Notes on the Symmetric QR Algorithm

Robert A. van de Geijn
Department of Computer Science
The University of Texas
Austin, TX 78712
rvdg@cs.utexas.edu

November 21, 2010

For simplicity we will focus on the symmetric eigenvalue problem and throughout this note we will assume that A is symmetric and real valued.

Recall:

Theorem 1. *If $A \in \mathbb{R}^{n\times n}$ then there exists unitary matrix Q and diagonal matrix Λ such that $A = Q\Lambda Q^T$.*

We will partition $Q = \left(\begin{array}{c|c|c} q_0 & \cdots & q_{n-1} \end{array}\right)$ and assume that $\Lambda = \mathrm{diag}(\lambda_0, \cdots, \lambda_{n-1})$.

1 Subspace iteration

We start with a matrix $V \in \mathbb{R}^{n\times r}$ with normalized columns and iterate something like

$V^{(0)} = V$
for $k = 0, \ldots$ convergence
  $V^{(k+1)} = AV^{(k)}$
  Normalize the columns to be of unit length.
end for

The problem with this approach is that all columns will converge to an eigenvector associated with the dominant eigenvalue, since the Power Method is being applied to all columns. We will lead the reader through a succession of insights towards a practical algorithm.

Let us examine what AV looks like, for the simple case where $V = \left(\begin{array}{c|c|c} v_0 & v_1 & v_2 \end{array}\right)$ (three columns).

Let $v_0 = \sum_{j=0}^{n-1} \psi_{0,j} q_j$, $v_1 = \sum_{j=0}^{n-1} \psi_{1,j} q_j$, and $v_2 = \sum_{j=0}^{n-1} \psi_{2,j} q_j$. Then

$$\begin{array}{rcl} AV & = & A\left(\begin{array}{c|c|c} v_0 & v_1 & v_2 \end{array}\right) \\ & = & A\left(\begin{array}{c|c|c} \sum_{j=0}^{n-1} \psi_{0,j} q_j & \sum_{j=0}^{n-1} \psi_{1,j} q_j & \sum_{j=0}^{n-1} \psi_{2,j} q_j \end{array}\right) \\ & = & \left(\begin{array}{c|c|c} \sum_{j=0}^{n-1} \psi_{0,j} \lambda_j q_j & \sum_{j=0}^{n-1} \psi_{1,j} \lambda_j q_j & \sum_{j=0}^{n-1} \psi_{2,j} \lambda_j q_j \end{array}\right) \end{array}$$

- If we happened to know λ_0, λ_1, and λ_2 then we could divide the columns by these, respectively, and get new vectors

$$\left(\begin{array}{c|c|c} \hat{v}_0 & \hat{v}_1 & \hat{v}_2 \end{array}\right) = \left(\begin{array}{c|c|c} \sum_{j=0}^{n-1} \psi_{0,j} \left(\frac{\lambda_j}{\lambda_0}\right) q_j & \sum_{j=0}^{n-1} \psi_{1,j} \left(\frac{\lambda_j}{\lambda_1}\right) q_j & \sum_{j=0}^{n-1} \psi_{2,j} \left(\frac{\lambda_j}{\lambda_2}\right) q_j \end{array}\right)$$

$$= \left(\begin{array}{c|c|c} & \psi_{1,0}\left(\frac{\lambda_0}{\lambda_1}\right) q_0 + & \psi_{2,0}\left(\frac{\lambda_0}{\lambda_2}\right) q_0 + \psi_{2,1}\left(\frac{\lambda_1}{\lambda_2}\right) q_1 + \\ \psi_{0,0} q_0 + & \psi_{1,1} q_1 + & \psi_{2,2} q_2 + \\ \sum_{j=1}^{n-1} \psi_{0,j}\left(\frac{\lambda_j}{\lambda_0}\right) q_j & \sum_{j=2}^{n-1} \psi_{1,j}\left(\frac{\lambda_j}{\lambda_1}\right) q_j & \sum_{j=3}^{n-1} \psi_{2,j}\left(\frac{\lambda_j}{\lambda_2}\right) q_j \end{array} \right) \quad (1)$$

- Assume that $|\lambda_0| > |\lambda_1| > |\lambda_2| > |\lambda_3| \leq \cdots \geq \lambda_{n-1}$. Then, like for the power method, the first column will see components in the direction of $\{q_1, \ldots, q_{n-1}\}$ shrink relative to the component in the direction of q_0. The second column will see components in the direction of $\{q_2, \ldots, q_{n-1}\}$ shrink relative to the component in the direction of q_1, but the component in the direction of q_0 increases, relatively, since $|\lambda_0/\lambda_1| > 1$, etc..

- If we happen to know q_0, then we could subtract out the component of $\hat{v}_1 = \psi_{1,0}\frac{\lambda_0}{\lambda_1} q_0 + \psi_{1,1} q_1 + \sum_{j=2}^{n-1} \psi_{1,j}\frac{\lambda_j}{\lambda_1} q_j$ in the direction of q_0: $\hat{v}_1 - q_0^T \hat{v}_1 q_0 = \psi_{1,1} q_1 + \sum_{j=2}^{n-1} \psi_{1,j}\frac{\lambda_j}{\lambda_1} q_j$ so that we are left with the component in the direction of q_1 and components in directions of $q_2, \ldots, q_{n-1}$ that are suppressed every time through the loop. Similarly, if we also know q_1, the components of $\hat{v}_2$ in the direction of q_0 and q_1 could be subtracted from that vector.

- We do not know λ_0, λ_1, and λ_2 but from the discussion about the Power Method we remember that we can just normalize $\hat{v}_0$, $\hat{v}_1$, and $\hat{v}_2$ to have unit length.

- We do not know q_0 and q_1, but we can informally argue that if we keep iterating, $\hat{v}_1$ and $\hat{v}_2$ will eventually point in the directions of q_0 and q_1, respectively. So, we use $\hat{v}_0$ and $\hat{v}_1$ in place of q_0 and q_1 in the above discussion.

- What we recognize is that normalizing $\hat{v}_0$, subtracting out the component of $\hat{v}_1$ in the direction of $\hat{v}_0$, and then normalizing $\hat{v}_1$, etc., is exactly what the Gram-Schmidt process does. And thus, we can use any convenient (and stable) QR factorization method. This also shows how the method can be generalized to more than three columns.

The algorithm now becomes:

$V^{(0)} = I^{n \times p}$ ($I^{n \times p}$ represents the first p columns of I)
for $k = 0, \ldots$ convergence
  $AV^{(k)} \rightarrow V^{(k+1)} R^{(k+1)}$ (QR factorization with $R^{(k+1)} \in \mathbb{R}^{p \times p}$)
end for

Now consider again (1), focusing on the third column:

$$\left(\begin{array}{l} \psi_{2,0}\left(\frac{\lambda_0}{\lambda_2}\right) q_0 + \psi_{2,1}\left(\frac{\lambda_1}{\lambda_2}\right) q_1 + \\ \psi_{2,2} q_2 + \\ \sum_{j=3}^{n-1} \psi_j \left(\frac{\lambda_j}{\lambda_2}\right) q_j \end{array} \right) = \left(\begin{array}{l} \psi_{2,0}\left(\frac{\lambda_0}{\lambda_2}\right) q_0 + \psi_{2,1}\left(\frac{\lambda_1}{\lambda_2}\right) q_1 + \\ \psi_{2,2} q_2 + \\ \psi_j \boxed{\left(\frac{\lambda_3}{\lambda_2}\right)} q_3 + \sum_{j=4}^{n-1} \psi_{2,j}\left(\frac{\lambda_j}{\lambda_2}\right) q_j \end{array} \right).$$

This shows that, if the components in the direction of q_0 and q_1 are subtracted out, it is the component in the direction of q_3 that is deminished in length the most slowly, dictated by the ratio $\left|\frac{\lambda_3}{\lambda_2}\right|$. This, of course, generalizes: the jth column of $V^{(k)}$, $v_i^{(k)}$ will have a component in the direction of q_{j+1}, of length $|q_{j+1}^T v_j^{(k)}|$, that can be expected to shrink most slowly.

We demonstrate this in Fig. 1, which shows the execution of the algorithm with $p = n$ for a 5×5 matrix, and shows how $|q_{j+1}^T v_j^{(k)}|$ converge to zero as as function of k.

