THE AUSTRALIAN NATIONAL UNIVERSITY

TR-CS-98-11

An Error Analysis of a Unitary Hessenberg QR Algorithm

Michael Stewart

December 1998

Joint Computer Science Technical Report Series

Department of Computer Science
Faculty of Engineering and Information Technology

Computer Sciences Laboratory
Research School of Information Sciences and Engineering

This technical report series is published jointly by the Department of Computer Science, Faculty of Engineering and Information Technology, and the Computer Sciences Laboratory, Research School of Information Sciences and Engineering, The Australian National University.

Please direct correspondence regarding this series to:

Technical Reports
Department of Computer Science
Faculty of Engineering and Information Technology
The Australian National University
Canberra ACT 0200
Australia

or send email to:

`Technical.Reports@cs.anu.edu.au`

A list of technical reports, including some abstracts and copies of some full reports may be found at:

`http://cs.anu.edu.au/techreports/`

Recent reports in this series:

TR-CS-98-10 Peter Strazdins. *Optimal load balancing techniques for block-cyclic decompositions for matrix factorization*. September 1998.

TR-CS-98-09 Jim Grundy, Martin Schwenke, and Trevor Vickers (editors). *International Refinement Workshop & Formal Methods Pacific '98 — Work-in-progress papers of IRW/FMP'98, 29 September – 2 October 1998, Canberra, Australia*. September 1998.

TR-CS-98-08 Jim Grundy and Malcolm Newey (editors). *Theorem Proving in Higher Order Logics: Emerging Trends — Proceedings of the 11th International Conference, TPHOLs'98, Canberra, Australia, September – October 1998, Supplementary Proceedings*. September 1998.

TR-CS-98-07 Peter Strazdins. *A comparison of lookahead and algorithmic blocking techniques for parallel matrix factorization*. July 1998.

TR-CS-98-06 M. Manzur Murshed. *Optimal computation of the contour of maximal elements on mesh-connected computers*. July 1998.

An Error Analysis of a Unitary Hessenberg QR Algorithm

Michael Stewart[1]

Several direct implementations of the QR algorithm for a unitary Hessenberg matrix are numerically unstable. In this paper we give an analysis showing how the instability in a particular rational form of the algorithm specialized to the case of a unimodular shift comes from two sources: loss of accuracy due to cancellation in a particular formula and a dynamic instability in the propagation of the normalization conditions on the Schur parameters and complementary parameters used to represent the matrix. The first problem can be fixed through the use of an alternate formula proposed by Gragg. The second problem can be controlled by not relying on the fact that the matrix is numerically unitary to enforce implicitly the unimodularity of the computed shift; if the shift is explicitly normalized then experiments suggest that the algorithm is stable in practice although stability cannot be proven. A third small modification, introduced to eliminate a potential for a relatively slow exponential growth in normalization errors leads to a provably stable algorithm. This stable rational algorithm for computing the eigenvalues leads directly to a stable algorithm for computing a complete eigenvalue decomposition.

1 Introduction

Any unitary Hessenberg matrix with real positive subdiagonal has the form

$$H = H(a,b) := \begin{pmatrix} -\overline{a}_0 a_1 & -\overline{a}_0 b_1 a_2 & -\overline{a}_0 b_1 b_2 a_3 & \cdots & -\overline{a}_0 b_1 \cdots b_{n-1} a_n \\ b_1 & -\overline{a}_1 a_2 & -\overline{a}_1 b_2 a_3 & \cdots & -\overline{a}_1 b_2 \cdots b_{n-1} a_n \\ & b_2 & -\overline{a}_2 a_3 & & \vdots \\ & & \ddots & \ddots & \vdots \\ & & & b_{n-1} & -\overline{a}_{n-1} a_n \end{pmatrix} \tag{1}$$

where b_k is real, $b_k > 0$, $|a_k|^2 + b_k^2 = 1$ for $1 \leq k < n$, $a_0 = 1$ and $|a_n| = 1$. The a_k are the *Schur parameters* of the unitary Hessenberg matrix and the b_k are the *complementary parameters.* The assumption that the subdiagonal is real and positive guarantees that the Schur parameters are sufficient to determine the complementary parameters and consequently the entire matrix. Throughout this paper, all references to a unitary Hessenberg matrix assume a real, positive subdiagonal.

A unitary Hessenberg matrix may also be written as a product of modified (column permuted) elementary rotations

$$H = G_1(a_1) G_2(a_2) \cdots G_n(a_n) \tag{2}$$

[1] Computer Sciences Laboratory, RSISE, Australian National University, Canberra ACT 0200, Australia, email: stewart@discus.anu.edu.au

where

$$G_j(a_j) = I_{k-1} \oplus \begin{pmatrix} -a_k & b_k \\ b_k & \overline{a}_k \end{pmatrix} \oplus I_{n-k-1}$$

for $1 \leq j < n$ and

$$G_n(a_n) = I_{n-1} \oplus (-a_n).$$

Both (1) and (2) are completely general formulations. The parameters in the elementary rotations are the same as the parameters in (1).

Direct and inverse unitary eigenvalue problems arise in signal processing, [3], where the Schur parameters are known as *reflection coefficients.* Such eigenvalue problems occur in frequency estimation, including Pisarenko's method [2]. Least squares approximation by trigonometric polynomials has a natural solution through the solution of a unitary inverse eigenvalue problem, [10, 4]. Efficient algorithms achieve a reduction in computational complexity by working with the Schur parameters and complementary parameters instead of with the entire matrix H.

Along these lines, it is possible to develop an $O(n^2)$ shifted QR algorithm for a unitary Hessenberg matrix by working with only the Schur parameters. However such an algorithm will be numerically unstable if the b_k are not used: although the Schur parameters determine the matrix mathematically, knowing the a_k to high relative accuracy does not accurately determine the corresponding b_k. If b_k is small, then $b_k = \sqrt{1 - |a_k|^2}$ will be sensitive to small relative perturbations in a_k. This is illustrated by the following expansion

$$\sqrt{1 - |a_k(1+\epsilon)|^2} = \sqrt{1 - |a_k|^2} + \frac{\mathrm{Re}(\epsilon)}{\sqrt{1 - |a_k|^2}} + O(\epsilon^2).$$

It follows that if $\sqrt{1 - |a_k|^2}$ is small for some k, then an algorithm that is backward stable in the sense of finding eigenvalues corresponding to a small perturbation of the a_k does not necessarily find eigenvalues corresponding to a small perturbation of H. Consequently the Bauer-Fike theorem does not directly give any information about the accuracy of eigenvalues. Such a "backward stable" algorithm would not be stable in the usual sense applied to matrix algorithm; the complementary parameters must be used to obtain accurate eigenvalues.

A unitary Hessenberg QR algorithm was first presented in [6]. The basic idea behind the algorithm is very natural and direct: given a set of Schur parameters and complementary parameters, the rotations that would be performed in one step of the QR algorithm can be efficiently computed with $O(n)$ operations using the Szegö recursions. One iteration of the QR algorithm results in another unitary Hessenberg matrix with Schur parameters which can be computed in a similarly efficient manner. No claims were made for the stability of this method and, as described in [6], the algorithm is unstable.

An alternate formulation of the unitary eigenvalue problem was given in [1]. It can be shown that a unitary similarity transformation gives

$$Q^H H Q = H_o H_e^H$$

where

$$H_o = G_1(a_1)G_3(a_3)G_5(a_5)\cdots G_{2\lfloor(n+1)/2\rfloor-1}(a_{2\lfloor(n+1)/2\rfloor-1})$$

and

$$H_e = G_2(a_2)G_4(a_4)G_6(a_6)\cdots G_{2\lfloor n/2\rfloor}(a_{2\lfloor n/2\rfloor})$$

with $\lfloor x\rfloor = \max\{i \in \mathbb{N} | i \leq x\}$. Consequently the eigenvalues of H are equal to the eigenvalues of the pencil $H_o - \lambda H_e$. An $O(n^2)$ implicit QR algorithm for finding the eigenvalues of $H_o - \lambda H_e$ was given in [5]. The paper also described an algorithm for reducing a general unitary matrix represented by the pencil $U - \lambda I$ to a *Schur parameter pencil*, $H_o - \lambda H_e$. Both the initial reduction procedure and the implicit QR iteration are numerically stable. In its direct use of orthogonal transformations and an implicit shift, the method can lay claim to being a very natural implementation of the QR algorithm for a unitary Hessenberg matrix.

However, despite the use of the Szegö recursions, the method of [6] is mathematically equivalent to the unsymmetric QR algorithm applied to H. Further it propagates a set of parameters that seem to be the minimum number necessary to accurately determine H and its eigenvalues in the presence of numerical errors.

While the initial Schur parameter pencil involves only the Schur parameters and complementary parameters, the implicit QR iteration preserves the block structure of H_e and H_o while destroying the rotation structure of each 2×2 block. A unitary diagonal scaling can restore this structure, but the algorithm most naturally works with a parameterization that is both mathematically and numerically redundant. So in a slightly different sense, the method of [6] can also claim to be the most natural implementation of the QR algorithm for a unitary Hessenberg matrix.

The point of this paper is to show that with a formula proposed in [7] and careful treatment of the normalization of a unimodular shift and the normalization of the Schur parameters and complementary parameters, a rational version of the algorithm from [6] can also be made numerically stable. Except for the additional emphasis on normalization, this is essentially a verification of a stability conjecture made in [7]. For completeness, we present a condensed derivation of five variants of the algorithm based on the development in [6], [8] and [7].

Let

$$\chi_k(z) := \det(zI - H_k), \qquad 1 \leq k \leq n$$

where, in MATLAB notation, $H_k = H(1:k, 1:k)$ and $\chi_0(z) \equiv 1$. The $\chi_k(z)$ are clearly monic. Using elementary properties of determinants, it is not difficult to verify that

$$\chi_k(z) \equiv z\chi_{k-1}(z) + a_k\overline{a}_{k-1}\chi_{k-1}(z) + a_k\sum_{j=1}^{k-1}\overline{a}_{j-1}(b_jb_{j+1}\cdots b_{k-1})^2\chi_{j-1}(z).$$

If we define the auxiliary polynomials

$$\tilde{\chi}_k(z) := \overline{a}_k\chi_k(z) + \sum_{j=1}^{k}\overline{a}_{j-1}(b_jb_{j+1}\cdots b_k)^2\chi_{j-1}(z)$$

then

$$\chi_k(z) \equiv z\chi_{k-1}(z) + a_k\tilde{\chi}_{k-1}(z) \tag{3}$$

and

$$\begin{aligned}
\tilde{\chi}_k(z) &\equiv \overline{a}_k(z\chi_{k-1}(z) + a_k\tilde{\chi}_{k-1}(z)) + b_k^2\left(\overline{a}_{k-1}\chi_{k-1}(z) + \sum_{j=1}^{k-1}\overline{a}_{j-1}(b_j\cdots b_{k-1})^2\chi_{j-1}(z)\right) \\
&\equiv \overline{a}_k z\chi_{k-1}(z) + (|a_k|^2 + b_k^2)\tilde{\chi}_{k-1}(z) \\
&\equiv \overline{a}_k z\chi_{k-1}(z) + \tilde{\chi}_{k-1}(z). \qquad (4)
\end{aligned}$$

Relations (3) and (4) are the *Szegö recursions.* The polynomials $\chi_k(z)$ are *Szegö polynomials* and they are orthogonal with respect to an inner product on the unit circle.

