

Last time: eigenvalue problems

$A \in \mathbb{C}^{n \times n}$, find $\lambda \in \mathbb{C}$, $v \in \mathbb{C}^n$ s.t.

$$Av = \lambda v \quad (v \neq 0)$$

- Power iteration with Rayleigh quotient

Given $x_0 \in \mathbb{C}^n$

For $k = 1, 2, 3, \dots$

$$\hat{x}_k = Ax_{k-1}$$

$$x_k = \frac{\hat{x}_k}{\|\hat{x}_k\|_2}$$

$$m_k = R(x_k) := x_k^* A x_k$$

When $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots$ or $\lambda_1 = \lambda_2 = \dots = \lambda_r$ but $|\lambda_1| > |\lambda_{r+1}|$

$$|m_k - \lambda_1| \approx \alpha |m_{k-1} - \lambda_1| \quad \alpha = \frac{|\lambda_2|}{|\lambda_1|} \text{ or } \frac{|\lambda_{r+1}|}{|\lambda_1|}$$

No convergence in general when $|\lambda_1| = |\lambda_2|$

- Rayleigh Quotient iteration

Given $x_0 \in \mathbb{C}^n$, $\mu \in \mathbb{C}$

For $k = 1, 2, 3, \dots$

$$m_k = R(x_k)$$

$$\text{Solve } (A - m_k I) \hat{x}_k = x_{k-1}$$

$$x_k = \frac{\hat{x}_k}{\|\hat{x}_k\|_2}$$

Almost always globally converges for normal A

then $|m_k - \lambda_j| \approx O(|m_k - \lambda_j|^3)$ locally cubically

- Simultaneous power iteration

Given $Q_0 \in \mathbb{C}^{n \times n}$

For $k = 1, 2, 3, \dots$

$$X_k = A Q_{k-1}$$

Compute QR fact. $X_k = Q_k R_k$

$$T_k = Q_k^* A Q_k$$

When $Q_k \rightarrow Q$, as $k \rightarrow +\infty$,

$$\text{we know } T_k = Q_k^* A Q_k = Q_k^* A Q_{k-1} (Q_{k-1}^* Q_k) \approx R_k$$

thus $T_k \rightarrow T = Q^* A Q$ as $k \rightarrow +\infty$, the Schur form of A

Convergence is guaranteed when $|\lambda_1| > |\lambda_2| > \dots > |\lambda_n|$

the i^{th} eig. val. $\rightarrow |\lambda_i^k - \lambda_i| \approx \frac{|\lambda_{i+1}|}{|\lambda_i|} |\lambda_i^{k-1} - \lambda_i|$

lower Δ entries $\rightarrow |(T_k)_{ij}| \approx \frac{|\lambda_{j+1}|}{|\lambda_j|} |(T_{k-1})_{ij}|, \forall i > j$

- QR iteration

Given $Q_0 \in \mathbb{C}^{n \times n}$, $T_0 = Q_0^* A Q$

For $k = 1, 2, 3, \dots$

$$\begin{aligned} T_{k-1} &= Q_{k-1}^* A Q_{k-1} \\ &= (Q_{k-1}^* Q_k) R_k \\ T_k &= Q_k^* A Q_k \\ &= Q_k^* A (Q_{k-1} Q_{k-1}^* Q_k) \\ &= Q_k^* Q_k R_k Q_{k-1}^* Q_k \\ &= R_k (Q_{k-1}^* Q_k) \end{aligned}$$

Compute QR fact. $T_{k-1} = Q_k R_k \leftarrow \frac{4}{3}n^3$ (complex) flops

$$T_{k+1} = R_k Q_k$$

$\leftarrow n^3$ (complex) flops

Reduce cost? Faster convergence?

