Backward Stability of the QR Algorithm

Françoise Tisseur*

Technical Report N° 239
UMR 5585 Lyon Saint-Etienne
October 1996.

Abstract

It is often said that the QR algorithm is *backward stable* because each of its component steps has been proved to be backward stable. We derive the standard Wilkinson backward error bound in modern notation, allowing for the use of the multishift QR algorithm.

Résumé

Il est souvent dit que l'algorithme QR est " *inversement stable*" parce que chacune des étapes qui le compose a été prouvée inversement stable. Nous reprenons, à l'aide de notations modernes, l'analyse de la propagation des erreurs d'arrondis proposée par Wilkinson et donnons des bornes d'erreurs inverses pour l'algorithme QR. Nos résultats prennent en compte les versions multishiftées de QR.

1 Introduction

The QR algorithm, developed first by Francis and Kublanovskaya in the sixties, is the method of choice for computing the eigenvalues of a dense real or complex matrix A of moderate size. In exact arithmetic this iterative process builds a sequence of matrices that, when it converges, tends toward the Schur form T of A. Suppose that $\widehat{T}$ is an approximation to T computed in arithmetic of precision $\mathbf{u}$. One way to measure the quality of $\widehat{T}$ is through the absolute and relative errors of $\widehat{T}$ in some appropriate norm:

$$E_{abs}(\widehat{T}) = \|\widehat{T} - T\| \ , \ E_{rel}(\widehat{T}) = \frac{\|\widehat{T} - T\|}{\|T\|},$$

which are called *forward errors*. Instead of focusing on forward error, we can ask: *For what set of data have we actually solved our problem?,* that is, we consider that $\widehat{T}$ is an exact Schur form of a perturbed matrix $A + \Delta A$

*Equipe d'Analyse Numérique, Université de St-Etienne, 23, rue Dr Paul Michelon, 42023, St-Etienne, cedex 02. e-mail :`ftisseur@anumsun1.univ-st-etienne.fr`

and ask for a bound on ΔA. The norm of ΔA is called the *backward error. Backward error analysis* is the process of bounding the backward error of a computed solution. If the backward error is no larger than uncertainties inherent in the data, then the computed solution can hardly be criticized.

Since its introduction the QR algorithm has undergone various modifications. Basic QR iterations have been replaced by implicit multishift QR iterations that depend only on Householder matrices. A way to obtain a bound for the backward error is to use rounding error analysis of Householder computation, developed first by Wilkinson thirty years ago [12, 13]. The aim of this work is to give an outline of the rounding error analysis of the QR algorithm using modern notation and a unified format as developed in [7].

The next section describes the evolution of the QR algorithm. Section 3 recalls background for finite precision arithmetic computation and error analysis of Householder transformations. Section 4 gives bounds for the backward error of the multishift QR algorithm and conclusions are presented in Section 5.

2 QR iterations

2.1 Basic QR algorithm

Let $A \in \mathbb{R}^{n\times n}$ be a matrix whose eigensystem is desired. The basic QR algorithm is as follows:

$A_1 = A$
for $k = 1, 2, \ldots$
  $A_k = Q_k R_k$ (QR factorization)
  $A_{k+1} = R_k Q_k$
end

We have

$$A_{k+1} = Q_k^T A_k Q_k = Q_k^T Q_{k-1}^T \ldots Q_1 A Q_1 \ldots Q_{k-1} Q_k.$$

Under suitable conditions [13], the sequence of orthogonally similar matrices A_k converges to quasi triangular form, revealing the eigenvalues on its diagonal blocks. Obviously, the basic QR algorithm is too inefficient to be an attractive tool, but two refinements suffice to make it competitive [13]:

1- A preliminary reduction to Hessenberg form radically decreases the cost of each QR step. The Hessenberg form is preserved under basic QR iteration.

2- The use of shifts, when properly chosen, reduces the total number of QR steps required to attain convergence.

2.2 Hessenberg QR iteration with explicit shift

The Hessenberg QR iteration with explicit shifts is described by

$$
\begin{array}{l}
H_1 = Q_0^T A Q_0 \quad \text{(Hessenberg reduction)} \\
\text{for } k = 1, 2, \ldots \\
\quad \text{determine a scalar } \mu_k \\
\quad H_k - \mu_k I \; = Q_k R_k \quad \text{(QR factorization)} \\
\quad H_{k+1} \qquad = R_k Q_k + \mu_k I \\
\text{end}
\end{array}
$$

The shifts μ_k are used to accelerate the convergence. If $\mu \equiv \mu_k$ is fixed from iteration to iteration and if the eigenvalues are ordered such that

$$|\lambda_1 - \mu| > |\lambda_2 - \mu| > \cdots > |\lambda_n - \mu|,$$

then the ith subdiagonal entry in the sequence of matrices H_k converges to zero linearly with rate

$$\frac{|\lambda_{i+1} - \mu|}{|\lambda_i - \mu|}.$$

If μ is closer to λ_n than to the other eigenvalues, the $(n-1)$st subdiagonal entry converges to zero rapidly. If $\mu = h_{nn}^{(k)}$ and if $h_{n,n-1}^{(k)}$ converges to zero, then it can be shown that the convergence is quadratic [6].