Starting with $\tilde{\chi}_0(z) \equiv 1$ and $\chi_0(z) \equiv 1$, it is simple to prove that

$$\tilde{\chi}_k(z) \equiv z^k\chi_k^*(1/z) \tag{5}$$

where $\chi_k^*(\cdot)$ represents the polynomial formed by conjugation of the coefficients of $\chi_k(\cdot)$. This follows inductively from (3) and (4) since if $\tilde{\chi}_{k-1}(z) \equiv z^{k-1}\chi_{k-1}^*(1/z)$ then

$$\begin{aligned}
\tilde{\chi}_k(z) &\equiv \overline{a}_k z\chi_{k-1}(z) + z^{k-1}\chi_{k-1}^*(1/z) \\
&\equiv z^k\left(\overline{a}_k(1/z)^{k-1}\chi_{k-1}(z) + (1/z)\chi_{k-1}^*(1/z)\right) \\
&\equiv z^k\left(\overline{a}_k\tilde{\chi}_{k-1}^*(1/z) + (1/z)\chi_{k-1}^*(1/z)\right).
\end{aligned}$$

Consequently (5) follows directly from an application of (3).

We now consider the shifted QR algorithm applied to H. For a particular choice of shift, z, let

$$H - zI = QR$$

and

$$\hat{H} = RQ + zI = Q^H HQ.$$

It is not hard to see that Q^H can be formed from a sequence of plane rotations computed to zero the subdiagonal of H so that Q is a unitary Hessenberg matrix with

$$Q = G_1(c_1)G_2(c_2)\cdots G_n(c_n)$$

for $|c_n|^2 + s_n^2 = 1$ with $s_n = 0$ and $s_j \geq 0$. Direct matrix multiplication shows that

$$b_k = s_k r_k, \qquad 1 \leq k \leq n-1 \tag{6}$$

where the r_k for $k = 1, 2, \ldots, n$ are the diagonal elements of R. Similarly, since $\hat{H} = RQ + zI$, we can verify that $\hat{H}$ is Hessenberg with subdiagonal

$$\hat{b}_k = s_k r_{k+1}, \qquad 1 \leq k \leq n-1. \tag{7}$$

We partition the equation $H - zI = QR$ as

$$\begin{pmatrix} H_k - zI & X \\ b_k e_1 e_k^H & X \end{pmatrix} = \begin{pmatrix} Q_k & X \\ X & X \end{pmatrix} \begin{pmatrix} R_k & X \\ 0 & X \end{pmatrix}$$

where each X represents an unspecified but distinct block. From the unitary Hessenberg structure of Q represented as a product of matrices $G_k(c_k)$ with $\det(G_k(c_k)) = -1$, it is not difficult to see that $\det(Q_k) = (-1)^k c_k$. Thus

$$\chi_k(z) = \det(zI - H_k) = (-1)^k \det(H_k - zI) = \det(R_k) c_k = r_1 r_2 \cdots r_k c_k.$$

If we define d_k by

$$\tilde{\chi}_k(z) = r_1 r_2 \cdots r_k d_k$$

then the Szegö recursions become

$$r_k c_k = z c_{k-1} + a_k d_{k-1} \tag{8}$$

and

$$r_k d_k = \overline{a}_k z c_{k-1} + d_{k-1}. \tag{9}$$

Define $p_k = r_k c_k$ and $q_k = r_k d_k$. Then using (6) gives

$$|p_k|^2 = r_k^2(1 - s_k^2) = r_k^2 - b_k^2$$

so that

$$r_k^2 = |p_k|^2 + b_k^2. \tag{10}$$

To form the algorithm, we put together (8), (9) and (10) together with (6) for the computation of s_k, (7) for the computation of $\hat{b}_k$ and the relations

$$c_k = p_k / r_k, \qquad d_k = q_k / r_k.$$

Together these equations allow the computation of $\hat{b}_k$, the complementary Schur parameters of $\hat{H}$, from a_k and b_k. To complete the unitary Hessenberg QR algorithm we need a formula for computing $\hat{a}_k$. We state without duplicating the proof in [6] that

$$\hat{a}_k = c_k \overline{d}_k - \overline{z} s_k^2 a_{k+1}, \qquad 1 \le k < n \tag{11}$$

and

$$\hat{a}_n = a_n.$$

With appropriate initialization for $k = 0$ and termination when $k = n$, we have the unitary Hessenberg QR algorithm as originally given in [6].

Algorithm UHQR1

```
    $z$ = shift,    $c_0 = d_0 = 1$,    $s_0 = 0$
    for $k = 1 : n - 1$
        $p_k = zc_{k-1} + a_k d_{k-1}$
        $q_k = \overline{a}_k zc_{k-1} + d_{k-1}$
        $r_k = \mathrm{sqrt}(| p_k |^2 + b_k^2)$
        $\hat{b}_{k-1} = r_k s_{k-1}$,    $c_k = p_k / r_k$
        $d_k = q_k / r_k$,    $s_k = b_k / r_k$
        $\hat{a}_k = c_k \overline{d}_k - \overline{z} s_k^2 a_{k+1}$
    end
    $p_n = z c_{n-1} + a_n d_{n-1}$
    $r_n = | p_n |$,    $\widehat{b}_{n-1} = r_n s_{n-1}$
    $\widehat{a}_n = a_n$
    if $r_n > 0$ then $c_n = \mathrm{sign}(p_n)$
              else $c_n = 1$
```

This algorithm is known to be numerically unstable. We do not attempt to fix the stability problems for such a general version of the algorithm. Instead we consider three rational variants of UHQR1, based on an algorithm described in [7], that assume a unimodular shift and propagate real, squared values where appropriate to avoid square roots. These algorithms compute the eigenvalues but do not compute an eigenvalue decomposition. Using a numerically stable rational algorithm as inspiration, we will also present a stable alternative to UHQR1 for computing a complete eigenvalue decomposition of H.

If we define

$$f_k = c_k / d_k = p_k / q_k.$$

then we may compute f_k from

$$\begin{aligned} f_k &= \frac{p_k}{q_k} \\ &= \frac{zc_{k-1} + a_k d_{k-1}}{\overline{a}_k z c_{k-1} + d_{k-1}} \\ &= \frac{z f_{k-1} + a_k}{\overline{a}_k z f_{k-1} + 1} \\ &= \frac{\overline{w}_k g_k^2}{|g_k|^2}. \end{aligned}$$

If the shift is unimodular then (5) implies that $|c_k| = |d_k|$ and consequently

$$\hat{a}_k = |d_k|^2 f_k - \overline{z} s_k^2 a_{k+1}.$$

Further $|p_k| = |q_k|$ and if we only desire eigenvalues and are not otherwise interested in computing c_k then it is only necessary to compute $|p_k|^2$, $|c_k|^2$, r_k^2 and s_k^2 instead of p_k, c_k, r_k and s_k. These observations result in the rational unitary Hessenberg QR algorithm for unimodular shifts as described in [7].

Algorithm UHQRR1

```
$f_0 \;=\; |\,c_0\,|^2 \;=\; 1, \qquad s_0^2 \;=\; 0$
for $k \;=\; 1 \;:\; n \;-\; 1$
    $w_k \;=\; z\,f_{k-1}, \qquad g_k \;=\; w_k \;+\; a_k$
    $|\,p_k\,|^2 = |\,c_{k-1}\,|^2\,|\,g_k\,|^2$
    $r_k^2 \;=\; |\,p_k\,|^2 \;+\; b_k^2, \qquad \widehat{b}_{k-1}^2 \;=\; r_k^2\,s_{k-1}^2$
    $|\,c_k\,|^2 \;=\; |\,p_k\,|^2/r_k^2, \qquad s_k^2 \;=\; b_k^2/r_k^2$
    $f_k \;=\; \overline{w}_k\,g_k^2/\,|\,g_k\,|^2$
    $\hat{a}_k \;=\; |\,c_k\,|^2\,f_k \;-\; \overline{z}\,s_k^2\,a_{k+1}$
end
$w_n \;=\; z\,f_{n-1}, \qquad g_n \;=\; w_n \;+\; a_n$
$r_n^2 \;\;=\; |\,c_{n-1}\,|^2\,|\,g_n\,|^2$
$\hat{b}_{n-1}^2 \;=\; r_n^2\,s_{n-1}^2, \qquad \hat{a}_n \;=\; a_n.$
```

UHQRR1 does not involve the use of any square roots and it is is somewhat similar to the algorithm of Reinsch for the case of a symmetric tridiagonal matrix, [9].

As the algorithm was derived with the explicit assumption that $|z| = 1$, we must assume that the shift is computed in such a way that it is unimodular except for numerical errors. In fact, since we are concerned in this paper with issues of numerical stability and not convergence, unimodularity is the only thing we will assume about the shift.

Despite this apparent indifference to the details of its computation, it is very important to note that the shift must be numerically unimodular in a very strict sense. There is a major difference between enforcing unimodularity implicitly through a computation guaranteed to produce a unimodular shift when H is exactly unitary and enforcing unimodularity through explicit normalization. Taking the Wilkinson shift as a model, we might choose the eigenvalue of

$$\begin{pmatrix} -\overline{a}_{n-2}a_{n-1}/|a_{n-2}| & -\overline{a}_{n-2}b_{n-1}a_n/|a_{n-2}| \\ b_{n-1} & -\overline{a}_{n-1}a_n \end{pmatrix} \tag{12}$$

that is closest to $-\overline{a}_{n-1}a_n$ as the shift. If $|a_{n-1}|^2 + b_{n-1}^2 = 1$ and $|a_n| = 1$ then $|z| = 1$, and we might suppose that this method is sufficient to enforce implicitly unimodularity. Unfortunately this results in a very severe instability: any deviation from unimodularity in the shift results in larger errors in the normalization condition $|\hat{a}_{n-1}|^2 + \hat{b}_{n-1}^2 = 1$ which makes the next shift even less unimodular. The result is an unstable algorithm which fails after only a few iterations. To deal with this problem we must explicitly normalize z instead of relying on the orthogonality of (12) to guarantee normalization.[2]

However, even with explicit normalization of z, UHQRR1 is unstable. On the assumption that this instability might be due to cancellation in the computation of g_k an alternate formula was proposed for its computation in [7].

[2] In fact, there are several other ways of dealing with the instability. Explicit normalization of w_k or f_k will also keep the Schur parameters normalized and hence implicitly keep the shift unimodular. However, explicitly normalizing the shift outside the loop seems to be the most natural and most efficient remedy for the problem.

Let $t_k = \overline{a}_k z f_{k-1}$. Note that $|f_{k-1}| = 1$ so that

$$g_k = zf_{k-1} + a_k = \overline{zf_{k-1}}(1 + \overline{t}_k).$$

If $\mathrm{Re}(t_k) \geq 0$ then g_k is computed to high relative accuracy by the formula $g_k = zf_{k-1} + a_k$. However, if $\mathrm{Re}(t_k) < 0$, cancellation can occur, sometimes leading to a significant loss of relative accuracy in the computed g_k.

Note that

$$\begin{aligned} g_k &= zf_{k-1} + a_k \\ &= \frac{(zf_{k-1} + a_k)\overline{(zf_{k-1} - a_k)}}{\overline{(zf_{k-1} - a_k)}} \\ &= \frac{1 - |a_k|^2 - 2i\,\mathrm{Imag}(\overline{a}_k z f_{k-1})}{\overline{(zf_{k-1} - a_k)}} \\ &= \frac{b_k^2 - 2i\,\mathrm{Imag}(t_k)}{\overline{(zf_{k-1} - a_k)}}. \end{aligned} \tag{13}$$

When $\mathrm{Re}(t_k) < 0$, (13) involves the possibility of cancellation only in the computation of $\mathrm{Imag}(t_k)$. In addition, the new equation incorporates b_k into the central recurrence of the algorithm. Given the earlier discussion of the essential nature of the information in b_k for determining the eigenvalues, this is a promising feature of the new equation.

The following algorithm is the result of these modifications.

