The QR algorithm

➤ The most common method for solving small (dense) eigenvalue problems. The basic algorithm:

QR without shifts

1. Until Convergence Do:
2. Compute the QR factorization $A = QR$
3. Set $A := RQ$
4. EndDo

➤ "Until Convergence" means "Until A becomes close enough to an upper triangular matrix"

➤ Note: $A_{new} = RQ = Q^H(QR)Q = Q^H AQ$

➤ *A_{new} is Unitarily similar to A* $\rightarrow$ Spectrum does not change

➤ Convergence analysis complicated – but insight: we are implicitly doing a QR factorization of A^k:

	QR-Factorize:	Multiply backward:	
Step 1	$A_0 = Q_0R_0$	$A_1 = R_0Q_0$	
Step 2	$A_1 = Q_1R_1$	$A_2 = R_1Q_1$	
Step 3:	$A_2 = Q_2R_2$	$A_3 = R_2Q_2$	Then:

$$\begin{aligned}[Q_0Q_1Q_2][R_2R_1R_0] &= Q_0Q_1A_2R_1R_0 \\ &= Q_0(Q_1R_1)(Q_1R_1)R_0 \\ &= Q_0A_1A_1R_0, \qquad A_1 = R_0Q_0 \rightarrow \\ &= \underbrace{(Q_0R_0)}_{A}\underbrace{(Q_0R_0)}_{A}\underbrace{(Q_0R_0)}_{A} = A^3\end{aligned}$$

➤ $[Q_0Q_1Q_2][R_2R_1R_0]$ == QR factorization of A^3

➤ This helps analyze the algorithm (details skipped)

➤ Above basic algorithm is never used as is in practice. Two variations:

(1) Use shift of origin and

(2) Start by transforming A into an Hessenberg matrix

Practical QR algorithms: Shifts of origin

Observation: (from theory): Last row converges fastest. Convergence is dictated by $\frac{|\lambda_n|}{|\lambda_{n-1}|}$

➤ We will now consider only the real symmetric case.

➤ Eigenvalues are real.

➤ $A^{(k)}$ remains symmetric throughout process.

➤ As k goes to infinity the last column and row (except $a_{nn}^{(k)}$) converge to zero quickly.,,

➤ and $a_{nn}^{(k)}$ converges to lowest eigenvalue.

$$A^{(k)} = \left(\begin{array}{ccccc|c} . & . & . & . & . & a \\ . & . & . & . & . & a \\ . & . & . & . & . & a \\ . & . & . & . & . & a \\ . & . & . & . & . & a \\ \hline a & a & a & a & a & a \end{array}\right)$$

- Idea: Apply QR algorithm to $A^{(k)} - \mu I$ with $\mu = a_{nn}^{(k)}$. Note: eigenvalues of $A^{(k)} - \mu I$ are shifted by μ (eigenvectors unchanged). $\rightarrow$ Shift matrix by $+\mu I$ after iteration.

QR with shifts

1. Until row $a_{in}, 1 \leq i < n$ converges to zero DO:
2. Obtain next shift (e.g. $\mu = a_{nn}$)
3. $A - \mu I = QR$
5. Set $A := RQ + \mu I$
6. EndDo

- Convergence (of last row) is cubic at the limit! [for symmetric case]

- Result of algorithm:

$$A^{(k)} = \left(\begin{array}{ccccc|c} . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ \hline 0 & 0 & 0 & 0 & 0 & \lambda_n \end{array}\right)$$

- Next step: deflate, i.e., apply above algorithm to $(n-1) \times (n-1)$ upper block.

Practical algorithm: Use the Hessenberg Form

Recall: Upper Hessenberg matrix is such that

$$a_{ij} = 0 \text{ for } i > j + 1$$

Observation: The QR algorithm preserves Hessenberg form (tridiagonal form in symmetric case). Results in substantial savings.

Transformation to Hessenberg form

- Want $H_1 A H_1^T = H_1 A H_1$ to have the form shown on the right
- Consider the first step only on a 6×6 matrix

$$\begin{pmatrix} \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star & \star \end{pmatrix}$$

- Choose a w in $H_1 = I - 2ww^T$ to make the first column have zeros from position 3 to n. So $w_1 = 0$.
- Apply to left: $B = H_1 A$
- Apply to right: $A_1 = BH_1$.

Main observation: the Householder matrix H_1 which transforms the column $A(2:n,1)$ into e_1 works only on rows 2 to n. When applying the transpose H_1 to the right of $B = H_1 A$, we observe that only columns 2 to n will be altered. So the first column will retain the desired pattern (zeros below row 2).

- Algorithm continues the same way for columns 2, ...,$n-2$.

QR for Hessenberg matrices

- Need the "Implicit Q theorem"

Suppose that $Q^T AQ$ is an unreduced upper Hessenberg matrix. Then columns 2 to n of Q are determined uniquely (up to signs) by the first column of Q.

- In other words if $V^T AV = G$ and $Q^T AQ = H$ are both Hessenberg and $V(:,1) = Q(:,1)$ then $V(:,i) = \pm Q(:,i)$ for $i = 2:n$.

Implication: To compute $A_{i+1} = Q_i^T AQ_i$ we can:

- Compute 1st column of Q_i [== scalar $\times A(:,1)$]
- Choose other columns so Q_i = unitary, and A_{i+1} = Hessenberg.

- W'll do this with Givens rotations:

Example: With $n = 5$:

$$A = \begin{pmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{pmatrix}$$

1. Choose $G_1 = G(1,2,\theta_1)$ so that $(G_1^T A_0)_{21} = 0$

- $$A_1 = G_1^T AG_1 = \begin{pmatrix} * & * & * & * & * \\ * & * & * & * & * \\ + & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{pmatrix}$$

2. Choose $G_2 = G(2,3,\theta_2)$ so that $(G_2^T A_1)_{31} = 0$

- $$A_2 = G_2^T A_1 G_2 = \begin{pmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & + & * & * & * \\ 0 & 0 & 0 & * & * \end{pmatrix}$$

3. Choose $G_3 = G(3,4,\theta_3)$ so that $(G_3^T A_2)_{42} = 0$

- $$A_3 = G_3^T A_2 G_3 = \begin{pmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & + & * & * \end{pmatrix}$$

4. *Choose* $G_4 = G(4, 5, \theta_4)$ *so that* $(G_4^T A_3)_{53} = 0$

- $$A_4 = G_4^T A_3 G_4 = \begin{pmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{pmatrix}$$

- Process known as "Bulge chasing"
- Similar idea for the symmetric (tridiagonal) case