9. QR algorithm

- basic QR algorithm
- QR iteration with tridiagonal matrices
- reduction to tridiagonal form
- QR algorithm with shifts

QR algorithm

- the standard method for computing eigenvalues and eigenvectors
- we discuss the algorithm for symmetric eigendecomposition

$$A = Q\Lambda Q^T = \sum_{i=1}^{n} \lambda_i q_i q_i^T$$

there are two stages

1. reduction of A to tridiagonal form by an orthogonal similarity transformation

$$Q_1^T A Q_1 = T, \qquad T \text{ tridiagonal}, \quad Q_1 \text{ orthogonal}$$

2. a fast iterative algorithm to compute eigendecomposition of a tridiagonal matrix

$$T = Q_2 \Lambda Q_2^T$$

   the product $Q = Q_1 Q_2$ is the matrix of eigenvectors of A

the purpose of stage 1 is to reduce the complexity of stage 2

Necessity of iterative methods

algorithms for computing eigenvalues of matrices of order $n \geq 5$ must be iterative

- roots of polynomial $\lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_1\lambda + a_0$ are eigenvalues of $n \times n$ matrix

$$A = \begin{bmatrix} -a_{n-1} & -a_{n-2} & -a_{n-3} & \cdots & -a_1 & -a_0 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

- no algebraic formula exists for the roots of a general polynomial of degree $n \geq 5$

  ("algebraic" means involving the four basic arithmetic operations and kth roots)

- hence, no finite algorithm exists for eigenvalues of general matrix of order $n \geq 5$

Outline

- **basic QR algorithm**
- QR iteration with tridiagonal matrices
- reduction to tridiagonal form
- QR algorithm with shifts

QR algorithm

suppose A is a symmetric $n \times n$ matrix

Basic QR iteration: start at $A_1 = A$ and repeat for $k = 1, 2, \ldots,$

- compute QR factorization $A_k = Q_k R_k$
- compute $A_{k+1} = R_k Q_k$

QR algorithm and QR factorization

- algorithm is called *QR algorithm,* because it is based on QR factorization
- singular A_k (in step 1) are handled by allowing zeros on diagonal of R_k

Convergence: for most matrices,

- A_k converges to a diagonal matrix of eigenvalues of A
- $U_k = Q_1 Q_2 \cdots Q_k$ converges to matrix of eigenvectors

Some immediate properties

$$A_1 = A, \qquad A_k = Q_k R_k, \quad A_{k+1} = R_k Q_k \quad \text{(for } k \geq 1\text{)}$$

- the matrices A_k are symmetric: the first matrix $A_1 = A$ is symmetric and

$$A_{k+1} = R_k Q_k = Q_k^T A_k Q_k$$

- continuing recursively, we see that an orthogonal similarity relates A_k and A:

$$\begin{aligned} A_{k+1} &= (Q_1 Q_2 \cdots Q_k)^T A (Q_1 Q_2 \cdots Q_k) \\ &= U_k^T A U_k \end{aligned}$$

  therefore the matrices A_k all have the same eigenvalues as A

- the orthogonal matrices $U_k = Q_1 Q_2 \cdots Q_k$ and the upper triangular R_k satisfy

$$A U_{k-1} = U_{k-1} A_k = U_{k-1} Q_k R_k = U_k R_k$$

Simultaneous iteration

a related algorithm generates the matrices U_k, R_k from last property on page 9.5:

$$AU_{k-1} = U_k R_k$$

note that the right-hand side is a QR factorization

Simultaneous iteration: start at $U_0 = I$ and repeat for $k = 1, 2, \ldots$,

- multiply with A: compute $V_k = AU_{k-1}$
- compute QR factorization $V_k = U_k R_k$

if the matrices U_k converge to U, then R_k converges to a diagonal matrix, since

$$R_k = U_k^T V_k = U_k^T A U_{k-1}$$

so the limit of R_k is both symmetric ($U^T A U$) and triangular, hence diagonal

Power iteration

simultaneous iteration is a matrix extension of the *power iteration*

Power iteration: start at n-vector u_0 with $\|u_0\| = 1$, and repeat for $k = 1, 2, \ldots$,