2.3 Explicit multishift QR iteration

For a given matrix A reduced to Hessenberg form $H = Q_0^T A Q_0$, m steps of the shifted QR algorithm with explicit shifts $\mu_1, \ldots, \mu_m$ produce the following relationships, where we define $H_1 = H$

$$
\begin{aligned}
H_{m+1} &= R_m Q_m + \mu_m I \\
&= Q_m^T H_m Q_m \\
&= Q_m^T Q_{m-1}^T \ldots Q_1^T H_1 Q_1 \ldots Q_{m-1} Q_m.
\end{aligned} \tag{2.1}
$$

Let us define three matrices that we will use below:

$$
\begin{gathered}
p_m(H) = (H - \mu_m I) \ldots (H - \mu_1 I), \\
Q = Q_1 \ldots Q_{m-1} Q_m, \quad R = R_m R_{m-1} \ldots R_1,
\end{gathered} \tag{2.2}
$$

so that Q is orthogonal and R is upper triangular. Then, from (2.1),

$$H_{m+1} = Q^T H Q.$$

We note that

$$
\begin{aligned}
Q^T p_m(H) &= Q_m^T \dots Q_1^T (H - \mu_m I) Q_1 \dots Q_{m-1} Q_{m-1}^T \dots Q_1^T (H - \mu_{m-1} I) \\
&\qquad \dots (H - \mu_2 I) Q_1 Q_1^T (H - \mu_1 I) \qquad (2.3)\\
&= Q_m^T (H_m - \mu_m I) Q_{m-1}^T (H_{m-1} - \mu_{m-1} I) \dots Q_1^T (H_1 - \mu_1 I) \\
&= R_m R_{m-1} \dots R_1 \\
&= R.
\end{aligned}
$$

This shows that Q in (2.2) gives a QR factorization of $p_m(H)$. If $p_m(H)$ is nonsingular, that is, if none of the shifts is an eigenvalue of H and if we normalize the diagonal of R to be real and positive, Q and R are uniquely defined by the QR factorization of $p_m(H)$. So, the Q in the QR factorization of $p_m(H)$ is essentially equal to the Q in (2.2) that is, the columns of these two matrices are identical up to a scalar multiplier of modulus unity. We can work directly with $Q = Q_1 \dots Q_{m-1} Q_m$ without forming the individual Q_i matrices and perform simultaneously m iterations of the QR algorithm. Note that H_{m+1} is again a Hessenberg matrix. This operation is called an *explicit QR iteration of multiplicity m.*

2.4 Implicit multishift QR iteration

If $p_m(H)$ is computed explicitly, the algorithm is too expensive. The implicit QR algorithm manages to perform the transformation $H_{m+1} = Q^* H Q$ more efficiently by avoiding the explicit computation of $p_m(H)$.

Let P_1 be a Householder matrix chosen to zero all but the first element of the first column of $p_m(H)$ (see section 3.2 for the construction of P_1):

$$P_1^T p_m(H) e_1 = \alpha_1 e_1, \quad e_1 = (1, 0, \dots, 0)^T. \qquad (2.4)$$

Then form $P_1^T H P_1$, which disturbs the upper Hessenberg form. The matrix $P_1^T H P_1$ is transformed back to upper Hessenberg form G using $n-2$ Householder matrices $P_2, \dots, P_{n-1}$ each having $(1,1)$ element unity. We have

$$G = P^T H P, \quad P = P_1 P_2 \dots P_{n-1}.$$

Watkins and Elsner [10] say that the transformation $P_1^T H P_1$ creates a *bulge* of size $m+1$ at the top of the matrix (see Figure 2.1) and the rest of the implicit QR iteration consists of *chasing the bulge* by a standard Hessenberg reduction (see Figure 2.2). As the columns are cleared one by one, new nonzero entries are added to the bottom of the bulge and the bulge is chased towards the bottom right of the matrix and eventually out of the matrix. The role of the bulge is to carry the shifts.

We will show that G and H_{m+1} defined by (2.1) are essentially equal. First, note that from (2.4) and the form of the P_i, we have

$$p_m(H) e_1 = P_1 \alpha_1 e_1 = P \alpha_1 e_1,$$

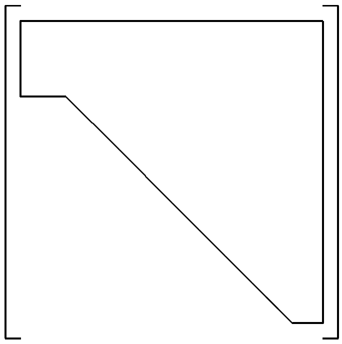

Figure 2.1: Hessenberg matrix with a bulge at the top left.

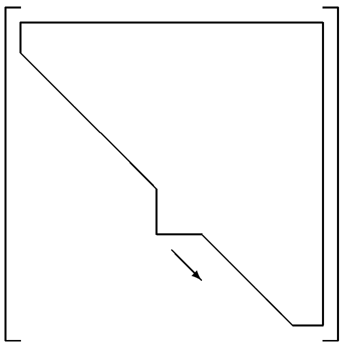

Figure 2.2: Hessenberg matrix with a bulge in the process of being chased.

and from (2.3),

$$p_m(H)e_1 = QRe_1,$$

so that the first columns of P and Q are identical up to a scalar multiplier of modulus unity.

Theorem 1 (Implicit Q theorem) *Let $H \in \mathbb{R}^{n\times n}$. Suppose that $Q \in \mathbb{R}^{n\times n}$ and $P \in \mathbb{R}^{n\times n}$ are orthogonal matrices with the property that both Q^THQ and P^THP are unreduced upper Hessenberg matrices. If $Pe_1 = Qe_1$ then $P = QD$ where $D = \mathrm{diag}(\pm 1, \dots, \pm 1)$.*

Proof. Let $K = Q^THQ$, $G = P^THP$ and $W = P^TQ = (w_1, \dots, w_n)$. Then, $WK = GW$ and for $i = 2, \dots, n$, we have

$$k_{i,i-1}w_i = Gw_{i-1} - \sum_{j=1}^{i-1} k_{j,i-1}w_j.$$

Since $w_1 = P^TQe_1 = P^TPe_1 = e_1$ and G is upper Hessenberg, we conclude that W is upper triangular. As W is orthogonal, we have $w_i = \pm e_i$ for $i = 2, \dots, n$. But $w_i = P^Tq_i$ so that $p_i = \pm q_i$ for $i = 1, \dots, n-1$. $\square$

The implicit Q theorem shows that H_{m+1} and G are essentially equal:

$$\begin{aligned} G &= P^THP \\ &= D^TQ^THQD \quad \text{with } Q \text{ as in (2.2)} \\ &= D^TH_{m+1}D. \end{aligned}$$

It follows that we can effect the transition H to H_{m+1} in $O(n^2)$ flops if we:

- compute $p_m(H)e_1$, the first column of $p_m(H) = (H - \mu_mI)\dots(H - \mu_1I)$.
- compute a Householder matrix P_1 from $p_m(H)e_1$ and form $P_1^THP_1$.
- transform $P_1^THP_1$ back to upper Hessenberg form with $n-2$ Householder matrices.

