# A Fast and Stable Nonsymmetric Eigensolver for Certain Structured Matrices

David Bindel Shivkumar Chandresekaran Jim Demmel
David Garmire Ming Gu

May 31, 2005

**Abstract**

We consider the set of matrices which differ from symmetric, skew symmetric, or unitary by only a low rank modification. This class of matrices includes companion matrices, arrow matrices, and matrices corresponding to mechanical oscillators with localized damping. We show that Hessenberg and Schur forms for such matrices have *semi-separable structure*, and we use this fact to construct a backward stable eigensolver for such matrices which requires $O(n)$ space and runs in $O(n^2)$ time. We evaluate the performance and accuracy of our approach for several test problems, including the computation of all the roots of a polynomial or a matrix polynomial.

## 1 Introduction

Since its invention by Francis and Kublanovskaya in the 1960s, the QR iteration has become the standard workhorse for solving dense nonsymmetric eigenvalue problems. The algorithm is backward stable and, despite the lack of a global convergence theory, remarkably robust. It was recently named one of the "algorithms of the century" for its importance [9]. However, while the QR algorithm runs in $O(n^2)$ time for symmetric tridiagonal matrices, it takes $O(n^3)$ time for nonsymmetric problems.

Despite the cost, the Matlab `roots` function finds the roots of a polynomial by using QR iteration to compute the Schur factorization of a related companion matrix. Edelman and Murakami cited an article by Cleve Moler [7] which observed that:

> it uses $O(n^2)$ storage and $O(n^3)$ time. An algorithm designed specifically for polynomial roots might use order $n$ storage and order $n^2$ time. And the roundoff errors introduced correspond to perturbations in the elements of the companion matrix, not the polynomial coefficients.

They then went on to show that, to first order, the roundoff errors introduced can be attributed to normwise small perturbations in the polynomial coefficients [4].

In this paper, we describe the design and performance of a fast QR variant for the companion matrix and a broad class of other structured nonsymmetric matrices. We review related work on structured eigensolvers and on polynomial root finding via companion matrices in Section 2. In Section 3, we show why all Hessenberg forms which are unitarily similar to such matrices have *semi-separable structure*. In Sections 4 and 5, we describe how to use this semi-separable structure to devise a backward stable QR-based algorithm for computing the Schur form which requires only $O(n)$ storage and runs in $O(n^2)$ time. We briefly discuss stability in Section 6.

While general diagonal similarities destroy the structure we use in our fast algorithm, we can still optimally balance companion matrices in a sense we make clear in Section 7. In Section 8, we describe the details of our implementation, and in Section 9, we describe our evaluation of the accuracy and speed of the algorithm in finding the roots of polynomials and matrix polynomials. We describe further extensions and applications to our ideas in Section 10, and conclude in Section 11.

# 2 Related work

Unitary, symmetric, and Hessenberg forms are all preserved under QR iteration. The preservation of the upper Hessenberg form is what makes the standard QR algorithm practical, since it reduces the overall cost of the algorithm from $O(n^4)$ to $O(n^2)$. Because the iteration also preserves the symmetric tridiagonal form, QR iteration has been used very successfully to find all the eigenvalues of a symmetric tridiagonal in $O(n^2)$ time [8]. Like the set of symmetric tridiagonals, the set of unitary Hessenberg matrices can be described by $O(n)$ parameters (the "Schur parameters"), and the set is closed under the QR iteration. Gragg has exploited this structure to construct a fast QR algorithm for the unitary Hessenberg eigenvalue problem [6]. More recently, Bini, Gemignani, and Pan have described an $O(n^2)$ algorithm for another set of structured matrices which is closed under QR iteration [].

Van Dooren and Dewilde described using QR iteration (actually, the related QZ iteration) on a companion matrix to solve polynomial eigenvalue problems from systems theory []. They note that though the standard stability theory for QR attributes rounding rounding error to a small perturbation of the entire matrix, the backward error in QR on a companion matrix can be attributed to perturbations in the polynomial coefficients alone. Edelman and Murikami have refined the result and put it in a geometric framework [4]. Trefethen and Toh compared several different root finding methods, including QR iteration on companion matrices, and also the Jenkins-Traub method, which they note was originally derived as a generalized Rayleigh quotient iteration on a companion matrix [], []. In the conclusion to their paper, Trefethen and Toh note that the

$O(n^3)$ cost of QR iteration on a companion matrix is not critically important, since

> polynomial zerofinding is a rather unusual problem in scientific computing: one is not interested in large N! It is very hard to conceive of a genuine application where it would be a good idea to compute zeros of high-degree polynomials specified by their coefficients.

Nevertheless, a recent paper describes several root finding algorithms in the context of signal processing, including an algorithm which has been used to factor polynomials of degree as high as a quarter million [].

The computation of roots of polynomials and eigenvalues of structured matrices remains an area of active research. Despite the potential instability, nonorthogonal structure-preserving similarity transformations have seen recent use in the HR algorithm for Hamiltonian eigenvalue problems [] and the SR algorithm for symplectic eigenvalue problems []. Dongarra and Geist proposed the LR algorithm be revived as the basis for fast algorithms to compute the eigenvalues of a nonsymmetric tridiagonal matrix [3], and Gemignani suggested an LR-based polynomial root finder []. Bini, Gemignani, and Tisseur proposed that the roots of a nonsymmetric tridiagonal be computed using the Ehrlich-Aberth iteration to simultaneously compute all roots of the characteristic polynomial, and used the structure of the tridiagonal and of its inverse to efficiently and accurately evaluate the polynomial and its derivative.

In the remainder of the paper, we develop a fast QR iteration for low rank modifications to symmetric, skew symmetric, and orthogonal matrices. Our running example of such a matrix is a companion matrix. However, matrices with the same structure occur naturally. Furthermore, the theory and algorithm we develop extend and illustrate from a different perspective the ideas used in some existing structured eigensolvers.