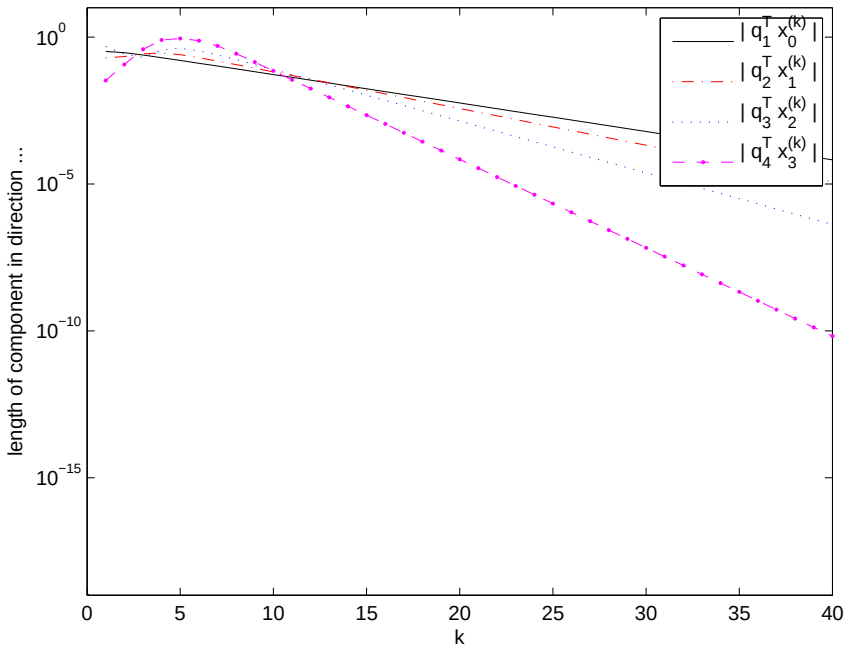

Figure 1: Convergence of the subspace iteration for a 5×5 matrix.

Next, we observe that if $V \in \mathbb{R}^{n \times n}$ in the above iteration, then AV yields a next-to last column of the form

$$\begin{pmatrix} \sum_{j=0}^{n-3} \gamma_{n-2,j} \left(\frac{\lambda_j}{\lambda_{n-2}} \right) q_j + \\ \psi_{n-2,n-2} q_{n-2} + \\ \psi_{n-2,n-1} \boxed{\left(\frac{\lambda_{n-1}}{\lambda_{n-2}} \right)} q_{n-1} \end{pmatrix},$$

where $\psi_{i,j} = q_j^T v_i$. Thus, given that the components in the direction of q_j, $j = 0, \ldots, n-2$ can be expected in later iterations to be greatly reduced by the QR factorization that subsequently happens with AV, we notice that it is $\left| \frac{\lambda_{n-1}}{\lambda_{n-2}} \right|$ that dictates how fast the component in the direction of q_{n-1} disappears from $v_{n-2}^{(k)}$. This is a ratio we also saw in the Inverse Power Method and that we noticed we could accelerate in the Rayleigh Quotient Iteration: At each iteration we should shift the matrix to $(A - \mu_k I)$ where $\mu_k \approx \lambda_{n-1}$. Since the last column of $V^{(k)}$ is supposed to be converging to q_{n-1}, it seems reasonable to use $\mu_k = v_{n-1}^{(k)T} A v_{n-1}^{(k)}$ (recall that $v_{n-1}^{(k)}$ has unit length, so this is the Rayleigh quotient.)

The above discussion motivates the iteration

$V^{(0)} := I \qquad (V^{(0)} \in \mathbb{R}^{n \times n}!)$
for $k := 0, \ldots$ convergence
  $\mu_k := v_{n-1}^{(k)T} A v_{n-1}^{(k)}$
  $(A - \mu_k I) V^{(k)} \rightarrow V^{(k+1)} R^{(k+1)} \qquad$ (QR factorization)
end for

Notice that this does not require the inverse of $(A - \mu_k I)$ to be computed, unlike in the Rayleigh Quotient Iteration. However, it does require a QR factorization, which is $O(n^3)$.

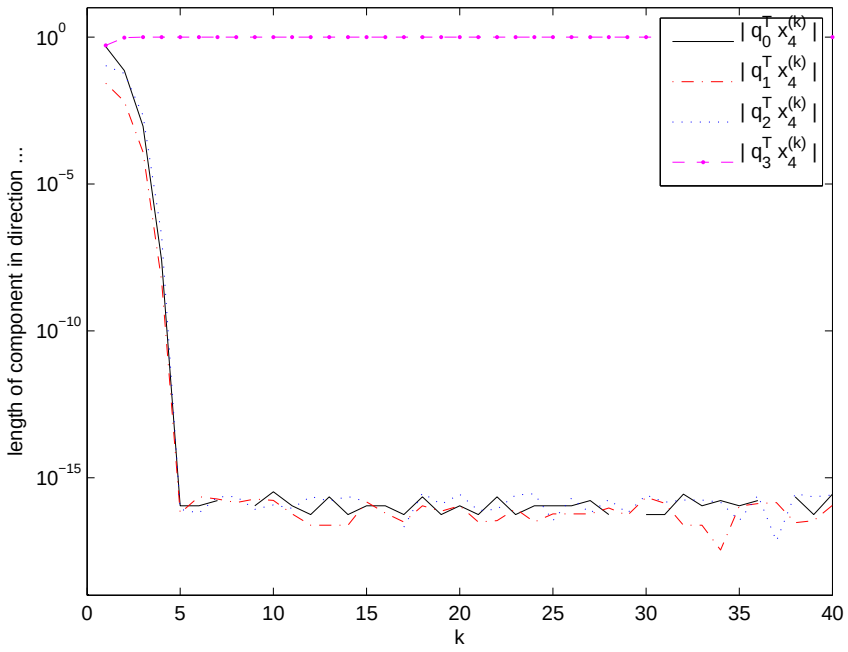

Figure 2: Convergence of the shifted subspace iteration for a 5×5 matrix.

We demonstrate the convergence in Fig. 2, which shows the execution of the algorithm with a 5×5 matrix and illustrates how $|q_j^T v_{n-1}^{(k)}|$ converge to zero as as function of k.

2 The QR Algorithm

The QR algorithm is a classic algorithm for computing all eigenvalues and eigenvectors of a matrix. While we explain it for the symmetric eigenvalue problem, it generalizes to the nonsymmetric eigenvalue problem as well.

2.1 A basic (unshifted) QR algorithm

We have informally argued that the columns of the orthogonal matrices $V^{(k)} \in \mathbb{R}^{n \times n}$ generated by the (unshifted) subspace iteration converge to eigenvectors of matrix A. (The exact conditions under which this happens have not been fully discussed.) In Figure 3 (left), we restate the subspace iteration. In it, we denote matrices $V^{(k)}$ and $R^{(k)}$ from the subspace iteration by $\hat{V}^{(k)}$ and $\hat{R}$ to distinguish them from the ones computed by the algorithm on the right. The algorithm on the left also computes the matrix $\hat{A}^{(k)} = V^{(k)T} A V^{(k)}$, a matrix that hopefully converges to Λ, the diagonal matrix with the eigenvalues of A on its diagonal. To the right is the QR algorithm. The claim is that the two algorithms compute the same quantities.

Exercise 2. *Prove that in Figure 3,* $\hat{V}^{(k)} = V^{(k)}$*, and* $\hat{A}^{(k)} = A^{(k)}$*,* $k = 1, \ldots$.

We conclude that if $\hat{V}^{(k)}$ converges to the matrix of orthonormal eigenvectors when the subspace iteration is applied to $V^{(0)} = I$, then $A^{(k)}$ converges to the diagonal matrix with eigenvectors along the diagonal.