Algorithm UHQRR2

```
$z = z \,/\, |z|$
$f_0 = |\, c_0 \,|^2 = 1, \qquad s_0^2 = 0$
**for** $k = 1 : n - 1$
    $w_k = z\, f_{k-1}, \qquad t_k = \overline{a}_k\, w_k$
    **if** $\mathrm{Re}(t_k) \geq 0$ **then** $g_k = w_k + a_k$
                         **else** $g_k = (b_k^2 - 2\,i\,\mathrm{Imag}(t_k)) \,/\, (\overline{w}_k - \overline{a}_k)$
    $|\, p_k \,|^2 = |\, c_{k-1} \,|^2 \,|\, g_k \,|^2$
    $r_k^2 = |\, p_k \,|^2 + b_k^2, \qquad \widehat{b}_{k-1}^2 = r_k^2\, s_{k-1}^2$
    $|\, c_k \,|^2 = |\, p_k \,|^2 / r_k^2, \qquad s_k^2 = b_k^2 / r_k^2$
    $f_k = \overline{w}_k\, g_k^2 / \,|\, g_k \,|^2$
    $\hat{a}_k = |\, c_k \,|^2\, f_k - \overline{z}\, s_k^2\, a_{k+1}$
**end**
$w_n = z\, f_{n-1}, \qquad t_n = \overline{a}_n\, w_n$
**if** $\mathrm{Re}(t_n) \geq 0$ **then** $g_n = w_n + a_n$
                     **else** $g_n = -\,2\,i\,\mathrm{Imag}(t_n) \,/\, (\overline{w}_n - \overline{a}_n)$
$r_n^2 = |\, c_{n-1} \,|^2 \,|\, g_n \,|^2$
$\hat{b}_{n-1}^2 = r_n^2\, s_{n-1}^2, \qquad \hat{a}_n = a_n.$
```

UHQRR2 is very reliable numerically and a careful analysis suggests that the algorithm is likely to be stable in practice. However, the analysis also reveals the apparent possibility that over the course of j QR iterations the errors in the relation $|a_k|^2 + b_k^2 = 1$ might grow exponentially at a rate of $O(\frac{9}{8}^j)$. This growth would appear in the backward error bounds.

Fortunately the 9/8 growth rate represents an upper bound on an expression that determines how the normalization errors are propagated from one iteration to the next. It is typically a very pessimistic upper bound. The circumstances under which normalization errors can be magnified are not particularly common and don't tend to be repeated over consecutive iterations. Consequently, exponential growth of normalization errors does not tend to happen in practice and has never been observed.

Nevertheless, we introduce the following algorithm incorporating a minor change to prevent any possibility of exponential error growth. The main result of this paper is a proof that the following algorithm is numerically stable and that UHQRR2 is stable if the normalization errors do not grow excessively.

Algorithm UHQRR3

```
$z = z / |z|$
$f_0 = |c_0|^2 = 1, \qquad s_0^2 = 0$
for $k = 1 : n - 1$
    $w_k = z f_{k-1}, \qquad t_k = \overline{a}_k w_k$
    if $\mathrm{Re}(t_k) \geq 0$ then $g_k = w_k + a_k$
                   else $g_k = (b_k^2 - 2i\,\mathrm{Imag}(t_k)) / (\overline{w}_k - \overline{a}_k)$
    $|p_k|^2 = |c_{k-1}|^2 |g_k|^2$
    $r_k^2 = |p_k|^2 + b_k^2$
    if $\mathrm{Re}(t_k) < 0$ and $(2b_k^2|c_{k-1}|^2/|w_k - a_k|^2 + 1)s_{k-1}^2 > 1$
        then $\hat{b}_{k-1}^2 = 1 - |\hat{a}_{k-1}|^2$
        else $\widehat{b}_{k-1}^2 = r_k^2 s_{k-1}^2$
    $|c_k|^2 = |p_k|^2/r_k^2, \qquad s_k^2 = b_k^2/r_k^2$
    $f_k = \overline{w}_k g_k^2 / |g_k|^2$
    $\hat{a}_k = |c_k|^2 f_k - \overline{z} s_k^2 a_{k+1}$
end
$w_n = z f_{n-1}, \qquad t_n = \overline{a}_n w_n$
if $\mathrm{Re}(t_n) \geq 0$ then $g_n = w_n + a_n$
               else $g_n = -2i\,\mathrm{Imag}(t_n) / (\overline{w}_n - \overline{a}_n)$
$r_n^2 = |c_{n-1}|^2 |g_n|^2$
$\hat{b}_{n-1}^2 = r_n^2 s_{n-1}^2, \qquad \hat{a}_n = a_n.$
```

The development and analysis of stable rational variants of the tridiagonal QR algorithm has been a difficult problem, [9]. Because such algorithms do not compute and apply plane rotations in the usual stable manner, their stability properties are not at all obvious. By analogy, one might expect the rational variants of the unitary Hessenberg QR algorithm to be similarly difficult relative to more direct implementations. However, by relying on the Szegö recursions, even UHQR1 does not compute and apply plane rotations in a direct manner. Further [7] presents results of numerical experiments showing the apparent stability of the modified rational algorithm UHQRR2 without directly considering the stability of a non-rational algorithm. For these reasons, this paper will emphasize the rational algorithms and even go so far as to use UHQRR2 as a basis for deriving a stable nonrational decomposition algorithm.

In [7], when estimating backward errors, UHQRR2 is not treated as a purely rational algorithm. The following code may be incorporated into UHQRR2 to accumulate sign(c_k).

```
$\mathrm{sign}(d_0) = 1$
for $k = 1 : n - 1$
    $\mathrm{sign}(c_k) = (g_k)\ \mathrm{sign}(d_{k-1})$
    $\mathrm{sign}(d_k) = \mathrm{sign}(\overline{f}_k)\ \mathrm{sign}(c_k)$
end
$\mathrm{sign}(c_n) = \mathrm{sign}(g_n)\ \mathrm{sign}(d_{n-1})$.
```

Together with square roots of $|c_k|^2$ and s_k^2, this code was used in [7] to verify experimentally the stability of UHQRR2 by computing a decomposition $H + E = QDQ^H$ for some small backward error E. The apparent stability of this approach motivates a QR algorithm that keeps as many features of UHQRR2 as possible while computing c_k instead of $|c_k|^2$.

Algorithm UHQR2

```
$z = z\,/\,|z|$
$f_0 = c_0 = 1, \qquad s_0 = 0$
for $k = 1 : n - 1$
    $w_k = z\,f_{k-1}, \qquad t_k = \overline{a}_k\,w_k$
    if $\mathrm{Re}(t_k) \geq 0$ then $g_k = w_k + a_k$
                    else $g_k = (b_k^2 - 2\,i\,\mathrm{Imag}(t_k))\,/\,(\overline{w}_k - \overline{a}_k)$
    $p_k = g_k\,\overline{f}_{k-1}\,c_{k-1}$
    $r_k = \mathrm{sqrt}(|\,p_k\,|^2 + b_k^2), \qquad \widehat{b}_{k-1} = r_k\,s_{k-1}$
    $c_k = p_k\,/r_k, \qquad s_k = b_k/r_k$
    $f_k = \overline{w}_k\,g_k^2/\,|\,g_k\,|^2$
    $\hat{a}_k = |\,c_k\,|^2\,f_k - \overline{z}\,s_k^2\,a_{k+1}$
end
$w_n = z\,f_{n-1}, \qquad t_n = \overline{a}_n\,w_n$
if $\mathrm{Re}(t_n) \geq 0$ then $g_n = w_n + a_n$
                else $g_n = -\,2\,i\,\mathrm{Imag}(t_n)\,/\,(\overline{w}_n - \overline{a}_n)$
$p_n = g_n\overline{f}_{n-1}c_{n-1}$
$r_n = |\,p_n\,|, \qquad \hat{b}_{n-1} = r_n s_{n-1}$
$\hat{a}_n = a_n$
if $r_n > 0$ then $c_n = \mathrm{sign}(p_n)$
          else $c_n = 1$
```

Since UHQR2 is a fairly direct adaptation of UHQRR2, we will be able to apply most of the analysis of the rational algorithms to UHQR2. The conclusion will be that the general stability properties of UHQR2 are identical to those of UHQRR2: the algorithm is backward stable if the normalization errors do not become large. As with UHQRR2, there are solid theoretical reasons for believing that the normalization errors will not grow significantly. The algorithm is very reliable in practice.

In §2 we consider the propagation of normalization errors in UHQRR2 and UHQRR3. Although the conclusions involve a good deal of error analysis, the results of §2 will say nothing about the backward errors or the stability of either algorithm. Instead the normalization results provide a necessary background for the backward error analysis to be presented in §3.

2 The Normalization Errors

The stability of all of the algorithms depends both on the numerical stability of a single QR iteration carried out on improperly normalized a_k, b_k and z as well as on the dynamic properties of the algorithm with respect to the propagation of errors in the normalization of a_k and b_k over a sequence of QR iterations. Since the algorithm was derived on the assumption that $|a_k|^2 + b_k^2 = 1$ and $|z| = 1$, the errors δ_k in the error relation

$$|a_k|^2 + b_k^2 = 1 + \delta_k \tag{14}$$

and the errors in the normalization of z will both contribute to new errors introduced in any iteration that makes use of the improperly normalized quantities. If the algorithm computes $\hat{a}_k$ and $\hat{b}_k$ with

$$|\hat{a}_k|^2 + \hat{b}_k^2 = 1 + \hat{\delta}_k$$

then $\hat{\delta}_k$ will depend on δ_k. If the error propagation is such that $|\hat{\delta}_k|$ can be significantly larger than $|\delta_k|$ then the local errors will be even larger for the following iteration starting from $\hat{a}_k$ and $\hat{b}_k$. If the shift is not explicitly normalized, then this feedback of normalization errors through the QR iteration is the source of a severe instability in UHQRR1.

The unimodularity of f_k is also significant for numerical stability. In all the rational algorithms, f_k is computed from the recursion

```
for $k = 1 : n - 1$
    $w_k = z f_{k-1}$,    $g_k = w_k + a_k$
    $f_k = \overline{w}_k g_k^2 / |g_k|^2$
end.
```

We also have the corresponding recursion for UHQRR2 and UHQRR3

```
for $k = 1 : n - 1$
    $w_k = z f_{k-1}$,    $t_k = \overline{a}_k w_k$
    if $\mathrm{Re}(t_k) \geq 0$ then $g_k = w_k + a_k$
                       else $g_k = (b_k^2 - 2 i \,\mathrm{Imag}(t_k)) / (\overline{w}_k - \overline{a}_k)$
    $f_k = \overline{w}_k g_k^2 / |g_k|^2$
end.
```

From these recursions, it is clear that in exact arithmetic, f_k is unimodular whenever f_{k-1} and z are unimodular. To ensure stability, we would like to guarantee that the computed f_k is the exact f_k for slightly perturbed $a_{k'}$ and $b_{k'}$ where $1 \leq k' \leq k$.

In particular let a_k and b_k satisfy (14) for $1 \leq k \leq n$ and let

$$z = \hat{z}(1 + \delta_z) \tag{15}$$

with $|\hat{z}| = 1$ and $\mathrm{Imag}(\delta_z) = 0$. The reason for the assumption that $\mathrm{Imag}(\delta_z) = 0$ is that (15) implies that

$$|\hat{z}| = |z|(1 + \mathrm{Re}(\delta_z)) + O(|\delta_z|^2).$$

Thus the choice $\mathrm{Imag}(\delta_z) = 0$ still allows $\hat{z}$ to be unimodular to first order whenever z is unimodular within a small relative perturbation. Since we are not analyzing convergence properties, we are concerned with the deviation of the shift from unimodularity and not with its accuracy compared to an exact shift computed to give a high rate of convergence.

We will construct $\tilde{a}_k$, $\tilde{b}_k$ such that

$$f_k = (1 + \eta_k)\tilde{f}_k(\tilde{a}_1, \tilde{a}_2, \ldots, \tilde{a}_k, \tilde{b}_1, \tilde{b}_2, \ldots, \tilde{b}_k, \hat{z}) = (1 + \eta_k)\tilde{f}_k \tag{16}$$

for some small η_k. Here we take f_k to be the quantity computed from the a_j, b_j and z with numerical errors and $\tilde{f}_k$ to be the quantity computed from the $\tilde{a}_j$, $\tilde{b}_j$ and $\hat{z}$ with no numerical errors. It is always possible to choose some η_k satisfying (16). In §3 we will show that η_k and the relative errors

$$|\tilde{a}_k - a_k|/|a_k|, \qquad |\tilde{b}_k - b_k|/|b_k|$$

can all be chosen to be not much larger than the machine precision, ϵ.