# 3 Semi-separable Hessenberg forms

We will call a matrix $A$ *rank-symmetric* if $\text{rank}(A_{12}) = \text{rank}(A_{21})$ for any 2-by-2 block partitioning of $A$ where $A_{11}$ and $A_{22}$ are square. Obviously, symmetric and skew-symmetric matrices are rank-symmetric. Orthogonal matrices are symmetric as a result of the CS decomposition [5]. Matrices which are *$J$-orthogonal* (i.e. matrices $Q \in \mathbb{R}^{n\times n}$ such that $QJQ^T = J$ for a signature matrix $J = \text{diag}(\pm 1)$) are rank-symmetric as a result of an analogous hyperbolic CS decomposition [11]. Complex Hermitian, skew-Hermitian, and unitary matrices are rank-symmetric, too, but for simplicity we will restrict ourselves to the real case in this paper.

We note that symmetric, skew-symmetric, and orthogonal matrices actually satisfy a property stronger than rank symmetry. Not only do corresponding off-diagonal blocks have the same rank; they have the same singular values. Therefore, if we perturb these matrices slightly, the resulting matrices will still

be *numerically* rank-symmetric. The same cannot necessarily be said of $J$-orthogonal matrices.

If $A$ is a rank-symmetric matrix and $L$ has low rank, then the rank of corresponding subdiagonal and superdiagonal blocks in $A+L$ cannot differ by much. In particular, we have the following:

**Lemma 1.** *Let $A \in \mathbb{R}^{n\times n}$ be rank symmetric, and let $L \in \mathbb{R}^{n\times n}$ have rank $k$. If we partition $A$ as*

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

*where $A_{11}$ and $A_{22}$ square, and partition $L$ conformally, then*

$$|\operatorname{rank}(A_{12} + L_{12}) - \operatorname{rank}(A_{21} + L_{21})| \leq 2k$$

*Proof.* The rank of $L_{ij}$ is bounded by $k$, so $|\operatorname{rank}(A_{ij} + L_{ij}) - \operatorname{rank}(A_{ij})| \leq k$. Since $\operatorname{rank}(A_{ij}) = \operatorname{rank}(A_{ji})$, the result holds by the triangle inequality. □

If $A$ is symmetric or skew, any congruent matrix $X^TAX$ will also be symmetric or skew. If $A$ is orthogonal, then products of $A$ with orthogonal matrices will also be orthogonal. Therefore, the set of symmetric, skew, and orthogonal matrices is closed under orthogonal similarity. The structure of a low rank modification to a symmetric, skew, or orthogonal matrix is also preserved under orthogonal similarity, both mathematically and numerically.

Suppose $A$ is (skew) symmetric or orthogonal, $L$ has low rank, and $B = A + L$. Let $Q$ be an orthogonal matrix such that $Q^TBQ = H$ is upper Hessenberg. Any subdiagonal block of $H$ has rank at most one, since there is only one possible nonzero in the block. Therefore, by the lemma, the superdiagonal blocks of $H$ can have rank no greater than $2\operatorname{rank}(L) + 1$. If $H$ is upper triangular, the rank of the superdiagonal blocks can be no greater than $2\operatorname{rank}(L)$.

The low rank structure in the upper triangle of $H$ admits a compact representation. In a *sequentially semi-separable* (SSS) representation of $H$, the matrix is partitioned into block form, and each superdiagonal block is written as

$$H_{ij} = U_i W_{i+1} W_{i+1} \dots W_{j-1} V_j^T$$

where $U_i \in \mathbb{R}^{m_i\times k}$, $V_j \in \mathbb{R}^{m_j\times k}$, and $W_{i+1}, \dots, W_{j-1} \in \mathbb{R}^{k\times k}$ [2]. Here, $m_i$ is the size of the $i$th diagonal block and $k$ is the maximum rank of any superdiagonal block. We store diagonal blocks and subdiagonal entries of $H$ explicitly. For a fixed $k$, the space required to store an SSS representation of $H$ is $O(n)$.

For the monomial $p(x) = x^n + a_{n-1}x^{n-1} + \ldots + a_1x + a_0$, the corresponding companion matrix is

$$C := \begin{bmatrix} -a_{n-1} & -a_{n-2} & \dots & -a_1 & -a_0 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ & & \ddots & & \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix}.$$

We can write $C$ as the sum of a permutation matrix and a rank one matrix:

$$C := \begin{bmatrix} 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ & & \ddots & & \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{bmatrix} a_{n-1} & a_{n-2} & \dots & a_1 & a_0 + 1 \end{bmatrix}$$

Thus, any Hessenberg form orthogonally similar to $C$ (a set of matrices which includes $C$ itself) has off-diagonal rank no greater than three.

# 4 Fast QR iteration

The standard Hessenberg QR iteration consists of "bulge chasing" passes. Each pass begins with a local orthogonal similarity which only affects the first few rows and the first few columns of the matrix. The resulting small "bulge" of nonzero below the subdiagonal is pushed down and right in the matrix by successive orthogonal similarities until the bulge is pushed off the end of the matrix. At the end of the pass, the matrix has Hessenberg form once more. Each local similarity in a bulge chasing pass takes $O(n)$ time. There are $O(n)$ such local similarities per pass, and convergence occurs in $O(n)$ passes. Therefore the total cost of the algorithm is $O(n^3)$ operations.

Suppose $H$ is a companion matrix stored in SSS form. By the results of the previous section, every intermediate matrix generated by the QR algorithm will have semi-separable structure. We will show shortly how to apply local similarity transforms to an SSS representation; otherwise, our fast QR iteration for companion matrices is *the same* as the standard iteration. The only difference is in the matrix representation.

To apply a local similarity to $H$ which affects only the $i$th block row and block column, we use the following procedure:

1. Transform the explicitly stored subdiagonal and diagonal blocks. For instance, we overwrite $H_{ii}$ with $Q^T H_{ii} Q$.
2. To transform $H_{ij} = U_i W_{i+1} \dots W_{j-1} V_j^T$ on the left for $j = i+1 : n$, we overwrite $U_i$ with $Q^T U_i$.
3. To transform $H_{ji} = U_j W_{j+1} \dots W_{i-1} V_i^T$ on the left for $j = 1 : i-1$, we overwrite $V_i$ with $Q^T V_i$.

