assume

$$H^k = \tilde{Q}_k \hat{R}_k$$

for an arbitrary k.

Then

$$H^{k+1} = H\tilde{Q}_k \hat{R}_k$$
$$= \tilde{Q}_{k+1} \underbrace{\tilde{R}_{k+1} \hat{R}_k}_{\hat{R}_{k+1}}$$

where $\hat{R}_{k+1}$ is upper triangular. $\square$

(2) Under mild assumptions

$$\mathrm{Col}(H^k(:,1:j)) \longrightarrow \mathrm{span}\{v^{(1)}, \dots, v^{(j)}\}$$
$$j = 1, \dots, n$$

It follows from (+) that, assuming H is invertible,

$$\mathrm{Col}(\tilde{Q}_k(:,1:j)) = \mathrm{Col}(H^k(:,1:j))$$
$$j = 1, \dots, n$$

$$\boxed{\begin{array}{l}\text{Conclusion from (3)} \\ H_k = \tilde{Q}_k^* H \tilde{Q}_k \\ \text{becomes upper triangular} \\ \text{as } k \to \infty\end{array}}$$

(3) For j, p s.t. $p > j$

$$\begin{cases} H\tilde{Q}_k(:,j) \in \mathrm{span}\{v^{(1)}, \dots, v^{(j)}\} \\ \quad \text{as } k \to \infty \end{cases}$$

and

$$\begin{cases} \mathrm{span}\{v^{(1)}, \dots, v^{(j)}\} \perp \tilde{Q}_k(:,p) \\ \quad \text{as } k \to \infty \end{cases}$$

$$\Longrightarrow \tilde{Q}_k^*(:,p)^* H \tilde{Q}_k(:,j) = [\tilde{Q}_k^* H \tilde{Q}_k]_{pj} \to 0$$
$$\text{as } k \to \infty$$