Subspace iteration	QR algorithm
$\hat{A}^{(0)} := A$	$A^{(0)} := A$
$\hat{V}^{(0)} := I$	$V^{(0)} := I$
for $k := 0, \ldots$ until convergence	for $k := 0, \ldots$ until convergence
$A\hat{V}^{(k)} \rightarrow \hat{V}^{(k+1)}\hat{R}^{(k+1)}$ (QR factorization)	$A^{(k)} \rightarrow Q^{(k+1)}R^{(k+1)}$ (QR factorization)
$\hat{A}^{(k+1)} := \hat{V}^{(k+1)T}A\hat{V}^{(k+1)}$	$A^{(k+1)} := R^{(k+1)}Q^{(k+1)}$
	$V^{(k+1)} := V^{(k)}Q^{(k+1)}$
end for	end for

Figure 3: Basic subspace iteration and basic QR algorithm.

Subspace iteration	QR algorithm
$\hat{A}^{(0)} := A$	$A^{(0)} := A$
$\hat{V}^{(0)} := I$	$V^{(0)} := I$
for $k := 0, \ldots$ until convergence	for $k := 0, \ldots$ until convergence
$\hat{\mu}_k := v_{n-1}^{(k)\,T} A v_{n-1}^{(k)}$	$\mu_k = \alpha_{n-1,n-1}^{(k)}$
$(A - \hat{\mu}I)\hat{V}^{(k)} \rightarrow \hat{V}^{(k+1)}\hat{R}^{(k+1)}$ (QR factorization)	$A^{(k)} - \mu_k I \rightarrow Q^{(k+1)}R^{(k+1)}$ (QR factorization)
$\hat{A}^{(k+1)} := \hat{V}^{(k+1)T}A\hat{V}^{(k+1)}$	$A^{(k+1)} := R^{(k+1)}Q^{(k+1)} + \mu_k I$
	$V^{(k+1)} := V^{(k)}Q^{(k+1)}$
end for	end for

Figure 4: Basic shifted subspace iteration and basic shifted QR algorithm.

2.2 A basic shifted QR algorithm

In Figure 4 (left), we restate the subspace iteration with shifting. In it, we denote matrices $V^{(k)}$ and $R^{(k)}$ from the subspace iteration by $\hat{V}^{(k)}$ and $\hat{R}$ to distinguish them from the ones computed by the algorithm on the right. The algorithm on the left also computes the matrix $\hat{A}^{(k)} = V^{(k)T}AV^{(k)}$, a matrix that hopefully converges to Λ, the diagonal matrix with the eigenvalues of A on its diagonal. To the right is the shifted QR algorithm. The claim is that the two algorithms compute the same quantities.

Exercise 3. *Prove that in Figure 4, $\hat{V}^{(k)} = V^{(k)}$, and $\hat{A}^{(k)} = A^{(k)}$, $k = 1, \ldots$.*

We conclude that if $\hat{V}^{(k)}$ converges to the matrix of orthonormal eigenvectors when the shifted subspace iteration is applied to $V^{(0)} = I$, then $A^{(k)}$ converges to the diagonal matrix with eigenvectors along the diagonal.

The convergence of the basic shifted QR algorithm is illustrated below. Pay particular attention to the convergence of the last row and column.

$$A^{(0)} = \begin{pmatrix} 2.01131953448 & 0.05992695085 & 0.14820940917 \\ 0.05992695085 & 2.30708673171 & 0.93623515213 \\ 0.14820940917 & 0.93623515213 & 1.68159373379 \end{pmatrix} \quad A^{(1)} = \begin{pmatrix} 2.21466116574 & 0.34213192482 & 0.31816754245 \\ 0.34213192482 & 2.54202325042 & 0.57052186467 \\ 0.31816754245 & 0.57052186467 & 1.24331558383 \end{pmatrix}$$

$$A^{(2)} = \begin{pmatrix} 2.63492207667 & 0.47798481637 & 0.07654607908 \\ 0.47798481637 & 2.35970859985 & 0.06905042811 \\ 0.07654607908 & 0.06905042811 & 1.00536932347 \end{pmatrix} \quad A^{(3)} = \begin{pmatrix} 2.87588550968 & 0.32971207176 & 0.00024210487 \\ 0.32971207176 & 2.12411444949 & 0.00014361630 \\ 0.00024210487 & 0.00014361630 & 1.00000004082 \end{pmatrix}$$

$$A^{(4)} = \begin{pmatrix} 2.96578660126 & 0.18177690194 & 0.00000000000 \\ 0.18177690194 & 2.03421339873 & 0.00000000000 \\ 0.00000000000 & 0.00000000000 & 1.00000000000 \end{pmatrix} \quad A^{(5)} = \begin{pmatrix} 2.9912213907 & 0.093282073553 & 0.00000000000 \\ 0.0932820735 & 2.008778609226 & 0.00000000000 \\ 0.0000000000 & 0.000000000000 & 1.00000000000 \end{pmatrix}$$

Once the off-diagonal elements of the last row and column have converged (are sufficiently small), the problem can be deflated by applying the following theorem:

Theorem 4. *Let*

$$A = \left(\begin{array}{c|c|c|c} A_{0,0} & A_{01} & \cdots & A_{0N-1} \\ \hline 0 & A_{1,1} & \cdots & A_{1,N-1} \\ \hline \vdots & \vdots & \ddots & A_{00} \\ \hline 0 & 0 & \cdots & A_{N-1,N-1} \end{array}\right),$$

where $A_{k,k}$ are all square. Then $\lambda(A) = \cup_{k=0}^{N-1} \lambda(A_{k,k})$.

Exercise 5. *Prove the above theorem.*

In other words, once the last row and column have converged, the algorithm can continue with the submatrix that consists of the first $n-1$ rows and columns.

The problem with the QR algorithm, as stated, is that each iteration requires $O(n^3)$ operations, which is too expensive given that many iterations are required to find all eigenvalues and eigenvectors.

3 Reduction to tridiagonal form

In the next section, we will see that if $A^{(0)}$ is a tridiagonal matrix, then so are all $A^{(k)}$. This reduces the cost of each iteration from $O(n^3)$ to $O(n)$. We first show how unitary similarity transformations can be used to reduce a matrix to tridiagonal form.

3.1 Householder transformations (reflectors)

Definition 6. *Let $u \in \mathbb{R}^n$, $\tau \in \mathbb{R}$. Then $H = H(u) = I - uu^T/\tau$, where $\tau = \frac{1}{2}u^T u$, is said to be a reflector or Householder transformation.*

We observe:

- Let z be any vector that is perpendicular to u. Applying a Householder transform $H(u)$ to z leaves the vector unchanged: $H(u)z = z$.
- Let any vector x be written as $x = z + u^T x u$, where z is perpendicular to u and $u^T x u$ is the component of x in the direction of u. Then $H(u)x = z - u^T x u$.

This can be interpreted as follows: The space perpendicular to u acts as a "mirror": any vector in that space (along the mirror) is not reflected, while any other vector has the component that is orthogonal to the space (the component outside and orthogonal to the mirror) reversed in direction. Notice that a reflection preserves the length of the vector. Also, it is easy to verify that:

1. $HH = I$ (reflecting the reflection of a vector results in the original vector);
2. $H = H^T$, and so $H^T H = HH^T = I$ (a reflection is an orthogonal matrix and thus preserves the norm); and
3. if $H_0, \cdots, H_{k-1}$ are Householder transformations and $Q = H_0 H_1 \cdots H_{k-1}$, then $Q^T Q = QQ^T = I$ (an accumulation of reflectors is an orthogonal matrix).

As part of the reduction to condensed form operations, given a vector x we will wish to find a Householder transformation, $H(u)$, such that $H(u)x$ equals a vector with zeroes below the first element: $H(u)x = \mp\|x\|_2 e_0$ where e_0 equals the first column of the identity matrix. It can be easily checked that choosing $u = x \pm \|x\|_2 e_0$

yields the desired $H(u)$. Notice that any nonzero scaling of u has the same property, and the convention is to scale u so that the first element equals one. Let us define $[u, \tau, h] = \text{Hous}(x)$ to be the function that returns u with first element equal to one, $\tau = \frac{1}{2}u^T u$, and $h = H(u)x$.

3.2 Algorithm

The first step towards computing the eigenvalue decomposition of a symmetric matrix is to reduce the matrix to tridiagonal form.

The basic algorithm for reducing a symmetric matrix to tridiagonal form, overwriting the original matrix with the result, can be explained as follows. We assume that symmetric A is stored only in the lower triangular part of the matrix and that only the diagonal and subdiagonal of the symmetric tridiagonal matrix is computed, overwriting those parts of A. Finally, the Householder vectors used to zero out parts of A overwrite the entries that they annihilate (set to zero).