Since this transformation only affects a small amount of data in the diagonal and subdiagonal blocks, $U_i$, and $V_i$, it runs in $O(1)$ time rather than $O(n)$ time.

To apply transformations which affects both the $i$th and $i+1$st block rows and columns of $H$, we *merge* the block rows and columns into a single block row and a single block column. After we finish those transformations, we will *split* the merged column to restore the original block partitioning. By successively

merging and splitting during bulge chasing, we can keep a "window" of a single (possibly merged) block row and block column that contains the bulge. The merge operation is needed to ensure the window contains the entire bulge; the split operation is needed to ensure the window does not grow too large. We describe both operations in the next section.

# 5 Merging and splitting SSS blocks

Let $H$ be a block upper Hessenberg matrix represented in SSS form. That is, for $j > i$,

$$H_{ij} = U_i W_{i+1} \dots W_{j-1} V_j^T$$

To merge blocks $i$ with $i+1$, we compute

$$\begin{aligned}
H^{\text{merge}} &:= \begin{bmatrix} H_{ii} & H_{i\ i+1} \\ H_{i+1\ i} & H_{i+1\ i+1} \end{bmatrix} \\
U^{\text{merge}} &:= \begin{bmatrix} U_i W_{i+1} \\ U_{i+1} \end{bmatrix} \\
V^{\text{merge}} &:= \begin{bmatrix} V_i \\ V_{i+1} W_i^T \end{bmatrix} \\
W^{\text{merge}} &:= W_i W_{i+1}
\end{aligned}$$

While the merging formulas can be derived by inspection, it is more complicated to split one block into two in a stable manner. To do so, we introduce the notion of *proper forms* for SSS representations.

The $i$th *Hankel block* of $H$ is the submatrix

$$\begin{bmatrix} H_{1,i+1} & \dots & H_{1,n} \\ \vdots & & \vdots \\ H_{i,i+1} & \dots & H_{i,n} \end{bmatrix}.$$

In outer product form, $i$th Hankel block is $C_i G_{i+1}^T$, where $C_j$ and $G_j$ are defined by the recurrences

$$\begin{aligned}
C_1 &= U_1 \\
C_j &= \begin{bmatrix} C_{j-1} W_j \\ U_j \end{bmatrix} \text{ for } j > 1
\end{aligned}$$

and

$$\begin{aligned}
G_n &= V_n \\
G_j &= \begin{bmatrix} V_v \\ G_{j+1} W_j^T \end{bmatrix} \text{ for } j < n.
\end{aligned}$$

We say the representation is in *left proper form* if the Hankel block column bases $C_1, \dots, C_{n-1}$ are all orthonormal bases. In left proper form, $U_1 = C_1$ is

column orthogonal and we can write $C_j$ for $j > 1$ as

$$C_j = \begin{bmatrix} C_{j-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} W_j \\ U_j \end{bmatrix}.$$

So the matrix is in left proper form if and only if $U_1$ is orthonormal and $\begin{bmatrix} W_j \\ U_j \end{bmatrix}$ is orthonormal for $j > 1$. Similarly, we have a *right proper form* if the column bases $G_2, \ldots, G_n$ are all orthogonal. Equivalently, right proper form means that $V_n$ is orthonormal and $\begin{bmatrix} V_j \\ W_j^T \end{bmatrix}$ is orthonormal for $j < n$.

Now suppose we wish to split the $i$th block in an SSS representation into blocks $i^\alpha$ and $i^\beta$. Consider the submatrix

$$\begin{bmatrix} H_{1,i} & \ldots & H_{1,n} \\ \vdots & & \vdots \\ H_{i,i} & \ldots & H_{i,n} \end{bmatrix}$$

Because of the SSS structure, we can factor this submatrix as

$$\begin{bmatrix} C_{i-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} V_i^T & W_i \\ H_{ii} & U_i \end{bmatrix} \begin{bmatrix} 0 & G_{i+1} \\ I & 0 \end{bmatrix}^T.$$

If the matrix is somewhere between left and right proper form, so that $C_1, \ldots C_{i-1}$ and $G_{i+1}, \ldots, G_n$ are orthogonal, then this is a complete orthogonal factorization of the submatrix. To split block $i$, we equate

$$\begin{aligned} \begin{bmatrix} V_i^T & W_i \\ H_{ii} & U_i \end{bmatrix} &= \begin{bmatrix} V_i^{\alpha T} & W_i^\alpha V_i^{\beta T} & W_i^\alpha W_i^\beta \\ H_{ii}^{\alpha\alpha} & U_i^\alpha V_i^{\beta T} & U_i^\alpha W_i^\beta \\ H_{ii}^{\beta\alpha} & H_{ii}^{\beta\beta} & U_i^\beta \end{bmatrix} \\ &= \begin{bmatrix} V_i^{\alpha T} & \begin{bmatrix} W_i^\alpha \\ U_i^\alpha \end{bmatrix} \begin{bmatrix} V_i^{\beta T} & W_i^\beta \end{bmatrix} \\ H_{ii}^{\alpha\alpha} & \\ H_{ii}^{\beta\alpha} & H_{ii}^{\beta\beta} \quad U_i^\beta \end{bmatrix}. \end{aligned}$$

We compute $W_i^\alpha$, $W_i^\beta$, $U_i^\alpha$ and $V_i^\beta$ by a rank-revealing decomposition; we can extract $V_i^\alpha$ and $U_i^\beta$ from the top half of $V_i$ and the bottom half of $U_i$ with no computation. Note that the $i^\alpha$th Hankel block is

$$\begin{bmatrix} C_{i-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} W_i^\alpha \\ U_i^\alpha \end{bmatrix} \begin{bmatrix} V_i^{\beta T} & W_i^\beta \end{bmatrix} \begin{bmatrix} 0 & G_{i+1} \\ I & 0 \end{bmatrix}^T.$$

Since the matrices on the left and right are unitary, the singular values of

$$\begin{bmatrix} W_i^\alpha \\ U_i^\alpha \end{bmatrix} \begin{bmatrix} V_i^{\beta T} & W_i^\beta \end{bmatrix}$$

must be the same as those of the $i^\alpha$ Hankel block, and a normwise small error in the rank-revealing decomposition corresponds to a small error in the affected Hankel block.