- multiply with A: compute $v_k = Au_{k-1}$
- normalize: $u_k = v_k / \|v_k\|$

this is a simple iteration for computing an eigenvector with the largest eigenvalue

- suppose the eigenvalues of A satisfy $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$
- expand u_0 as $u_0 = \alpha_1 q_1 + \cdots + \alpha_n q_n$ where q_i is a normalized eigenvector for λ_i
- after k power iterations, u_k is a normalized scalar multiple of the vector

$$A^k u_0 = \lambda_1^k \left(\alpha_1 q_1 + \alpha_2 (\lambda_2/\lambda_1)^k q_2 + \cdots + \alpha_n (\lambda_n/\lambda_1)^k q_n \right)$$

- if $\alpha_1 \neq 0$, the vector $\pm u_k$ converges to q_1, and $u_k^T A u_k$ converges to λ_1

Simultaneous iteration as matrix power iteration

Simultaneous iteration:

$$U_0 = I, \qquad AU_{k-1} = U_k R_k \quad \text{(for } k \geq 1\text{)}$$

with U_k orthogonal, R_k upper triangular

Interpretation as QR factorization of powers of A: after k steps,

$$A^k = U_k S_k \qquad \text{where } S_k = R_k R_{k-1} \cdots R_1$$

- the product $S_k = R_k R_{k-1} \cdots R_1$ is upper triangular
- follows from repeated substitution:

$$A = U_1 R_1, \qquad A^2 = AU_1 R_1 = U_2 R_2 R_1, \qquad A^3 = AU_2 R_2 R_1 = U_3 R_3 R_2 R_1, \quad \ldots$$

Convergence of simultaneous iteration

eigendecomposition of A

$$A = \sum_i \lambda_i q_i q_i^T = Q \Lambda Q^T$$

Assumptions

- $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_{n-1}| > |\lambda_n|$ with strict inequalities
- the $j \times j$ leading principal submatrices of Q are nonsingular for $j = 1, \ldots, n$:

$$\begin{bmatrix} Q_{11} & \cdots & Q_{1j} \\ \vdots & & \vdots \\ Q_{j1} & \cdots & Q_{jj} \end{bmatrix} = \begin{bmatrix} e_1^T q_1 & \cdots & e_1^T q_j \\ \vdots & & \vdots \\ e_j^T q_1 & \cdots & e_j^T q_j \end{bmatrix} \quad \text{is nonsingular}$$

Convergence: in simultaneous iteration (and QR iteration),

$$U_k^T A U_k \quad \longrightarrow \quad \Lambda = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{bmatrix}$$

(Outline of) proof

- eigendecomposition of A^k is $A^k = \sum_{i=1}^n \lambda_i^k q_i q_i^T$
- first j columns of factor U_k in the QR factorization $A^k = U_k S_k$ span the range of

$$
\begin{aligned}
A^k \begin{bmatrix} e_1 & \cdots & e_j \end{bmatrix} \quad &= \quad \begin{bmatrix} q_1 & \cdots & q_j \end{bmatrix} \begin{bmatrix} \lambda_1^k & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_j^k \end{bmatrix} \begin{bmatrix} q_1^T e_1 & \cdots & q_1^T e_j \\ \vdots & & \vdots \\ q_j^T e_1 & \cdots & q_j^T e_j \end{bmatrix} \\
&+ \begin{bmatrix} q_{j+1} & \cdots & q_n \end{bmatrix} \begin{bmatrix} \lambda_{j+1}^k & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n^k \end{bmatrix} \begin{bmatrix} q_{j+1}^T e_1 & \cdots & q_{j+1}^T e_j \\ \vdots & & \vdots \\ q_n^T e_1 & \cdots & q_n^T e_j \end{bmatrix}
\end{aligned}
$$