In this section a key concern will be the size of $\mathrm{Re}(\eta_k)$. Since $|\tilde{f}_k| = 1$

$$|f_k| = |1 + \mathrm{Re}(\eta_k)|.$$

Thus $\mathrm{Re}(\eta_k)$ measures the degree to which $\mathrm{Re}(\eta_k)$ has departed from unimodularity. We will show that $\mathrm{Re}(\eta_k)$ can be bounded regardless of the choice of $\tilde{a}_k$ and $\tilde{b}_k$ and leave the construction of $\tilde{a}_k$ and $\tilde{b}_k$ and the derivation of a bound on $\mathrm{Imag}(\eta_k)$ for §3.

The rational algorithms naturally use and naturally compute b_k^2 and $\hat{b}_k^2$ rather than b_k and and $\hat{b}_k$. We will assume that the squared values are the ones that are stored so that there are no additional computations and errors involved in squaring b_k or taking square roots to obtain $\hat{b}_k$. All backward errors take the form of relative perturbations on the Schur parameters and complementary parameters. Since a small relative perturbation of b_k^2 corresponds to a small relative perturbation of b_k, the storage of and use of the squared quantities has no effect on the general conclusions about the stability of the algorithm.

The purpose of the analysis will be to prove the existence of reasonable bounds on the backward error rather than to give a tight upper bound. To simplify the appearance of the bounds, the model we use for complex floating point arithmetic will assume that

$$\mathrm{fl}(x \,\mathrm{op}\, y) = (x \,\mathrm{op}\, y)(1 + \epsilon_{\mathrm{op}})$$

where $|\epsilon_{op}| \leq \epsilon$. This holds for real arithmetic and holds with complex arithmetic to first order if ϵ is replaced by $4\sqrt{2}\epsilon$. The analysis is first order and we will freely ignore second order terms without commenting on the fact.

We use ϵ_l with an appropriate integer, non-variable subscript to represent a particular numerical error with $|\epsilon_l| \leq \epsilon$. We use x_l to represent a number satisfying $|x_l| \leq 1$. To avoid double-digit subscripts, we use subscripts that are local to the proof of a theorem or lemma. Equal subscripts indicate equal values of these variables within a proof, but the same subscripts can be used to denote different quantities in another proof. We also subscript error variables with non-integer variables to indicate the quantities they perturb; these subscripts are consistent throughout the paper unless otherwise noted.

The following simple lemma implies that with a numerically unimodular shift, the unimodularity of f_k is preserved to numerical accuracy.

Lemma 1 *If $f_k = (1+\eta_k)\tilde{f}_k$ then*

$$|Re(\eta_k)| \leq k|\delta_z| + 5k\epsilon.$$

Consequently

$$||f_k| - 1| \leq k|\delta_z| + 5k\epsilon.$$

Proof: Since η_k is chosen so that $|\tilde{f}_k| = 1$ we have

$$|f_k| = |1+\eta_k| = 1 + \mathrm{Re}(\eta_k) + O(\epsilon^2).$$

It is easy to verify that the computed f_k also satisfies

$$f_k = \frac{\overline{z f_{k-1}} g_k^2}{|g_k|^2}(1+5\epsilon_1).$$

Thus

$$|f_k| = |z f_{k-1}| \cdot (1+5\epsilon_1) = (1 + \mathrm{Re}(\eta_{k-1}) + \delta_z + 5\mathrm{Re}(\epsilon_1)).$$

We have used the fact that δ_z is real. Comparing this with $|f_k| = 1 + \mathrm{Re}(\eta_k)$ we get

$$\mathrm{Re}(\eta_k) = \mathrm{Re}(\eta_{k-1}) + 5\mathrm{Re}(\epsilon_1) + \delta_z.$$

Upon noting that $|f_0| = 1$, this directly implies the inequality stated in the lemma. ■

The following lemma bounds errors in the relation $|\hat{a}_k|^2 + \hat{b}_k^2 = 1$ for all the rational algorithms when $\mathrm{Re}(t_{k+1}) \geq 0$.

Lemma 2 *For $k < n$, if $Re(t_{k+1}) \geq 0$ then the $\hat{a}_k$ and $\hat{b}_k$ computed by UHQRR3 satisfy*

$$|\hat{a}_k|^2 + \hat{b}_k^2 = 1 + \hat{\delta}_k$$

with

$$\hat{\delta}_k = s_k^2 \delta_{k+1} + x\left[(15k+19)\epsilon + 3(k+1)|\delta_z|\right]$$

for some $|x| \leq 1$.

Proof: From the relations

$$s_k^2 = \mathrm{fl}\left(\frac{b_k^2}{r_k^2}\right) \qquad |c_k|^2 = \mathrm{fl}\left(\frac{|p_k|^2}{r_k^2}\right)$$

and

$$r_k^2 = \mathrm{fl}\left(b_k^2 + |p_k|^2\right)$$

it is easy to see that

$$|c_k|^2 + s_k^2 = 1 + 2\epsilon_1.$$

Define

$$\hat{f}_k = (1 - \mathrm{Re}(\eta_k)) f_k$$

so that to first order $|\hat{f}_k| = 1$ and

$$z f_k \overline{\hat{z}\hat{f}_k} = \hat{z}\hat{f}_k \overline{\hat{z}\hat{f}_k}(1 + \mathrm{Re}(\eta_k) + \delta_z) = 1 + \mathrm{Re}(\eta_k) + \delta_z.$$

Since $\mathrm{Re}(t_{k+1}) \geq 0$ and $|g_{k+1}| \leq 2$

$$g_{k+1} = (z f_k + a_{k+1}) + 3\epsilon_2 = \hat{z}\hat{f}_k \left(z f_k \overline{\hat{z}\hat{f}_k} + a_{k+1}\overline{\hat{z}\hat{f}_k}\right) + 3\epsilon_2.$$

Thus

$$|g_{k+1}|^2 = \left|1 + a_{k+1}\overline{\hat{z}\hat{f}_k}\right|^2 + x_1(4|\mathrm{Re}(\eta_k)| + 4|\delta_z| + 12\epsilon)$$

for some $|x_1| \leq 1$. It is easily verified that $|c_k|^2 s_k^2 < 1/4 + O(\epsilon)$ and

$$\left|\mathrm{fl}\left(|g_{k+1}|^2\right) - |g_{k+1}|^2\right| \leq \epsilon |g_{k+1}|^2$$

so that

$$\begin{aligned} \hat{b}_k^2 &= \mathrm{fl}\left(\left(|c_k|^2 \mathrm{fl}\left(|g_{k+1}|^2\right) + b_{k+1}^2\right) s_k^2\right) \\ &= \left(|c_k|^2 \left|1 + a_{k+1}\overline{\hat{z}\hat{f}_k}\right|^2 + b_{k+1}^2\right) s_k^2 + x_2\left(|\mathrm{Re}(\eta_k)| + |\delta_z| + 7\epsilon\right). \end{aligned}$$

The computed $\hat{a}_k$ satisfies

$$\hat{a}_k = |c_k|^2 f_k - \overline{z} s_k^2 a_{k+1} + 4\epsilon_3 = |c_k|^2 \hat{f}_k - \overline{\hat{z}} s_k^2 a_{k+1} + x_3(4\epsilon + |\mathrm{Re}(\eta_k)| + |\delta_z|)$$

and thus

$$\begin{aligned} |\hat{a}_k|^2 + \hat{b}_k^2 &= |c_k|^4 + |a_{k+1}|^2 s_k^4 - 2\mathrm{Re}\left(|c_k|^2 s_k^2 \hat{f}_k \hat{z} \overline{a}_{k+1}\right) + \left(|c_k|^2 \left|1 + a_{k+1}\overline{\hat{z}\hat{f}_k}\right|^2 + b_{k+1}^2\right) s_k^2 + \\ &\quad x_4\left(3|\mathrm{Re}(\eta_k)| + 3|\delta_z| + 15\epsilon\right). \end{aligned}$$

Using $|c_k|^2 + s_k^2 = 1 + 2\epsilon_1$ we use this to show that $|\hat{\delta}_k|$ is not much larger than $|\delta_{k+1}|$. In particular

$$\begin{aligned}
|\hat{a}_k|^2 + \hat{b}_k^2 &= |c_k|^4 + |a_{k+1}|^2 s_k^4 - 2|c_k|^2 s_k^2 \mathrm{Re}\left(\hat{f}_k \hat{z} \overline{a}_{k+1}\right) + \\
&\quad \left(|c_k|^2 \left(1 + 2\mathrm{Re}\left(\hat{z}\hat{f}_k \overline{a}_{k+1}\right) + |a_{k+1}|^2\right) + b_{k+1}^2\right) s_k^2 + \\
&\quad x_4 \left(3|\mathrm{Re}(\eta_k)| + 3|\delta_z| + 15\epsilon\right) \\
&= |c_k|^2 \left(|c_k|^2 + s_k^2\right) + |a_{k+1}|^2 s_k^2 \left(|c_k|^2 + s_k^2\right) + b_{k+1}^2 s_k^2 + \\
&\quad x_4 \left(3|\mathrm{Re}(\eta_k)| + 3|\delta_z| + 15\epsilon\right) \\
&= |c_k|^2 + s_k^2 \left(|a_{k+1}|^2 + b_{k+1}^2\right) + 2\left(|c_k|^2 + |a_{k+1}|^2 s_k^2\right)\epsilon_1 + \\
&\quad x_4 \left(3|\mathrm{Re}(\eta_k)| + 3|\delta_z| + 15\epsilon\right) \\
&= 1 + s_k^2 \delta_{k+1} + 2\left(1 + |c_k|^2 + |a_{k+1}|^2 s_k^2\right)\epsilon_1 + x_4 \left(3|\mathrm{Re}(\eta_k)| + 3|\delta_z| + 15\epsilon\right)
\end{aligned}$$

The lemma follows from an application of Lemma 1 using the fact that

$$|c_k|^2 + |a_{k+1}|^2 s_k^2 \le 1. \blacksquare$$

If $\mathrm{Re}(t_k) < 0$ then the corresponding result is somewhat less satisfactory for UHQRR2.

