Welcome to Ma221! (Mar 17)

power method ⇒ orthogonal iteration
⇒ QR iteration ⇒ add inverse iteration

Let's us converge to any eval for which we have an approximation (to be supplied by alg)

QR Iteration
Given $A_0 = A$
$i = 0$
repeat
  factor $A_i = Q_i R_i$
  $A_{i+1} = R_i Q_i$
  $i = i+1$
until convergence

$$A_{i+1} = R_i Q_i = Q_i^T \underbrace{Q_i R_i}_{A_i} Q_i = Q_i^T A_i Q_i$$

Thm A_i from QR iter. same as $Z_i^T A Z_i$ from orthogonal iter.
($Z_0 = I$, $AZ_i = Z_{i+1} R_{i+1}$: QR decomp)

⇒ A_i converges to Schur form if evals all have different absolute values

proof (induction) assume $A_i = Z_i^T A Z_i$
One step of Orthog Iter:
$A_i Z_i = Z_{i+1} R_{i+1}$ ... QR decomp

$$A_i = Z_i^T A Z_i = \underbrace{Z_i^T Z_{i+1}}_{\text{orthog}} \underbrace{R_{i+1}}_{\text{tri}}$$

$= QR$ decomp of A_i by uniqueness

$$Z_{i+1}^T A Z_{i+1} = (Z_{i+1}^T A Z_i)(Z_i^T Z_{i+1})$$
$$= R_{i+1} \cdot (Z_i^T Z_{i+1})$$
$$= RQ$$
$$= A_{i+1}$$

Add inverse iteration

QR iteration with a shift

$A_0 = A$

$i = 0$

repeat

choose shift σ_i near an eval

factor $A_i - \sigma_i I = Q_i R_i$

$A_{i+1} = R_i Q_i + \sigma_i I$

$i = i+1$

until convergence

Lemma A_i and A_{i+1} are orthog. similar

pf: $A_{i+1} = R_i Q_i + \sigma_i I$

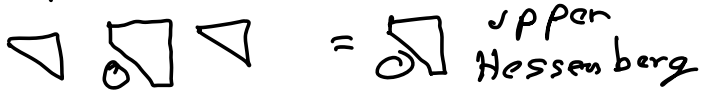
$$= Q_i^T Q_i R_i Q_i + \sigma_i I$$
$$= Q_i^T (A_i - \sigma_i I) Q_i + \sigma_i I$$
$$= Q_i^T A_i Q_i - \sigma_i Q_i^T Q_i + \sigma_i I$$

(cancel)

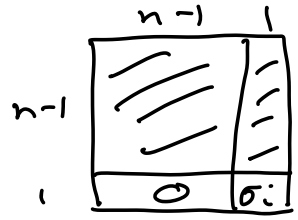
$$= Q_i^T A_i Q_i$$

Note: if R_i nonsingular we can also write

$$\begin{aligned} A_{i+1} &= R_i Q_i + \sigma_i I \\ &= R_i Q_i R_i R_i^{-1} + \sigma_i I \\ &= R_i (A_i - \sigma_i I) R_i^{-1} + \sigma_i I \\ &\qquad \text{cancel} \\ &= R_i A_i R_i^{-1} \end{aligned}$$



If σ_i exact eval of A,
QR with shift converges in one step
$A_i - \sigma_i I$ singular $\Rightarrow R_i$ singular
$\Rightarrow$ some $R_i(k,k) = 0$
Suppose $R_i(n,n) = 0$
$\Rightarrow$ last row of $R_i = 0$
$\Rightarrow$ last row of $R_i Q_i = 0$
$\Rightarrow$ last row of $A_{i+1} = R_i Q_i + \sigma_i I$
is zero except $A_{i+1}(n,n) = \sigma_i$



block upper triangular
algorithm proceeds
on leading $n-1 \times n-1$
submatrix

If σ_i not exact eval, declare
convergence if $\|A_{i+1}(n, 1:n-1)\|$
$= O(\varepsilon) \cdot \|A\|$
set it to zero, backward stable

Previous analysis: expect

$A_{i+1}(n, 1:n-1)$ to shrink by factor:

(λ_k = closest eval to shift σ_i)

$$\frac{|\lambda_k - \sigma_i|}{\min_{j \neq k} |\lambda_j - \sigma_i|}$$

from inverse iteration

(suppose eval real)

$$A_i - \sigma_i I = Q_i R_i$$

$$\Rightarrow Q_i^T (A_i - \sigma_i I) = R_i$$

$\Rightarrow$ if σ_i were exact eval

$R_i(n,n) = 0 \Rightarrow$

last row of $Q_i^T (A_i - \sigma_i I)$ is 0

$\Rightarrow$ last column of Q_i (left evec of A_i for eval σ_i

Now suppose σ_i just close to eval

$$A_i - \sigma_i I = Q_i R_i$$

$$(A_i - \sigma_i I)^{-1} = R_i^{-1} Q_i^T$$

$$(A_i - \sigma_i I)^{-T} = Q_i R_i^{-T}$$

$$(A_i - \sigma_i I)^{-T} R_i^T = Q_i$$

Δ

$(A_i - \sigma_i I)^{-T} \cdot e_n \cdot R_i(n,n) =$ last col of Q_i

$\Rightarrow$ inverse iteration starting with e_n to get last col of Q_i

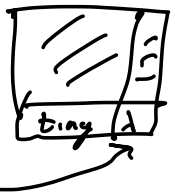
$\Rightarrow$ last col of Q_i closer to an evec of A_i^T

$\Rightarrow$ last col of $A_i^T Q_i$ closer to $\lambda \cdot$ last col of Q_i

$\Rightarrow$ last col $Q_i^T A_i^T Q_i$ closer to $\lambda \cdot e_n$

$\Rightarrow$ last row of $Q_i^T A_i Q_i$ closer to $\lambda\, e_n^T$

where do we get shift σ_i?



use $\sigma_i = A_i(n,n)$

$\Rightarrow$ Quadratic convergence

$\| A_i(n, 1:n-1) \|_2 = \epsilon << 1$

$| A_i(n,n) - \lambda |_2 = O(\epsilon)$

By last analysis, expect $\|A_i(n, 1:n-1)\|_2$ to get multiplied by

$$\frac{|\lambda_k - \sigma_i|}{\min_{j \neq k} |\lambda_j - \sigma_i|} = O(\epsilon)$$

$\Rightarrow \| A_{i+1}(n, 1:n-1) \|_2 = O(\epsilon^2)$

(matlab demo, see code in typed notes)

Making QR iteration practical

1) Each iteration costs 1 QR decomp + 1 mat mul $= O(n^3) \Rightarrow$ if we did $O(1)$ iterations per eval $\Rightarrow$ $O(n^4)$ cost total, want $O(n^3)$

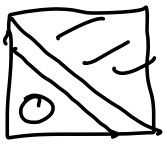
2) How to shift to converge to real Schur form?

3) How to decide convergence

4) How to minimize communication?

Answers:

(1) preprocess $A = Q H Q^T$ where Q orthog and H upper Hessenberg

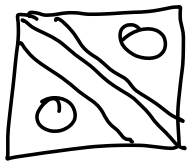


QR iteration on H keeps it upper Hessenberg, lowers cost to $O(n^2)$

$\Rightarrow$ total cost $= O(n^3)$

$A = A^T \Rightarrow H = H^T \Rightarrow H$ tridiagonal

$\Rightarrow$ lowers cost to $O(n)$



(Chap 5)

(2) Converge to real Schur form:

since evals of real matrices appear in pairs $\lambda, \bar{\lambda}$

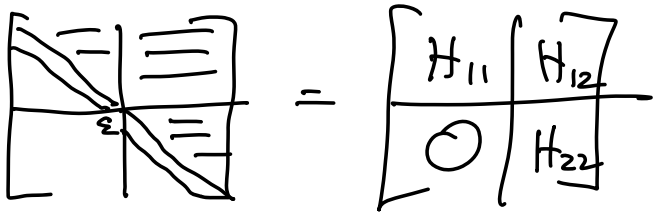
turns out one step of QR iteration with λ as shift followed by $\bar{\lambda}$ as shift keeps A_{i+2} real $\Rightarrow$ combine two

steps, no need to compute imaginary parts

(3) Detect convergence?

If any $H(i+1,i)$ small enough, $O(\varepsilon)\|A\|$, set it to zero

set ε to zero, problem splits into H_{11} and H_{22}



for example, Real matrix with complex evals, will split off 2x2 H_{22}
$\Rightarrow$ real Schur form

(4) Reducing communication:

no known way to attain $O\left(\frac{n^3}{\sqrt{\text{cachesize}}}\right)$ using deterministic alg, just randomized, but too expensive so far