In the paragraph above, we assumed that $C_1, \ldots, C_{i-1}$ and $G_{i+1}, \ldots, G_n$ are both orthogonal. When we perform the rank revealing decomposition during the splitting process, we choose to make $\begin{bmatrix} W_i^\alpha \\ U_i^\alpha \end{bmatrix}$ orthogonal. That way, when we move the "window" down and right, we will have the correct balance of right proper form and left proper form needed to perform the next split. Thus, we start each iteration in right proper form; convert to left proper form during the process of bulge chasing; and restore right proper form at the end of the iteration in preparation for the next sweep.

# 6 Stability analysis

Most of the steps in our algorithm are obviously backward stable. Applying orthogonal similarities yields small backward error by the usual analysis. The iterations to restore the representation to proper form involve only QR decompositions, which are also backward stable. Because of the way we manage proper forms, all the matrix products involved in computing a merge remain modest; it is not difficult to show that merging is backward stable. The only step that requires analysis is the splitting step.

Let $B = A + L + E$, where $A$ is symmetric, orthogonal, or skew; $L$ has rank $k$; and $E$ is a perturbation. We partition $B$ into two-by-two block form with square diagonal blocks, and partition $A$, $L$, and $E$ conformally. We know that $A_{12}$ and $A_{21}$ have the same singular values; write SVDs of these matrices as

$$\begin{aligned} A_{12} &= U_1 \Sigma V_2^T \\ A_{21} &= U_2 \Sigma V_1^T. \end{aligned}$$

Now we express $\Sigma$ in terms of quantities in the 12 block:

$$\begin{aligned} \Sigma &= U_2^T A_{21} V_1 \\ &= U_2^T (B_{21} - L_{21} - E_{21}) V_1. \end{aligned}$$

Therefore,

$$\begin{aligned} B_{12} &= A_{12} + L_{12} + E_{12} \\ &= U_1 \Sigma V_2^T + L_{12} + E_{12} \\ &= (U_1 U_2^T)(B_{21} - L_{21} - E_{21})(V_1 V_2^T) + L_{12} + E_{12} \\ &= \left\{ (U_1 U_2^T) B_{21} (V_1 V_2^T) + L_{12} - (U_1 U_2^T) L_{21} (V_1 V_2)^T \right\} + \\ &\quad \left\{ E_{12} - (U_1 U_2^T) E_{21} (V_1 V_2)^T \right\}. \end{aligned}$$

Suppose $B_{12}$ is rank one. Then $B_{12}$ consists of a matrix which is actually rank $2k+1$ plus an error term which is bounded in norm by $\|E_{12}\| + \|E_{21}\|$. As we described in the previous section, splitting involves computing a truncated SVD of a superdiagonal block; if we truncate $B_{12}$, we could in principle get an error as large as $\|E_{12}\| + \|E_{21}\|$ from truncation, in addition to the usual error. Thus,

our bound allows the possibility of exponential growth in error as successive splitting operations cause the matrix to drift from having orthogonal-plus-low rank structure.

We note that in Stewart's analysis [10] of variant of Gragg's unitary Hessenberg QR algorithm [6], he mentions a similar possibility of exponential backward error growth due to loss of normalization in the Schur parameters describing the unitary Hessenberg matrix. In that algorithm, the theoretical backward error growth bound appears to be very pessimistic, and the behavior is stable in general. Stewart also suggests a rearranged version of that algorithm which is provably backward stable. It may be possible to use similar stabilizing techniques to guarantee that the iterates cannot drift from the structure of a low rank modification to something symmetric, skew, or orthogonal. However, in practice we observe little growth in backward error in our procedure.

# 7 Balancing

Prior to the QR iteration, we would usually usually preprocess a nonsymmetric matrix by a balancing transformation. Suppose $A$ is irreducible; then balancing replaces $A$ by $DAD^{-1}$ for some diagonal matrix $D$. The diagonal matrix $D$ is often chosen to make the norm of each row of $DAD^{-1}$ nearly equal to the norm of the corresponding column. The entries of $D$ are typically chosen to be a power of the machine radix, so that no error is committed in forming $DAD^{-1}$. Balancing tends to reduce the norm of $DAD^{-1}$.

Unfortunately, the structure we require for fast QR iteration is not preserved under arbitrary diagonal similarities. For instance, consider the companion matrix $C$; if $D = \mathrm{diag}(d_1, \ldots d_n)$, then

$$DCD^{-1} = \begin{bmatrix} -a_{n-1} & -\frac{d_1}{d_2}a_{n-2} & \cdots & -\frac{d_1}{d_{n-1}}a_1 & -\frac{d_1}{d_n}a_0 \\ \frac{d_2}{d_1} & 0 & \cdots & 0 & 0 \\ 0 & \frac{d_3}{d_2} & \cdots & 0 & 0 \\ & & \ddots & & \\ 0 & 0 & \cdots & \frac{d_n}{d_{n-1}} & 0 \end{bmatrix}.$$

For a general $D$, Hessenberg forms orthogonally similar to $DCD^{-1}$ need no longer have low rank Hankel blocks. However, the low rank structure is preserved if $D$ is a geometric scaling ($d_i = \alpha^i$ for some $\alpha$). The elements of the first row of $DCD^{-1}$ are the negatives of the trailing coefficients of $p(\alpha x)/\alpha^{n-1}$; the remaining rows are simply multiplied by $\alpha$. Where $C$ was a rank one modification to a cyclic permutation matrix $P$, $DCD^{-1}$ is a rank one modification to $\alpha P$. Since rank symmetry of a scalar multiple of an orthogonal matrix is preserved under orthogonal similarity, the fast algorithm still works.