Computationally, this implicit approach is much less expensive than if we were to explicitly form $p_m(H)$. These operations form an *implicit QR iteration of multiplicity m*. The shifts are used to compute the first column of $p_m(H)$. This is done in $O(m^3)$ flops because H is upper Hessenberg and only the first $m+1$ entries of $p_m(H)e_1$ are nonzero. In general, $m \ll n$. We can use two complex conjugate shifts and stay in the real field.

From this description, we see that the analysis of the propagation of rounding errors in the implicit multishift QR iteration depends only on the rounding error analysis of similarity transformation based on Householder matrices.

3 Error analysis of Householder transformations

Computations with Householder matrices are very stable. Wilkinson [13, p. 152–160] showed that the computed Householder vector (reflector) is very close to the exact one and application of a Householder matrix to a given matrix is the exact update of a tiny normwise perturbation of the original matrix. Wilkinson's analysis assumes that we use extra (double) precision accumulation of inner products [11, 14]. We do not make this assumption in our analysis because such accumulation is rarely used nowadays. Parlett [9, Sections 6.5, 6.6] gives also a good discussion of the backward stability of a sequence of unitary transformations.

In this section, we give an error analysis of Householder matrix computation using a unified format as developed by Higham [7].

3.1 Background

In order to analyze the effect of rounding errors on a given algorithm we use the *standard model of floating point arithmetic:*

$$fl(x \operatorname{op} y) = (x \operatorname{op} y)(1+\delta), \qquad |\delta| \leq \mathbf{u}, \quad \operatorname{op} = +, -, *, /, \tag{3.1}$$

where $\mathbf{u}$ is the *unit roundoff.* In the widely used IEEE double precision arithmetic, this model holds with $\mathbf{u} = 2^{-53} \approx 1.1 \times 10^{-16}$. For certain machines that lack a guard digit in addition/subtraction, (3.1) must be replaced for $+, -$ by

$$fl(x \pm y) = x(1+\delta_1) \pm y(1+\delta_2), \qquad |\delta_i| \leq \mathbf{u},\ i = 1, 2. \tag{3.2}$$

All the results below remain valid for (3.2) with minor changes to the constant terms.

Bounds for the rounding errors in the basic complex arithmetic operations are of the same form as for the standard model (3.1) for real arithmetic, but with larger constants [7, Lemma 3.5]. We will work with real arithmetic, but our bounds are valid for complex arithmetic provided that the constants are increased appropriately.

The following lemma is a basic tool used in our analysis as a convenient way to keep track of higher order terms in the unit roundoff $\mathbf{u}$.

Lemma 1 *If $|\delta_i| \leq \mathbf{u}$, $\rho_i = \pm 1$ for $i = 1, \ldots, n$ and $n\mathbf{u} < 1$ then*

$$\prod_{i=1}^{n} (1+\delta_i)^{\rho_i} = 1 + \theta_n, \qquad |\theta_n| \leq \frac{n\mathbf{u}}{1 - n\mathbf{u}}.$$

Proof. The proof is by induction. Suppose that the result holds for $n-1$. Then, for $\rho_n = 1$ we have

$$\prod_{i=1}^{n}(1+\delta_i)^{\rho_i} = (1+\delta_n)(1+\theta_{n-1}) = 1+\theta_n$$

with $\theta_n = \delta_n + (1+\delta_n)\theta_{n-1}$, and

$$\begin{aligned} |\theta_n| &\leq \mathbf{u} + (1+\mathbf{u})\frac{(n-1)\mathbf{u}}{1-(n-1)\mathbf{u}} \\ &= \frac{\mathbf{u}(1-(n-1)\mathbf{u}) + (1+\mathbf{u})(n-1)\mathbf{u}}{1-(n-1)\mathbf{u}} \\ &\leq \frac{n\mathbf{u}}{1-n\mathbf{u}}. \end{aligned}$$

For $\rho_n = -1$ we have

$$\prod_{i=1}^{n}(1+\delta_i)^{\rho_i} = (1+\delta_n)^{-1}(1+\theta_{n-1}) = 1+\theta_n$$

with $\theta_n = \dfrac{\theta_{n-1}-\delta_n}{1+\delta_n}$, and

$$\begin{aligned} |\theta_n| &\leq \frac{\frac{(n-1)\mathbf{u}}{1-(n-1)\mathbf{u}} + \mathbf{u}}{1-\mathbf{u}} \\ &= \frac{(n-1)\mathbf{u} + \mathbf{u}(1-(n-1)\mathbf{u})}{(1-(n-1)\mathbf{u})(1-\mathbf{u})} \\ &\leq \frac{n\mathbf{u}}{1-n\mathbf{u}}. \qquad \square \end{aligned}$$

As in [7], we introduce for convenience the constant

$$\gamma_{\mathbf{u}}(n) = \frac{n\mathbf{u}}{1-n\mathbf{u}},$$

and we make the reasonable assumption that $n\mathbf{u} < 1$. Other styles of writing bounds are possible (see, for example, Forsythe and Moler [5, p. 92]), but the notations θ_n and $\gamma_{\mathbf{u}}(n)$ are convenient for keeping track of terms of second and higher order in $\mathbf{u}$. The next lemma provides the necessary rules to manipulate the $1+\theta_n$ and $\gamma_{\mathbf{u}}(n)$ terms.

