Last time: eigenvalue problems

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

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

- Power iteration with Rayleigh quotient

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

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

$$\hat{x}_k = A x_{k-1}$$

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

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

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

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

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

- Rayleigh Quotient iteration

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

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

$$m_k = R(x_k)$$

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 $k = 1, 2, 3, \cdots$

  $X_k = A Q_{k-1}$

  Compute QR fact. $X_k = Q_k R_k$

  $T_k = Q_k^* A Q_k$

when $Q_k \to Q$, as $k \to +\infty$,

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

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

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

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

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

- QR iteration

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

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

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

  $T_{k+1} = R_k Q_k$ ← n^3 (complex) flops

Reduce cost? Faster convergence?

$T_{k-1} = Q_{k-1}^* A Q_{k-1} = (Q_{k-1}^* Q_k) R_k$

$T_k = Q_k^* A Q_k = Q_k^* A Q_{k-1} Q_{k-1}^* Q_k = Q_k^* Q_k R_k Q_{k-1}^* Q_k = R_k (Q_{k-1}^* Q_k)$

Today: Practical QR iteration

Idea: Reduce A to simpler form before applying QR

- Two phase algorithm:

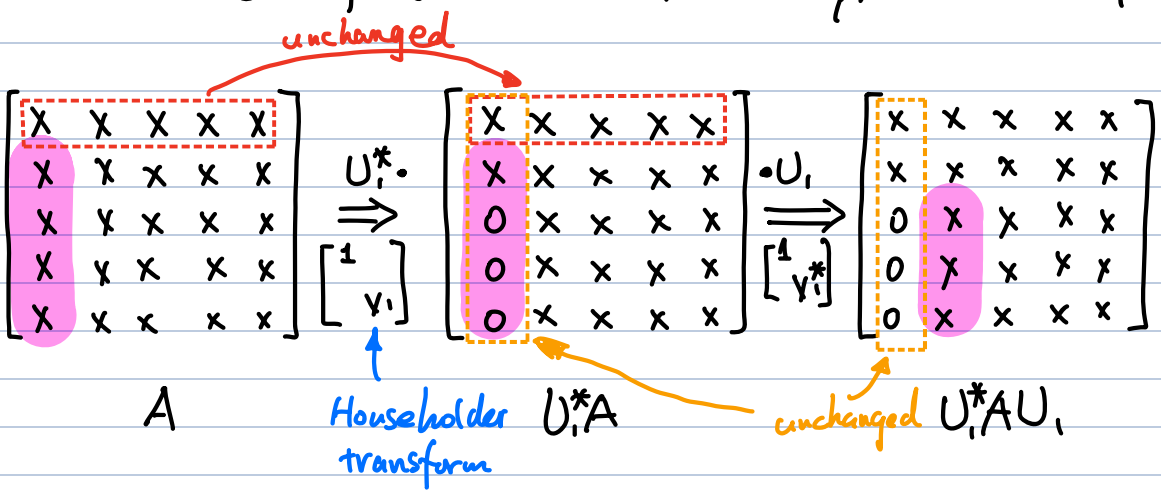
Step 1: Transform A to an upper Hessenberg matrix.