Chen developed balancing algorithms based on the approximation of Perron vectors of $|A|$ []. If $A$ is irreducible and $x$ and $y$ are the right and left Perron vectors of $|A|$, then

- $D = \text{diag}(1/x_1, \ldots, 1/x_n)$ minimizes $\|DAD^{-1}\|_\infty$
- $D = \text{diag}(\sqrt{y_1/x_1}, \ldots, \sqrt{y_n/x_n})$ minimizes $\|DAD^{-1}\|_2$.

Since $|C|$ is still a companion matrix, it has a right Perron vector with entries $x_i = \alpha^i$, where $\alpha$ is the maximum positive eigenvalue of $|C|$, or equivalently the largest positive root of

$$x^n - \sum_{i=0}^{n-1} |a_i| x^i.$$

Therefore, geometric scaling is the optimal infinity-norm balancing for $C$.

Symmetry, skew symmetry, and orthogonality are all destroyed by general diagonal similarities. If $A$ is a low rank modification to such a matrix, there may exist some diagonal similarities that preserve that structure. Also, if $A$ differs from symmetric, skew, or orthogonal by a rank $k$ matrix and all but $m$ of the entries of $D$ are one, then $DAD^{-1}$ will differ from symmetric, skew, or orthogonal by at most rank $k + m$. Therefore, it may be possible to trade the increased cost of higher-rank Hankel blocks for improved accuracy from balancing.

# 8 Implementation

Our initial implementation is based on the LAPACK routine `dlahqr`, which computes the Schur factorization of a Hessenberg matrix using the basic double-shift QR algorithm. The multi-shift QR routine `dhseqr` uses `dlahqr` as a helper function; we eventually intend to implement a multi-shift version of our code based on `dlahqr`, but we have not done so yet.

We store the elements of $H$ near the diagonal in a band structure; the superdiagonal blocks are stored in an SSS representation as described above. Most of the logic in `dlahqr` only references entries of $H$ very near the diagonal; other elements are referenced only when we transform the entire matrix. By adjusting the stated leading dimension of the band array, we can access elements near the diagonal of $H$ in our modified algorithm using exactly the same code that we used in the dense case. Consequently, our fast code differs from our dense test code in only a few line; those lines call routines that shift the window for transformations, maintain the correct proper forms, apply transformations to the SSS part of the matrix, and estimate the norms of submatrices during convergence testing.

The code is summarized as Algorithm 1.

# 9 Evaluation

# 10 Related applications

Though we have focused on structured eigenvalue computations, and in particular on the decomposition of companion matrices, the same ideas we used to

**Algorithm 1** Fast QR algorithm

**INPUT:** A Hessenberg matrix $H$ in compressed form
**OUTPUT:** A compressed Schur form $T$ and a list of eigenvalues
$I := N$
Convert representation to right proper form
**while** $I > 1$ **do**
  **repeat**
    **if** any subdiagonal in rows 2 through $L$ is negligible **then**
      $L :=$ row of last negligible subdiagonal element
    **else**
      $L := 1$
    **end if**
    Choose a shift (Francis, Wilkinson, or exceptional)
    Find $L < M \leq I - 2$ to make $H(M, M-1)$ negligible, or set $M = L$
    Compute a reflection $G$ to apply starting at column $M$
    Convert rows 1 to $M - 1$ to left proper form.
    **for** $K = M$ to $I - 1$ **do**
      **if** $K > M$ **then**
        Compute $G$ to restore Hessenberg form to column $K - 1$
      **end if**
      **if** $G$ affects rows outside the current window **then**
        **if** current window consists of two merged blocks **then**
          Split, extending the partial left proper form by one block
        **end if**
        Merge the blocks affected by $G$
      **end if**
      Apply $G$ to the compressed representation
    **end for**
    Restore right proper form.
  **until** a submatrix of order 1 or 2 splits off the bottom
  Store converged eigenvalues and decrease upper bound $I$
**end while**

make a fast eigensolver can be used for related problems. In this section, we briefly describe N such applications: X, Y, Z.

## 10.1 Matrix polynomial eigenvalues

In the previous sections, we illustrated fast QR iteration on a companion matrix to find polynomial roots. The same ideas works with few modifications for the solution of matrix polynomial eigenvalue problems.

We now consider the matrix polynomial

$$p(z) = Iz^n + A_{n-1}z_{n-1} + \dots A_0 \in \mathbb{R}^{d\times d}.$$

If $p(\lambda)x = 0$, then $(x, \lambda)$ is a right eigenpair for the polynomial eigenvalue problem. The eigenvalues of $p$ are precisely the eigenvalues of the block companion matrix

$$C^{\text{block}} := \begin{bmatrix} -A_{n-1} & -A_{n-2} & \dots & -A_1 & -A_0 \\ I & 0 & \dots & 0 & 0 \\ 0 & I & \dots & 0 & 0 \\ & & \ddots & & \\ 0 & 0 & \dots & I & 0 \end{bmatrix}.$$

The block matrix is the sum of a cyclic permutation matrix and a matrix of rank $d$, where $d$ is the dimension of the coefficient matrices. Therefore, the theory from the previous sections tells us that superdiagonal and subdiagonal blocks of matrices which are orthogonally similar to $C^{\text{block}}$ can differ in rank by at most $2d$. The block companion matrix is in block upper Hessenberg form, and iterates in the QR iteration on $C^{\text{block}}$ remain in upper Hessenberg form. Therefore, the superdiagonal blocks in QR iteration on $C$ all have rank of at most $3d$. Using the same split-and-merge ideas as before, we can perform bulge chasing on a compressed representation of $C$ with a cost of $O(d^3n)$ flops per QR iteration and $O(d^2n)$ storage.