- Partition $A \rightarrow \left(\begin{array}{c|c} \alpha_{11} & a_{21}^T \\ \hline a_{21} & A_{22} \end{array}\right)$.
- Let $[u_{21}, \tau, a_{21}] := \text{Hous}(a_{21})$.[1]
- Update

$$\left(\begin{array}{c|c} \alpha_{11} & a_{21}^T \\ \hline a_{21} & A_{22} \end{array}\right) := \left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & H \end{array}\right)\left(\begin{array}{c|c} \alpha_{11} & a_{21}^T \\ \hline a_{21} & A_{22} \end{array}\right)\left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & H \end{array}\right) = \left(\begin{array}{c|c} \alpha_{11} & a_{21}^T H \\ \hline Ha_{21} & HA_{22}H \end{array}\right)$$

  where $H = H(u_{21})$. Note that $a_{21} := Ha_{21}$ need not be executed since this update was performed by the instance of Hous above.[2] Also, a_{12}^T is not stored nor updated. Finally, only the lower triangular part of $HA_{22}H$ is computed, overwriting A_{22}. The update of A_{22} warrants closer scrutiny:

$$\begin{aligned}
A_{22} &:= (I - \frac{1}{\tau}u_{21}u_{21}^T)A_{22}(I - \frac{1}{\tau}u_{21}u_{21}^T) \\
&= (A_{22} - \frac{1}{\tau}u_{21}\underbrace{u_{21}^T A_{22}}_{y_{21}^T})(I - \frac{1}{\tau}u_{21}u_{21}^T) \\
&= A_{22} - \frac{1}{\tau}u_{21}y_{21}^T - \frac{1}{\tau}\underbrace{Au_{21}}_{y_{21}}u_{21}^T + \frac{1}{\tau^2}u_{21}\underbrace{y_{21}^T u_{21}}_{2\beta}u_{21}^T \\
&= A_{22} - \left(\frac{1}{\tau}u_{21}y_{21}^T + \frac{\beta}{\tau^2}u_{21}u_{21}^T\right) - \left(\frac{1}{\tau}y_{21}u_{21}^T + \frac{\beta}{\tau^2}u_{21}u_{21}^T\right) \\
&= A_{22} - u_{21}\underbrace{\frac{1}{\tau}\left(y_{21}^T + \frac{\beta}{\tau}u_{21}^T\right)}_{w_{21}^T} - \underbrace{\frac{1}{\tau}\left(y_{21} + \frac{\beta}{\tau}u_{21}\right)}_{w_{21}}u_{21}^T \\
&= \underbrace{A_{22} - u_{21}w_{21}^T - w_{21}u_{21}^T}_{\begin{array}{c}\text{symmetric} \\ \text{rank-2 update}\end{array}}.
\end{aligned}$$

- Continue this process with the updated A_{22}.

[1] Note that the semantics here indicate that a_{21} is overwritten by Ha_{21}.

[2] In practice, the zeros below the first element of Ha_{21} are not actually written. Instead, the implementation overwrites these elements with the corresponding elements of the vector u_{21}.

Algorithm: $[A] := \text{TRIRED\_UNB}(b, A)$

Partition $A \rightarrow \left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right)$

where A_{TL} is 0×0

while $m(A_{TL}) < m(A)$ **do**

Repartition

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c|c|c} A_{00} & a_{01} & A_{02} \\ \hline a_{10}^T & \alpha_{11} & a_{12}^T \\ \hline A_{20} & a_{21} & A_{22} \end{array}\right)$$

where α_{11} is a scalar

$[u_{21}, \tau, a_{21}] := \text{Hous}(a_{21})$

$y_{21} := A_{22} u_{21}$

$\beta := u_{21}^T y_{21}/2$

$w_{21} := (y_{21} - \beta u_{21}/\tau)/\tau$

$A_{22} := A_{22} - \text{TRIL}(u_{21} w_{21}^T + w_{21} u_{21}^T)$ (symmetric rank-2 update)

Continue with

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \leftarrow \left(\begin{array}{c|c|c} A_{00} & a_{01} & A_{02} \\ \hline a_{10}^T & \alpha_{11} & a_{12}^T \\ \hline A_{20} & a_{21} & A_{22} \end{array}\right)$$

endwhile

Figure 5: Basic algorithm for reduction of a symmetric matrix to tridiagonal form.

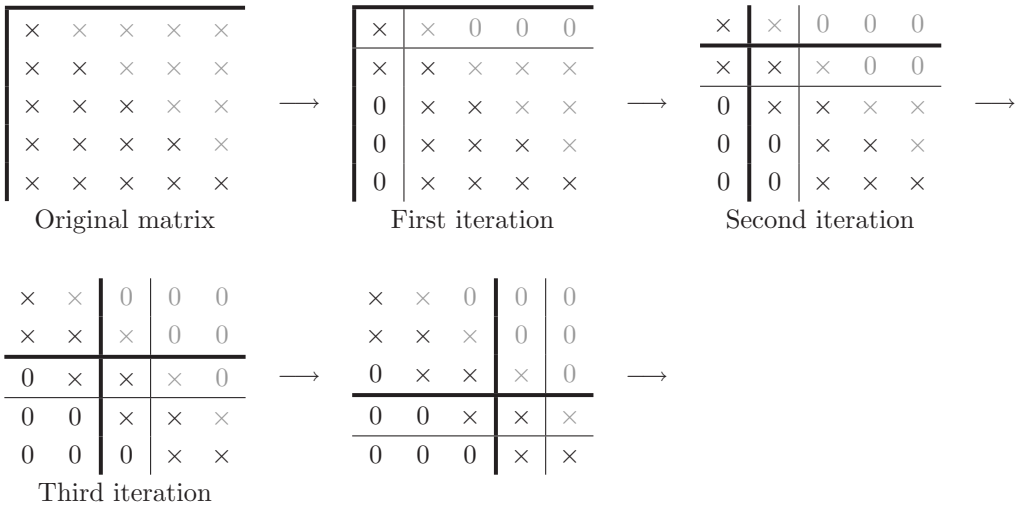

Figure 6: Illustration of reduction of a symmetric matrix to tridiagonal form. The ×s denote nonzero elements in the matrix. The gray entries above the diagonal are not actually updated.

This is captured in the algorithm in Figure 5. It is also illustrated in Figure 6.

The total cost for reducing $A \in \mathbb{R}^{n\times n}$ is approximately

$$\sum_{k=0}^{n-1} \left(4(n-k-1)^2\right) \text{ flops} \approx \frac{4}{3}n^3 \text{flops}.$$

4 The QR algorithm with a tridiagonal matrix

We are now ready to describe an algorithm for the QR algorithm with a tridiagonal matrix.

4.1 Givens' rotations

First, we introduce Givens' rotations. Given a vector $x = \begin{pmatrix} \chi_1 \\ \chi_2 \end{pmatrix} \in \mathbb{R}^2$, there exists an orthogonal matrix G such that $G^T x = \begin{pmatrix} \pm\|x\|_2 \\ 0 \end{pmatrix}$. The Householder transformation is one example of such a matrix G. An alternative is the Givens' rotation: $G = \begin{pmatrix} \gamma & -\sigma \\ \sigma & \gamma \end{pmatrix}$ where $\gamma^2 + \sigma^2 = 1$. (Notice that γ and σ can be thought of as the cosine and sine of an angle.) Then

$$G^T G = \begin{pmatrix} \gamma & -\sigma \\ \sigma & \gamma \end{pmatrix}^T \begin{pmatrix} \gamma & -\sigma \\ \sigma & \gamma \end{pmatrix} = \begin{pmatrix} \gamma & \sigma \\ -\sigma & \gamma \end{pmatrix} \begin{pmatrix} \gamma & -\sigma \\ \sigma & \gamma \end{pmatrix} = \begin{pmatrix} \gamma^2+\sigma^2 & -\gamma\sigma+\gamma\sigma \\ \gamma\sigma-\gamma\sigma & \gamma^2+\sigma^2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

which means that a Givens' rotation is unitary.