More detail on Hessenberg QR

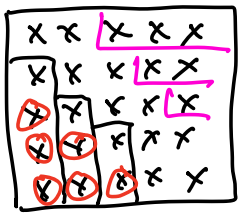
How to reduce $A = QHQ^T$

$H = Q^T A Q$

Analogous to QR with Householder transforms

zero if symmetric ⇒ tridiagonal

$$\begin{bmatrix} I_3 & \\ & H \end{bmatrix} \begin{bmatrix} I_2 & 0 \\ 0 & H^{3\times 3} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & H^{4\times 4} \end{bmatrix}$$



$$\begin{bmatrix} 1 & 0 \\ 0 & H^{4\times 4T} \end{bmatrix} \begin{bmatrix} I_2 & \\ & H^{3\times 3} \end{bmatrix} \begin{bmatrix} I_3 & \\ & H^{T} \end{bmatrix}$$

upper Hessenberg 

Cost: $\frac{10}{3}n^3 + O(n^2)$ just for H

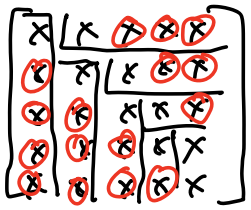
or $\frac{14}{3}n^3 + O(n^2)$ to get Q

much more than QR or LU already

(cheap part of overall alg)

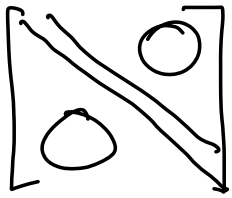
SVD similar: reduce to bidiagonal form

$$\begin{bmatrix} I^3 & \\ & H^{3\times 3} \end{bmatrix} \begin{bmatrix} I^2 & \\ & H^{3\times 3} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & H^{4\times 4} \end{bmatrix} H^{5\times 5}$$



$$\begin{bmatrix} 1 & \\ & H^{4\times 4} \end{bmatrix} \begin{bmatrix} I_2 & \\ & H^{3\times 3} \end{bmatrix} \begin{bmatrix} I_3 & \\ & H_{22} \end{bmatrix}$$

bidiagonal



← then compute SVD of this

QR Iteration on upper Hessenberg matrix in $O(n^2)$ flops

Lemma: upper Hessenberg preserved by QR iteration

pf: A upper Hess $\Rightarrow$ $A - \sigma I$ upper Hess

$\Rightarrow A - \sigma I = QR$, Q upper Hess

col i of Q_i = linear comb of cols $1:i$ of A

then $RQ =$  also upper Hess

How to do one step of QR iteration in $O(n^2)$ flops:

Def: H upper Hess <u>unreduced</u> if all $H(i+1,i) \neq 0$ (else split)

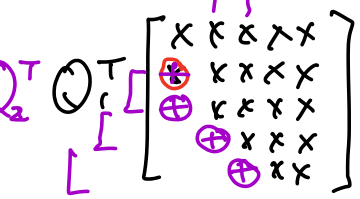
Implict Q Theorm: suppose $Q^T A Q$ upper Hess and unreduced. Then columns 2 through n of Q are uniquely determined by col 1 (up to factors ± 1)

1 step of QR in $O(n^2)$ flops!

$A - \sigma I = QR$, first col is $\begin{bmatrix} A(1,1)-\sigma \\ A(2,1) \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

Let Q_1 be Givens rotation s.t.

first col of $Q \rightarrow Q_1^T \begin{bmatrix} A(1,1)-\sigma \\ A(2,1) \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} x \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

$Q_4^T Q_3^T Q_2^T Q_1^T$  $Q_1 Q_2 Q_3 Q_4$ "bulge chasing" until + falls off bottom

cost = $O(n^2)$

Proof of implicit Q theorm:

let q_i be col i of Q

$Q^T A Q = H \Rightarrow AQ = QH$

Col 1: $A q_1 = H(1,1) \cdot q_1 + H(2,1) q_2$

$\Rightarrow$ determines $H(1,1)$, $H(2,1)$, q_2

via QR $[q_1, Aq_1] = [q_1, q_2] \cdot \begin{bmatrix} 1 & H(1,1) \\ 0 & H(2,1) \end{bmatrix}$

More generally:

Suppose we have $q_1, q_2, q_3 \cdots q_i$

and columns $1:i-1$ of H

Get next columns: $(AQ = QH)_i$

$q_j^T \left(Aq_i = \sum_{j=1}^{i+1} q_j H(j,i) \right)$

$q_j^T A q_i = H(j,i)$ for $j = 1:i$

$$A q_i - \sum_{j=1}^{i} q_j H(j,i) = q_{i+1} \cdot H(i+1,i)$$

gives us q_{i+1} and $H(i+1,i)$

used in LAPACK ~~xGEES~~

(Schur decomp) or xGEEV

(evals + evecs)

eig(A) in Matlab