Lemma 3 *If Re(t_k) < 0 then UHQRR2 computes $\hat{a}_k$ and $\hat{b}_k$ for which*

$$\hat{\delta}_k = \left(s_k^2 + \frac{2b_{k+1}^2 |c_k|^2 s_k^2}{|zf_k - a_{k+1}|^2}\right)\delta_{k+1} + x\left[(30k + 26)\epsilon + 6(k+1)|\delta_z|\right]$$

where $|x| < 1$. This implies that

$$|\hat{\delta}_k| \le \frac{9}{8}|\delta_{k+1}| + (30k + 26)\epsilon + 6(k+1)|\delta_z|.$$

Proof: Assuming that b_{k+1}^2 is stored, if $\mathrm{Re}(t_k) < 0$ then $|f_k z - a_{k+1}| > 1$ and the computed g_{k+1} satisfies

$$g_{k+1} = \frac{b_{k+1}^2 - 2i\,\mathrm{Imag}(\overline{a}_{k+1} f_k z)}{\overline{f_k z} - \overline{a}_{k+1}} + 10\epsilon_1.$$

In terms of the unimodular $\hat{f}_k$ and $\hat{z}$ we have

$$g_{k+1} = \frac{b_{k+1}^2 - 2i\,\mathrm{Imag}(\overline{a}_{k+1} \hat{f}_k \hat{z})}{\overline{\hat{f}_k \hat{z}} - \overline{a}_{k+1}} + x_1(4|\mathrm{Re}(\eta_k)| + 4|\delta_z| + 10\epsilon).$$

Thus

$$\begin{aligned}
\hat{z}\hat{f}_k + a_{k+1} &= \frac{(\hat{z}\hat{f}_k + a_{k+1})\overline{(\hat{z}\hat{f}_k - a_{k+1})}}{\overline{(\hat{z}\hat{f}_k - a_{k+1})}} \\
&= \frac{1 - |a_{k+1}|^2 - 2i\,\mathrm{Imag}(\overline{a}_{k+1}\hat{f}_k \hat{z})}{\overline{(\hat{z}\hat{f}_k - a_{k+1})}}.
\end{aligned}$$

Consequently

$$\begin{aligned}\left|1 + a_{k+1}\overline{\hat{z}\hat{f}_k}\right|^2 &= |\hat{z}\hat{f}_k + a_{k+1}|^2 \\ &= \frac{\left(b_{k+1}^2 - \delta_{k+1}\right)^2 + 4\,\mathrm{Imag}(\overline{a}_{k+1}\hat{f}_k\hat{z})^2}{|\hat{z}\hat{f}_k - a_{k+1}|^2} \\ &= |g_{k+1}|^2 + x_2(16\mathrm{Re}(\eta_k) + 16|\delta_z| + 40\epsilon) - \frac{2b_{k+1}^2}{|\hat{z}\hat{f}_k - a_{k+1}|^2}\delta_{k+1}\end{aligned}$$

and since $|c_k|^2 s_k^2 \leq 1/4$

$$\begin{aligned}\hat{b}_k^2 &= \mathrm{fl}\left(\left(|c_k|^2 \mathrm{fl}\left(|g_{k+1}|^2\right) + b_{k+1}^2\right) s_k^2\right) \\ &= \left(|c_k|^2 \left|1 + a_{k+1}\overline{\hat{z}\hat{f}_k}\right|^2 + b_{k+1}^2\right) s_k^2 + x_3\left(4|\mathrm{Re}(\eta_k)| + 4|\delta_z| + 14\epsilon\right) + \frac{2b_{k+1}^2|c_k|^2 s_k^2}{|\hat{z}\hat{f}_k - a_{k+1}|^2}\delta_{k+1}.\end{aligned}$$

Using the expression for a_k from the proof of Lemma 2 we get

$$\begin{aligned}|\hat{a}_k|^2 + \hat{b}_k^2 = {} & |c_k|^4 + |a_{k+1}|^2 s_k^4 - 2\mathrm{Re}\left(|c_k|^2 s_k^2 \hat{f}_k \hat{z} \overline{a}_{k+1}\right) + \left(|c_k|^2 \left|1 + a_{k+1}\overline{\hat{z}\hat{f}_k}\right|^2 + b_{k+1}^2\right) s_k^2 + \\ & x_4\left(6|\mathrm{Re}(\eta_k)| + 6|\delta_z| + 22\epsilon\right) + \frac{2b_{k+1}^2|c_k|^2 s_k^2}{|\hat{z}\hat{f}_k - a_{k+1}|^2}\delta_{k+1}.\end{aligned}$$

A repetition of the proof of Lemma 2 gives the equality for $\hat{\delta}_k$. The inequality follows from the easily verified fact that

$$s_k^2 + 2|c_k|^2 s_k^2 < \frac{9}{8}. \blacksquare$$

Note that if the shift is properly normalized, Lemma 3 gives an equality of the form

$$\hat{\delta}_k = \left(s_k^2 + \frac{2b_{k+1}^2|c_k|^2 s_k^2}{|zf_k - a_{k+1}|^2}\right)\delta_{k+1} + O(\epsilon).$$

where the $O(\epsilon)$ term hides only small local errors. This is not just an upper bound: if δ_{k+1} is reasonably large relative to the bounded $O(\epsilon)$ term and

$$s_k^2 + \frac{2b_{k+1}^2|c_k|^2 s_k^2}{|zf_k - a_{k+1}|^2} > 1 \tag{17}$$

then we can guarantee that $|\hat{\delta}_k| > |\delta_{k+1}|$. Thus there appears to be a very real possibility of unstable propagation of normalization errors in UHQRR2. Fortunately (17) holds quite rarely. Although the analysis suggests a potential instability, constructing a unitary Hessenberg matrix for which this unstable propagation of errors happens often enough to impact the accuracy of the overall algorithm is very difficult. In practice, normalization errors don't seem to grow.

Nevertheless, we have introduced modifications into UHQRR3 to prevent any possibility of error growth. Whenever (17) holds, we use a safe alternate formula for $\hat{b}_k^2$. The alternate formula takes advantage of the fact that if (17) holds then $\hat{b}_k^2$ must be large enough to be computed to high relative accuracy by $\hat{b}_k^2 = 1 - |\hat{a}_k|^2$. We will cover this point in greater detail when performing the actual backward error analysis.

UHQRR3 guarantees that in all cases

$$|\hat{\delta}_k| \leq |\delta_{k+1}| + 6(k+1)|\delta_z| + (30k+26)\epsilon \tag{18}$$

Given the numerical unimodularity of δ_z, this is clearly sufficient to give an $O(jn\epsilon)$ bound on δ_k for iteration j of the QR algorithm. In considering Lemma 2 and Lemma 3, we first see the need for explicit normalization of z. If δ_z were dependent on δ_k then neither lemma would be sufficient to guarantee stability in the propagation of the normalization errors.

Although, in practice the normalization errors in UHQRR2 are not typically any larger than those in UHQRR3, it is only for UHQRR3 that we can prove the following theorem.

Theorem 1 *For an $n \times n$ numerically unitary Hessenberg matrix with Schur parameters $a_{k,0}$ and complementary parameters $b_{k,0}$ satisfying $|a_{k,0}|^2 + b_{k,0}^2 = 1 + \delta_{k,0}$ with $|\delta_{k,0}| < \delta$ for each k we get for the computed Schur parameters $a_{k,j}$ and complementary parameters $b_{k,j}$ after j steps of UHQRR3*

$$|a_{k,j}|^2 + b_{k,j}^2 = 1 + \delta_{k,j}$$

with

$$|\delta_{k,j}| \leq \delta + j(48n+44)\epsilon$$

and

$$||f_k| - 1| \leq 8k\epsilon.$$

Proof: The results follow from (18) and Lemma 1 upon noting that in the simplified model of complex floating point arithmetic, UHQRR3 normalizes z so that $|\delta_z| \leq 3\epsilon$. ■

3 Error Analysis

Having established that for UHQRR3, all normalization conditions hold to within a reasonable multiple of the machine precision, we are now ready to give a complete error analysis. The analysis naturally breaks into four parts.

1. We will need to obtain a bound on the backward errors on a_k and b_k and on the forward error $|\eta_k|$. Since (16) is satisfied for some η_k whatever the choice of $\tilde{a}_k$ and $\tilde{b}_k$, we will approach the analysis by defining $\tilde{a}_k$ and $\tilde{b}_k$ for several distinct special cases and for a particular k and then derive the corresponding bounds on $|\eta_k|$ for each case. The general rules used to construct these perturbed quantities for a particular k assume the existence of a bound on $|\eta_{k'}|$ for $1 \leq k' < k$. Consequently we will derive a recursion in k that can be used to bound η_k for each $1 \leq k \leq n$.

2. In addition to f_{k-1} and the Schur parameters and complementary parameters, iteration k from UHQRR3 depends on $|c_{k-1}|^2$ and s_{k-1}^2 from the previous iteration. The quantity s_{k-1}^2 is only used to compute $\hat{b}_{k-1}^2$, but $|c_{k-1}|^2$ is used to compute $|p_k|^2$. Since $|c_k|^2$ is computed from $|p_k|^2$, the computation of $|c_k|^2$ is recursive and there is an apparent possibility for the unstable propagation of errors in $|c_k|^2$. We will show that the error propagation is stable and that the computed $|c_k|^2$ is a slightly perturbed version of the quantity computed from $\tilde{a}_{k'}$ and $\tilde{b}_{k'}$ for $1 \leq k' \leq k$ without numerical errors.

3. Except for $\hat{a}_k$ the other quantities are computed from equations that don't involve cancellation. Consequently it is easy to show that they are relative perturbations of the corresponding quantities computed from $\tilde{a}_{k'}$ and $\tilde{b}_{k'}$ without errors. The computation of $\hat{a}_k$ can involve cancellation, but because the quantities involved can never be larger than 1, the absolute errors in $\hat{a}_k$ are small. We will make these observations precise by providing bounds for errors in all the quantities computed by the algorithm.

4. Finally, we will extend the analysis of the rational algorithms to prove the existence of a bound on the backward error E in $H + E = Q\hat{H}Q^H$ for $\hat{H}$ computed by UHQR2 and orthogonal Q formed by accumulating c_k and s_k. Backward stability really only holds if the normalization errors do not grow, but, as with UHQRR2, there is strong evidence that these errors are not likely to grow. The analysis is essentially the same as for the rational algorithms except that we must reconsider the propagation of errors in the recursive computation of c_k.

We will proceed with each of these points in order.

3.1 Backward Errors

Define

$$\delta_a = \frac{\delta_k}{2|a_k|^2}$$

and

$$\delta_b = \frac{\delta_k}{2b_k^2}.$$

Let $\breve{a}_k = (1-\delta_a)a_k$ and $\breve{b}_k = b_k$ when $|a_k|^2 \geq \sqrt{2}/2$ and $\breve{a}_k = a_k$ and $\breve{b}_k = (1-\delta_b)b_k$ otherwise.

To first order $|\breve{a}_k|^2 + \breve{b}_k^2 = 1$. The relative perturbation of a_k or b_k never exceeds $|\delta_k|$. Thus

$$\frac{|a_k - \breve{a}_k|}{|a_k|} \leq |\delta_k|$$

and

$$\frac{|b_k - \breve{b}_k|}{|b_k|} \leq |\delta_k|.$$

The choice of $\breve{a}_k$ and $\breve{b}_k$ to satisfy the normalization condition $|\breve{a}_k|^2 + \breve{b}_k^2 = 1$ completely determines the real part of the relative backward error on a_k and b_k. The only other freedom we will make use of will be in choosing the imaginary part of the relative backward error for a_k so that

$$\tilde{a}_k = (1 + i\,\eta_a)\breve{a}_k$$

and

$$\tilde{b}_k = \breve{b}_k.$$

Define

$$t_k := \mathrm{fl}(\overline{a}_k w_k) = (1+\epsilon_t)\overline{a}_k w_k = (1+\epsilon_t+\epsilon_w)\overline{a}_k z f_{k-1} := (1+2\epsilon_{tw})\overline{a}_k z f_{k-1}.$$

We choose η_a according to two special cases.

1. If $\mathrm{Re}(t_k) < 0$, $|a_k| \geq \sqrt{2}/2$ and $|\mathrm{Re}(t_k)| \geq \sqrt{2}|t_k|/2$ so that there is cancellation that makes g_k significantly smaller a_k then
$$\eta_a = -\frac{\mathrm{Imag}(t_k)}{\mathrm{Re}(t_k)}\mathrm{Re}(\delta_a + \delta_z + \eta_{k-1} + 2\epsilon_{tw}) - \mathrm{Imag}(\eta_{k-1} + 2\epsilon_{tw}).$$
2. Otherwise
$$\eta_a = -\mathrm{Imag}(\eta_{k-1}).$$

Case 1 involves cancellation that might destroy the relative accuracy of g_k. The analysis of Case 1 is more delicate numerically, but the analysis of Case 2 is also somewhat involved: we must consider the computation of g_k using both possible formula depending on whether $\mathrm{Re}(t_k) \geq 0$ or $\mathrm{Re}(t_k) < 0$.