- if the two assumptions on the previous page hold,

$$
\text{range}(\begin{bmatrix} U_k e_1 & \cdots & U_k e_j \end{bmatrix}) = \text{range}(\begin{bmatrix} A^k e_1 & \cdots & A^k e_j \end{bmatrix}) \quad \longrightarrow \quad \text{range}(\begin{bmatrix} q_1 & \cdots & q_j \end{bmatrix})
$$

- the fact that this holds for every $j = 1, \ldots, n$ implies that $U_k^T A U_k \longrightarrow \Lambda$

Outline

- basic QR algorithm
- **QR iteration with tridiagonal matrices**
- reduction to tridiagonal form
- QR algorithm with shifts

Complexity of QR iteration

QR iteration: start at $A_1 = A$ and repeat for $k = 1, 2, \ldots,$

- compute QR factorization $A_k = Q_k R_k$
- compute $A_k = R_k Q_k$

Complexity

- for general symmetric A, cost per iteration is order n^3
- we'll see that for tridiagonal A, cost per iteration is only order n
- this motivates stage 1 (page 9.2): first reduce A to tridiagonal form (at cost n^3)

QR factorization of tridiagonal matrix

suppose A is $n \times n$ and tridiagonal, with QR factorization

$$A = QR$$

then Q and R have a special structure:

$$\begin{bmatrix} \bullet & \bullet & & & & \\ \bullet & \bullet & \bullet & & & \\ & \bullet & \bullet & \bullet & & \\ & & \bullet & \bullet & \bullet & \\ & & & \bullet & \bullet & \bullet \\ & & & & \bullet & \bullet \end{bmatrix} = \begin{bmatrix} \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ & \bullet & \bullet & \bullet & \bullet & \bullet \\ & & \bullet & \bullet & \bullet & \bullet \\ & & & \bullet & \bullet & \bullet \\ & & & & \bullet & \bullet \end{bmatrix} \begin{bmatrix} \bullet & \bullet & \bullet & & & \\ & \bullet & \bullet & \bullet & & \\ & & \bullet & \bullet & \bullet & \\ & & & \bullet & \bullet & \bullet \\ & & & & \bullet & \bullet \\ & & & & & \bullet \end{bmatrix}$$

(dots indicate possibly nonzero elements)

- Q is zero below the first subdiagonal ($Q_{ij} = 0$ if $i > j + 1$)

  column k is column k of A orthogonalized with respect to previous columns

- R is zero above second superdiagonal ($R_{ij} = 0$ if $j > i + 2$)

  follows from considering $R = Q^T A$ and the property of Q

QR iteration with tridiagonal A

now suppose A in the basic QR iteration on page 9.4 is tridiagonal and symmetric

- we already noted that matrices A_k are symmetric if A is symmetric (page 9.5):

$$A_{k+1} = R_k Q_k = Q_k^T A_k Q_k$$

- Q-factor of a tridiagonal matrix is zero below the first subdiagonal (page 9.12)
- this implies that the product $R_k Q_k = A_{k+1}$ is zero below the first subdiagonal:

$$\begin{bmatrix} \bullet & \bullet & \bullet & & & \\ & \bullet & \bullet & \bullet & & \\ & & \bullet & \bullet & \bullet & \\ & & & \bullet & \bullet & \bullet \\ & & & & \bullet & \bullet \\ & & & & & \bullet \end{bmatrix} \begin{bmatrix} \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ & \bullet & \bullet & \bullet & \bullet & \bullet \\ & & \bullet & \bullet & \bullet & \bullet \\ & & & \bullet & \bullet & \bullet \\ & & & & \bullet & \bullet \end{bmatrix} = \begin{bmatrix} \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ & \bullet & \bullet & \bullet & \bullet & \bullet \\ & & \bullet & \bullet & \bullet & \bullet \\ & & & \bullet & \bullet & \bullet \\ & & & & \bullet & \bullet \end{bmatrix}$$