As in the scalar case, we can optimally balance $C^{\text{block}}$ in the $\infty$-norm without destroying the structure needed for fast iteration. The right Perron vector of $|C^{\text{block}}|$ has the form

$$x = \begin{bmatrix} x_1 \\ \alpha x_1 \\ \alpha^2 x_1 \\ \vdots \\ \alpha^{n-1} x_1 \end{bmatrix}$$

where $x_1 \in \mathbb{R}^d$. If we balance $C^{\text{block}}$ by a diagonal matrix $D$ whose coefficients are the elements of $x$, the resulting matrix is $\alpha$ times a cyclic permutation matrix plus a matrix of rank $d$.

## 10.2 Limited storage Arnoldi iteration

Suppose that $A$ is symmetric, skew, or orthogonal plus a low-rank modification. Then the Arnoldi process converts $A$ to a Hessenberg form where the superdiagonal Hankel blocks have known maximum rank $k$. Using this knowledge, we can construct a variant of the Arnoldi procedure in which only a limited number of vectors of the Arnoldi basis are maintained. While we have not analyzed the behavior of this algorithm in finite precision, we expect some form of partial reorthogonalization will be necessary. Similar ideas lie behind the Lanczos process for symmetric matrices and the isometric Arnoldi procedure for unitary matrices [1], which both involve only limited orthogonalization against previous vectors when extending the Arnoldi basis.

At the beginning, we run an ordinary Arnoldi process. After some number of iterations, we will have a partial Hessenberg form, which we write in blocked form as

$$A \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \\ & H_{32} \end{bmatrix} \begin{bmatrix} Q_1 & Q_2 & Q_3 \end{bmatrix}$$

where $H_{12}$ has numerical rank $k$. Generically, this should take place after generating $2k+1$ Arnoldi vectors, or the first $2k$ columns of $H$. In practice, though, the block $H_{12}$ may have numerical rank less than $k$; if this is so, we simply extend the factorization by a few more columns.

We factor $H_12 = U_1 V_2^T$, where $U_1$ is an orthonormal basis for $H_12$, and so for the entire Hankel block of which $H_12$ is the leftmost part. We now extend the Arnoldi factorization to

$$A \begin{bmatrix} Q_1 & Q_2 & Q_3 \end{bmatrix} \begin{bmatrix} H_{11} & C_1 V_1^T & C_1 X_2^T \\ H_{21} & H_{22} & H_{23} \\ & H_{32} & H_{33} \\ & & H_{43} \end{bmatrix} \cdot \begin{bmatrix} Q_1 & Q_2 & Q_3 & Q_4 \end{bmatrix}.$$

In the standard Arnoldi process, we would explicitly orthogonalize each generated vector against all previous columns of $Q$. Here, though, we orthogonalize against $Q_1 C_1$ instead of against $Q_1$. The coefficients produced by orthogonalizing against $Q_1 C_1$ are the elements of $X_2$.

Once the block $H_{23}$ has rank $k$, we factor

$$\begin{bmatrix} W_2 \\ U_3 \end{bmatrix} V_3^T = \begin{bmatrix} X_2^T \\ H_{23} \end{bmatrix}$$

where the column vectors $\begin{bmatrix} W_2 \\ U_3 \end{bmatrix}$ are orthonormal. Then

$$C_2 = \begin{bmatrix} C_1 & \\ & I \end{bmatrix} \begin{bmatrix} W_2 \\ U_3 \end{bmatrix}$$

is an orthonormal basis for a Hankel block of $H$. Now, when we extend the Arnoldi factorization, we can orthogonalize against

$$\begin{bmatrix} Q_1 Q_2 \end{bmatrix} C_2$$

rather than orthogonalizing against $Q_1$ and $Q_2$ explicitly.

The remaining steps are similar. After constructing $i$ blocks, we will have an orthonormal column basis $C_i$ for a Hankel block of $H$. We extend the factorization by orthogonalizing against

$$Z_i = \begin{bmatrix} Q_1 & \dots & Q_i \end{bmatrix} C_i$$

in lieu of explicitly orthogonalizing against $\begin{bmatrix} Q_1 & \dots & Q_i \end{bmatrix}$. Let the matrix $X_i$ consist of the coefficients from orthogonalizing against $Z_i$. After a few steps, when $H_{i,i+1}$ has rank $k$, we can extend $C_i$ and $Z_i$ by factoring

$$\begin{bmatrix} W_i \\ U_{i+1} \end{bmatrix} V_{i+1}^T = \begin{bmatrix} X_i^T \\ H_{i,i+1} \end{bmatrix}$$

Then

$$\begin{aligned} C_{i+1} &= \begin{bmatrix} C_i W_i \\ U_{i+1} \end{bmatrix} \\ Z_{i+1} &= \begin{bmatrix} Q_1 & \dots & Q_{i+1} \end{bmatrix} C_{i+1} \\ &= \begin{bmatrix} Z_i & Q_{i+1} \end{bmatrix} \begin{bmatrix} W_i \\ U_{i+1} \end{bmatrix}. \end{aligned}$$

Thus at any step we will keep the superdiagonal blocks of the Hessenberg matrix generated by Arnoldi in an SSS structure in left proper form.

## 10.3 Quadratic eigenvalue problems

Consider a large quadratic eigenvalue problem from the discretization of a continuous mechanical system with damping. The damping may be localised if, for example, it is the result of frictional contact at a small interface surface, or if it comes from an active control system. Similar low-rank damping terms occur due to radiation boundary conditions in some electromagnetic and acoustic scattering problems. If we scale the system to momentum coordinates so that the mass matrix is $I$, such a system can be described by a quadratic eigenvalue problem:

$$(\lambda^2 I + \lambda C + K)x = 0.$$

Assume $K$ is symmetric and positive definite and $C$ has low rank. Let $L$ be a Cholesky factor of $K$. Then

$$\begin{bmatrix} 0 & L \\ -L^T & -C \end{bmatrix} \begin{bmatrix} Lx \\ \lambda x \end{bmatrix} = \begin{bmatrix} Lx \\ \lambda x \end{bmatrix} \lambda$$

is a linearization of the eigenproblem which consists of a low-rank modification to a skew-symmetric matrix. We can compute the dominant eigenvalues of this system by, for example, using the storage-limited Arnoldi process described in the previous section.