Lemma 2 *For any positive integer k, let θ_k and $\gamma_{\mathbf{u}}(k)$ be defined as before. The following relations hold for positive integers j and k :*

$$(1+\theta_k)(1+\theta_j) = 1+\theta_{k+j}, \tag{3.3}$$

$$
1 + \theta_k \over 1 + \theta_j} = \begin{cases} 1 + \theta_{k+j} & j \le k, \\ 1 + \theta_{k+2j} & j > k, \end{cases} \tag{3.4}
$$

$$
k\gamma_{\mathbf{u}}(j) \le \gamma_{\mathbf{u}}(jk), \tag{3.5}
$$

$$
\gamma_{\mathbf{u}}(k) + \mathbf{u} \le \gamma_{\mathbf{u}}(k+1), \tag{3.6}
$$

$$
\gamma_{\mathbf{u}}(k) + \gamma_{\mathbf{u}}(j) + \gamma_{\mathbf{u}}(k)\gamma_{\mathbf{u}}(j) \le \gamma_{\mathbf{u}}(k+j). \tag{3.7}
$$

Proof. Relations (3.3), (3.5), and (3.6) are trivial. For the division result, we have

$$
\frac{1 + \theta_k}{1 + \theta_j} = 1 + \theta \quad \text{with} \quad \theta = \frac{\theta_k - \theta_j}{1 + \theta_j}.
$$

So,

$$
|\theta| \le \frac{\frac{k\mathbf{u}}{1-k\mathbf{u}} + \frac{j\mathbf{u}}{1-j\mathbf{u}}}{1 - \frac{j\mathbf{u}}{1-j\mathbf{u}}} = \frac{(k+j)\mathbf{u} - 2kj\mathbf{u}^2}{(1-2j\mathbf{u})(1-k\mathbf{u})}.
$$

If $j \le k$,

$$
\frac{(k+j)\mathbf{u} - 2kj\mathbf{u}^2}{(1-2j\mathbf{u})(1-k\mathbf{u})} \le \frac{(k+j)\mathbf{u}}{1-(k+j)\mathbf{u}} = \gamma_{\mathbf{u}}(k+j),
$$

else,

$$
\begin{aligned}
\frac{(k+j)\mathbf{u} - 2kj\mathbf{u}^2}{(1-2j\mathbf{u})(1-k\mathbf{u})} &= \frac{(k+j)\mathbf{u} - 2kj\mathbf{u}^2}{1-(2j+k)\mathbf{u} + 2jk\ u^2} \\
&\le \frac{(k+2j)\mathbf{u}}{1-(k+2j)\mathbf{u}} = \gamma_{\mathbf{u}}(k+2j).
\end{aligned}
$$

For the last result,

$$
\begin{aligned}
\gamma_{\mathbf{u}}(k) + \gamma_{\mathbf{u}}(j) + \gamma_{\mathbf{u}}(k)\gamma_{\mathbf{u}}(j) &= \frac{k\mathbf{u}(1-j\mathbf{u}) + j\mathbf{u}(1-k\mathbf{u}) + kj\mathbf{u}^2}{(1-k\mathbf{u})(1-j\mathbf{u})} \\
&= \frac{(k+j)\mathbf{u} - kj\mathbf{u}^2}{1-(k+j)\mathbf{u} + kj\mathbf{u}^2} \\
&\le \frac{(k+j)\mathbf{u}}{1-(k+j)\mathbf{u}} = \gamma_{\mathbf{u}}(k+j). \qquad \square
\end{aligned}
$$

In the following, computed quantities will be denoted by hats. The next lemma provides an error bound for a floating point inner product.

Lemma 3 *Consider the inner product $x^T y$ where $x, y \in \mathbb{R}^n$. Then,*

$$
fl(x^T y) = x^T(y + \Delta y), \qquad |\Delta y| \le \gamma_{\mathbf{u}}(n)|y|.
$$

Proof. Using the standard model (3.1), we have

$$
\widehat{s}_1 = fl(x_1 y_1) = x_1 y_1 (1 + \delta_1), \quad |\delta_1| \le \mathbf{u},
$$

and for $k = 2, \dots, n$,

$$\widehat{s}_k = fl(\widehat{s}_{k-1} + x_k y_k) = (\widehat{s}_{k-1} + x_k y_k(1+\delta_k))(1+\varepsilon_k), \quad |\delta_k| \le \mathbf{u}, |\varepsilon_k| \le \mathbf{u}.$$

Overall, we have

$$fl(x^T y) = x_1 y_1 (1+\delta_1) \prod_{j=2}^{n} (1+\varepsilon_j) + \sum_{k=2}^{n} x_k y_k (1+\delta_k) \prod_{j=k}^{n} (1+\varepsilon_j)$$

and applying Lemma 2 we obtain

$$fl(x^T y) = x_1 y_1 (1+\theta_n) + x_2 y_2 (1+\theta_n^{'}) + x_3 y_3 (1+\theta_{n-1}) + \cdots + x_n y_n (1+\theta_2),$$

where $|\theta_k| \le \gamma_{\mathbf{u}}(k), k = 2, \dots, n$. This shows that the computed inner product is the exact one for a perturbed set of data $x_1, x_2, \dots, x_n, y_1(1+\theta_n), y_2(1+\theta_n^{'}), \dots, y_n(1+\theta_2)$. Thus,

$$fl(x^T y) = x^T(y + \Delta y), \quad |\Delta y| \le \gamma_{\mathbf{u}}(n)|y|. \qquad \square$$

3.2 Elementary Householder matrices in floating point

A Householder matrix is a matrix of the form

$$P = I - \beta v v^T, \qquad \beta = \frac{2}{v^T v} \quad (v \ne 0).$$

Given any two vectors $x, y \in \mathbb{R}^n$ with $\|x\|_2 = \|y\|_2$ and $x \ne y$ there exists a Householder matrix $P \in \mathbb{R}^{n \times n}$ such that $Px = y$. It is easy to check that $v = x - y$ defines such a P. Our interest is in the choice $y = \pm\|x\|_2 e_1$.