- since A_{k+1} is also symmetric, it is tridiagonal

hence, symmetric tridiagonal structure of A is preserved in A_k during QR iteration

Computing tridiagonal QR factorization

QR factorization of $n \times n$ tridiagonal A takes order n operations

$$Q^T A = R$$

for example, in the Householder algorithm (133A lecture 6)

- Q^T is a product of reflectors $H_k = I - v_k v_k^T$ that make A upper triangular

$$H_{n-1} \cdots H_1 \begin{bmatrix} A_{11} & A_{12} & 0 & \cdots & 0 & 0 \\ A_{21} & A_{22} & A_{23} & \cdots & 0 & 0 \\ 0 & A_{32} & A_{33} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A_{n-1,n-1} & A_{n-1,n} \\ 0 & 0 & 0 & \cdots & A_{n,n-1} & A_{nn} \end{bmatrix} = R$$

  if A is tridiagonal, each vector v_k has only two nonzero elements

- Q is stored in factored form (the reflectors v_k are stored)
- we can allow zeros on diagonal of R, to extend QR factorization to singular A

Outline

- basic QR algorithm
- QR iteration with tridiagonal matrices
- **reduction to tridiagonal form**
- QR algorithm with shifts

Reflector

$$Q = I - vv^T \qquad \text{with } \|v\| = \sqrt{2}$$

- Qx is reflection of x through the hyperplane $\{z \mid v^T z = 0\}$
- Q is symmetric and orthogonal
- for m-vectors x, v, matrix–vector product Qx can be computed in $4m$ flops, as

$$Qx = x - (v^T x)v$$

Reflection to multiple of first unit vector

- an easily constructed reflector maps a given y to a multiple of e_1
- if $y \neq 0$, choose the reflector defined by

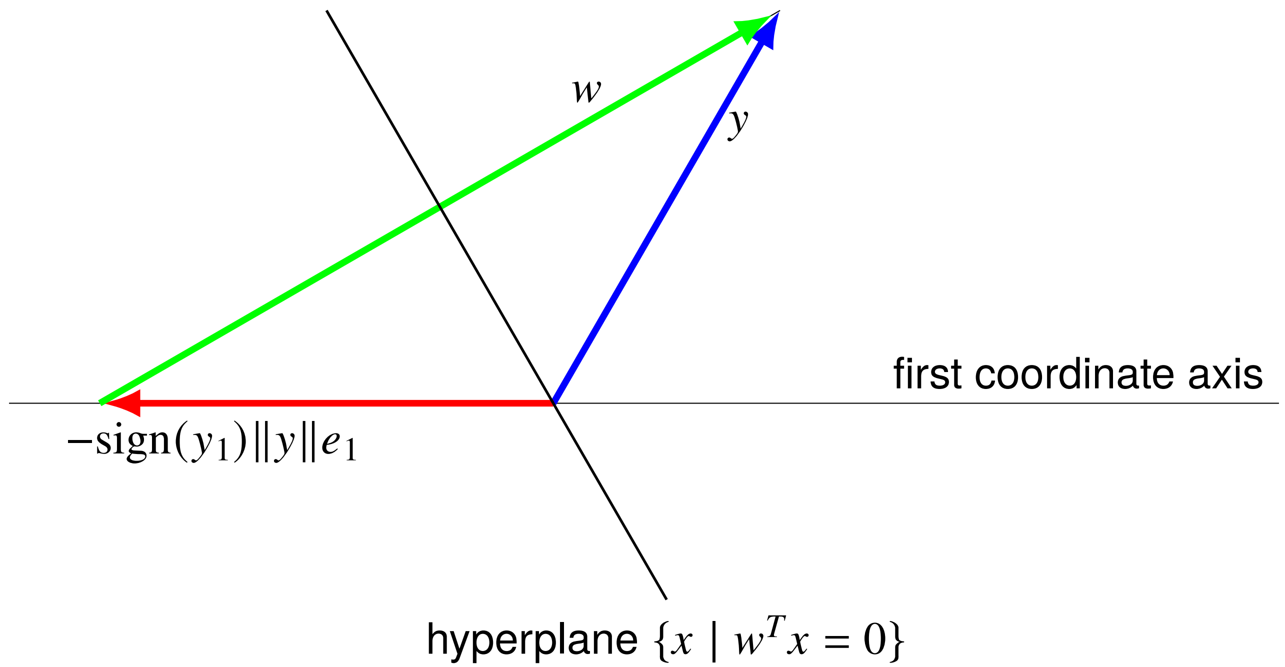
$$v = \frac{\sqrt{2}}{\|w\|} w, \qquad w = y + \mathrm{sign}(y_1)\|y\| e_1 = \begin{bmatrix} y_1 + \mathrm{sign}(y_1)\|y\| \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