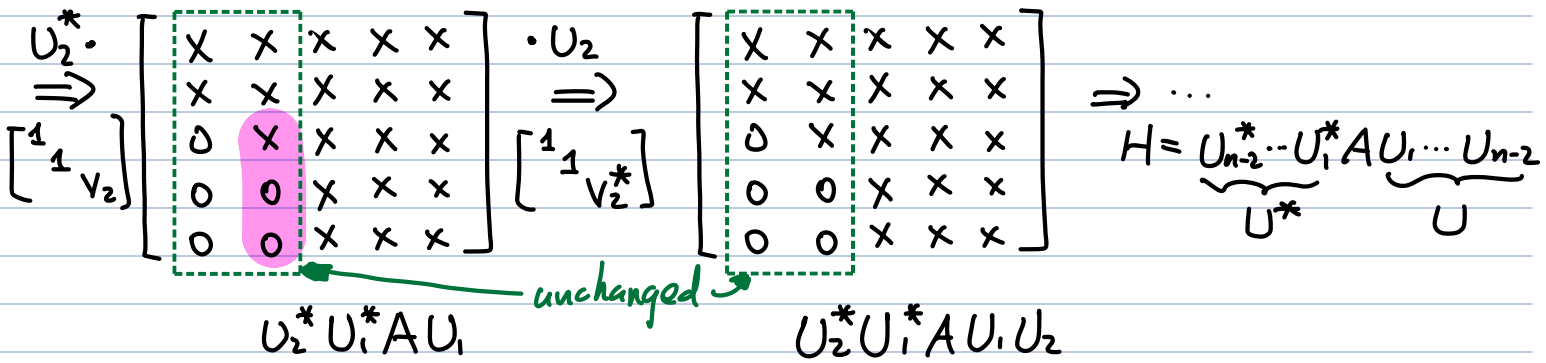
i.e. $h_{ij} = 0$ for $i > j+1$

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

This is possible by applying unitary transform to A

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





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

A is Hermitian, A is reduced to tridiagonal

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

$\approx \frac{4}{3} n^3$ (complex) flop for Hermitian A

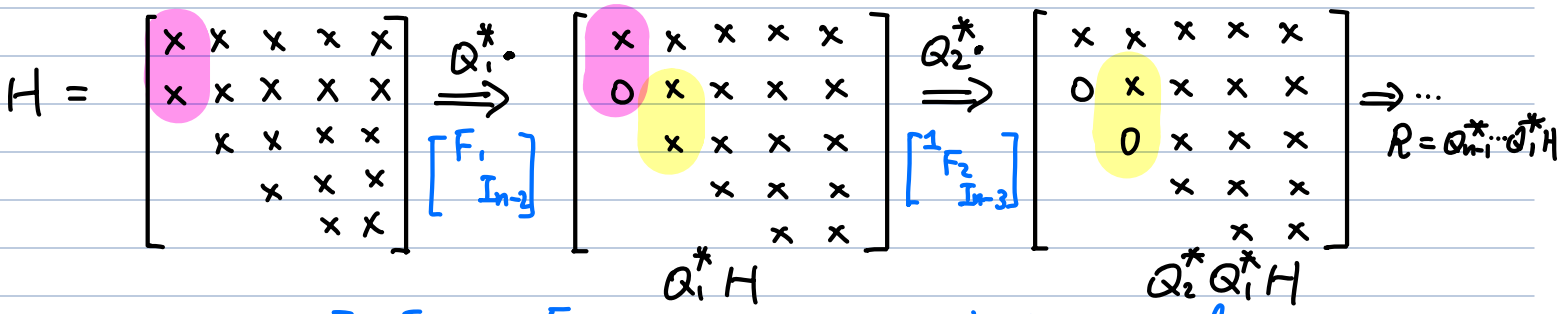
(it suffices to work on lower triangular part)

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



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

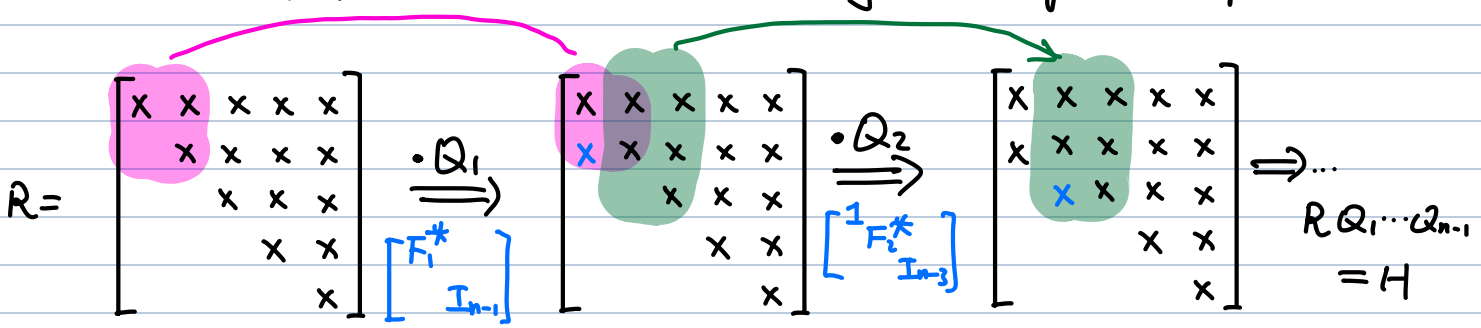
Note that Q_k^* only changes k^{th} and $k+1^{th}$ row

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

For Hermitian A (H is tridiagonal), 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?



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

When A is general, RQ is still upper Hessenberg!