Today: Practical QR iteration

Idea: Reduce A to simpler form before applying QR

- Two phase algorithm:

Step 1: Transform A to an upper Hessenberg matrix.

i.e. $h_{ij} = 0$ for $i > j+1$

$$H = \begin{bmatrix} x & x & x & \cdots & x \\ x & x & x & \cdots & x \\ x & x & \cdots & x \\ \ddots & \ddots & \vdots \\ x & x \end{bmatrix} \in \mathbb{C}^{n \times n}$$

This is possible by applying unitary transform to A

i.e. find $U \in \mathbb{C}^{n \times n}$, unitary, $U^*AU = H$

The diagram illustrates the reduction of matrix A to an upper Hessenberg matrix U^*AU through a series of unitary transformations U_i . Matrix A is shown with a red dashed box around its top-left corner. A pink shaded region indicates the part of A that remains unchanged during the process. The first step shows A being transformed by U_1^* (indicated by a blue arrow) into U_1^*A (indicated by a blue arrow). The second step shows U_1^*A being transformed by U_2 (indicated by a green arrow) into $U_1^*AU_2$ (indicated by a green arrow). The third step shows $U_1^*AU_2$ being transformed by U_3 (indicated by an orange arrow) into $U_1^*AU_2U_3$ (indicated by an orange arrow). The final result is an upper Hessenberg matrix U^*AU with a pink shaded region indicating the unchanged part.

The diagram illustrates the iterative reduction of matrix A to an upper Hessenberg matrix H using multiple Householder transforms. It shows the transformation $A \rightarrow U_1^*AU_1$ (indicated by a blue arrow), followed by $U_1^*AU_1 \rightarrow U_1^*AU_1U_2$ (indicated by a green arrow), and then $U_1^*AU_1U_2 \rightarrow U_1^*AU_1U_2U_3$ (indicated by a green arrow). The final result is an upper Hessenberg matrix H with a green dashed box around its top-left corner, labeled U^*AU .

— If A is non-Hermitian, A is reduced to upper Hessenberg

A is Hermitian, A is reduced to tridiagonal

— Total cost: $\approx 2 \cdot \frac{4}{3} n^3$ (complex) flop for general A

$\approx \frac{4}{3} n^3$ (complex) flop for Hermitian A
(it suffices to work on lower triangular part)

Step 2: Hessenberg QR iteration

QR iteration applied to H is significantly faster than A !

- Compute $H = QR$

We only need to zero out one subdiagonal entry each time

$$H = \begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{bmatrix} \xrightarrow{Q_1^T} \begin{bmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{bmatrix} \xrightarrow{Q_2^T} \begin{bmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 1 & x & x & x \\ 0 & x & x & x & x \\ x & x & x & x & x \end{bmatrix} \Rightarrow \dots$$

$R = Q_{n-1}^T \cdots Q_1^T H$

$Q_1^T H$ $Q_2^T Q_1^T H$

F_1, F_2, \dots, F_{n-1} are 2×2 Householder transform

Note that Q_k^T only changes k^{th} and $k+1^{th}$ row

For general A (H is upper Hessenberg), cost of $H = QR \approx 3n^2$ (complex) flops

For Hermitian A (H is tri-diagonal), cost of $H = QR \approx 6n$ (complex) flops

- Compute RQ

Apply Q in the last step to R get RQ , but are the nice properties (upper Hessenberg / tridiagonal) preserved?

$$R = \begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{bmatrix} \xrightarrow{\cdot Q_1} \begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{bmatrix} \xrightarrow{\cdot Q_2} \begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{bmatrix} \Rightarrow \dots R Q_1 \dots Q_{n-1} = H$$

Note that Q_k only changes k^{th} and $k+1^{\text{th}}$ columns.

When A is general, PQ is still upper Hessenberg !

(Hermitian)

(triangular)

Cost of QR step: for general A , $\approx 3n^2$ (complex) flops

Hermitian A , $\approx 6n$ (complex) flops

Total cost of step 2 $\approx \# \text{iterations} \cdot 6n^2$ flops (general)

..... $\cdot 12n$ flops (Hermitian)

Note: By choosing suitable shifting, convergence of QR iteration is usually very fast (in a few iterations), so the dominate cost comes from step 1 $\approx O(n^3)$

Note: Apply QR iter. to a Hessenberg matrix. the p^{th} subdiagonal entry in H converges to zero with linear rate $\frac{|h_{p+1}|}{|h_p|}$

- Deflation

In practice, when a subdiagonal entry in H is sufficiently small,

for example, $|h_{p+1,p}| \leq c \cdot \text{Emach} (|h_{pp}| + |h_{p+1,p+1}|)$

We can justifiably set it to be zero, (b.c. comparable to rounding error)

In this case, we can decouple the problem into two small problems:

suppose we have at some step reduced H to block upper Δ

$$H = \begin{bmatrix} H_{11} & H_{12} \\ 0 & H_{22} \end{bmatrix}_{n-p}^p, \quad 1 \leq p < n$$

we may proceed with computing Schur form for submatrices:

$$H_{11} = Q_1 T_{11} Q_1^*, \quad H_{22} = Q_2 T_{22} Q_2^*$$

then

$$H = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} \begin{bmatrix} T_{11} & T_{12} \\ & T_{22} \end{bmatrix} \begin{bmatrix} Q_1^* \\ Q_2^* \end{bmatrix} \quad \text{with } T_{12} := Q_1^* H_{12} Q_2$$

is the Schur fact. of H

If this happens when $p=n-1$ or $n-2$ we call it deflation.