We start with some definitions. If $\mathrm{Re}(t_k) \geq 0$ then let

$$\hat{g}_k = z f_{k-1} + a_k$$

and if $\mathrm{Re}(t_k) < 0$ then let

$$\hat{g}_k = \frac{b_k^2 - 2i\,\mathrm{Imag}(\overline{a}_k z f_{k-1})}{\overline{z f_{k-1}} - \overline{a}_k}.$$

Thus whatever the sign of $\mathrm{Re}(t_k)$, $\hat{g}_k$ is an exact quantity corresponding to the computed g_k. Similarly, if $\mathrm{Re}(t_k) \geq 0$ then let

$$\breve{g}_k = z f_{k-1} + \breve{a}_k$$

and if $\mathrm{Re}(t_k) < 0$ then let

$$\breve{g}_k = \frac{\breve{b}_k^2 - 2i\,\mathrm{Imag}(\overline{\breve{a}}_k z f_{k-1})}{\overline{z f_{k-1}} - \overline{\breve{a}}_k}.$$

We first consider Case 2 and the numerical errors involved in the computation of g_k. The key point is that for Case 2, if g_k is computed by the correct formula determined by the sign of $\mathrm{Re}(t_k)$ then it is not sensitive to small relative perturbations in any of the quantities used in the computation. If $\mathrm{Re}(t_k) < 0$ then the numerical errors take the form

$$g_k = \frac{b_k^2 - 2(1+\epsilon_1) i \,\mathrm{Imag}(\overline{a}_k z f_{k-1}(1+2\epsilon_2))}{(\overline{z f_{k-1}}(1+\epsilon_3) - \overline{a}_k)(1+\epsilon_4)}. \tag{19}$$

Because of the absence of cancellations except possibly in the computation of t_k, the perturbations ϵ_1, ϵ_3 and ϵ_4 all lead to small relative perturbations in g_k. To deal with ϵ_2, we note that since we are considering Case 2, either $b_k \geq \sqrt{2}/2$ or $|\mathrm{Imag}(t_k)| \geq \sqrt{2}|t_k|/2$. If $b_k \geq \sqrt{2}/2$ then

$$\left| b_k^2 - 2i \,\mathrm{Imag}(\overline{a}_k z f_{k-1}) \right| \geq \frac{1}{2}. \tag{20}$$

If $|\mathrm{Imag}(t_k)| \geq \sqrt{2}|t_k|/2$ then

$$\left| b_k^2 - 2i \,\mathrm{Imag}(\overline{a}_k z f_{k-1}) \right| \geq \sqrt{2}|t_k|. \tag{21}$$

Even if there is cancellation in the computation of $\mathrm{Imag}(t_k)$, the errors from ϵ_2 can be cast as absolute errors proportional to $4|t_k|\epsilon_5$ for some $|\epsilon_5| < \epsilon$. If $b_k \geq \sqrt{2}/2$ then $|t_k| = |a_k| < \sqrt{2}/2$ and for any ϵ_5

$$b_k^2 - 2i \,\mathrm{Imag}(\overline{a}_k z f_{k-1}) + 4|t_k|\epsilon_5 = \left(b_k^2 - 2i \,\mathrm{Imag}(\overline{a}_k z f_{k-1})\right)\left(1 + 4\sqrt{2}\epsilon_6\right)$$

for some suitable ϵ_6. This also applies if $|\mathrm{Imag}(t_k)| \geq \sqrt{2}|t_k|/2$. Thus in either case, since

$$|z f_{k-1} - a_k| \geq |z f_{k-1}| = 1 + O(\epsilon),$$

we can conclude from (19) that

$$g_k = \frac{b_k^2 - 2i \,\mathrm{Imag}(\overline{a}_k z f_{k-1})}{\overline{z f_{k-1}} - \overline{a}_k}\left(1 + \left(3 + 4\sqrt{2}\right)\epsilon_5\right) = \left(1 + \left(3 + 4\sqrt{2}\right)\epsilon_5\right)\hat{g}_k.$$

If $\mathrm{Re}(t_k) \geq 0$ then

$$g_k = (z f_{k-1}(1+\epsilon_1) + a_k)(1+\epsilon_2) = (1 + 2\epsilon_3)\hat{g}_k.$$

Thus for Case 2 there always exists $|\epsilon_g| \leq \epsilon$ such that

$$g_k = \left(1 + \left(3 + 4\sqrt{2}\right)\epsilon_g\right)\hat{g}_k.$$

In a similar manner we can conclude from (20) and (21) that if $\mathrm{Re}(t_k) < 0$ then

$$\hat{g}_k = \left(1 + \left(1 + 2\sqrt{2}\right)\breve{\delta}\right)\breve{g}_k$$

for some $|\breve{\delta}| \leq |\delta_k|$. This also holds if $\mathrm{Re}(t_k) \geq 0$.

We now consider the errors η_a, η_{k-1} and δ_z. If $\mathrm{Re}(t_k) < 0$ then it follows from $\breve{b}_k^2 = \tilde{b}_k^2$ and $\eta_a = -\mathrm{Imag}(\eta_{k-1})$ that

$$\begin{aligned}
\breve{g}_k &= \frac{\breve{b}_k^2 - 2i\,\mathrm{Imag}(\overline{\breve{a}}_k z f_{k-1})}{\overline{z f_{k-1}} - \overline{\breve{a}}_k} \\
&= \frac{\tilde{b}_k^2 - 2i\,\mathrm{Imag}\left(\overline{\tilde{a}}_k \hat{z}\tilde{f}_{k-1}(1 + \mathrm{Re}(\eta_{k-1}) + \delta_z)\right)}{\left(\overline{\hat{z}\tilde{f}_{k-1}}(1 + \mathrm{Re}(\eta_{k-1}) + \delta_z) - \overline{\tilde{a}}_k\right)(1 - i\,\mathrm{Imag}(\eta_{k-1}))} \\
&= (1 + i\,\mathrm{Imag}(\eta_{k-1}) + (1+2\sqrt{2})\delta_g)\frac{\tilde{b}_k^2 - 2i\,\mathrm{Imag}(\overline{\tilde{a}}_k \hat{z}\tilde{f}_{k-1})}{\overline{\hat{z}\tilde{f}_{k-1}} - \overline{\tilde{a}}_k} \\
&:= (1 + i\,\mathrm{Imag}(\eta_{k-1}) + (1+2\sqrt{2})\delta_g)\tilde{g}_k.
\end{aligned}$$

where the inequality

$$|\delta_g| \leq (|\mathrm{Re}(\eta_{k-1})| + |\delta_z|)\,.$$

follows from (20) and (21).

If $\mathrm{Re}(t_k) \geq 0$ then

$$\begin{aligned}
\breve{g}_k &= z f_{k-1} + \breve{a}_k \\
&= \left(\hat{z}\tilde{f}_{k-1}(1 + \mathrm{Re}(\eta_{k-1}) + \delta_z) + \tilde{a}_k\right)(1 + i\,\mathrm{Imag}(\eta_{k-1})) \\
&= \left(\hat{z}\tilde{f}_{k-1} + \tilde{a}_k\right)(1 + i\,\mathrm{Imag}(\eta_{k-1}) + \delta_g) \;:=\; (1 + i\,\mathrm{Imag}(\eta_{k-1}) + \delta_g)\tilde{g}_k
\end{aligned}$$

where

$$|\delta_g| \leq |\mathrm{Re}(\eta_{k-1})| + |\delta_z|.$$

We have shown that whatever the sign of $\mathrm{Re}(t_k)$ in Case 2

$$\begin{aligned}
g_k &= (1 + (3+4\sqrt{2})\epsilon_g)\hat{g}_k = \left(1 + (3+4\sqrt{2})\epsilon_g + (1+2\sqrt{2})\breve{\delta}\right)\breve{g}_k \\
&= \left(1 + (3+4\sqrt{2})\epsilon_g + (1+2\sqrt{2})\breve{\delta} + i\mathrm{Imag}(\eta_{k-1}) + (1+2\sqrt{2})\delta_g\right)\tilde{g}_k
\end{aligned}$$

where $|\epsilon_g| \leq \epsilon$, $|\breve{\delta}| \leq |\delta_k|$ and $|\delta_g| \leq |\mathrm{Re}(\eta_{k-1})| + |\delta_z|$.

For Case 1 we verify that η_a is chosen so that

$$\mathrm{Imag}(t_k) = \mathrm{Imag}\left(\overline{\tilde{a}}_k \hat{z}\tilde{f}_{k-1}\right)$$

Since

$$\mathrm{Imag}\left(\overline{\tilde{a}}_k \hat{z}\tilde{f}_{k-1}\right) = \mathrm{Imag}\left((1 - \delta_a - i\eta_a - \delta_z - \eta_{k-1} - 2\epsilon_{tw})t_k\right)$$

this is equivalent to

$$\mathrm{Imag}(t_k) = \mathrm{Imag}(t_k) - \mathrm{Re}(\delta_a + \delta_z + \eta_{k-1} + 2\epsilon_{tw})\mathrm{Imag}(t_k) - \mathrm{Imag}(i\eta_a + \eta_{k-1} + 2\epsilon_{tw})\mathrm{Re}(t_k).$$

Clearly η_a has been chosen to satisfy this relation. Thus

$$\begin{aligned}
g_k &= \frac{b_k^2 - 2(1+\epsilon_1)i\,\mathrm{Imag}(\overline{\tilde{a}}_k\hat{z}\tilde{f}_{k-1})}{\left(\overline{zf_{k-1}}(1+\epsilon_2) - a_k\right)(1+\epsilon_3)} \\
&= \frac{b_k^2 - 2i\,\mathrm{Imag}(\overline{\tilde{a}}_k\hat{z}\tilde{f}_{k-1})}{\overline{zf_{k-1}} - a_k}(1+3\epsilon_g) \\
&= \frac{\breve{b}_k^2 - 2i\,\mathrm{Imag}(\overline{\tilde{a}}_k\hat{z}\tilde{f}_{k-1})}{\overline{zf_{k-1}} - \breve{a}_k}(1+3\epsilon_g+2\breve{\delta}') \\
&= \frac{\tilde{b}_k^2 - 2i\,\mathrm{Imag}(\overline{\tilde{a}}_k\hat{z}\tilde{f}_{k-1})}{\left(\overline{\hat{z}\tilde{f}_{k-1}} - \overline{\tilde{a}}_k\right)(1 - i\,\mathrm{Imag}(\eta_{k-1}))}(1+3\epsilon_g+3\breve{\delta}+\delta_g)
\end{aligned}$$

where $|\breve{\delta}|, |\breve{\delta}'| \le |\delta_k|$ and

$$|\delta_g| \le 2|\mathrm{Re}(\eta_{k-1})| + 2|\delta_z| + 4|\epsilon_{tw}| \le 2|\mathrm{Re}(\eta_{k-1})| + 2|\delta_z| + 4|\epsilon|.$$

The bound on $|\delta_g|$ follows from the form of η_a, the fact that there is no cancellation in $\hat{z}\tilde{f}_{k-1} - \tilde{a}_k$ and the fact that for Case 1 $|\mathrm{Imag}(t_k)/\mathrm{Re}(t_k)| < 1$. Thus

$$g_k = \left(1 + 3\epsilon_g + 3\breve{\delta} + \delta_g + i\,\mathrm{Imag}(\eta_{k-1})\right)\tilde{g}_k.$$

All that is relevant in this analysis of the computation of g_k can be summarized in the following lemma.

Lemma 4 *There exist $\tilde{a}_k$ and $\tilde{b}_k$ with*

$$\frac{|\tilde{a}_k - a_k|}{|a_k|} \le 2|\delta_k| + |\delta_z| + |Re(\eta_{k-1})| + |Imag(\eta_{k-1})| + 4\epsilon, \qquad \frac{|\tilde{b}_k - b_k|}{|b_k|} \le |\delta_k|$$

such that UHQRR3 computes

$$g_k = \left(1 + (3+4\sqrt{2})\epsilon_g + (1+2\sqrt{2})\breve{\delta} + iImag(\eta_{k-1}) + (1+2\sqrt{2})\delta_g\right)\tilde{g}_k$$

where $|\epsilon_g| < \epsilon$, $|\breve{\delta}| \le |\delta_k|$, $|\delta_g| \le |Re(\eta_{k-1})| + |\delta_z|$ and $\tilde{g}_k$ is the exact value determined from $\tilde{a}_k$, $\tilde{b}_k$, $\tilde{f}_{k-1}$ and $\hat{z}$ without any rounding error.