Lemma 4 *Let $x \in \mathbb{R}^m$. Consider the construction of $\beta \in \mathbb{R}$ and $v \in \mathbb{R}^m$ such that $Px = -\operatorname{sign}(x_1)\|x\|_2 e_1$ where $P = I - \beta v v^T$ is a Householder matrix with $\beta = \dfrac{2}{v^T v}$:*

$$\begin{aligned} v &= x, \\ s &= \operatorname{sign}(x_1)\|x\|_2, \\ v_1 &= v_1 + s, \\ \beta &= \frac{1}{s v_1}. \end{aligned}$$

In floating point arithmetic, the computed $\widehat{\beta}$ and $\widehat{v}$ satisfy

$$\widehat{\beta} = \beta + \Delta\beta, \quad |\Delta\beta| \le \gamma_{\mathbf{u}}(2m+3)|\beta|, \tag{3.8}$$

$$\widehat{v} = v + \Delta v, \quad |\Delta v| \le \gamma_{\mathbf{u}}(m+1)|v|. \tag{3.9}$$

Proof. It is straightforward to verify that the given formulas for v and β do indeed construct the desired Householder matrix. Lemma 3 gives

$$fl(x^T x) = x^T x(1+\theta_m), \quad |\theta_m| \le \gamma_{\mathbf{u}}(m),$$

so that

$$fl(\|x\|_2) = (x^T x)^{1/2}(1+\theta_m)^{1/2}(1+\delta), \quad |\delta| \le \mathbf{u},$$

and then, for $m > 1$ we can show that

$$\widehat{s} = s(1+\theta_m^{'}) \quad \text{where} \quad |\theta_m^{'}| \le \gamma_{\mathbf{u}}(m).$$

For notational convenience, we define $w = v_1 + s$. Hence,

$$\begin{aligned}
\widehat{w} &= (v_1 + \widehat{s})(1+\delta) \quad \text{with} \quad |\delta| \le \mathbf{u} \\
&= w(1+\theta_{m+1}) \quad \text{where} \quad |\theta_{m+1}| \le \gamma_{\mathbf{u}}(m+1).
\end{aligned}$$

Finally, using equality (3.3) and (3.4) of Lemma 2,

$$\begin{aligned}
fl(\widehat{s}\widehat{w}) &= sw(1+\theta_{2m+2}) \quad \text{where} \quad |\theta_{2m+2}| \le \gamma_{\mathbf{u}}(2m+2), \\
fl(1/(\widehat{s}\widehat{w})) &= \frac{1+\delta}{sw(1+\theta_{2m+2})} = \beta(1+\theta_{4m+5}),
\end{aligned}$$

where $|\theta_{4m+5}| \le \gamma_{\mathbf{u}}(4m+5)$. $\square$

Lemma 5 *Let $x, v \in \mathbb{R}^m$. We consider the construction of $y = Px = x - \beta(v^T x)v$ where $P = I - \beta v v^T$ is a Householder matrix. Assume that the computation is performed using computed $\widehat{\beta}$ and $\widehat{v}$ that satisfy* (3.8) *and* (3.9). *The computed $\widehat{y}$ satisfies*

$$\widehat{y} = P(x + \Delta x), \quad \|\Delta x\|_2 \le \gamma_{\mathbf{u}}(cm)\|x\|_2,$$

where c is a small integer constant.

Proof. We give a sketch of the proof, omitting the details. By Lemma 3, we have

$$fl(\widehat{v}^T x) = \widehat{v}^T(x + \Delta x), \quad |\Delta x| \le \gamma_{\mathbf{u}}(m)|x|,$$

so that

$$fl(\widehat{v}^T x) = v^T x + w, \quad |w| \le \gamma_{\mathbf{u}}(2m+1)|v||v^T||x|,$$

and

$$\widehat{w} := fl(\widehat{\beta}\widehat{v}(\widehat{v}^T x)) = \beta v(v^T x) + \Delta w, \quad |\Delta w| \le \gamma_{\mathbf{u}}(4m+5)\beta|v||v^T||x|.$$

Then

$$\widehat{y} = fl(x - \widehat{w}) = x - \beta v(v^T x) + \Delta y, \quad |\Delta y| \le \mathbf{u}|x|\gamma_{\mathbf{u}}(4m+6)\beta|v||v^T||x|.$$

As $\|I+\beta|v||v^T|\|_2 \le 3$, we have,

$$\|\Delta y\|_2 \le \gamma_{\mathbf{u}}(12m+18)\ \|x\|_2. \qquad \square$$

We mention that one can also consider the Householder matrix P such that $Px = +\operatorname{sign}(x_1)\|x\|_2 e_1$ (the other choice of sign) provided that the formulae in Lemma 4 are modified appropriately to avoid cancellation [7, p. 383]. Lemma 5 and our subsequent results remain true for this alternative choice of P.

In a computer implementation of the Householder transformation $y = Px = -\operatorname{sign}(x_1)\|x\|_2 e_1$, we do not explicitly compute $y_2, \dots, y_m$, but rather set them to zero. It is easy to see that the bound of Lemma 5 is still valid in this situation (in fact, by forcing $y_2 = \cdots = y_m = 0$ we are making the error smaller).

