Numerical Analysis

Raphael Hauser
with thanks to Endre Süli

Oxford Mathematical Institute

HT 2019

The Symmetric QR Algorithm

We consider only the case where A is symmetric.

The Symmetric QR Algorithm

We consider only the case where A is symmetric.

Recall: a symmetric matrix A is said to be *similar* to B if there is a nonsingular matrix P for which $A = P^{-1}BP$.

The Symmetric QR Algorithm

We consider only the case where A is symmetric.

Recall: a symmetric matrix A is said to be *similar* to B if there is a nonsingular matrix P for which $A = P^{-1}BP$.

Similar matrices have the same eigenvalues.

The Symmetric QR Algorithm

We consider only the case where A is symmetric.

Recall: a symmetric matrix A is said to be *similar* to B if there is a nonsingular matrix P for which $A = P^{-1}BP$.

Similar matrices have the same eigenvalues. Indeed: if $A = P^{-1}BP$,

The Symmetric QR Algorithm

We consider only the case where A is symmetric.

Recall: a symmetric matrix A is said to be *similar* to B if there is a nonsingular matrix P for which $A = P^{-1}BP$.

Similar matrices have the same eigenvalues. Indeed: if $A = P^{-1}BP$, then

$$0 = \det(A - \lambda I) = \det(P^{-1}(B - \lambda I)P) = \det(P^{-1})\det(P)\det(B - \lambda I),$$

The Symmetric QR Algorithm

We consider only the case where A is symmetric.

Recall: a symmetric matrix A is said to be *similar* to B if there is a nonsingular matrix P for which $A = P^{-1}BP$.

Similar matrices have the same eigenvalues. Indeed: if $A = P^{-1}BP$, then

$$0 = \det(A - \lambda I) = \det(P^{-1}(B - \lambda I)P) = \det(P^{-1})\det(P)\det(B - \lambda I),$$

so

$$\det(A - \lambda I) = 0 \qquad \text{if, and only if,} \qquad \det(B - \lambda I) = 0.$$

The basic **QR algorithm** is:

```
Set A_1 = A.
for k = 1, 2, ...
  form the QR factorization A_k = Q_k R_k
  and set A_{k+1} = R_k Q_k
end
```

$$\texttt{Set } A_1 = A.$$
$$\texttt{for } k = 1, 2, \ldots$$
$$\quad \texttt{form the QR factorization } A_k = Q_k R_k$$
$$\quad \texttt{and set } A_{k+1} = R_k Q_k$$
$$\texttt{end}$$

Proposition

The symmetric matrices $A_1, A_2, \ldots, A_k, \ldots$ *generated by the QR algorithm are all similar and thus have the same eigenvalues.*

Proposition

The symmetric matrices $A_1, A_2, \ldots, A_k, \ldots$ generated by the QR algorithm are all similar and thus have the same eigenvalues.

Proof. Since

$$A_{k+1} = R_k Q_k = (Q_k^{-1} Q_k) R_k Q_k = Q_k^{-1} (Q_k R_k) Q_k = Q_k^{-1} A_k Q_k,$$

A_{k+1} is similar to A_k.
Furthermore, since $Q_k^{-1} = Q_k^{\mathrm{T}}$, the matrix A_{k+1} is symmetric if A_k is.
By hypothesis, $A_1 := A$ is symmetric; therefore each of the matrices A_k, $k = 1, 2, \ldots$, is symmetric. □

This basic QR algorithm works since $A_k \to$ a diagonal matrix as $k \to \infty$, the diagonal entries of which are the eigenvalues.

This basic QR algorithm works since $A_k \to$ a diagonal matrix as $k \to \infty$, the diagonal entries of which are the eigenvalues.

However, a really practical, fast algorithm is based on some refinements.

Reduction to tridiagonal form:

the idea is to apply explicit similarity transformations $QAQ^{-1} = QAQ^{\mathrm{T}}$, with Q orthogonal, so that QAQ^{T} is tridiagonal.

Reduction to tridiagonal form:

the idea is to apply explicit similarity transformations $QAQ^{-1} = QAQ^{\mathrm{T}}$, with Q orthogonal, so that QAQ^{T} is tridiagonal.

Note: direct reduction to triangular form would reveal the eigenvalues, but is not possible.

Reduction to tridiagonal form:
the idea is to apply explicit similarity transformations $QAQ^{-1} = QAQ^{\mathrm{T}}$, with Q orthogonal, so that QAQ^{T} is tridiagonal.

Note: direct reduction to triangular form would reveal the eigenvalues, but is not possible.

Indeed, if

$$H(w)A = \begin{pmatrix} \times & \times & \cdots & \times \\ 0 & \times & \cdots & \times \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \times & \cdots & \times \end{pmatrix}$$

Reduction to tridiagonal form:

the idea is to apply explicit similarity transformations $QAQ^{-1} = QAQ^{\mathrm{T}}$, with Q orthogonal, so that QAQ^{T} is tridiagonal.