Now, if $\gamma = \chi_1/\|x\|_2$ and $\sigma = \chi_2/\|x\|_2$, then $\gamma^2 + \sigma^2 = (\chi_1^2 + \chi_2^2)/\|x\|_2^2 = 1$ and

$$\begin{pmatrix} \gamma & -\sigma \\ \sigma & \gamma \end{pmatrix}^T \begin{pmatrix} \chi_1 \\ \chi_2 \end{pmatrix} = \begin{pmatrix} \gamma & \sigma \\ -\sigma & \gamma \end{pmatrix} \begin{pmatrix} \chi_1 \\ \chi_2 \end{pmatrix} = \begin{pmatrix} (\chi_1^2+\chi_2^2)/\|x\|_2 \\ (\chi_1\chi_2-\chi_1\chi_2)/\|x\|_2 \end{pmatrix} = \begin{pmatrix} \|x\|_2 \\ 0 \end{pmatrix}.$$

5 QR factorization of a tridiagonal matrix, Part I

Now, consider the 4×4 tridiagonal matrix

$$\begin{pmatrix} \alpha_{0,0} & \alpha_{0,1} & 0 & 0 \\ \alpha_{1,0} & \alpha_{1,1} & \alpha_{1,2} & 0 \\ 0 & \alpha_{2,1} & \alpha_{2,2} & \alpha_{2,3} \\ 0 & 0 & \alpha_{3,2} & \alpha_{3,3} \end{pmatrix}$$

From $\begin{pmatrix} \alpha_{0,0} \\ \alpha_{1,0} \end{pmatrix}$ one can compute $\gamma_{1,0}$ and $\sigma_{1,0}$ so that $\begin{pmatrix} \gamma_{1,0} & -\sigma_{1,0} \\ \sigma_{1,0} & \gamma_{1,0} \end{pmatrix}^T \begin{pmatrix} \alpha_{0,0} \\ \alpha_{1,0} \end{pmatrix} = \begin{pmatrix} \hat{\alpha}_{0,0} \\ 0 \end{pmatrix}$. Then

$$\left(\begin{array}{cc|c|c} \hat{\alpha}_{0,0} & \hat{\alpha}_{0,1} & \hat{\alpha}_{0,2} & 0 \\ 0 & \hat{\alpha}_{1,1} & \hat{\alpha}_{1,2} & 0 \\ \hline 0 & \alpha_{2,1} & \alpha_{2,2} & \alpha_{2,3} \\ \hline 0 & 0 & \alpha_{3,2} & \alpha_{3,3} \end{array}\right) = \left(\begin{array}{cc|cc} \gamma_{1,0} & \sigma_{1,0} & 0 & 0 \\ -\sigma_{1,0} & \gamma_{1,0} & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right) \left(\begin{array}{cc|c|c} \alpha_{0,0} & \alpha_{0,1} & 0 & 0 \\ \alpha_{1,0} & \alpha_{1,1} & \alpha_{1,2} & 0 \\ \hline 0 & \alpha_{2,1} & \alpha_{2,2} & \alpha_{2,3} \\ \hline 0 & 0 & \alpha_{3,2} & \alpha_{3,3} \end{array}\right)$$

Next, from $\begin{pmatrix} \hat{\alpha}_{1,1} \\ \alpha_{2,1} \end{pmatrix}$ one can compute $\gamma_{2,1}$ and $\sigma_{2,1}$ so that $\begin{pmatrix} \gamma_{2,1} & -\sigma_{2,1} \\ \sigma_{2,1} & \gamma_{2,1} \end{pmatrix}^T \begin{pmatrix} \hat{\alpha}_{1,1} \\ \alpha_{2,1} \end{pmatrix} = \begin{pmatrix} \hat{\hat{\alpha}}_{1,1} \\ 0 \end{pmatrix}$.
Then

$$\left(\begin{array}{c|ccc} \hat{\alpha}_{0,0} & \hat{\alpha}_{0,1} & \hat{\alpha}_{0,2} & 0 \\ \hline 0 & \hat{\hat{\alpha}}_{1,1} & \hat{\hat{\alpha}}_{1,2} & \hat{\hat{\alpha}}_{1,3} \\ \hline 0 & 0 & \hat{\alpha}_{2,2} & \hat{\alpha}_{2,3} \\ \hline 0 & 0 & \alpha_{3,2} & \alpha_{3,3} \end{array}\right) = \left(\begin{array}{c|ccc} 1 & 0 & 0 & 0 \\ \hline 0 & \gamma_{2,1} & \sigma_{2,1} & 0 \\ 0 & -\sigma_{2,1} & \gamma_{2,1} & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \left(\begin{array}{c|ccc} \hat{\alpha}_{0,0} & \hat{\alpha}_{0,1} & \hat{\alpha}_{0,2} & 0 \\ \hline 0 & \hat{\alpha}_{1,1} & \hat{\alpha}_{1,2} & 0 \\ 0 & \alpha_{2,1} & \alpha_{2,2} & \alpha_{2,3} \\ \hline 0 & 0 & \alpha_{3,2} & \alpha_{3,3} \end{array}\right)$$

Finally, from $\begin{pmatrix} \hat{\alpha}_{2,2} \\ \alpha_{3,2} \end{pmatrix}$ one can compute $\gamma_{3,2}$ and $\sigma_{3,2}$ so that $\begin{pmatrix} \gamma_{3,2} & -\sigma_{3,2} \\ \sigma_{3,2} & \gamma_{3,2} \end{pmatrix}^T \begin{pmatrix} \hat{\alpha}_{2,2} \\ \alpha_{3,2} \end{pmatrix} = \begin{pmatrix} \hat{\hat{\alpha}}_{2,2} \\ 0 \end{pmatrix}$.
Then

$$\left(\begin{array}{cc|cc} \hat{\alpha}_{0,0} & \hat{\alpha}_{0,1} & \hat{\alpha}_{0,2} & 0 \\ 0 & \hat{\hat{\alpha}}_{1,1} & \hat{\hat{\alpha}}_{1,2} & \hat{\hat{\alpha}}_{1,3} \\ \hline 0 & 0 & \hat{\hat{\alpha}}_{2,2} & \hat{\hat{\alpha}}_{2,3} \\ 0 & 0 & 0 & \hat{\alpha}_{3,3} \end{array}\right) = \left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 1 & 0 & \gamma_{3,2} & \sigma_{3,2} \\ 0 & 1 & -\sigma_{3,2} & \gamma_{3,2} \end{array}\right) \left(\begin{array}{cc|cc} \hat{\alpha}_{0,0} & \hat{\alpha}_{0,1} & \hat{\alpha}_{0,2} & 0 \\ 0 & \hat{\hat{\alpha}}_{1,1} & \hat{\hat{\alpha}}_{1,2} & \hat{\hat{\alpha}}_{1,3} \\ \hline 0 & 0 & \hat{\alpha}_{2,2} & \hat{\alpha}_{2,3} \\ 0 & 0 & \alpha_{3,2} & \alpha_{3,3} \end{array}\right)$$

The matrix Q is the orthogonal matrix that results from multiplying the different Givens' rotations together:

$$Q = \left(\begin{array}{cc|cc} \gamma_{1,0} & -\sigma_{1,0} & 0 & 0 \\ \sigma_{1,0} & \gamma_{1,0} & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right) \left(\begin{array}{c|ccc} 1 & 0 & 0 & 0 \\ \hline 0 & \gamma_{2,1} & -\sigma_{2,1} & 0 \\ 0 & \sigma_{2,1} & \gamma_{2,1} & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & \gamma_{3,2} & -\sigma_{3,2} \\ 0 & 0 & \sigma_{3,2} & \gamma_{3,2} \end{array}\right). \tag{2}$$

However, it is typically not explicitly formed.

