5.5 QR-algorithm

Let A be an $n \times n$ (possibly complex) matrix. We want to find eigenvalues.

- Schur's Thm: A is unitarily similar to a triangular matrix.

$$\underset{\substack{\uparrow \\ \text{unitary}}}{U} A U^* = \underset{\substack{\uparrow \\ \text{triangular}}}{T}$$

QR-algorithm

$$A_1 = A \qquad A_k = \underset{\substack{\uparrow \\ \text{unitary}}}{Q_k} \underset{\substack{\uparrow \\ \text{upper-triangular} \\ \text{w/ } (R_k)_{ii} \geq 0}}{R_k} \qquad A_{k+1} = R_k Q_k$$

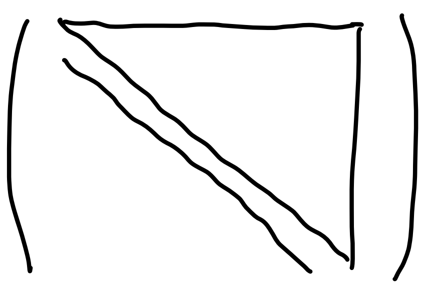
$$A_{k+1} = Q_k^* A_k Q_k$$

- Direct calculation of $Q_k R_k$ costs $O(n^3)$ in one iteration.

  Too expensive!

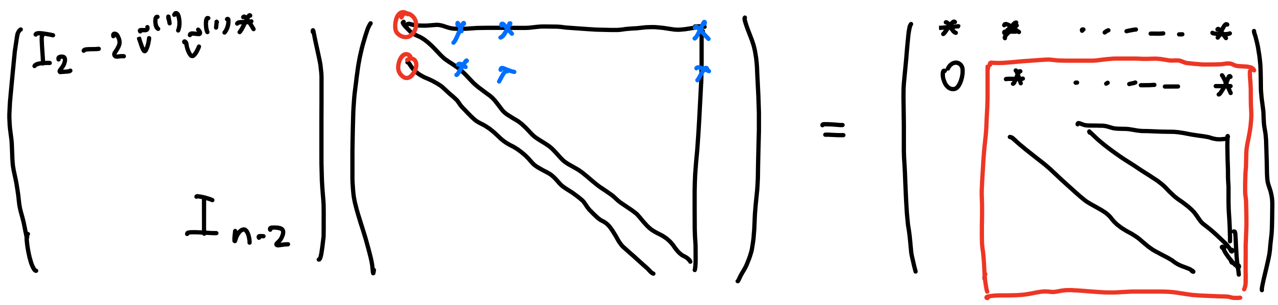
Reduction to upper Hessenberg form

Def An $n \times n$ matrix H is upper Hessenberg if $H_{ij} = 0 \ \forall\, i > j+1$



- QR-factorization for an upper Hessenberg matrix costs $O(n^2)$

$U^{(1)} H^{(1)} = H^{(2)}$

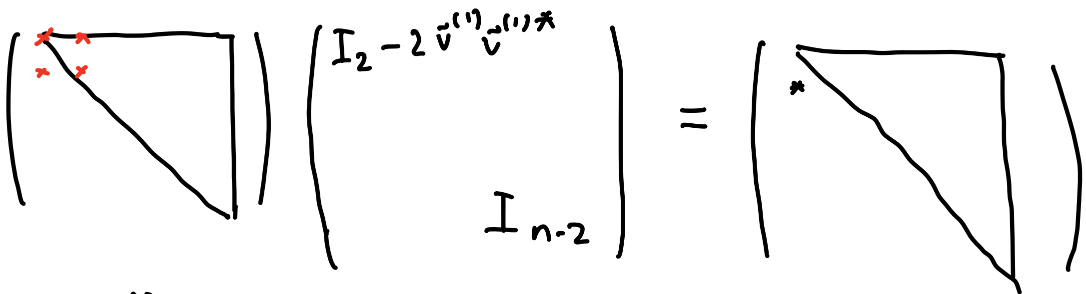


$$U^{(2)} = \begin{pmatrix} I_1 & & \\ & I_2 - 2\vec{v}^{(2)}\vec{v}^{(2)*} & \\ & & I_{n-3} \end{pmatrix}$$