- How to incorporate shifting in QR?

If apply QR directly to $A - \mu I$. i.e.

Given $Q_0 \in \mathbb{C}^{n \times n}$ unitary, Let $T_0 := Q_0^* (A - \mu I) Q_0$.

For $k = 1, 2, 3, \dots$

Compute QR fact. $T_{k-1} = Q_k R_k$ } Problem: How to
 $T_k = R_k Q_k$ change μ during iter. ?

Instead, let $H_k := T_k + \mu I$.

Shifted QR Iteration:

Given $Q_0 \in \mathbb{C}^{n \times n}$ unitary. Let $H_0 := Q_0^* A Q_0$.

For $k = 1, 2, 3, \dots$

Determine a scalar $\mu_k \in \mathbb{C}$

Compute QR fact. $H_{k-1} - \mu_k I = Q_k R_k$

$H_k = R_k Q_k + \mu_k I$

If we order eig. val. of A so that

$$|\lambda_1 - \mu_k| \geq \dots \geq |\lambda_n - \mu_k|$$

then the p^{th} subdiagonal entry in H converges to zero

with rate $\frac{|\lambda_{p+1} - \mu_k|}{|\lambda_p - \mu_k|}$

In the extreme case, when μ is an eig. val of A ,
we get the exact eig. val. in a single step:

Then let μ be an eig. val. of a Hessenberg matrix H

with all $h_{i+1,i} \neq 0$, $i = 1, 2, \dots, n-1$ ← called "unreduced".

Then after a single shifted QR step,

we have $h_{n,n-1} = 0$ and $h_{nn} = \mu$

Pf: Since $H - \mu I = QR$ and $H - \mu I$ is singular,

we know $r_{11} \cdots r_{nn} = 0$.

Since H is unreduced, the first $n-1$ columns of H is

linearly independent. thus $r_{ii} \neq 0$ if $i = 1, \dots, n-1$

$$\Rightarrow r_{nn} = 0$$

$$\Rightarrow \mu I + RQ = \begin{bmatrix} * & * \\ 0 & \mu \end{bmatrix} \quad \boxed{\text{}}$$

- How to choose μ_k ?

For simplicity, we assume A is Hermitian.

Option 1: Rayleigh quotient shift

$$Q_K = [q_K^{(1)} \cdots q_K^{(n)}]$$

$$\mu_k := R(q_K^{(n)}) = q_K^{(n)*} A q_K^{(n)} = (T_K)_{n,n}$$

Recall $Q_K^* H Q_K = T_K$

Rayleigh quotient shift gives (local) cubic convergence

in generic case (global convergence and cubic local convergence can fail in some corner cases)

ex. $A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

$$\mu = 0, \text{ QR fact. } A = QR = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$RQ = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = A$$

Option 2: Wilkinson's shift

If $H = \begin{bmatrix} a_1 & b_1 & \dots & 0 \\ b_1 & a_2 & \dots & \vdots \\ \vdots & \ddots & \ddots & a_{n-1} & b_{n-1} \\ 0 & \dots & b_{n-1} & a_n \end{bmatrix}$

Wilkinson's shift choose μ to be eigenvalue of

$$T(n-1:n, n-1:n) = \begin{bmatrix} a_{n-1} & b_{n-1} \\ b_{n-1} & a_n \end{bmatrix}$$

and set μ to be the eig. val. of $T(n-1:n, n-1:n)$
that is closer to a_n .

$$\mu = a_n + d - \text{sign}(d) \sqrt{d^2 + b_{n-1}^2}$$

$$= a_n - \frac{\text{sign}(d) b_{n-1}^2}{|d| + \sqrt{d^2 + b_{n-1}^2}} \quad (\text{numerically stable})$$

with $d = (a_{n-1} - a_n) / 2$

If is guaranteed that Wilkinson's shift always converges
(globally)
and at least locally quadratically, and almost always
cubically.