The next question is how compute RQ given the QR factorization of the tridiagonal matrix discussed in Section 5:

$$\underbrace{\left(\begin{array}{cc|cc} \hat{\alpha}_{0,0} & \hat{\alpha}_{0,1} & \hat{\alpha}_{0,2} & 0 \\ 0 & \hat{\hat{\alpha}}_{1,1} & \hat{\hat{\alpha}}_{1,2} & \hat{\hat{\alpha}}_{1,3} \\ \hline 0 & 0 & \hat{\hat{\alpha}}_{2,2} & \hat{\hat{\alpha}}_{2,3} \\ 0 & 0 & 0 & \hat{\alpha}_{3,3} \end{array}\right) \left(\begin{array}{cc|cc} \gamma_{1,0} & -\sigma_{1,0} & 0 & 0 \\ \sigma_{1,0} & \gamma_{1,0} & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right)}_{\left(\begin{array}{cc|cc} \tilde{\alpha}_{0,0} & \tilde{\alpha}_{0,1} & \hat{\alpha}_{0,2} & 0 \\ \tilde{\alpha}_{1,0} & \tilde{\hat{\alpha}}_{1,1} & \hat{\hat{\alpha}}_{1,2} & \hat{\hat{\alpha}}_{1,3} \\ \hline 0 & 0 & \hat{\hat{\alpha}}_{2,2} & \hat{\hat{\alpha}}_{2,3} \\ 0 & 0 & 0 & \hat{\alpha}_{3,3} \end{array}\right)} \left(\begin{array}{c|ccc} 1 & 0 & 0 & 0 \\ \hline 0 & \gamma_{2,1} & -\sigma_{2,1} & 0 \\ 0 & \sigma_{2,1} & \gamma_{2,1} & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & \gamma_{3,2} & -\sigma_{3,2} \\ 0 & 0 & \sigma_{3,2} & \gamma_{3,2} \end{array}\right)$$

$$\underbrace{\left(\begin{array}{cc|cc} \tilde{\alpha}_{0,0} & \tilde{\alpha}_{0,1} & \hat{\alpha}_{0,2} & 0 \\ \tilde{\alpha}_{1,0} & \tilde{\hat{\alpha}}_{1,1} & \hat{\hat{\alpha}}_{1,2} & \hat{\hat{\alpha}}_{1,3} \\ \hline 0 & 0 & \hat{\hat{\alpha}}_{2,2} & \hat{\hat{\alpha}}_{2,3} \\ 0 & 0 & 0 & \hat{\alpha}_{3,3} \end{array}\right)}_{\left(\begin{array}{c|ccc} \tilde{\alpha}_{0,0} & \tilde{\tilde{\alpha}}_{0,1} & \cancel{\tilde{\alpha}_{0,2}} & 0 \\ \hline \tilde{\alpha}_{1,0} & \tilde{\tilde{\alpha}}_{1,1} & \tilde{\alpha}_{1,2} & \hat{\hat{\alpha}}_{1,3} \\ 0 & \tilde{\alpha}_{2,1} & \tilde{\alpha}_{2,2} & \hat{\hat{\alpha}}_{2,3} \\ \hline 0 & 0 & 0 & \hat{\alpha}_{3,3} \end{array}\right)} \underbrace{\left(\begin{array}{c|ccc} \tilde{\alpha}_{0,0} & \tilde{\tilde{\alpha}}_{0,1} & \cancel{\tilde{\alpha}_{0,2}} & 0 \\ \hline \tilde{\alpha}_{1,0} & \tilde{\tilde{\alpha}}_{1,1} & \tilde{\tilde{\alpha}}_{1,2} & \cancel{\tilde{\alpha}_{1,3}} \\ \hline 0 & \tilde{\alpha}_{2,1} & \tilde{\tilde{\alpha}}_{2,2} & \tilde{\alpha}_{2,3} \\ 0 & 0 & \tilde{\alpha}_{3,2} & \tilde{\alpha}_{3,3} \end{array}\right)}_{\left(\begin{array}{cc|cc} \tilde{\alpha}_{0,0} & \tilde{\tilde{\alpha}}_{0,1} & \cancel{\tilde{\alpha}_{0,2}} & 0 \\ \tilde{\alpha}_{1,0} & \tilde{\tilde{\alpha}}_{1,1} & \tilde{\tilde{\alpha}}_{1,2} & \cancel{\tilde{\alpha}_{1,3}} \\ \hline 0 & \tilde{\alpha}_{2,1} & \tilde{\tilde{\alpha}}_{2,2} & \tilde{\alpha}_{2,3} \\ 0 & 0 & \tilde{\alpha}_{3,2} & \tilde{\alpha}_{3,3} \end{array}\right)}}.$$

A symmetry argument can be used to motivate that $\tilde{a}_{0,2} = \tilde{a}_{1,3} = 0$.

6 The implictly shifted QR algorithm

6.1 The Implicit Q Theorem

The following theorem sets up one of the most remarkable algorithms in numerical linear algebra, which allows us to greatly simplify the implementation of the shifted QR algorithm where A is tridiagonal.

Theorem 7 (Implicit Q Theorem)**.** *Let $A, B \in \mathbb{R}^{n\times n}$ where B is tridiagonal and has only positive elements on its first subdiagonal and assume there exists an orthogonal matrix Q such that $Q^T AQ = B$. Then Q and B are uniquely determined by A and the first column of Q.*

Proof: Notice that $AQ = QB$. Let $Q = \left(\begin{array}{c|c|c} q_0 & q_1 & Q_2 \end{array}\right)$ and $B = \left(\begin{array}{c|c|c} \psi_{00} & \star & \star \\ \hline \psi_{10} & \psi_{11} & \star \\ \hline 0 & \psi_{21}e_0 & B_{22} \end{array}\right)$ and focus on the first column of both sides of $AQ = QB$: $Aq_0 = \left(\begin{array}{c|c} q_0 & q_1 \end{array}\right)\left(\begin{array}{c} \psi_{00} \\ \hline \psi_{10} \end{array}\right) = \psi_{00}q_0 + \psi_{10}q_0$. By orthogonality of q_0 and q_1 we find that $\psi_{00} = q_0^T Aq_0$ and $\psi_{10}q_1 = \hat{q}_1 = Aq_0 - \psi_{00}q_0$. Since $\psi_{10} > 0$ we deduce that $\hat{q}_1 \neq 0$. Since $\|q_1\|_2 = 1$ we conclude that $\psi_{10} = \|\hat{q}_1\|_2$ and $q_1 = \hat{q}_1/\psi_{10}$. The point is that ψ_{00} and ψ_{10} are prescribed, as is q_1. An inductive proof can be constructed to similarly show that the rest of the elements of B and Q are uniquely determined.
□

Notice the similarity between the above proof and the proof of the existence and uniqueness of the QR factorization!

Exercise 8. *Complete the above proof.*

Note: the Implicit Q Theorem also holds if B is assumed to be upperHessenberg (upper triangular and one subdiagonal nonzero) rather than tridiagonal.

6.2 The Francis QR Step

The Francis QR Step combines the steps $(A^{(k-1)} - \mu_k I) \rightarrow Q^{(k)}R^{(k)}$ and $A^{(k+1)} := R^{(k)}Q^{(k)} + \mu_k I$ into a single step.

Now, consider the 4×4 tridiagonal matrix

$$\left(\begin{array}{cccc} \alpha_{0,0} & \alpha_{0,1} & 0 & 0 \\ \alpha_{1,0} & \alpha_{1,1} & \alpha_{1,2} & 0 \\ 0 & \alpha_{2,1} & \alpha_{2,2} & \alpha_{2,3} \\ 0 & 0 & \alpha_{3,2} & \alpha_{3,3} \end{array}\right) - \mu I$$

The first Givens' rotation is computed from $\left(\begin{array}{c} \alpha_{0,0} - \mu \\ \alpha_{1,0} \end{array}\right)$, yielding $\gamma_{1,0}$ and $\sigma_{1,0}$ so that

$$\left(\begin{array}{cc} \gamma_{1,0} & -\sigma_{1,0} \\ \sigma_{1,0} & \gamma_{1,0} \end{array}\right)^T \left(\begin{array}{c} \alpha_{0,0} - \mu I \\ \alpha_{1,0} \end{array}\right)$$

From: Gene H Golub <golub@stanford.edu>
Date: Sun, 19 Aug 2007 13:54:47 -0700 (PDT)
Subject: John Francis, Co-Inventor of QR

Dear Colleagues,

For many years, I have been interested in meeting J G F Francis, one of the co-inventors of the QR algorithm for computing eigenvalues of general matrices. Through a lead provided by the late Erin Brent and with the aid of Google, I finally made contact with him.