Note: direct reduction to triangular form would reveal the eigenvalues, but is not possible.

Indeed, if

$$H(w)A = \begin{pmatrix} \times & \times & \cdots & \times \\ 0 & \times & \cdots & \times \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \times & \cdots & \times \end{pmatrix}$$

then $H(w)AH(w)^{\mathrm{T}}$ is generally full, i.e., all zeros created by pre-multiplication are destroyed by the post-multiplication.

However, if

$$A = \begin{pmatrix} \gamma & u^{\mathrm{T}} \\ u & C \end{pmatrix}$$

(as $A = A^{\mathrm{T}}$)

However, if

$$A = \begin{pmatrix} \gamma & u^{\mathrm{T}} \\ u & C \end{pmatrix}$$

(as $A = A^{\mathrm{T}}$) and

$$w = \begin{pmatrix} 0 \\ \hat{w} \end{pmatrix} \text{ where } H(\hat{w})u = \begin{pmatrix} \alpha \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

However, if

$$A = \begin{pmatrix} \gamma & u^{\mathrm{T}} \\ u & C \end{pmatrix}$$

(as $A = A^{\mathrm{T}}$) and

$$w = \begin{pmatrix} 0 \\ \hat{w} \end{pmatrix} \text{ where } H(\hat{w})u = \begin{pmatrix} \alpha \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

it follows that

$$H(w)A = \begin{pmatrix} \gamma & & u^{\mathrm{T}} & \\ \alpha & \times & \vdots & \times \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \times & \vdots & \times \end{pmatrix},$$

i.e., the u^{T} part of the first row of A is unchanged.

i.e., the u^{T} part of the first row of A is unchanged.

However, then

$$H(w)AH(w)^{-1} = H(w)AH(w)^{\mathrm{T}} = H(w)AH(w)$$

i.e., the u^{T} part of the first row of A is unchanged.

However, then

$$H(w)AH(w)^{-1} = H(w)AH(w)^{\mathrm{T}} = H(w)AH(w) = \left(\begin{array}{c|cccc} \gamma & \alpha & 0 & \cdots & 0 \\ \hline \alpha & & & & \\ 0 & & & & \\ \vdots & & & B & \\ 0 & & & & \end{array}\right),$$

i.e., the u^{T} part of the first row of A is unchanged.

However, then

$$H(w)AH(w)^{-1} = H(w)AH(w)^{\mathrm{T}} = H(w)AH(w) = \left(\begin{array}{c|cccc} \gamma & \alpha & 0 & \cdots & 0 \\ \hline \alpha & & & & \\ 0 & & & & \\ \vdots & & & B & \\ 0 & & & & \end{array}\right),$$

where $B = H(\hat{w})CH^{\mathrm{T}}(\hat{w})$, as $u^{\mathrm{T}}H(\hat{w})^{\mathrm{T}} = (\alpha, \ 0, \ \cdots, \ 0)$.

i.e., the u^{T} part of the first row of A is unchanged.

However, then

$$H(w)AH(w)^{-1} = H(w)AH(w)^{\mathrm{T}} = H(w)AH(w) = \left(\begin{array}{c|cccc} \gamma & \alpha & 0 & \cdots & 0 \\ \hline \alpha & & & & \\ 0 & & & & \\ \vdots & & & B & \\ 0 & & & & \end{array}\right),$$

where $B = H(\hat{w})CH^{\mathrm{T}}(\hat{w})$, as $u^{\mathrm{T}}H(\hat{w})^{\mathrm{T}} = (\alpha, \ 0, \ \cdots, \ 0)$.

Note that $H(w)AH(w)^{\mathrm{T}}$ is symmetric as A is.

Now we inductively apply this to the smaller matrix B, as described for the QR factorization but using post- as well as pre-multiplications.

Now we inductively apply this to the smaller matrix B, as described for the QR factorization but using post- as well as pre-multiplications.

The result of $n-2$ such Householder similarity transformations is the matrix

$$H(w_{n-2}) \cdots H(w_2) H(w) A H(w) H(w_2) \cdots H(w_{n-2}),$$

which is tridiagonal.