(Hermitian) (tridiagonal)

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

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

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

$\cdots\cdots\cdots\cdots\cdot$ $12n$ flops ($\overset{\text{Hermitian}}{A}$)

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 $\frac{|\lambda_{p+1}|}{|\lambda_p|}$

- Deflation

In practice, when a subdiagonal entry in H is sufficiently small, for example, $|h_{p+1,p}| \leq C\,\varepsilon_{mach}\,(|h_{pp}| + |h_{p+1,p+1}|)$

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

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

$\qquad\quad p \qquad n-p$

we may proceed with computing Schur form for submatrices:

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

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 <u>deflation.</u>

- How to incorperate 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 $k = 1, 2, 3, \cdots$

Compute QR fact. $T_{k-1} = Q_k R_k$

$T_k = R_k Q_k$

} Problem: How to change μ during iter. ?

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

<u>Shifted QR Iteration :</u>

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

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

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

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

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:

Thm Let μ be an eig. val of a Hessenberg matrix H with all $h_{i+1,i} \neq 0$, $i = 1, 2, \cdots, n-1$ ← called "unreduced". If this is not true, we can decouple the problem

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 $r_{ii} \neq 0 \quad \forall i = 1, \cdots, n-1$

$\Rightarrow r_{nn} = 0$

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

- How to choose μ_k ?

For simplicity, we assume A is Hermitian.

Option 1: Rayleigh quotient shift

$Q_k = [\, q_k^{(1)} \cdots q_k^{(n)} \,]$

Recall $Q_k^* H Q_k = T_k$

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

Rayleigh quotient shift gives (local) cubic convergence

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

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

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

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.

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

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

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

It is gauranteed 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 .....

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 sucessive shifts

combine QR fact. and RQ in a single step (good when $\mu \Rightarrow a_{ii}$ 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 $U \in \mathbb{C}^{m \times m}$, $V \in \mathbb{C}^{n \times n}$.

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

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. $A = \begin{bmatrix} 1 & 1 \\ \sqrt{\varepsilon_{mach}} & 0 \\ 0 & \sqrt{\varepsilon_{mach}} \end{bmatrix}$ $\left(\kappa_2(A) = \sqrt{\frac{\varepsilon_{mach}+2}{\varepsilon_{mach}}} \to 1 \right)$

compute on floating point system → $fl((A^*A) = fl\left(\begin{bmatrix} 1 & \sqrt{\varepsilon_{mach}} & 0 \\ 1 & 0 & \sqrt{\varepsilon_{mach}} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ \sqrt{\varepsilon_{mach}} & 0 \\ 0 & \sqrt{\varepsilon_{mach}} \end{bmatrix} \right)$

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

Remark: The relative perturbation of the smallest eigenvalue of A^*A can be amplified by $\kappa(A)^2$ in forming A^*A

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

$A = \begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix} \xrightarrow{U_1^* \cdot} \begin{bmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & x & x & x \\ 0 & x & x & x \\ 0 & x & x & x \\ 0 & x & x & x \end{bmatrix} \xrightarrow{\cdot V_1} \begin{bmatrix} x & x & 0 & 0 \\ 0 & x & x & x \\ 0 & x & x & x \\ 0 & x & x & x \\ 0 & x & x & x \\ 0 & x & x & x \end{bmatrix}$

F_1 $\quad \begin{bmatrix} 1 & \\ & F_2 \end{bmatrix}$

U_1^*A $\qquad U_1^*AV_1$

$\xrightarrow{U_2^* \cdot} \begin{bmatrix} x & x & 0 & 0 \\ 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & x & x \\ 0 & 0 & x & x \\ 0 & 0 & x & x \end{bmatrix} \xrightarrow{\cdot V_2} \begin{bmatrix} x & x & 0 & 0 \\ 0 & x & x & 0 \\ 0 & 0 & x & x \\ 0 & 0 & x & x \\ 0 & 0 & x & x \\ 0 & 0 & x & x \end{bmatrix} \Rightarrow \cdots \Rightarrow \begin{bmatrix} x & x & 0 & 0 \\ 0 & x & x & 0 \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

$\begin{bmatrix} I_2 & \\ & F_3 \end{bmatrix}$ $\quad \begin{bmatrix} I_3 & \\ & F_4 \end{bmatrix}$

$U_2^*U_1^*AV_1$ $\qquad U_2^*U_1^*AV_1V_2$ $\qquad U_4^*U_3^*U_2^*U_1^*AV_1V_2 =: B$

Work for Golub-Kahan bidiagonalization

$\approx$ twice of QR fact.

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

Remark: Cost can be reduced by first computing QR fact. of $A = QR$, then perform bidiagonalization to R. This is saving when $m \gg n$.

(R-SVD)
(Lawson-Hanson-Chan Bidiagonalization)

Step 2: Apply QR iter. to B^*B ← bidiagonal form

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

Need to apply QR and RQ step implicitly.

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

Mature algorithms can be found in LAPACK