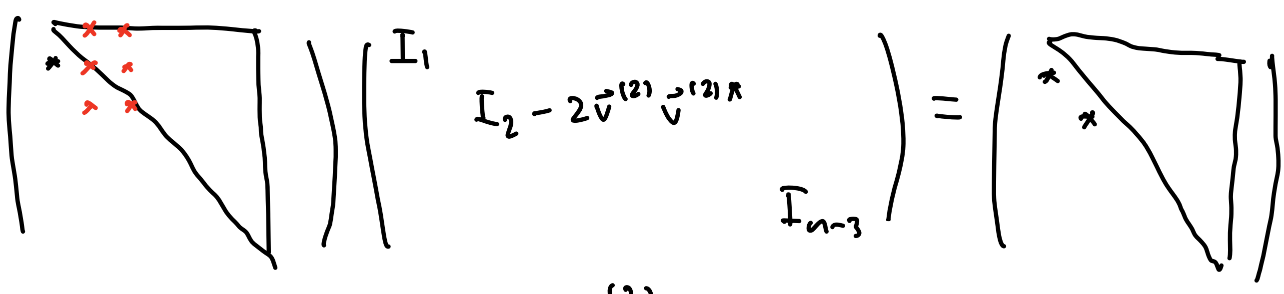
$\cdots$

Calculation of RQ also costs $O(n^2)$

RQ is also upper Hessenberg $\quad Q = U^{(1)} \cdots U^{(n-1)}$



$R \qquad U^{(1)}$



$RU^{(1)} \qquad U^{(2)}$

- Starting from a general A, we want to unitarily transform it into upper Hessenberg form. (cost $O(n^3)$)

Step 1:

$$A = A^{(1)} = \begin{pmatrix} \multicolumn{5}{c}{\rule{3cm}{0.4pt}} \\ | & | & | & \cdots & | \end{pmatrix}$$

Apply Householder

$$U^{(1)} = \begin{pmatrix} I_1 & \\ & I_{n-1} - 2\vec{v}^{(1)}\vec{v}^{(1)*} \end{pmatrix}$$

$$U^{(1)} A^{(1)} = \begin{pmatrix} \multicolumn{5}{c}{\rule{3cm}{0.4pt}} \\ * & | & | & \cdots & | \\ 0 & & & & \\ \vdots & & & & \\ 0 & & & & \end{pmatrix}$$

$$A^{(2)} = U^{(1)} A^{(1)} \underset{= U^{(1)*}}{\underset{\uparrow}{U^{(1)}}} = \begin{pmatrix} \multicolumn{5}{c}{\rule{3cm}{0.4pt}} \\ * & | & | & \cdots & | \\ 0 & & & & \\ \vdots & & & & \\ 0 & & & & \end{pmatrix} \begin{pmatrix} I_1 & \\ & I_{n-1} - 2\vec{v}^{(1)}\vec{v}^{(1)*} \end{pmatrix}$$

$$= \begin{pmatrix} \multicolumn{4}{c}{\rule{3cm}{0.4pt}} \\ * & \multicolumn{3}{c}{\multirow{4}{*}{$\boxed{\begin{matrix} | & | & \cdots & | \end{matrix}}$}} \\ 0 & & & \\ \vdots & & & \\ 0 & & & \end{pmatrix}$$

Step 2: ...

A heuristic proof of convergence of QR-algorithm

$$A_{k+1} = Q_k^* A_k Q_k = Q_k^* Q_{k-1}^* A_{k-1} Q_{k-1} Q_k$$

$$= \cdots = Q_k^* \cdots Q_1^* A \underbrace{Q_1 \cdots Q_k}_{P_k} = P_k^* A P_k$$

Denote $T_k = R_k \cdots R_1$ (upper-triangular)