Now, we consider premultiplication of a matrix by an approximate Householder matrix.

Lemma 6 *Let $A \in \mathbb{R}^{m\times m}$ and let $P = I - \beta vv^T \in \mathbb{R}^{m\times m}$ be a Householder matrix. Assume that the computation of PA is performed using computed $\widehat{\beta}$ and $\widehat{v}$ that satisfy* (3.8) *and* (3.9), *respectively. Then,*

$$fl(PA) = P(A+\Delta A), \qquad \|\Delta A\|_F \le \gamma_{\mathbf{u}}(cm), \tag{3.10}$$

$$fl(P^TAP) = P^T(A+\Delta A)P, \qquad \|\Delta A\|_F \le \gamma_{\mathbf{u}}(c'm). \tag{3.11}$$

Proof. Let a_j be the jth column of A. By Lemma 5 we have

$$fl(Pa_j) = P(a_j + \Delta a_j), \qquad \|\Delta a_j\|_2 \le \gamma_{\mathbf{u}}(cm)\|a_j\|_2.$$

Hence, $\widehat{B} := fl(PA) = P(A+\Delta A)$, where

$$\|\Delta A\|_F \le \sum_{j=1}^{m} \|\Delta a_j\|_2^2 \le \gamma_{\mathbf{u}}(cm)^2 \sum_{j=1}^{m} \|a_j\|_2^2 = \gamma_{\mathbf{u}}(cm)^2\|A\|_F^2,$$

that is, $\|\Delta A\|_F \le \gamma_{\mathbf{u}}(cm)\|A\|_F$. Similarly,

$$\begin{aligned} fl(\widehat{B}P) &= (\widehat{B} + \Delta\widehat{B})P \\ &= P^T(A + \Delta A + P\Delta\widehat{B})P \end{aligned}$$

with $\|\Delta\widehat{B}\|_F \le \gamma_{\mathbf{u}}(cm)\|\widehat{B}\|_F$ and

$$\begin{aligned} \|\Delta A + P\Delta\widehat{B}\|_F &\le \|\Delta A\|_F + \gamma_{\mathbf{u}}(cm)\|\widehat{B}\|_F \\ &\le (\gamma_{\mathbf{u}}(cm) + \gamma_{\mathbf{u}}(cm) + \gamma_{\mathbf{u}}(cm)^2)\|A\|_F \\ &\le \gamma_{\mathbf{u}}(2cm)\|A\|_F, \end{aligned}$$

by (3.7). $\square$

Note that Lemma 6 expresses the computed result as an exact orthogonal similarity transformation of a perturbed matrix, that is, in backward error terms.

We mention that for the computation of PA, Higham [7, Lemma 18.3] shows that $fl(PA) = P(A + \Delta A)$ with a componentwise bound of the form

$$|\Delta A| \leq \gamma_{\mathbf{u}}(cm^2)G|A|, \quad \|G\|_F = 1.$$

For two sided transformations P^TAP, this bound must be modified to the form

$$|\Delta A| \leq \gamma_{\mathbf{u}}(cm^2)G_1|A|G_2, \quad \|G_1\|_F = 1, \quad \|G_2\|_F = 1,$$

which is essentially the same as (3.11). Hence, there is no advantage in using componentwise analysis in the backward error analysis of the QR algorithm.

4 Error analysis of the QR algorithm

4.1 The Hessenberg reduction

The implicit multishift QR algorithm needs to be applied to a matrix A reduced to upper Hessenberg form H, that is, a matrix such that $h_{ij} = 0$ for $i > j + 1$. The matrix H is said to be unreduced if $h_{i+1,i} \neq 0$ for $i = 1, \ldots, n-1$.

Let

$$U_0^T A U_0 = H, \qquad U_0^T U_0 = I$$

be the Hessenberg reduction of $A \in \mathbb{R}^{n\times n}$. The transformation U_0 is a product of $n-2$ Householder matrices where the role of the kth matrix is to zero the kth column below the subdiagonal.

Theorem 2 *We consider the reduction of $A \in \mathbb{R}^{n\times n}$ to Hessenberg form: $H = U_0^T A U_0$. The computed Hessenberg form satisfies*

$$\widehat{H} = W^T(A + \Delta A)W \quad \textit{where} \quad \|\Delta A\|_F \leq \gamma_{\mathbf{u}}(cn^2),$$

where W is orthogonal and c is a small integer constant.

Proof. Let P_k be the matrix used to zero the kth column below the subdiagonal. Lemma 6 gives

$$\widehat{A}_1 = fl(P_1^T A P_1) = P_1^T(A + \Delta A_1)P_1, \qquad \|\Delta A_1\|_F \leq \gamma_{\mathbf{u}}(c(n-1))\|A\|_F.$$

Then,

$$\begin{aligned} \widehat{A}_2 = fl(P_2^T \widehat{A}_1 P_2) &= P_2^T(\widehat{A}_1 + \Delta\widehat{A}_1)P_2 \text{ where } \|\Delta\widehat{A}_1\|_F \leq \gamma_{\mathbf{u}}(c(n-2))\|\widehat{A}_1\|_F, \\ &= P_2^T P_1^T(A + \Delta A_2)P_1P_2 \end{aligned}$$

where

$$
\begin{aligned}
\|\Delta A_2\|_F &\le \|\Delta A_1\|_F + \|\Delta \widehat{A}_1\|_F \\
&\le [\gamma_{\mathbf{u}}(c(n-1)) + \gamma_{\mathbf{u}}(c(n-2))(1+\gamma_{\mathbf{u}}(c(n-1)))]\|A\|_F \\
&\le \gamma_{\mathbf{u}}(c[(n-1)+(n-2)])\|A\|_F,
\end{aligned}
$$

using (3.7). Continuing in this fashion, we find

$$
\begin{aligned}
\widehat{H} &= fl(P_{n-2}^T \widehat{A}_{n-3} P_{n-2}) \\
&= P_{n-2}^T P_{n-3}^T \dots P_1^T (A + \Delta A) P_1 \dots P_{n-3} P_{n-2} \\
&:= W^T (A + \Delta A) W,
\end{aligned}
$$

where W is an orthogonal matrix and $\|\Delta A\|_F \le \gamma_{\mathbf{u}}(q)$, with

$$
q = c \sum_{k=1}^{n-2} (n-k) \le c\, \frac{n(n-1)}{2} \le c' n^2. \qquad \square
$$