John Francis was born in 1934 in London and currently lives in Hove, near Brighton. His residence is about a quarter mile from the sea; he is a widower. In 1954, he worked at the National Research Development Corp (NRDC) and attended some lectures given by Christopher Strachey.
In 1955,'56 he was a student at Cambridge but did not complete a degree.
He then went back to NRDC as an assistant to Strachey where he got involved in flutter computations and this led to his work on QR.

After leaving NRDC in 1961, he worked at the Ferranti Corp and then at the University of Sussex. Subsequently, he had positions with various industrial organizations and consultancies. He is now retired. His interests were quite general and included Artificial Intelligence, computer languages, systems engineering. He has not returned to numerical computation.

He was surprised to learn there are many references to his work and that the QR method is considered one of the ten most important algorithms of the 20th century. He was unaware of such developments as TeX and Math Lab. Currently he is working on a degree at the Open University.

John Francis did remarkable work and we are all in his debt. Along with the conjugate gradient method, it provided us with one of the basic tools of numerical analysis.

Gene Golub

Figure 7: Posting by the late Gene Golub in NA Digest Sunday, August 19, 2007 Volume 07 : Issue 34. An article on the ten most important algorithms of the 20th century, published in SIAM News, can be found at http://www.uta.edu/faculty/rcli/TopTen/topten.pdf.

has a zero second entry. Now, to preserve eigenvalues, any orthogonal matrix that is applied from the left must also have its transpose applied from the right. Let us compute

$$\left(\begin{array}{cc|cc} \tilde{\alpha}_{0,0} & \hat{\alpha}_{1,0} & \hat{\alpha}_{2,0} & 0 \\ \hat{\alpha}_{1,0} & \hat{\alpha}_{1,1} & \hat{\alpha}_{1,2} & 0 \\ \hline \hat{\alpha}_{2,0} & \hat{\alpha}_{2,1} & \alpha_{2,2} & \alpha_{2,3} \\ 0 & 0 & \alpha_{3,2} & \alpha_{3,3} \end{array}\right) = \left(\begin{array}{cc|cc} \gamma_{1,0} & \sigma_{1,0} & 0 & 0 \\ -\sigma_{1,0} & \gamma_{1,0} & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right) \left(\begin{array}{cc|cc} \alpha_{0,0} & \alpha_{0,1} & 0 & 0 \\ \alpha_{1,0} & \alpha_{1,1} & \alpha_{1,2} & 0 \\ \hline 0 & \alpha_{2,1} & \alpha_{2,2} & \alpha_{2,3} \\ 0 & 0 & \alpha_{3,2} & \alpha_{3,3} \end{array}\right) \left(\begin{array}{cc|cc} \gamma_{1,0} & -\sigma_{1,0} & 0 & 0 \\ \sigma_{1,0} & \gamma_{1,0} & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right).$$

Next, from $\begin{pmatrix} \hat{\alpha}_{1,0} \\ \hat{\alpha}_{2,0} \end{pmatrix}$ one can compute $\gamma_{2,0}$ and $\sigma_{2,0}$ so that $\begin{pmatrix} \gamma_{2,0} & -\sigma_{2,0} \\ \sigma_{2,0} & \gamma_{2,0} \end{pmatrix}^T \begin{pmatrix} \hat{\alpha}_{1,0} \\ \hat{\alpha}_{2,0} \end{pmatrix} = \begin{pmatrix} \tilde{\alpha}_{1,0} \\ 0 \end{pmatrix}$.
Then

$$\left(\begin{array}{c|ccc} \tilde{\alpha}_{0,0} & \tilde{\alpha}_{1,0} & 0 & 0 \\ \hline \tilde{\alpha}_{1,0} & \tilde{\alpha}_{1,1} & \hat{\tilde{\alpha}}_{2,1} & \hat{\alpha}_{3,1} \\ 0 & \hat{\tilde{\alpha}}_{2,1} & \hat{\alpha}_{2,2} & \hat{\alpha}_{2,3} \\ \hline 0 & \hat{\alpha}_{3,1} & \hat{\alpha}_{3,2} & \alpha_{3,3} \end{array}\right) = \left(\begin{array}{c|ccc} 1 & 0 & 0 & 0 \\ \hline 0 & \gamma_{2,0} & \sigma_{2,0} & 0 \\ 0 & -\sigma_{2,0} & \gamma_{2,0} & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \left(\begin{array}{c|ccc} \tilde{\alpha}_{0,0} & \hat{\alpha}_{1,0} & \hat{\alpha}_{2,0} & 0 \\ \hline \hat{\alpha}_{1,0} & \hat{\alpha}_{1,1} & \hat{\alpha}_{1,2} & 0 \\ \hat{\alpha}_{2,0} & \hat{\alpha}_{2,1} & \alpha_{2,2} & \alpha_{2,3} \\ \hline 0 & 0 & \alpha_{3,2} & \alpha_{3,3} \end{array}\right) \left(\begin{array}{c|ccc} 1 & 0 & 0 & 0 \\ \hline 0 & \gamma_{2,0} & -\sigma_{2,0} & 0 \\ 0 & \sigma_{2,0} & \gamma_{2,0} & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right)$$

again preserves eigenvalues. Finally, from $\begin{pmatrix} \hat{\alpha}_{2,1} \\ \hat{\alpha}_{3,1} \end{pmatrix}$ one can compute $\gamma_{3,1}$ and $\sigma_{3,1}$ so that

$$\begin{pmatrix} \gamma_{3,1} & -\sigma_{3,1} \\ \sigma_{3,1} & \gamma_{3,1} \end{pmatrix}^T \begin{pmatrix} \hat{\alpha}_{2,1} \\ \hat{\alpha}_{3,1} \end{pmatrix} = \begin{pmatrix} \tilde{\alpha}_{2,1} \\ 0 \end{pmatrix}.$$

Then

$$\left(\begin{array}{cc|cc} \tilde{\alpha}_{0,0} & \tilde{\alpha}_{1,0} & 0 & 0 \\ \tilde{\alpha}_{1,0} & \tilde{\alpha}_{1,1} & \tilde{\alpha}_{2,1} & 0 \\ \hline 0 & \tilde{\alpha}_{2,1} & \tilde{\alpha}_{2,2} & \tilde{\alpha}_{2,3} \\ 0 & 0 & \tilde{\alpha}_{3,2} & \tilde{\alpha}_{3,3} \end{array}\right) = \left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 1 & 0 & \gamma_{3,2} & \sigma_{3,2} \\ 0 & 1 & -\sigma_{3,2} & \gamma_{3,2} \end{array}\right) \left(\begin{array}{cc|cc} \tilde{\alpha}_{0,0} & \tilde{\alpha}_{1,0} & 0 & 0 \\ \tilde{\alpha}_{1,0} & \tilde{\alpha}_{1,1} & \hat{\tilde{\alpha}}_{2,1} & \hat{\alpha}_{3,1} \\ \hline 0 & \hat{\tilde{\alpha}}_{2,1} & \hat{\alpha}_{2,2} & \hat{\alpha}_{2,3} \\ 0 & \hat{\alpha}_{3,1} & \hat{\alpha}_{3,2} & \alpha_{3,3} \end{array}\right) \left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 1 & 0 & \gamma_{3,1} & -\sigma_{3,1} \\ 0 & 1 & \sigma_{3,1} & \gamma_{3,1} \end{array}\right)$$

The matrix Q is the orthogonal matrix that results from multiplying the different Givens' rotations together:

$$Q = \left(\begin{array}{cc|cc} \gamma_{1,0} & -\sigma_{1,0} & 0 & 0 \\ \sigma_{1,0} & \gamma_{1,0} & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right) \left(\begin{array}{c|ccc} 1 & 0 & 0 & 0 \\ \hline 0 & \gamma_{2,0} & -\sigma_{2,0} & 0 \\ 0 & \sigma_{2,0} & \gamma_{2,0} & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & \gamma_{3,1} & -\sigma_{3,1} \\ 0 & 0 & \sigma_{3,1} & \gamma_{3,1} \end{array}\right).$$

It is important to note that the first columns of Q is given by $\begin{pmatrix} \gamma_{1,0} \\ \sigma_{1,0} \\ 0 \\ 0 \end{pmatrix}$, which is exactly the same first column had Q been computed as in Section 5 (Equation 2). Thus, by the Implicit Q Theorem, the tridiagonal matrix that results from this approach is equal to the tridiagonal matrix that would be computed by applying the QR factorization from Section 5 with $A - \mu I$, $A - \mu I \rightarrow QR$ followed by the formation of $RQ + \mu I$ using the algorithm for computing RQ in Section **??**.