Then $P_k T_k = Q_1 \cdots Q_k R_k \cdots R_1$

$$= Q_1 \cdots Q_{k-1} A_k R_{k-1} \cdots R_1$$

$$= P_{k-1} A_k T_{k-1}$$

$$A_k = P_{k-1}^* A P_{k-1} \quad \Rightarrow \quad P_{k-1} A_k = A P_{k-1}$$

Substitute into previous, we get

$$P_k T_k = A P_{k-1} T_{k-1} = \cdots = A^{k-1} P_1 T_1 = A^k$$

Suppose A can be diagonalized as $A = XDX^{-1}$ w/ D invertible

Let $X = \underset{\text{unitary}}{Q}\ \underset{\text{upper-triangular}}{R}$ and $X^{-1} = \underset{\text{lower-triangular}}{L}\ \underset{\text{upper-triangular}}{U}$ Then

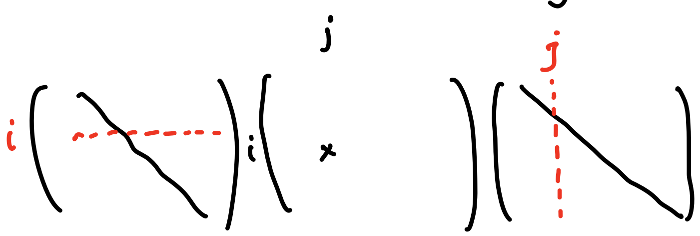
diagonal entries $= 1$

$$A^k = XD^kX^{-1} = QRD^kLU = QR(D^kLD^{-k})D^kU$$

<u>Lemma</u> If $D = \operatorname{diag}\{\lambda_1, \cdots, \lambda_n\}$ satisfies $|\lambda_1| > \cdots > |\lambda_n|$, then

$$\lim_{k\to\infty} D^k L D^{-k} = \underline{I}$$

<u>Proof</u> $(D^kLD^{-k})_{ij} = L_{ij}\lambda_i^k \lambda_j^{-k} = L_{ij}\left(\frac{\lambda_i}{\lambda_j}\right)^k$



$$\left|\frac{\lambda_i}{\lambda_j}\right| \begin{cases} = 1 & i = j \\ < 1 & i > j \end{cases}$$

✓

Then $\underset{\text{unitary}}{P_k}\ \underset{\text{upper-triangular}}{T_k} = A^k \approx \underset{\text{unitary}}{Q}\ \underbrace{RD^kU}_{\text{upper-triangular}} \quad \overset{\text{"Uniqueness" of QR-factor.}}{\Longrightarrow} \quad P_k \approx Q.$

Then, since $A = XDX^{-1} = QRDR^{-1}Q^* \approx P_k RDR^{-1} P_k^*$

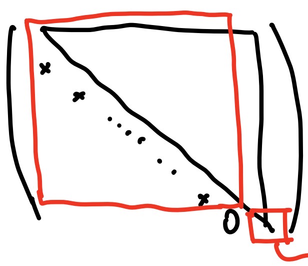
$A_{k+1} = P_k^* A P_k \approx R D R^{-1}$ upper-triangular.

Shifted QR algorithm

$$A_k - z_k I = Q_k R_k \ , \quad A_{k+1} = R_k Q_k + z_k I$$

$$A_{k+1} = Q_k^* (A_k - z_k I) Q_k + z_k I = Q_k^* A_k Q_k$$

- If z_k is exactly an eigenvalue of A_k, then $(A_{k+1})_{n,n-1} = 0$



get an eigenvalue, take out (deflation)

Generally, if $A = \begin{pmatrix} B_1 & B_2 \\ 0 & B_3 \end{pmatrix}$ w/ square B_1, B_3

then $\mathrm{eig}(A) = \mathrm{eig}(B_1) \cup \mathrm{eig}(B_3)$

$\Rightarrow$ Choose z_k as an approximation of an eigenvalue of A.

- Raleigh quotient shift : $z_k = (A_k)_{n,n}$
- Wilkinson shift : $z_k =$ eigenvalue of $\begin{pmatrix} (A_k)_{n-1,n-1} & (A_k)_{n-1,n} \\ (A_k)_{n,n-1} & (A_k)_{n,n} \end{pmatrix}$ that is closer to $(A_k)_{n,n}$.

When $|(A_k)_{n,n-1}| < TOL$, do deflation