Proof: We choose the largest of the bounds that we have proven for Case 1 and Case 2. ■

We are now in a position to find a bound on η_k. Clearly

$$\begin{aligned}
f_k &= \frac{\overline{zf_{k-1}}g_k^2}{|g_k|^2}(1+5\epsilon_f) \\
&= \frac{\overline{\hat{z}\tilde{f}_{k-1}}\tilde{g}_k^2}{|\tilde{g}_k|^2}\Big(1 + 5\epsilon_f + \eta_{k-1} + \delta_z + 2(3+4\sqrt{2})i\,\mathrm{Imag}(\epsilon_g) + 2(1+2\sqrt{2})i\,\mathrm{Imag}(\breve{\delta}) + \\
&\quad 2(1+2\sqrt{2})i\,\mathrm{Imag}(\delta_g)\Big) \\
&= \tilde{f}_k(1+\eta_k)
\end{aligned}$$

where the second equality follows from Lemma 4 and the third equality is a restatement of the original definition of $\tilde{f}_k$ and η_k. Applying the bounds on $|\breve{\delta}|$ and $|\delta_g|$ we get

$$|\eta_k| \leq |\eta_{k-1}| + (3+4\sqrt{2})|\delta_z| + (11+8\sqrt{2})\epsilon + 2(1+2\sqrt{2})|\delta_k| + 2(1+2\sqrt{2})|\mathrm{Re}(\eta_{k-1})|. \quad (22)$$

If $|\delta_z| < 3\epsilon$ and we use the bound on $|\mathrm{Re}(\eta_{k-1})|$ from Lemma 1 and the bound on $|\delta_k|$ from Theorem 1 then

$$\begin{aligned} |\eta_k| \leq\ & |\eta_{k-1}| + 2\left(1+2\sqrt{2}\right)\delta + \\ & \left(20 + 20\sqrt{2} + 16\left(1+2\sqrt{2}\right)(k-1) + 8j\left(1+2\sqrt{2}\right)(12n+11)\right)\epsilon \end{aligned}$$

where j is the QR iteration number, n is the size of the matrix and δ is the maximum normalization error in the initial Schur parameters and complementary parameters. Since we are not concerned with finding the tightest possible bounds, we use the fact that $\sqrt{2} < 3/2$ to get

$$|\eta_k| \leq |\eta_{k-1}| + 8\delta + (64k - 14)\epsilon + 32j(12n+11)\epsilon.$$

Since $f_0 = 1$ is exact, we can take $\eta_0 = 0$ for any j and this inequality is sufficient to show that η_k can never be very large. In fact, with $\eta_0 = 0$, we immediately obtain the following theorem.

Theorem 2 *Let $|1 - |a_{k,0}|^2 - b_{k,0}^2| \leq \delta$ for $1 \leq k \leq n$. After j steps of the QR algorithm, from Schur parameters $a_{k,j}$ and $b_{k,j}$ for $1 \leq k \leq n$, UHQRR3 computes f_k satisfying*

$$f_k = (1+\eta_k)\tilde{f}_k(\tilde{a}_1, \tilde{a}_2, \ldots, \tilde{a}_k, \tilde{b}_1, \tilde{b}_2, \ldots, \tilde{b}_k, \hat{z}) = (1+\eta_k)\tilde{f}_k$$

with

$$|\eta_k| \leq 8k\delta + \left(32jk(12n+11) + 32k^2 + 18k\right)\epsilon,$$

$$\frac{|\tilde{a}_k - a_k|}{|a_k|} \leq 2(4k-3)\delta + 8j(4k-3)(12n+11)\epsilon + (32k^2 - 38k + 13)\epsilon \quad (23)$$

and

$$\frac{|\tilde{b}_k - b_k|}{|b_k|} \leq \delta + 4j(12n+11)\epsilon.$$

We have proven that UHQRR3 computes a slightly perturbed version of an $\tilde{f}_k$ corresponding to slightly perturbed, perfectly normalized data. Moreover, although we have not emphasized the details, the same general conclusion applies to UHQRR2 if $\delta_{k'}$ is small for $1 \leq k' \leq k$.

3.2 The Error Propagation for $|c_k|^2$

Most of the other equations in UHQRR2 and UHQRR3 involve simple multiplication and compute their results to high relative accuracy. However, despite the impossibility of cancellation, the dependence of $|c_k|^2$ on $|p_k|^2 = |c_{k-1}|^2|g_k|^2$ means that the computation of $|c_k|^2$ is recursive. It is important to understand how errors propagate in this recursion.

Let

$$|c_k|^2 = (1+\gamma_k)|\tilde{c}_k|^2$$

where $|\tilde{c}_k|^2$ is the quantity computed from $\hat{z}$, $\tilde{a}_{k'}$ and $\tilde{b}_{k'}$ for $1 \le k' \le k$ in exact arithmetic. Then

$$\begin{aligned}
|c_k|^2 &= \frac{|c_{k-1}|^2|g_k|^2(1+2\epsilon_1)}{|c_{k-1}|^2|g_k|^2(1+2\epsilon_1)+b_k^2}(1+2\epsilon_2) \\
&= \frac{|c_{k-1}|^2|g_k|^2}{|c_{k-1}|^2|g_k|^2+b_k^2}\left(1+2\epsilon_2+2\epsilon_1-\frac{|c_{k-1}|^2|g_k|^2}{|c_{k-1}|^2|g_k|^2+b_k^2}2\epsilon_1\right) \\
&= \frac{|c_{k-1}|^2|g_k|^2}{|c_{k-1}|^2|g_k|^2+b_k^2}(1+2\epsilon_2+2\epsilon_1 s_k^2).
\end{aligned}$$

This follows directly from the round-off error model and the computation used to compute $|c_k|^2$. Note that this formula does not magnify the effect of the relative error, ϵ_1, on $|c_k|^2$.

In a similar manner, using Lemma 4 and noting that an imaginary relative perturbation of g_k does not have any first order effect on $|g_k|^2$, we obtain

$$\begin{aligned}
|c_k|^2 &= \frac{|\tilde{c}_{k-1}|^2|\tilde{g}_k|^2}{|\tilde{c}_{k-1}|^2|\tilde{g}_k|^2+\tilde{b}_k^2}\Big(1+s_k^2\gamma_{k-1}+2\epsilon_2+2\epsilon_1 s_k^2+s_k^2\Big(2\left(3+4\sqrt{2}\right)\mathrm{Re}(\epsilon_g)+ \\
&\quad 2\left(1+2\sqrt{2}\right)\breve{\delta}+2\left(1+2\sqrt{2}\right)\mathrm{Re}(\delta_g)\Big)+2\breve{\delta}'\Big) \\
&= (1+\gamma_k)|\tilde{c}_k|^2
\end{aligned}$$

where $|\breve{\delta}|, |\breve{\delta}'| \le |\delta_k|$ and with $|\epsilon_g| \le \epsilon$ and δ_g as in Lemma 4. Consequently it can be shown that

$$\gamma_k = s_k^2\gamma_{k-1} + x\left(10\delta + 64k\epsilon + 40j(12n+11)\epsilon\right)$$

for some $|x| < 1$. It follows that

$$|\gamma_k| \le (32(k^2+k) + 40jk(12n+11))\epsilon + 10k\delta. \tag{24}$$

and we can conclude that $|c_k|^2$ is always close to $|\tilde{c}_k|^2$ in a relative sense.

3.3 Updating the Schur Parameters

The rest of the error analysis is very straightforward. With b_k and a_k taken as given quantities, f_k and $|c_k|^2$ are the only computed quantities that are propagated from step $k-1$ to step k. We have shown that both the f_k and the c_k are close, in a relative sense, to exact quantities computed from a slightly perturbed $\tilde{a}_{k'}$ and $\tilde{b}_{k'}$ satisfying $|\tilde{a}_{k'}|^2+\tilde{b}_{k'}^2=1$ using the unimodular shift $\hat{z}$. We have shown that g_k is also close to the value that would be obtained from exact computation on the perturbed problem. Because they are computed directly from these quantities and because their respective computations are all performed with high relative accuracy, $|p_k|^2$, r_k^2 and s_k^2 are all close to the quantities that would be obtained from exact computation using $\tilde{a}_k$, $\tilde{b}_k$ and $\hat{z}$.

In particular, using Lemma 4 and the bound for η_k we can show that

$$g_k = \big(1 + x_g \left(4(2k-1)\delta + (32k^2-14k+3)\epsilon + 16j(2k-1)(12n+11)\epsilon\right)\big)\,\tilde{g}_k \qquad (25)$$

where $|x_g| < 1$. We know that $|p_k|^2$ is computed to high relative accuracy from $|c_{k-1}|^2$, $|g_k|^2$ and b_k^2. Consequently, using (25), (24) and the bound for the relative errors on b_k we can show that

$$r_k^2 = \big(1 + x_r \left(2(13k-8)\delta + (96k^2-60k+9)\epsilon + 8j(13k-8)(12n+11)\epsilon\right)\big)\,\tilde{r}_k^2 \qquad (26)$$

where $|x_r| \leq 1$ and $\tilde{r}_k^2$ is the quantity determined by $\tilde{a}_k$ and $\tilde{b}_k$ without numerical errors. This in turn implies that

$$s_k^2 = \big(1 + x_s \left(2(13k-7)\delta + (96k^2-60k+10)\epsilon + 8j(13k-7)(12n+11)\epsilon\right)\big)\,\tilde{s}_k^2 \qquad (27)$$

where $|x_s| \leq 1$.

The formula for $\hat{a}_k$ only computes $\hat{a}_k$ to high absolute, rather than relative, precision from $|c_k|^2$, f_k, z, s_k^2 and a_{k+1}. But since we know that a_{k+1} will be close to $\tilde{a}_{k+1}$ in a relative sense, this is at least sufficient to claim that $\hat{a}_k$ is close in an absolute sense to the $\hat{a}_k$ that would have been computed using $\tilde{a}_{k'}$, $\tilde{b}_{k'}$ for $1 \leq k' \leq k+1$ with shift $\hat{z}$. Using the bounds on η_k and γ_k together with (27) and (23) for the error on a_{k+1} we get

$$\hat{a}_k = |\tilde{c}_k|^2\tilde{f}_k - \bar{\hat{z}}\tilde{s}_k^2\tilde{a}_{k+1} + x_a\big((192k^2+16k+25)\epsilon + 4(13k-3)\delta + 16j(13k-3)(12n+11)\epsilon\big) \qquad (28)$$

where $|x_a| \leq 1$.