(we define $\mathrm{sign}(0) = 1$)

- this reflector maps y to

$$Qy = -\mathrm{sign}(y_1)\|y\| e_1 = \begin{bmatrix} -\mathrm{sign}(y_1)\|y\| \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Geometry



the reflection through the hyperplane $\{x \mid w^T x = 0\}$ with normal vector

$$w = y + \text{sign}(y_1)\|y\|e_1$$

maps y to the vector $-\text{sign}(y_1)\|y\|e_1$

Reduction to tridiagonal form

given an $n \times n$ symmetric matrix A, find orthogonal Q such that

$$Q^T A Q = \begin{bmatrix} a_1 & b_1 & 0 & \cdots & 0 & 0 & 0 \\ b_1 & a_2 & b_2 & \cdots & 0 & 0 & 0 \\ 0 & b_2 & a_3 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{n-2} & b_{n-2} & 0 \\ 0 & 0 & 0 & \cdots & b_{n-2} & a_{n-1} & b_{n-1} \\ 0 & 0 & 0 & \cdots & 0 & b_{n-1} & a_n \end{bmatrix}$$

- this can be achieved by a product of $n-2$ reflectors

$$Q = Q_1 Q_2 \cdots Q_{n-2}$$

- complexity is order n^3

First step

partition A as

$$A = \begin{bmatrix} a_1 & c_1^T \\ c_1 & B_1 \end{bmatrix}$$

c_1 is an $(n-1)$-vector, B_1 is $(n-1) \times (n-1)$

- find $(n-1) \times (n-1)$ reflector $I - v_1 v_1^T$ that maps c_1 to $b_1 e_1$ and define

$$Q_1 = \begin{bmatrix} 1 & 0 \\ 0 & I - v_1 v_1^T \end{bmatrix}$$

- multiply A with Q_1 to introduce zeros in positions $3, \ldots, n$ of 1st column and row

$$\begin{aligned} Q_1 A Q_1 &= \begin{bmatrix} a_1 & c_1^T (I - v_1 v_1^T) \\ (I - v_1 v_1^T) c_1 & (I - v_1 v_1^T) B_1 (I - v_1 v_1^T) \end{bmatrix} \\ &= \begin{bmatrix} a_1 & b_1 e_1^T \\ b_1 e_1 & B_1 - v_1 w_1^T - w_1 v_1^T \end{bmatrix} \end{aligned}$$

where $w_1 = B_1 v_1 - \dfrac{v_1^T B_1 v_1}{2} v_1$

- computation of $2, 2$ block requires order $4n^2$ flops

General step

after $k-1$ steps,

$$
Q_{k-1}\cdots Q_1 A Q_1 \cdots Q_{k-1} = \left[\begin{array}{cccccc|cc}
a_1 & b_1 & 0 & \cdots & 0 & 0 & 0 & 0 \\
b_1 & a_2 & b_2 & \cdots & 0 & 0 & 0 & 0 \\
0 & b_2 & a_3 & \cdots & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & a_{k-2} & b_{k-2} & 0 & 0 \\
0 & 0 & 0 & \cdots & b_{k-2} & a_{k-1} & b_{k-1} & 0 \\
\hline
0 & 0 & 0 & \cdots & 0 & b_{k-1} & a_k & c_k^T \\
0 & 0 & 0 & \cdots & 0 & 0 & c_k & B_k
\end{array}\right]
$$