4.2 Multishift QR algorithm

The next result gives a bound for the backward error of an implicit multishift QR iteration. Before each QR step, we set to zero all subdiagonal elements that satisfy a certain deflation criterion[1] and then find the largest nonnegative r and the largest nonnegative s such that

$$
H = \begin{array}{cc}
\begin{pmatrix} H_{11} & H_{12} & H_{13} \\ 0 & H_{22} & H_{23} \\ 0 & 0 & H_{33} \end{pmatrix} & \begin{array}{c} q \\ r \\ s \end{array} \\
\begin{array}{ccc} \quad q & \quad r & \quad s \end{array} &
\end{array}, \qquad q + r + s = n,
$$

where H_{33} is upper triangular and H_{22} is unreduced. Then, an implicit multishift QR iteration is performed on $H_{22} \in \mathbb{R}^{r \times r}$, producing

$$
G = P^T H P, \qquad P^T P = I,
$$

where $P = P_1 \dots P_{r-2}$ is a product of $r-2$ Householder matrices.

[1] The classical deflation criterion is $|h_{i,i-1}| \le (|h_{ii}| + |h_{i-1,i-1}|)$ [8]. This criterion yields backward stability but has no relation to the accuracy of the computed eigenvalues. Recently, a new criterion based on mathematical considerations has been developed in [1]. This criterion takes into account the size of the subdiagonal and off-diagonal entries and the distance between two consecutive diagonal elements.

Theorem 3 *One iteration of the implicit QR algorithm of multiplicity m applied to a Hessenberg matrix $H \in \mathbb{R}^{r\times r}$ computes the Hessenberg form $\widehat{G}$ such that*

$$\widehat{G} = U^T(H+\Delta H)U, \qquad \|\Delta H\|_F \le \gamma_{\mathbf{u}}(crm)\ \|H\|_F,$$

where U is orthogonal.

Proof. We give a sketch of the proof, omitting the details. One iteration of the implicit QR algorithm is done in two stages. First, we construct the matrix $P_1^T H P_1$ which creates a bulge at the top of the current matrix H. The computed matrix $\widehat{H}_1$ satisfies

$$\widehat{H}_1 = fl(P_1^T H P_1) = P_1(H+\Delta H_1)P_1, \quad \|\Delta H_1\|_F \le \gamma_{\mathbf{u}}(cm)\|H\|_F,$$

The second step consists of chasing the bulge of size $m+1$ with $r-2$ Householder matrices. Overall, the computed matrix satisfies

$$\widehat{G} := U^T(H+\Delta H)U \quad \|\Delta H\|_F \le \gamma_{\mathbf{u}}(c^{'}rm)\|H\|_F.$$

The proof is very similar to that of Theorem 2. □

Theorem 4 *The implicit QR algorithm of multiplicity m applied to a Hessenberg matrix $H \in \mathbb{R}^{n\times n}$ computes a quasi-triangular matrix $\widehat{T}$ such that*

$$\widehat{T} = Q^T(H+\Delta H)Q, \quad Q^TQ = I$$

and

$$\|\Delta H\|_F \le \sum_{j=1}^{p} \gamma_{\mathbf{u}}(cm\sum_{k=1}^{j} r_k)\ \|H\|_F$$

where p is the number of iterations and r_k is the size of the problem at iteration k.

Proof. Let $\widehat{H}_k$ be the computed matrix at the start of iteration k. Suppose that any elements $\widehat{h}^{(k)}_{i+1,i}$ that are set to zero on the kth stage satisfy

$$|\widehat{h}^{(k)}_{i+1,i}| \le c^{'}\mathbf{u}\|\widehat{H}_k\|_F.$$

We have

$$\widehat{H}_{k+1} = Q_k^T(\widehat{H}_k + \Delta H_k)Q_k$$

where ΔH_k contains the eventually neglected subdiagonal elements and the rounding errors made during the similarity transformations corresponding to the kth implicit Hessenberg QR iteration of multiplicity m and satisfies:

$$\begin{aligned}\|\Delta H_k\|_F &\le \gamma_{\mathbf{u}}(cr_km)\ \|\widehat{H}_k\|_F + c^{'}\mathbf{u}\|\widehat{H}_k\|_F \\ &\le \gamma_{\mathbf{u}}(c^{''}r_km)\ \|\widehat{H}_k\|_F\end{aligned}$$

