Lecture 28

HW6

Q: Why is tri-diagonal form special?

- Are QR transforms $R_i Q_i = Q_i^T A Q_i$ related to λ-vect $A = V \Lambda V^T$?
- Explain more about ARPACK? How does it grow? Shrink?
- For subspace, how do we know cols of X will converge to λ-vecs.
- Is householder analog for tri-diag faster than diagonalization?
- Convergence of QR iter?
- geev for Julia + python on sym vs non-sym?
- Complex λ-vals?
- Why start w/ $X_0 = I$?
- Subspace over QR iter?
- Confusion on step in Subspace ⟺ QR.

Tridiag form

- Efficient, compact, simple

Why does it exist? David's "guess": it's hiding a set of orthogonal polys for ___?

$A = A^T \Rightarrow$ all λ-vals are Real!

$A \neq A^T \Rightarrow$ Some or all λ-vals may be complex.

$A^T = A \to$ tridiag (sym.)

$A^T \neq A \to \boxed{\text{bulge upper tri (non-sym).}}$

$A \neq A^T \to$

$A \to H$ (hessenberg / bulge upper tri)

$H \to R$ (upper triangular)

$A \to Q R Q^*$ (Q is orthog.) R is upper tri.

A is real $\Rightarrow$ you always get complex conj. λ-vals

So if $\lambda = \boxed{x+iy} \Rightarrow \lambda = \boxed{x-iy}$ too.

⇓ you can avoid complex arithmetic.

2x2 block matrix

$\begin{bmatrix} a & b \\ -b & a \end{bmatrix} \Rightarrow$

Why does subspace iteration converge?

$A = A^T \Rightarrow$ orthogonal λ-vecs.

If v_1 = largest λ-vec. Then if I do power method and insist all iters are orthogonal to v_1.

x_0 = random, $x_0 = (x_0 - v_1(x_0^T v_1))$

for $i = 1$ to ...

$x_i = A x_{i-1}$

$x_i = x_i - v_1(x_i^T v_1)$

QR ⟺ Subspace

for $i = 1$ to ___	for $i = 1$ to ___
$Q_i R_i = qr(A_i)$	$Y = A X_i$
$A_{i+1} = R_i Q_i$	$X_{i+1} R = qr(Y)$

$A_i = X_i^T A X_i$, $X_i = Q_1 Q_2 \cdots Q_i$

$Q_i^T A_i = R_i$ = upper-tri.

$Q_i^T X_i^T A X_i$

If X_i is converged, then $X_i^T A X_i = \Lambda$

$A = V \Lambda V^T$

$V^T A V = \Lambda$

Is it worth it doing the tridiag reduction? Yes!

1. $qr(\text{tridiag}) \Rightarrow$ $O(n)$ if you don't need Q explicitly.

$O(n^2)$ to build Q explicitly.

$qr(T_i) = \boxed{\bar{Q}_i \bar{R}_i}$

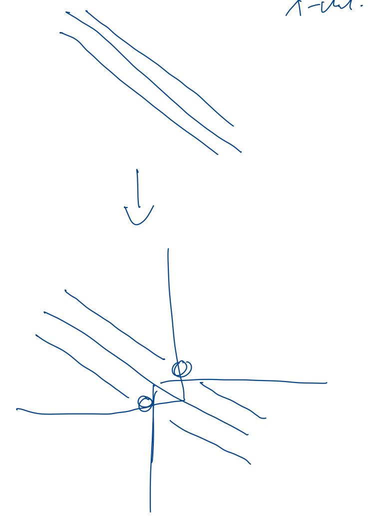
→ Compute $\underbrace{Q_1 \cdots Q_n \bar{Q}_i \Rightarrow Q}$

$O(n^2)$ time.

$\bar{R}_i \bar{Q}_i \Rightarrow T_{i+1}$

linear time ~~for most~~ ~~$O(n^2)$~~.

How many steps? Approx 5 per λ-val.



If you only want say 10–100 λ-vals, then use subsp.

↓

$Y = A X_i$, X_i is real, $X \in \mathbb{R}^{n \times k}$, k = small.

$Q = qr(Y)$

$M = Q^T A Q$, get $O(k^3)$

$M = V \Lambda V^T$ then use $X_{i+1} = QV$.

ARPACK

A: Only used as an operator Ax.

$A \xrightarrow{k+p}$ Steps of Lanczos w/ Reordering from b = rand. $V_{k+p}\ T_{k+p}$.

Then, it does an implicit restart

$V_{k+p}, T_{k+p} \to V_k, T_k$.

⟺ Lanczos of k steps start from b_2.

$V_k, T_k \to V_{k+p}, T_{k+p}$

$V_k, T_k \xleftarrow{b_3}$

Then compute λ-vals of T_k

↑ Should focus on the spectrum you want!