## 11 Conclusions and future work

## References


[1] A. Bunse-Gerstner and H. Fabender, *Error bounds in the isometric Arnoldi process*, J. Comput. Appl. Math., 86 (1997), pp. 53–72.

[2] S. Chandrasekaran, P. Dewilde, M. Gu, T. Pals, and A.-J. van der Veen, *Fast stable solvers for sequentially semi-separable linear systems of equations*. Submitted to SIAM J. Matrix Anal. Appl. Apr. 2002.

[3] J. J. Dongarra, G. A. Geist, and C. H. Romine, *Algorithm 710: FORTRAN subroutines for computing the eigenvalues and eigenvectors of a general matrix by reduction to general tridiagonal form*, ACM Transactions on Mathematical Software, 18 (1992), pp. 392–400.

[4] A. Edelman and H. Murakami, *Polynomial roots from companion matrix eigenvalues*, Mathematics of Computation, 64 (1995), pp. 763–776.

[5] G. Golub and C. V. Loan, *Matrix Computations*, The John Hopkins University Press, 1989.

[6] W. B. Gragg, *The QR algorithm for unitary Hessenberg matrices*, J. Comput. Appl. Math., 16 (1986), pp. 1–8.

[7] C. Moler, *Roots – of polynomials, that is*, The Mathworks Newsletter, 5 (1991), pp. 8–9.

[8] B. Parlett, *The Symmetric Eigenvalue Problem*, SIAM, 1997.

[9] B. Parlett, *The QR algorithm*, Computing in Science and Engineering, 2 (2000), pp. 38–42. Special Issue: Top 10 Algorithms of the Century.

[10] M. Stewart, *An error analysis of a unitary Hessenberg QR algorithm*, Tech. Rep. TR-CS-98-11, Department of Computer Science, Australian National University, Canberra 0200 ACT, Australia, 1998.

[11] M. Stewart and P. V. Dooren, *On the factorization of hyperbolic and unitary transformations into rotations*. Submitted to SIAM J. Matrix Anal. Appl.


## A Characterizing low rank modifications to symmetric, skew, and orthogonal matrices

Let $\text{symm}(B) = (B + B^T)/2$ denote the symmetric part of $B$ and $\text{skew}(B) = (B - B^T)/2$ denote the skew part. For simplicity, assume in each of the following statements that $2k \leq n$.

**Theorem 1.** *Suppose $B \in \mathbb{R}^{n\times n}$. Then the following statements are equivalent.*

1. *$B = A + L$ for some real symmetric matrix $A$ and real low rank matrix $L$ with* $\text{rank}(L) \leq k$

2. $\text{rank}(\text{skew}(B)) \leq 2k$

*Proof.* First suppose $B = A + L$ where $A$ is symmetric and $L$ has rank $k$. Then $\text{skew}(B) = \text{skew}(L) = L/2 + L^T/2$ is the sum of two rank $k$ matrices, and so has rank at most $2k$.

Now suppose $\text{skew}(B)$ has rank at most $2k$. Then there is a real Schur decomposition

$$\text{skew}(B) = QTQ^T$$

where $Q$ is orthogonal and $T$ is a block diagonal matrix with all nonzero entries occuring in skew-symmetric 2-by-2 diagonal blocks, each corresponding to a complex conjugate pair of eigenvalues. Since the rank of $\text{skew}(B)$ is at most $2k$, there can be at most $k$ such nonzero 2-by-2 skew blocks. Any 2-by-2 skew block is the sum of a rank one matrix and a symmetric matrix:

$$\begin{bmatrix} 0 & -s \\ s & 0 \end{bmatrix} = \begin{bmatrix} -s & -s \\ s & s \end{bmatrix} + \begin{bmatrix} s & 0 \\ 0 & -s \end{bmatrix}.$$

Therefore, we can write $\text{skew}(B) = \hat{A} + \hat{L}$, where $\hat{A}$ is a symmetric matrix and $\hat{L}$ is a matrix of rank at most $k$ (rank one for each nonzero 2-by-2 diagonal block in $B$). Consequently, we can write $B = A + L$, where $A = \text{symm}(B) + Q\hat{A}Q^T$ and $L = Q\hat{L}Q^T$.

□

**Theorem 2.** *Suppose $B \in \mathbb{R}^{n \times n}$. Then the following statements are equivalent.*

1. *$B = S + L$ for some real skew matrix $S$ and real low rank matrix $L$ with* $\text{rank}(L) \leq k$

2. $\text{symm}(B)$ *has no more than $k$ negative eigenvalues and no more than $k$ positive eigenvalues.*

*Proof.* First suppose $B = S + L$ where $S$ is skew and $\text{rank}(L) \leq k$. Then $\text{symm}(B) = (L + L^T)/2$ is a symmetric matrix, to which we can apply the Courant-Fischer minimax theorem. Let $\mathcal{N}$ denote an $n-k$-dimensional subspace of the null space of $L$. If the eigenvalues of $\text{symm}(B)$ are $\lambda_1 \leq \ldots \leq \lambda_n$, then

$$\begin{aligned} \lambda_{n-k} &= \min_{\dim(\mathcal{S})=n-k} \max_{v \in \mathcal{S}-\{0\}} \frac{v^T(L+L^T)v}{2v^Tv} \\ &\leq \max_{v \in \mathcal{N}-\{0\}} \frac{v^T(L+L^T)v}{2v^Tv} = 0. \end{aligned}$$

Therefore $\lambda_1 \leq \ldots \leq \lambda_{n-k} \leq 0$, and so $\text{symm}(B)$ can have at most $k$ positive eigenvalues. Similarly,

$$\begin{aligned} \lambda_{k+1} &= \max_{\dim(\mathcal{S})=n-k} \min_{v \in \mathcal{S}-\{0\}} \frac{v^T(L+L^T)v}{2v^Tv} \\ &\geq \min_{v \in \mathcal{N}-\{0\}} \frac{v^T(L+L^T)v}{2v^Tv} = 0. \end{aligned}$$

So $0 \leq \lambda_{k+1} \ldots \leq \lambda_n$, and so symm$(B)$ can have at most $k$ negative eigenvalues.