But

$$\begin{aligned}
\|\widehat{H}_k\|_F &\le \|\widehat{H}_{k-1}\|_F + \|\Delta H_{k-1}\|_F \\
&\le (1+\gamma_{\mathbf{u}}(cr_{k-1}m))\|\widehat{H}_{k-1}\|_F \\
&\le (1+\gamma_{\mathbf{u}}(cr_{k-1}m))(1+\gamma_{\mathbf{u}}(cr_{k-2}m))\|\widehat{H}_{k-2}\|_F \\
&\vdots \\
&\le \prod_{i=1}^{k-1}(1+\gamma_{\mathbf{u}}(cr_i m))\|H\|_F \\
&\le (1+\gamma_{\mathbf{u}}(cm\sum_{i=1}^{k-1} r_i))\|H\|_F,
\end{aligned}$$

so that, using (3.7),

$$\|\Delta H_k\|_F \le \gamma_{\mathbf{u}}(cm\sum_{i=1}^{k} r_i)\|H\|_F.$$

Finally, if p denotes the total number of QR iterations, the last iterated matrix satisfies

$$\begin{aligned}
\widehat{H}_{p+1} &= Q_p^T(\widehat{H}_p + \Delta H_p)Q_p \\
&= Q_p^T(Q_{p-1}^T(\widehat{H}_{p-1} + \Delta H_{p-1})Q_{p-1} + \Delta H_p)Q_p \\
&= Q_p^T Q_{p-1}^T(\widehat{H}_{p-1} + \Delta H_{p-1} + Q_{p-1}\Delta H_p Q_{p-1}^T)Q_{p-1}Q_p \\
&\vdots \\
&= Q_p^T \dots Q_1^T \left(\widehat{H}_1 + \Delta H_1 + \sum_{i=2}^{p}\left(\left(\prod_{j=1}^{i-1} Q_j\right)\Delta H_i \left(\prod_{j=1}^{i-1} Q_j\right)^T\right)\right) Q_1 \dots Q_p \\
&:= Q^T(H+\Delta H)Q
\end{aligned}$$

where

$$\|\Delta H\|_F \le \sum_{j=1}^{p}\|\Delta H_j\|_F \le \sum_{j=1}^{p}\gamma_{\mathbf{u}}(cm\sum_{k=1}^{j} r_k)\|H\|_F. \qquad \square$$

We see that the backward error bound for the implicit multishift QR algorithm depends essentially on the size of the problem at each iteration and the number of QR iterations.

Suppose that we use the double shift QR algorithm (also called Francis QR algorithm). Empirical observations have shown that, on average, only two iterations are required before the lower 1×1 or 2×2 trailing principal submatrix decouples. Then, with $m=2$ and $p \simeq 2n$, we certainly have

$$\|\Delta H\|_F \le \gamma_{\mathbf{u}}(c' n^3)\ \|H\|_F.$$

5 Conclusions

We have given bounds for the backward error for the implicit multishift QR algorithm which show that the algorithm is backward stable. This pleasing result is due to the use of similarity transformations based on Householder matrices. Of course, our results are applicable only when the QR algorithm converges; for examples of non-convergence of the shifted QR algorithm, see Day [4].

Nowadays, most linear algebra libraries use aggregated Householder transformations. One form is the WY representation of Bischof and Van Loan [3]. This kind of representation allows the use of matrix-matrix multiplications routines. Our result remains true with the use of the WY technique; moreover the use of fast BLAS-3 routines for applying the updates affects stability only through the constants in the backward errors bounds. See [7, Section 18.4] for details of error analysis for the WY technique.

Our error analysis produces a *bound* on the backward error. If we are interested only in $\widehat{T}$, then the backward error can be defined by

$$\mu_\nu(A) = \min_{Q^TQ} \|A - Q\widehat{T}Q^T\|_\nu, \quad \nu = 2, F.$$

Unfortunately, this Procrustes problem has no known closed-form solution, so it is a nontrivial computational task to evaluate $\mu_\nu(A)$.

Our analysis does not address the error in the computed eigenvalues. To bound these forward errors one needs perturbation theory, and to compute forward error bounds efficiently one needs to use condition estimation techniques [2].

Acknowledgements. I would like to thank Nick Higham for helpful comments and suggestions.

References

[1] Mario Ahues, Alain Largillier, and Françoise Tisseur. On partitioning and stopping criteria for the QR algorithm. Research Report N° 206, URA CNRS 740 Lyon St-Etienne, September 1995.

[2] Zhaojun Bai, James W. Demmel, and Alan McKenney. On computing condition numbers for the nonsymmetric eigenproblem. *ACM Trans. Math. Software*, 19(2):197–212–223, 1993.

[3] Christian H. Bischof and Charles F. Van Loan. The WY representation for products of Householder matrices. *SIAM J. Sci. Stat. Comput.*, 8(1):s2–s13, 1987.

[4] David Day. How the QR algorithm fails to converge and how to fix it. Sandia technical report 96-0913J, Sandia National Laboratory, April 1996.

[5] George E. Forsythe and Cleve B. Moler. *Computer Solution of Linear Algebraic Systems.* Prentice-Hall, Englewood Cliffs, NJ, USA, 1967.

[6] Gene H. Golub and Charles F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, MD, USA, second edition, 1989.

[7] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1996.

[8] R. S. Martin, G. Peters, and J. H. Wilkinson. The QR Algorithm for Real Hessenberg Matrices. *Numer. Math.*, 14:219–231, 1970.

[9] Beresford N. Parlett. *The Symmetric Eigenvalue Problem.* Prentice-Hall, Englewood Cliffs, NJ, USA, 1980.

[10] D. S. Watkins and L. Elsner. Chasing algorithms for the eigenvalue problem. *SIAM J. Matrix Anal. Appl.*, 12(2):374–384, 1991.

[11] J. H. Wilkinson. Error analysis of eigenvalue techniques based on orthogonal transformations. *J. Soc. Indust. Appl. Math.*, 10(1):162–195, 1962.

[12] J. H. Wilkinson. *Rounding Errors in Algebraic Processes.* Notes on Applied Science No. 32, Her Majesty's Stationery Office, London, 1963. Also published by Prentice-Hall, Englewood Cliffs, NJ, USA. Reprinted by Dover, New York, 1994.

[13] J. H. Wilkinson. *The Algebraic Eigenvalue Problem.* Oxford University Press, 1965.

[14] J. H. Wilkinson. Error analysis of transformations based on the use of matrices of the form $I - 2ww^H$. In Louis B. Rall, editor, *Error In Digital Computation*, volume 2, pages 77–101. Wiley, New York, 1965.