In determining the errors in $\hat{b}_k^2$ we must deal with the two special cases distinguished in UHQRR3. We note that if $\mathrm{Re}(t_k) < 0$ and $2b_k^2 s_{k-1}^2|c_{k-1}|^2/|w_k - a_k|^2 + s_{k-1}^2 > 1$ then

$$1 < \frac{2b_k^2 s_{k-1}^2 |c_{k-1}|^2}{|w_k - a_k|^2} + s_{k-1}^2 \leq 2b_k^2 s_{k-1}^2 |c_{k-1}|^2 + s_{k-1}^2.$$

But

$$2b_k^2 s_{k-1}^2 |c_{k-1}|^2 + s_{k-1}^2 > 1$$

only when

$$b_k^2 s_{k-1}^2 > \frac{1 - s_{k-1}^2}{2|c_{k-1}|^2} = \frac{1}{2}$$

which, since $b_k^2 s_{k-1}^2 = (s_k^2 r_k^2) s_{k-1}^2 = \hat{b}_{k-1}^2 s_k^2$, implies that

$$\hat{b}_{k-1}^2 > \frac{1}{2}$$

In this case, the formula $\hat{b}_{k-1}^2 = 1 - |\hat{a}_{k-1}|^2$ computes $\hat{b}_{k-1}^2$ to high relative precision from $\hat{a}_{k-1}$ so that

$$\hat{b}_{k-1}^2 = (1 + 2\epsilon_1)\left(1 - |\hat{a}_{k-1}|^2\right).$$

The exact value for the perturbed problem is $\tilde{r}_k^2 \tilde{s}_{k-1}^2$. Using (28) and the inequality $\hat{b}_{k-1}^2 > 1/2$ we find that

$$\hat{b}_k^2 = \left(1 + x_b\left(8(13k-3)\delta + (384k^2 + 32k + 52)\epsilon + 32j(13k-3)(12n+11)\epsilon\right)\right)\tilde{r}_{k+1}^2 \tilde{s}_k^2. \quad (29)$$

If $\mathrm{Re}(t_k) \geq 0$ or $2b_k^2 s_{k-1}^2 |c_{k-1}|^2 / |w_k - a_k|^2 + s_{k-1}^2 \leq 1$ then using (26) and (27) it can be shown that

$$\hat{b}_k^2 = \left(1 + x_b\left(4(13k-1)\delta + (192k^2 + 72k + 56)\epsilon + 16j(13k-1)(12n+11)\epsilon\right)\right)\tilde{r}_{k+1}^2 \tilde{s}_k^2. \quad (30)$$

errors.

Except for $\hat{a}_k$ every quantity computed by UHQRR3 is within a small relative perturbation of the quantity obtained by applying the algorithm to $\tilde{a}_k$, $\tilde{b}_k$, and $\hat{z}$ without rounding errors. We obtain the correct $\hat{a}_k$ for the perturbed problem to within a small absolute error. This is sufficient to prove stability: we obtain Schur parameters and complementary parameters that are close to those that would be obtained from the perturbed problem without rounding error.

Further, in showing that normalization errors on a_k and b_k can't grow with each iteration of the QR algorithm, we have shown that the backward errors we are required to place on a_k and b_k and the forward errors we place on the other quantities will not grow significantly with each iteration. We can expect both these types of errors to be comparable in magnitude for each iteration and to combine at most additively in their effect on the computed eigenvalues. These results are summarized in the following stability theorem.

Theorem 3 *Let $|1 - |a_{k,0}|^2 - b_{k,0}^2| \leq \delta$ for $1 \leq k \leq n$. After j steps of UHQRR3, producing Schur parameters $a_{k,j}$ and $b_{k,j}$ for $1 \leq k \leq n$, there exist $\tilde{a}_k$ and $\tilde{b}_k$ satisfying*

$$|\tilde{a}_k|^2 + \tilde{b}_k^2 = 1$$

such that

$$\frac{|\tilde{a}_k - a_k|}{|a_k|} \leq 2(4k-3)\delta + 8j(4k-3)(12n+11)\epsilon + (32k^2 - 38k + 13)\epsilon$$

$$\frac{|\tilde{b}_k - b_k|}{|b_k|} \leq \delta + 4j(12n+11)\epsilon.$$

and for which the computed $\hat{a}_k$ and $\hat{b}_k$ satisfy

$$\hat{a}_k = |\tilde{c}_k|^2 \tilde{f}_k - \overline{\tilde{z}} \tilde{s}_k^2 \tilde{a}_{k+1} + x_a \left((192k^2 + 16k + 25)\epsilon + 4(13k-3)\delta + 16j(13k-3)(12n+11)\epsilon \right)$$

and

$$\hat{b}_k^2 = \left(1 + x_b \left(4(26k-1)\delta + (384k^2 + 72k + 56)\epsilon + 16j(26k-1)(12n+11)\epsilon\right)\right) \tilde{r}_{k+1}^2 \tilde{s}_k^2$$

for $|x_a| \leq 1$ and $|x_b| \leq 1$. Thus $\hat{a}_k$ and $\hat{b}_k$ are close to the quantities obtained from applying UHQRR3 to $\tilde{a}_k$ and $\tilde{b}_k$ without any numerical errors.

The theorem shows that the new Schur parameters describe a matrix which is close to a numerically unitary Hessenberg matrix which is itself similar to an exactly unitary Hessenberg perturbed version of H. All the perturbations are small and the Bauer-Fike theorem then implies that the eigenvalues of H and $\hat{H}$ will be close. If the shifts are chosen appropriately so that b_{n-1} becomes comparable to ϵ, then a deflation can be carried out without adding significantly to the errors on the eigenvalues. Consequently, if z is chosen in a way that ensures convergence, UHQRR3 will compute the eigenvalues of H to high accuracy. The same general conclusions apply to UHQRR2 if δ_k, $1 \leq k \leq n$, is small in each QR iteration.

3.4 The Eigenvalue Decomposition

UHQR2 is virtually identical to UHQRR2 except for the recursive computation of c_k and p_k and the use of these values to compute $\hat{b}_{k-1}$ and s_k. Consequently, if we can show that c_k and p_k correspond to the quantities that would be obtained by applying exact computations to $\tilde{a}_{k'}$ and $\tilde{b}_{k'}$ then the entire analysis of the rational algorithm carries over to UHQR2. In particular, we will be able to conclude that in the absence of normalization error growth, UHQR2 is stable. The observation that normalization growth is unlikely applies to UHQR2 as readily as it does to UHQRR2. It is possible to ensure stability in all circumstances by modifying UHQR2 in the same way UHQRR2 was modified to give UHQRR3.

Since the analysis of the error propagation for c_k is not very different from the earlier analysis for the error propagation of $|c_k|^2$, we will be relatively informal. We will show that the propagation of errors on c_k is stable without proving detailed bounds.

UHQR2 computes a c_k that satisfies

$$c_k = \frac{g_k \overline{f}_{k-1} c_{k-1}}{\sqrt{|g_k|^2 |f_{k-1}|^2 |c_{k-1}|^2 + b_k^2}} (1 + 7\epsilon)$$

for some K that is not too large. The analysis of the rational algorithm shows that g_k and f_k are close to quantities that would be computed from $\tilde{a}_{k'}$ and $\tilde{b}_{k'}$ without numerical errors. Consequently

$$c_k = \frac{\tilde{g}_k \overline{\tilde{f}}_{k-1} c_{k-1}}{\sqrt{|\tilde{g}_k|^2 |c_{k-1}|^2 + \tilde{b}_k^2}} (1 + K\epsilon)$$

for some moderate constant K that is independent of the errors on c_{k-1}.

If

$$c_{k-1} = (1 + \alpha_{k-1})\tilde{c}_{k-1}$$

then

$$\begin{aligned} c_k &= \frac{\tilde{g}_k \overline{\tilde{f}}_{k-1} \tilde{c}_{k-1}(1 + \alpha_{k-1})}{\sqrt{|\tilde{g}_k|^2 |\tilde{c}_{k-1}|^2 (1 + 2\mathrm{Re}(\alpha_{k-1})) + \tilde{b}_k^2}}(1 + K\epsilon) \\ &= \frac{\tilde{g}_k \overline{\tilde{f}}_{k-1} \tilde{c}_{k-1}}{\sqrt{|\tilde{g}_k|^2 |\tilde{c}_{k-1}|^2 + \tilde{b}_k^2}}(1 + \alpha_{k-1} - |c_k|^2 \mathrm{Re}(\alpha_{k-1}) + K\epsilon). \end{aligned}$$

Since

$$|\alpha_{k-1} - |c_k|^2 \mathrm{Re}(\alpha_{k-1})| \leq |\alpha_{k-1}|$$

and K is independent of α we have

$$|\alpha_k| < |\alpha_{k-1}| + O(\epsilon).$$

As with the analysis of the error propagation for $|c_k|^2$, this implies that c_k is close to $\tilde{c}_k$. In turn, this implies that p_k is close to $\tilde{p}_k$. In fact, except for $\hat{a}_k$, all quantities computed by the algorithm are small relative perturbations of the corresponding exact quantities computed from $\tilde{a}_{k'}$ and $\tilde{b}_{k'}$. As before $\hat{a}_k$ is a small absolute perturbation of the corresponding exact value for the perturbed data.

It follows that the rotations formed from c_k and s_k can be accumulated in the usual stable manner to form Q for which

$$H + E = Q\hat{H}Q^H$$

for some small E. Thus UHQR2 can be used to compute a stable eigenvalue decomposition if the normalization errors are not large. Although we do not derive the bounds, the modification of the formula for $\hat{b}_k$ that was implemented in UHQRR3 eliminates the possibility of normalization error growth, leading to a provably stable algorithm.

4 Summary

This paper has analyzed several versions of the unitary Hessenberg QR algorithm for the special case in which the shift is assumed to be unimodular. Except for further modifications to take special care in handling the normalization of the shift and the normalization of the Schur parameters and complementary parameters, the modified rational algorithms were fully described in [7].

We presented two versions, UHQRR2 and UHQRR3, with comparable practical numerical properties. The latter is the one for which we give a full stability proof, although the stability analysis shows that UHQRR2 is also stable if the normalization errors in the relation $|a_k|^2 + b_k^2 = 1$ remain small. The possibility for instability involves at most an $O(\frac{9}{8}^n)$ error growth. This growth has never been observed; both the analysis and numerical experiments suggest that it will occur only in very exceptional circumstances or not at all. We feel safe in concluding that UHQRR2 is numerically reliable. Similar conclusions apply to the computation of a complete eigenvalue decomposition using UHQR2.

The restriction to unimodular shifts is not particularly limiting in practice: the choice of z as an eigenvalue of (12) seems to give the best practical convergence, [7]. The analysis does not extend in an obvious way to the general case of a nonunimodular shift. It is still not known with any certainty if there is a stable algorithm for solving the problem with a general shift, although positive experimental results along these lines were achieved for another modified version of UHQR in [7].

5 Acknowledgments

I would like to thank Bill Gragg for showing me the remarkable effect of his alternate formula for g_k and for encouraging my interest in this area.

References

[1] G. S. Ammar, W. B. Gragg, and L. Reichel. On the eigenproblem for orthogonal matrices. In *Proc. 25th IEEE Conference on Decision and Control*, pages 1963–1966, 1986.

[2] G. S. Ammar, W. B. Gragg, and L. Reichel. Determination of pisarenko frequency estimates as eigenvalues of an orthogonal matrix. In F. T. Luk, editor, *Advanced Algorithms and Architectures for Signal Processing II, Proc. SPIE 826*, pages 143–145, 1987.

[3] G. S. Ammar, W. B. Gragg, and L. Reichel. Direct and inverse unitary eigenvalue problems in signal processing: an overview. In M. S. Moonen, G. H. Golub, and B. L. R. De Moor, editors, *Linear Algebra for Large Scale and Real Time Applications*, NATO ASI Series, pages 341–343, Dordrecht, The Netherlands, 1993. Kluwer Academic Publishers.

[4] H. Faßbender. On numerical methods for discrete least-squares approximation by trigonometric polynomials. *Mathematics of Computation*, 66:719–741, 1997.

[5] A. Bunse-Gerstner and L. Elsner. Schur parameter pencils for solution of the unitary eigenproblem. *Linear Algebra and its Applications*, 154–156:741–778, 1991.

[6] W. B. Gragg. The QR algorithm for unitary Hessenberg matrices. *Journal of Computational and Applied Mathematics*, 16:1–8, 1986.

[7] W. B. Gragg. Stabilization of the uhqr algorithm. Report, Naval Postgraduate School, 1997.

[8] W. B. Gragg and T.-L. Wang. Convergence of the unitary Hessenberg QR algorithm. Report NPS-53-90-008, Naval Post Graduate School, 1990.

[9] B. N. Parlett. *The Symmetric Eigenvalue Problem.* Prentice-Hall, Englewood Cliffs, New Jersey, 1980.

[10] L. Reichel, G. S. Ammar, and W. B. Gragg. Discrete least squares approximation by trigonometric polynomials. *Mathematics of Computation*, 57:273–289, 1991.