Now suppose that $B$ is a matrix such that symm$(B)$ has at most $k$ positive eigenvalues and at most $k$ negative eigenvalues. Choose $Q$ such that $Q^T \text{symm}(B) Q = D$ is diagonal with diagonal entries

$$\begin{cases} d_i & \geq 0 \text{ for odd } i \leq 2k \\ d_i & \leq 0 \text{ for even } i \leq 2k \\ d_i & = 0 \text{ for } i > 2k \end{cases}$$

Now we add $k$ skew-symmetric rank one modifications to $D$, one to each of the first $k$ 2-by-2 blocks on the diagonal. Let $i$ be an odd number, $i \leq 2k$; then since $d_i d_{i+1} \leq 0$, we can find real $s_i = \sqrt{-d_i d_{i+1}}$. Thus, the diagonal block is rank one plus skew:

$$\begin{bmatrix} d_i & 0 \\ 0 & d_{i+1} \end{bmatrix} = \begin{bmatrix} d_i & s_i \\ -s_i & d_{i+1} \end{bmatrix} + \begin{bmatrix} 0 & -s_i \\ s_i & 0 \end{bmatrix}$$

Put together, we find that $D = \hat{S} + \hat{L}$, where $\hat{S}$ is skew and $\hat{L}$ is rank $k$. Therefore, $B = S + L$, where $S = \text{skew}(B) + Q\hat{S}Q^T$ is skew and $L = Q\hat{L}Q^T$ has rank $k$.

□

**Theorem 3.** *Suppose $B \in \mathbb{R}^{n \times n}$. Then the following statements are equivalent.*

1. *$B = Q + L$ for some real orthogonal matrix $Q$ and real low rank matrix $L$ with* rank$(L) \leq k$.
2. *$B$ has no more than $k$ singular values greater than one and no more than $k$ singular values less than one.*

*Proof.* First suppose $B = Q + L$ where $Q$ is orthogonal and rank$(L) \leq k$. Then $B^T B = I + Q^T L + L^T Q + L^T L$ is a symmetric matrix, to which we can apply the Courant-Fischer minimax theorem. Let $\mathcal{N}$ denote an $n - k$-dimensional subspace of the null space of $L$. If the eigenvalues of $B^T B$ are $\sigma_1^2 \leq \ldots \leq \sigma_n^2$, then

$$\begin{aligned} \sigma_{n-k}^2 &= \min_{\dim(\mathcal{S})=n-k} \max_{v \in \mathcal{S}-\{0\}} \frac{v^T(I + Q^T L + L^T Q + L^T L)v}{v^T v} \\ &\leq \max_{v \in \mathcal{N}-\{0\}} \frac{v^T(I + Q^T L + L^T Q + L^T L)v}{v^T v} = 1. \end{aligned}$$

Therefore $\sigma_1^2 \leq \ldots \leq \sigma_{n-k}^1 \leq 1$, and so $B$ can have at most $k$ singular values greater than one. Similarly,

$$\begin{aligned} \sigma_{k+1}^2 &= \max_{\dim(\mathcal{S})=n-k} \min_{v \in \mathcal{S}-\{0\}} \frac{v^T(I + Q^T L + L^T Q + L^T L)v}{v^T v} \\ &\geq \min_{v \in \mathcal{N}-\{0\}} \frac{v^T(I + Q^T L + L^T Q + L^T L)v}{v^T v} = 1. \end{aligned}$$

So $1 \leq \sigma_{k+1}^2 \ldots \leq \sigma_n^2$, and so $B$ can have at most $k$ singular values less than one.

Now suppose $B$ has at most $k$ singular values greater than one and at most $k$ singular values less than one. Suppose we write a reordered singular value decomposition $B = U\Sigma V^T$, where the first $k$ odd singular values are at most one, the first $k$ even singular values are at least one, and all the remaining singular values are exactly one. Consider the first $2k$ singular values in this order as $k$ 2-by-2 blocks. Each such block has the form

$$D = \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix}$$

such that $0 \leq d_1 \leq 1$ and $d_2 \geq 1$. Set $c = (1 + d_1 d_2)/(d_1 + d_2)$, and note that $0 \leq c \leq 1$ since both numerator and denominator are positive and

$$(1 + d_1 d_2) - (d_1 + d_2) = (1 - d_1)(1 - d_2) \leq 0.$$

We claim $D$ is within rank one of a plane rotation matrix $G$ with cosine and sine values $c$ and $s = \sqrt{1 - c^2}$ respectively:

$$\begin{aligned} \det(D - G) &= \det \begin{bmatrix} d_1 - c & -s \\ s & d_2 - c \end{bmatrix} \\ &= (d_1 - c)(d_2 - c) + s^2 \\ &= d_1 d_2 - (d_1 + d_2)c + (c^2 + s^2) \\ &= d_1 d_2 - (d_1 + d_2)c + 1 \\ &= 0. \end{aligned}$$

So by making $k$ rank one modifications to $\Sigma$, we can turn each pair of singular values straddling one into a 2-by-2 orthogonal block. Therefore we can write $\Sigma$ as

$$\Sigma = \hat{Q} + \hat{L}$$

where $\hat{Q}$ is orthogonal (diagonal with blocks which are either 1-by-1 or 2-by-2 orthogonal) and $\text{rank}(\hat{L}) \leq k$. Therefore $B = Q + L$ where $B = U\hat{Q}V^T$ is orthogonal and $L = U\hat{L}V^T$ has rank at most $k$.

□