The QR factorization of a tridiagonal matrix can now easily be achieved with $n-1$ Givens rotations:

The QR factorization of a tridiagonal matrix can now easily be achieved with $n-1$ Givens rotations: if A is tridiagonal

$$\underbrace{J(n-1,n)\cdots J(2,3)J(1,2)}_{Q^{\mathrm{T}}}\,A = R, \quad \text{upper triangular.}$$

The QR factorization of a tridiagonal matrix can now easily be achieved with $n-1$ Givens rotations: if A is tridiagonal

$$\underbrace{J(n-1,n)\cdots J(2,3)J(1,2)}_{Q^{\mathrm{T}}} A = R, \quad \text{upper triangular.}$$

Precisely, R has a diagonal and 2 super-diagonals,

$$R = \begin{pmatrix} \times & \times & \times & 0 & 0 & 0 & \cdots & 0 \\ 0 & \times & \times & \times & 0 & 0 & \cdots & 0 \\ 0 & 0 & \times & \times & \times & 0 & \cdots & 0 \\ \vdots & \vdots & & & & & & \vdots \\ 0 & 0 & 0 & 0 & \times & \times & \times & 0 \\ 0 & 0 & 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \times \end{pmatrix}$$

(exercise: check!).

The QR factorization of a tridiagonal matrix can now easily be achieved with $n-1$ Givens rotations: if A is tridiagonal

$$\underbrace{J(n-1,n)\cdots J(2,3)J(1,2)}_{Q^{\mathrm{T}}} A = R, \quad \text{upper triangular.}$$

Precisely, R has a diagonal and 2 super-diagonals,

$$R = \begin{pmatrix} \times & \times & \times & 0 & 0 & 0 & \cdots & 0 \\ 0 & \times & \times & \times & 0 & 0 & \cdots & 0 \\ 0 & 0 & \times & \times & \times & 0 & \cdots & 0 \\ \vdots & \vdots & & & & & & \vdots \\ 0 & 0 & 0 & 0 & \times & \times & \times & 0 \\ 0 & 0 & 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \times \end{pmatrix}$$

(exercise: check!).

In the QR algorithm, the next matrix in the sequence is RQ.

Lemma

In the QR algorithm applied to a tridiagonal matrix, the tridiagonal form is preserved when Givens rotations are used.

Lemma

In the QR algorithm applied to a tridiagonal matrix, the tridiagonal form is preserved when Givens rotations are used.

Proof. If

$$A_k = QR = J(1,2)^{\mathrm{T}} J(2,3)^{\mathrm{T}} \cdots J(n-1,n)^{\mathrm{T}} R$$

is tridiagonal,

Lemma

In the QR algorithm applied to a tridiagonal matrix, the tridiagonal form is preserved when Givens rotations are used.

Proof. If

$$A_k = QR = J(1,2)^{\mathrm{T}} J(2,3)^{\mathrm{T}} \cdots J(n-1,n)^{\mathrm{T}} R$$

is tridiagonal, then

$$A_{k+1} = RQ = RJ(1,2)^{\mathrm{T}} J(2,3)^{\mathrm{T}} \cdots J(n-1,n)^{\mathrm{T}}.$$

Lemma

In the QR algorithm applied to a tridiagonal matrix, the tridiagonal form is preserved when Givens rotations are used.

Proof. If

$$A_k = QR = J(1,2)^{\mathrm{T}} J(2,3)^{\mathrm{T}} \cdots J(n-1,n)^{\mathrm{T}} R$$

is tridiagonal, then

$$A_{k+1} = RQ = RJ(1,2)^{\mathrm{T}} J(2,3)^{\mathrm{T}} \cdots J(n-1,n)^{\mathrm{T}}.$$

Recall that:

> post-multiplication of a matrix by $J(i,i+1)^{\mathrm{T}}$ replaces columns i and $i+1$ by linear combinations of the pair of columns, while leaving columns $j \neq i, i+1$ alone.

Thus, since R is upper triangular, the only subdiagonal entry in $RJ(1,2)^{\mathrm{T}}$ is in position $(2,1)$.

Thus, since R is upper triangular, the only subdiagonal entry in $RJ(1,2)^{\mathrm{T}}$ is in position $(2,1)$.

Similarly, the only subdiagonal entries in

$$RJ(1,2)^{\mathrm{T}}J(2,3)^{\mathrm{T}} = (RJ(1,2)^{\mathrm{T}})J(2,3)^{\mathrm{T}}$$

are in positions $(2,1)$ and $(3,2)$.

Thus, since R is upper triangular, the only subdiagonal entry in $RJ(1,2)^{\mathrm{T}}$ is in position $(2,1)$.

Similarly, the only subdiagonal entries in

$$RJ(1,2)^{\mathrm{T}} J(2,3)^{\mathrm{T}} = (RJ(1,2)^{\mathrm{T}}) J(2,3)^{\mathrm{T}}$$

are in positions $(2,1)$ and $(3,2)$.

Inductively, the only subdiagonal entries in

$$\begin{aligned} &RJ(1,2)^{\mathrm{T}} J(2,3)^{\mathrm{T}} \cdots J(i-2,i-1)^{\mathrm{T}} J(i-1,i)^{\mathrm{T}} \\ &\quad = (RJ(1,2)^{\mathrm{T}} J(2,3)^{\mathrm{T}} \cdots J(i-2,i-1)^{\mathrm{T}}) J(i-1,i)^{\mathrm{T}} \end{aligned}$$

are in positions $(j, j-1)$, $j = 2, \ldots i$.