The successive elimination of elements $\hat{\alpha}_{i+1,i}$ is often referred to as chasing the bulge while the entire process that introduces the bulge and then chases it is known as a Francis Implicit QR Step. Obviously, the method generalizes to matrices of arbitrary size, as illustrated in Fig. 8. An algorithm for the chasing of the bulge is given in Fig. 9. (Note that in those figures T is used for A, something that needs to be made consistent eventually.)

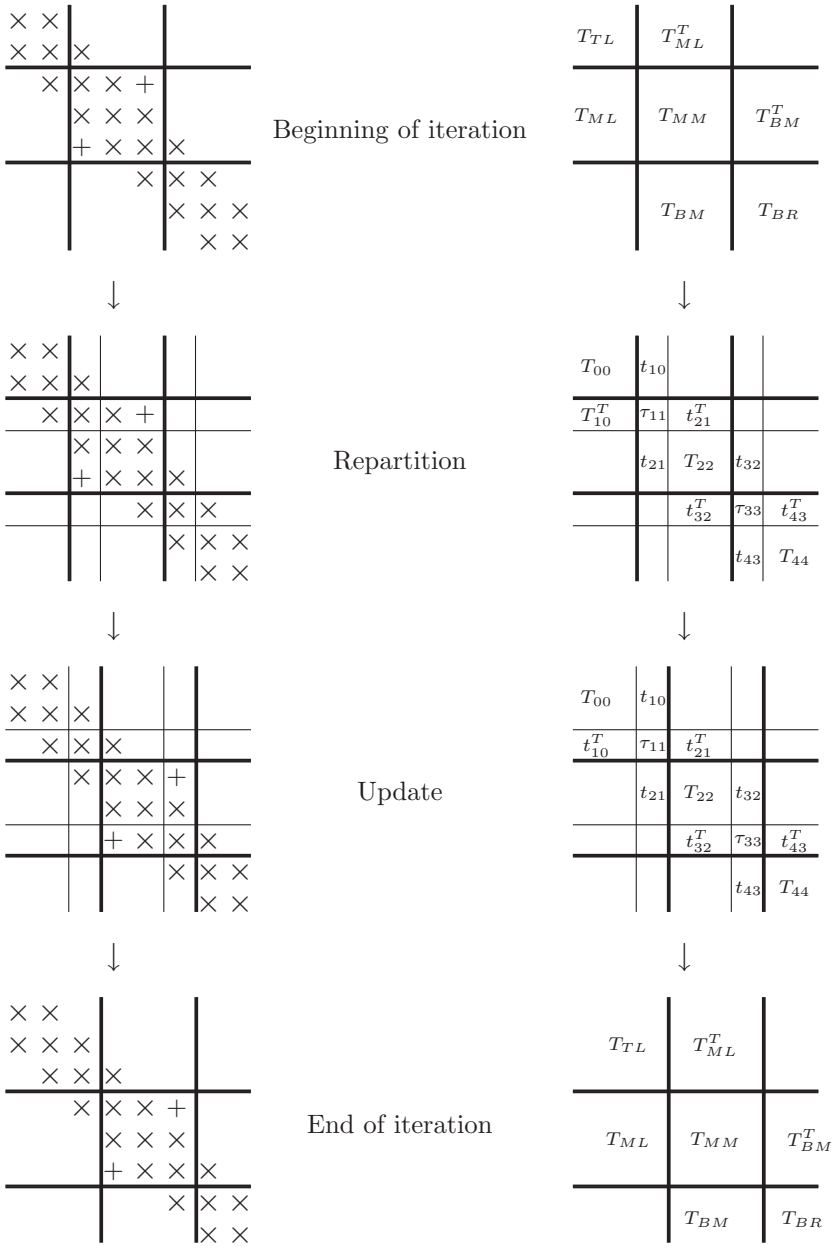

Figure 8: One step of "chasing the bulge" in the implicitly shifted symmetric QR algorithm.

6.3 A complete algorithm

This last section shows how one iteration of the QR algorithm can be performed on a tridiagonal matrix by implicitly shifting and then "chasing the bulge". All that is left to complete the algorithm is to note that

- The shift μ_k can be chosen to equal $\alpha_{n-1,n-1}$ (the last element on the diagonal, which tends to converge to the eigenvalue smallest in magnitude).
- If an element of the subdiagonal (and corresponding element on the superdiagonal) becomes small enough, it can be considered to be zero and the problem deflates (decouples) into two smaller tridiagonal

Algorithm: $T := \text{CHASEBULGE}(T)$

Partition $T \rightarrow \left(\begin{array}{c|c|c} T_{TL} & \star & \star \\ \hline T_{ML} & T_{MM} & \star \\ \hline 0 & T_{BM} & T_{BR} \end{array}\right)$

where T_{TL} is 0×0 and T_{MM} is 3×3

while $m(T_{BR}) > 0$ **do**

Repartition

$$\left(\begin{array}{c|c|c} T_{TL} & \star & 0 \\ \hline T_{ML} & T_{MM} & \star \\ \hline 0 & T_{BM} & T_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c|c|c|c|c} T_{00} & \star & 0 & 0 & 0 \\ \hline t_{10}^T & \tau_{11} & \star & 0 & 0 \\ \hline 0 & t_{21} & T_{22} & \star & 0 \\ \hline 0 & 0 & t_{32}^T & \tau_{33} & \star \\ \hline 0 & 0 & 0 & t_{43} & T_{44} \end{array}\right)$$

where τ_{11} and τ_{33} are scalars
(during final step, τ_{33} is 0×0)

Compute (γ, σ) s.t. $G_{\gamma,\sigma}^T t_{21} = \begin{pmatrix} \tau_{21} \\ 0 \end{pmatrix}$, and **assign** $t_{21} = \begin{pmatrix} \tau_{21} \\ 0 \end{pmatrix}$

$T_{22} = G_{\gamma,\sigma}^T T_{22} G_{\gamma,\sigma}$

$t_{32}^T = t_{32}^T G_{\gamma,\sigma}$ (not performed during final step)

Continue with

$$\left(\begin{array}{c|c|c} T_{TL} & \star & 0 \\ \hline T_{ML} & T_{MM} & \star \\ \hline 0 & T_{BM} & T_{BR} \end{array}\right) \leftarrow \left(\begin{array}{c|c|c|c|c} T_{00} & \star & 0 & 0 & 0 \\ \hline t_{10}^T & \tau_{11} & \star & 0 & 0 \\ \hline 0 & t_{21} & T_{22} & \star & 0 \\ \hline 0 & 0 & t_{32}^T & \tau_{33} & \star \\ \hline 0 & 0 & 0 & t_{43} & T_{44} \end{array}\right)$$

endwhile

Figure 9: Chasing the bulge.

matrices. Small is often taken to means that $|\alpha_{i+1,i}| \leq \epsilon(|\alpha_{i,i}| + |\alpha_{i+1,i+1}|)$ where ϵ is some quantity close to the machine epsilon (unit roundoff).

- If $A = QTQ^T$ reduced A to the tridiagonal matrix T before the QR algorithm commensed, then the Givens' rotations encountered as part of the implicitly shifted QR algorithm can be applied from the right to the appropriate columns of Q so that upon completion Q is overwritten with the eigenvalues of A. Notice that applying a Givens' rotation to a pair of columns of Q requires $O(n)$ computation per Givens rotation. For each Francis implicit QR step $O(n)$ Givens' rotations are computed, making the application of Givens' rotations to Q of cost $O(n^2)$ per iteration of the implicitly shifted QR algorithm. Typically a few (2-3) iterations are needed per eigenvalue that is uncovered (by deflation) meaning that $O(n)$ iterations are needed. Thus, the QR algorithm is roughly of cost $O(n^3)$ if the eigenvalues are accumulated (in addition to the cost of forming the Q from the reduction to tridiagonal form, which takes another $O(n^3)$ operations.)