Last time: "plain" QR iteration $\Longleftrightarrow$ block power iteration

$$A_0 = A$$
for $k = 1, 2, \dots$
$$Q_k, R_k = qr(A_k)$$
$$A_{k+1} = R_k Q_k$$

Why is it slow?

1. $QR = O(n^3)$ cost per QR factorization $\rightarrow$ Upper Hessenberg
2. Can require many iterations $\rightarrow$ shifts.

1. Upper Hessenberg form: pre-factorize A ($O(n^3)$ cost)

$\Rightarrow$ $A = UHU^*$ via Householder reflectors

w/ $H_{ij} = 0$ if $i > j+1$

$$H = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \end{bmatrix}$$

Construction via Householder reflectors.

$QR(H) \rightarrow$ using Householder structure or Givens rotations (for matrices w/ sparsity)

$\rightarrow$ $O(n)$ cost for QR

Recall Householder reflectors for QR:

$$Q_1 A = \left[\begin{array}{c|ccc} x & x & x & x \\ \hline 0 & x & x & x \\ & x & x & x \\ & x & x & x \end{array}\right]$$

Want: $$\underline{Q_1 A}\, Q_1^* = \left[\begin{array}{c|c} x & x \\ \hline 0 & X \end{array}\right] Q_1^*$$

$$= \left(Q_1 \left[\begin{array}{c|c} x & 0 \\ \hline x & X \end{array}\right] \right)^*$$

$=$ dense matrix = useless

Define Q_1 via Householder: $$A = \left[\begin{array}{c} — a_1 — \\ \hline A_1 \end{array}\right]$$

Let $$\tilde{Q}_1 A_1 = \left[\begin{array}{c|cccc} x & x & x & x & x \\ \hline 0 & x & x & x & x \\ & x & x & x & x \end{array}\right]$$

$$\underbrace{\left[\begin{array}{c|c} 1 & \\ \hline & \tilde{Q}_1 \end{array}\right]}_{Q_1} A = \begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ & x & x & x & x \\ 0 & x & x & x & x \\ & x & x & x & x \end{bmatrix}$$

$$Q_1 A Q_1^* = \underbrace{\begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ & x & x & x & x \\ 0 & x & x & x & x \\ & x & x & x & x \end{bmatrix}}_{Q_1 A} \left[\begin{array}{c|c} 1 & \\ \hline & \tilde{Q}_1 \end{array}\right]$$

$$Q_1 A Q_1^* = \left[\begin{array}{c|cccc} x & x & x & x & x \\ x & x & x & x & x \\ \hline 0 & x & x & x & x \\ 0 & x & x & x & x \\ \vdots & x & x & x & x \\ 0 & x & x & x & x \end{array}\right]$$

Repeat $\Rightarrow Q_2 Q_1 A Q_1^* Q_2^* = \begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & x & x & x \end{bmatrix}$

$$\Rightarrow \underbrace{Q_{n-2} \cdots Q_1}_{U^*} A \underbrace{Q_1^* Q_2^* \cdots Q_{n-2}^*}_{U} = \begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ & x & x & x & x \\ 0 & & x & x & x \\ & & & x & x \end{bmatrix}$$

$$U^* A U = H \Rightarrow A = U H U^*$$

$$QR(H) = O(n^2)$$

$\Rightarrow$ cost of $QR = O(n^2) \times$ number of iters

Practical QR algorithm:

$H_0 = U^* A U$

for $k = 0, 1, 2, \ldots$

$\quad Q_k R_k = qr(H_k - \mu_k I)$

$\quad H_{k+1} = R_k Q_k + \mu_k I$

Note $R_k Q_k =$ retains upper Hessenberg form (see homework)

$Q_k =$ upper Hessenberg.

Practical QR $\Longleftrightarrow$ shifted inverse iter.

① Practical QR $\Longleftrightarrow$ inverse iteration w/ out shifts.

QR iter: constructs $A^k = \tilde{Q}_k \tilde{R}_k$ where

$$\tilde{Q}_k = Q_1 \cdots Q_k$$
$$\tilde{R}_k = R_1 \cdots R_k$$

(Q_i, R_i from QR iter)

QR iter also factorizes A^{-k} implicitly!

Assume $A = A^T$ for simplicity

$A^{-k} = \tilde{R}_k^{-1} \tilde{Q}_k^{*}$. Since $A = A^T$,

$(A^{-k})^T = \tilde{Q}_k \tilde{R}_k^{-T}$. Note $\tilde{R}_k^{-T}$ = lower tri. matrix since R^{-1} = upper triangular

$$P = \begin{bmatrix} & & & 1 \\ & & 1 & \\ & 1 & & \\ 1 & & & \end{bmatrix} = \text{reverses order}$$

Note $P^2 = I$.

$$(A^{-k})^T P = \underbrace{\tilde{Q}_k P}_{\text{still orthonormal}} \underbrace{P \, \tilde{R}_k^{-T} P}$$

$$= P \begin{bmatrix} x & & \\ x & x & \\ x & x & x \end{bmatrix} P$$

$$= \begin{bmatrix} x & x & x \\ x & x & \\ x & & \end{bmatrix} P$$

$$P \tilde{R}_k^{-T} P = \begin{bmatrix} x & x & x \\ & x & x \\ & & x \end{bmatrix} \text{ upper tri.}$$

$\Rightarrow$ QR iter also constructing a "permuted" QR factorization of A^{-k}

<u>Shifts:</u>

$$Q_k R_k = A_{k-1} - \mu_k I$$
$$A_k = R_k Q_k + \mu_k I$$

$$R_k = Q_k^* A_{k-1} - \mu_k Q_k^*$$

$$A_k = (Q_k^* A_{k-1} - \mu_k Q_k^*) Q_k + \mu_k I$$
$$= Q_k^* A_{k-1} Q_k - \mu_k \underbrace{Q_k^* Q_k}_{I} + \mu_k I$$

$$\Rightarrow A_k = Q_k^* A_{k-1} Q_k$$

Turns out shifts change $A^k = \tilde{Q}_k \tilde{R}_k$ to

$$\tilde{Q}_k \tilde{R}_k = (A - \mu_k I)(A - \mu_{k-1} I) \cdots (A - \mu_1 I)$$

By relation to inverse iteration,

$$(\tilde{Q}_k P)(P \tilde{R}_k P) = \prod_{j=1}^{k} (A - \mu_j I)^{-1}$$

Recall RQI: shifted inverse iter. with

$$\mu_k = \frac{v_k^* A v_k}{v_{k-1}^* v_{k-1}}. \quad v_k \to \text{eigen vector as } k \to \infty$$

How to mimic w/ QR? Recall $\tilde{Q}_k$ columns will converge to eigenvectors for $A = A^T$

Since $\tilde{Q}_k = Q_1 \cdots Q_k$

and $A_k = \tilde{Q}_k^* A \tilde{Q}_k$. If $A_k \to$ diagonal ($A = A^T$)

$$\Rightarrow A_k = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} = \tilde{Q}_k^* A Q_k$$

$$\Rightarrow Q_k (*) Q_k^* = A$$

Consider $q_m = m$th col. of $\tilde{Q}_k$, Want $q_m^* A q_m = \mu_k$

But $(A_k)_{mm} = (\tilde{Q}_k^* A \tilde{Q}_k)_{mm}$

$e_m = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leftarrow$ mth entry $\quad = e_m^* \tilde{Q}_k^* A \tilde{Q}_k e_m$

$= q_m^* A q_m$

$\Rightarrow$ Pick a diag entry of A_k as the shift.