So, the lower triangular part of A_{k+1} only has nonzeros on its first subdiagonal.

So, the lower triangular part of A_{k+1} only has nonzeros on its first subdiagonal.

However, then since A_{k+1} is symmetric, it must be tridiagonal. $\square$

Using shifts.

One further and final step in making an efficient algorithm is the use of **shifts**:

```
for k = 1, 2, ...
  form the QR factorization of A_k − μ_k I = Q_k R_k
  and set A_{k+1} = R_k Q_k + μ_k I
end
```

For any chosen sequence of values of $\mu_k \in \mathbb{R}$, $\{A_k\}_{k=1}^{\infty}$ are symmetric and tridiagonal if A_1 has these properties, and similar to A_1:

For any chosen sequence of values of $\mu_k \in \mathbb{R}$, $\{A_k\}_{k=1}^{\infty}$ are symmetric and tridiagonal if A_1 has these properties, and similar to A_1:

$$\begin{aligned} A_2 &= R_1 Q_1 + \mu_1 I \\ &= Q_1^{\mathcal{T}}(Q_1 R_1) Q_1 + \mu_1 Q_1^{\mathcal{T}} Q_1 \\ &= Q_1^{\mathcal{T}}(A_1 - \mu_1 I) Q_1 + \mu_1 Q_1^{\mathcal{T}} Q_1 \\ &= Q_1^{\mathcal{T}} A_1 Q_1, \end{aligned}$$

For any chosen sequence of values of $\mu_k \in \mathbb{R}$, $\{A_k\}_{k=1}^{\infty}$ are symmetric and tridiagonal if A_1 has these properties, and similar to A_1:

$$\begin{aligned} A_2 &= R_1 Q_1 + \mu_1 I \\ &= Q_1^{\mathcal{T}}(Q_1 R_1) Q_1 + \mu_1 Q_1^{\mathcal{T}} Q_1 \\ &= Q_1^{\mathcal{T}}(A_1 - \mu_1 I) Q_1 + \mu_1 Q_1^{\mathcal{T}} Q_1 \\ &= Q_1^{\mathcal{T}} A_1 Q_1, \end{aligned}$$

$$\begin{aligned} A_3 &= R_2 Q_2 + \mu_2 I \\ &= Q_2^{\mathcal{T}}(Q_2 R_2) Q_2 + \mu_2 Q_2^{\mathcal{T}} Q_2 \\ &= Q_2^{\mathcal{T}} A_2 Q_2 = Q_2^{\mathcal{T}} Q_1^{\mathcal{T}} A_1 Q_1 Q_2, \end{aligned}$$

and similarly for $A_4, A_5, \ldots$

The simplest shift to use is $\mu_k = a_{n,n}$ for all k, which leads rapidly in almost all cases to

$$A_k = \left(\begin{array}{c|c} T_k & 0 \\ \hline 0^{\mathrm{T}} & \lambda \end{array} \right),$$

where T_k is $n-1$ by $n-1$ and tridiagonal, and λ is an eigenvalue of A_1.

The simplest shift to use is $\mu_k = a_{n,n}$ for all k, which leads rapidly in almost all cases to

$$A_k = \left(\begin{array}{c|c} T_k & 0 \\ \hline 0^{\mathrm{T}} & \lambda \end{array} \right),$$

where T_k is $n-1$ by $n-1$ and tridiagonal, and λ is an eigenvalue of A_1.

Inductively, once this form has been found, the QR algorithm with shift $a_{n-1,n-1}$ can be concentrated only on the $(n-1) \times (n-1)$ leading submatrix T_k.

The simplest shift to use is $\mu_k = a_{n,n}$ for all k, which leads rapidly in almost all cases to

$$A_k = \left(\begin{array}{c|c} T_k & 0 \\ \hline 0^{\mathrm{T}} & \lambda \end{array} \right),$$

where T_k is $n-1$ by $n-1$ and tridiagonal, and λ is an eigenvalue of A_1.

Inductively, once this form has been found, the QR algorithm with shift $a_{n-1,n-1}$ can be concentrated only on the $(n-1) \times (n-1)$ leading submatrix T_k.

This process is called **deflation**.

The overall algorithm

for calculating the eigenvalues of an n by n symmetric matrix:

```
reduce A to tridiagonal form by orthogonal
(Householder) similarity transformations.

for m = n, n-1, ..., 2
  while a_{m-1,m} > tol
    [Q, R] = qr(A - a_{m,m} * I)
    A = R * Q + a_{m,m} * I
  end while
  record eigenvalue λ_m = a_{m,m}
  A ← leading m-1 by m-1 submatrix of A
end
record eigenvalue λ_1 = a_{1,1}
```