Option 3: (bulge chasing) choose μ based on largest
off-diagonal entry of $H \dots \dots$

others : If stalled, perturb the shift to break the cycle ...

Other problems:

1) Perform QR iter. in real number ?

Schur fact. is not true in real number

Complex eig. val. how to choose real shifts ?
(Rayleigh Quotient Shift is poor in this case)

Double - Implicit - Shift strategy (Francis QR)

↑
two successive shifts ↑
combine QR fact.

and RQ in a single step (good when $\mu \gg a_{ii}$
for some i)

2) Compute selected eig. vectors ?

3) A lot others ...

• Computing the SVD

Goal : $A \in \mathbb{C}^{m \times n}$

Find unitary $U \in \mathbb{C}^{m \times m}$, $V \in \mathbb{C}^{n \times n}$.

$\Sigma = \text{diag}(\sigma_1, \dots, \sigma_{\min(m,n)}) \in \mathbb{R}^{m \times n}$

such that $A = U \Sigma V^*$

Naive algorithm

Step 1: Compute $C = A^*A$

Step 2: Apply QR iteration to C

Bad idea because : 1) Forming A^*A is costly

2) Information is lost in A^*A

ex. $A = \begin{bmatrix} 1 & 1 \\ \frac{1}{\sqrt{\text{mach}}} & 0 \\ 0 & \frac{1}{\sqrt{\text{mach}}} \end{bmatrix}$ ($\kappa_2(A) = \sqrt{\frac{\text{mach}+2}{\text{mach}}} \rightarrow 1$)

compute
on
floating
point system

$$f((A^*A)) = f(\begin{bmatrix} 1 & \frac{1}{\sqrt{\text{mach}}} \\ \frac{1}{\sqrt{\text{mach}}} & 0 \\ 0 & \frac{1}{\sqrt{\text{mach}}} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & \frac{1}{\sqrt{\text{mach}}} \\ 0 & 0 \end{bmatrix})$$

$$= \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad \text{singular!}$$

Mimicking the two-phase method of QR iteration,
the standard SVD solver for dense A is the following

Golub-Kahan Method

Idea: Implicitly apply QR iteration to A^*A

Step 1 Reduce A to upper bidiagonal form

Goal : Apply different unitary matrix on left and right of A

$$A = \begin{bmatrix} x & x & x & x \\ x & x & x & x \end{bmatrix} \xrightarrow{U_i^* \cdot F_i} \begin{bmatrix} x & x & x & x \\ 0 & x & x & x \end{bmatrix} \xrightarrow{\cdot V_i \quad [F_2]} \begin{bmatrix} x & x & 0 & 0 \\ 0 & x & x & x \end{bmatrix}$$

$$\xrightarrow{U_2^* \cdot \begin{bmatrix} I_2 \\ F_3 \end{bmatrix}} \begin{bmatrix} x & x & 0 & 0 \\ 0 & x & x & x \\ 0 & 0 & x & x \end{bmatrix} \xrightarrow{\cdot V_2 \quad [I_3 \quad F_4]} \begin{bmatrix} x & x & 0 & 0 \\ 0 & x & x & 0 \\ 0 & 0 & x & x \end{bmatrix} \xrightarrow{\dots} \begin{bmatrix} x & x & 0 & 0 \\ 0 & x & x & 0 \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$U_2^* U_1^* A V_1 \quad U_2^* U_1^* A V_1 V_2 \quad U_4^* U_3^* U_2^* U_1^* A V_1 V_2 \\ =: B$$

Work for Groeb-Kahan bidiagonalization

\approx twice of QR fact.

$\approx 2(2mn^2 - \frac{2}{3}n^3)$ complex flops

Remark: Cost can be reduced by first computing
(A-SVD)
(Lawson-Hanson-Chan) QR fact. of $A = QR$, then perform bidiagonalization
to R. This is saving when $m \gg n$.
Bidiagonalization

Step 2 : Apply QR iter. to $B^T B$ bidiagonal form

Again, forming $B^T B$ is not a wise choice from numerical standpoint.

Need to apply QR and RD step implicitly.

Details are omitted (Implicit Q theorem....)

Mature algorithms can be found in LAPACK