- find a reflector $I - v_k v_k^T$ that maps the $(n-k)$-vector c_k to $b_k e_1$ and define

$$
Q_k = \begin{bmatrix} I_{k-1} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & I - v_k v_k^T \end{bmatrix}, \qquad (I - v_k v_k^T) B_k (I - v_k v_k^T) = \begin{bmatrix} a_{k+1} & c_{k+1}^T \\ c_{k+1} & B_{k+1} \end{bmatrix}
$$

- complexity of step k is $4(n-k)^2$ plus lower order terms

Summary for 5×5 matrix

$$
\begin{aligned}
Q_3^T Q_2^T Q_1^T \begin{bmatrix} \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \end{bmatrix} Q_1 Q_2 Q_3
&= Q_3^T Q_2^T \begin{bmatrix} \bullet & \bullet & & & \\ \bullet & \bullet & \bullet & \bullet & \bullet \\ & \bullet & \bullet & \bullet & \bullet \\ & \bullet & \bullet & \bullet & \bullet \\ & \bullet & \bullet & \bullet & \bullet \end{bmatrix} Q_2 Q_3 \\
&= Q_3^T \begin{bmatrix} \bullet & \bullet & & & \\ \bullet & \bullet & \bullet & & \\ & \bullet & \bullet & \bullet & \bullet \\ & & \bullet & \bullet & \bullet \\ & & \bullet & \bullet & \bullet \end{bmatrix} Q_3 \\
&= \begin{bmatrix} \bullet & \bullet & & & \\ \bullet & \bullet & \bullet & & \\ & \bullet & \bullet & \bullet & \\ & & \bullet & \bullet & \bullet \\ & & & \bullet & \bullet \end{bmatrix}
\end{aligned}
$$

Complexity

- complexity for complete algorithm is dominated by

$$\sum_{k=1}^{n-2} 4(n-k)^2 \approx \frac{4}{3}n^3$$

- Q is stored in factored form (the vectors v_k are stored)

- if needed, assembling the matrix Q adds another order n^3 term

Outline

- basic QR algorithm
- QR iteration with tridiagonal matrices
- reduction to tridiagonal form
- **QR algorithm with shifts**

QR algorithm with shifts

in practice, a multiple of the identity is subtracted from A_k before factoring

QR iteration with shifts: start at $A_1 = A$ and repeat for $k = 1, 2, \ldots,$

- choose a shift μ_k
- compute QR factorization $A_k - \mu_k I = Q_k R_k$
- define $A_{k+1} = R_k Q_k + \mu_k I$

- iteration still preserves symmetry and tridiagonal structure in A_k
- with properly chosen shifts, the iteration always converges
- with properly chosen shifts, convergence is fast (usually cubic)

Complexity

overall complexity of QR method for symmetric eigendecomposition $A = Q\Lambda Q^T$

Eigenvalues: if eigenvectors are not needed, we can leave Q in factored form

- reduction of A to tridiagonal form costs $(4/3)n^3$
- for tridiagonal matrix, complexity of one QR iteration is linear in n
- on average, number of QR iterations is a small multiple of n

hence, cost is dominated by $(4/3)n^3$ for initial reduction to tridiagonal form

Eigenvalues and eigenvectors: if Q is needed, order n^3 terms are added

- reduction to tridiagonal form and accumulating orthogonal matrix costs $(8/3)n^3$
- finding eigenvalues and eigenvectors of tridiagonal matrix costs $6n^3$

hence, total cost is $(26/3)n^3$ plus lower order terms

References

- Lloyd N. Trefethen and David Bau, III, *Numerical Linear Algebra* (1997).

  lectures 26–29 in this book discuss the QR iteration

- James W. Demmel, *Applied Numerical Linear Algebra* (1997).

  page 213 of this book gives details for the complexity figures on page 9.24