SIAM J. MATRIX ANAL. APPL.
Vol. 12, No. 2, pp. 374–384, April 1991

013

CHASING ALGORITHMS FOR THE EIGENVALUE PROBLEM*

D. S. WATKINS† AND L. ELSNER‡

Abstract. A generic chasing algorithm for the matrix eigenvalue problem is introduced and studied. This algorithm includes, as special cases, the implicit, multiple-step QR and LR algorithms and similar bulge-chasing algorithms for the standard eigenvalue problem. The scope of the generic chasing algorithm is quite broad; it encompasses a number of chasing algorithms that cannot be analyzed by the traditional (e.g., implicit Q theorem) approach. These include the LR algorithm with partial pivoting and other chasing algorithms that employ pivoting for stability, as well as hybrid algorithms that combine elements of the LR and QR algorithms. The main result is that each step of the generic chasing algorithm amounts to one step of the generic GR algorithm. Therefore the convergence theorems for GR algorithms that were proven in a previous work [D. S. Watkins and L. Elsner, *Linear Algebra Appl.*, 143 (1991), pp. 19–47] also apply to the generic chasing algorithm.

Key words. eigenvalue, QR algorithm, GR algorithm, chasing the bulge, subspace iteration

AMS(MOS) subject classifications. 65F15, 15A18

1. Introduction.

Two of the best known algorithms for calculating eigenvalues and eigenvectors of matrices are the QR and LR algorithms [15], [12]. There are other, not so well known, algorithms of the same type, e.g., the SR algorithm [8], [9], [6] and the HR algorithm [5], [7], [6], which can be useful in special situations. In [14] we developed a general convergence theory of GR algorithms that includes the QR, LR, SR, HR, and similar algorithms as special cases. In this paper we consider in general terms the question of how such algorithms can be implemented.

Algorithms in this class are usually implemented implicitly, as chasing algorithms: The matrix whose eigenvalues we would like to know is first reduced to upper Hessenberg form. Then the chasing algorithm is set in motion by a similarity transformation that introduces a bulge in the Hessenberg form near the upper left-hand corner of the matrix. A sequence of similarity transformations then chases the bulge downward and to the right, until the Hessenberg form is restored. At this point the first chasing step is complete. Chasing steps are repeated until (hopefully) the matrix converges to upper triangular or block triangular form, exposing the eigenvalues. There are a number of types of similarity transformations that can be used to chase the bulge. For example, certain unitary transformations can be used, in which case each step of the chasing algorithm amounts to a step of the QR algorithm. If, on the other hand, lower triangular transformation matrices are used, each step of the chasing algorithm amounts to a step of the LR algorithm.

In this paper we introduce and study a generic chasing algorithm. After describing the algorithm at the beginning of §2, we state and prove the main result, which is that no matter what kind of transformations are used to chase the bulge, each chasing step amounts to one step of the generic GR algorithm [14]. Consequently all of the observations that we made in [14] concerning the generic GR algorithm apply to the chasing algorithm as well. To wit, each step of the chasing algorithm amounts to a step of nested subspace iteration combined with a change of coordinate system. All of the convergence theorems of [14] apply. Roughly speaking, the chasing

* Received by the editors March 6, 1989; accepted for publication (in revised form) May 24, 1990.
† Department of Pure and Applied Mathematics, Washington State University, Pullman, Washington 99164-3113 (watkins@wsumath.bitnet). The research of this author was supported by National Science Foundation grant DMS-8800437.
‡ Fakultät für Mathematik, Universität Bielefeld, Postfach 8640, D-4800 Bielefeld 1, Federal Republic of Germany (umatf105@dbuni11.bitnet).

algorithm will converge, provided that (i) reasonable choices of shifts are made, and (ii) the condition numbers of the transforming matrices are kept under control. If the generalized Rayleigh quotient shifting strategy is used, quadratic, and in some cases cubic, convergence can be achieved. We close §2 with a brief discussion of some of the types of transformation that can be used to implement the chasing algorithm.

Our approach to chasing algorithms differs from the traditional one. For purposes of illustration, let us consider the standard way of justifying the implicit QR algorithm. A QR step consists of a similarity transformation $B = Q^{-1}AQ$, where the transforming matrix Q is unitary. One can show that the unitary Q is more or less uniquely determined by its first column. (This fact is known as the *implicit Q theorem*; see, for example, [12, Thm. 7.4.2].) The implicit QR (chasing) algorithm performs a different similarity transformation $\tilde{B} = \tilde{Q}^{-1}A\tilde{Q}$, but $\tilde{Q}$ is constructed in such a way that its first column is proportional to the first column of Q. It follows from the implicit Q theorem that Q and $\tilde{Q}$ are essentially the same, and consequently B and $\tilde{B}$ are essentially the same.

By contrast, our generic chasing algorithm performs repeated similarity transformations $B = G^{-1}AG$, where the nature of G is left unspecified, except that it is nonsingular and its first column is given. In this more general context, we cannot assert that G is more or less uniquely determined. All we can say is that no matter how the chasing step is carried out, it effects one step of the generic GR algorithm. But this is all we need!

Our approach has the following advantages: (i) It covers implicit variants of the QR, LR, SR, and HR algorithms all at once. (ii) It covers the implicit LR algorithm with partial pivoting and other chasing algorithms that employ pivoting for stability, none of which are covered by the traditional approach. (iii) It covers hybrid chasing algorithms as well. For example, an algorithm that uses a mixture of unitary and lower triangular transformations to chase the bulge is a QR-LR hybrid that cannot be analyzed by the traditional approach. Thus our approach encompasses a much broader class of chasing algorithms.

The theorems associated with the traditional approach (e.g., implicit Q theorem) can be derived via our approach by considering the effect of restricting the types of transformations that can be used to do the chasing. This is the main business of §3. Here we restrict our attention, for clarity, to the nonsingular case. This is the generic case, in which none of the shifts (defined in §2) are eigenvalues of A.

In §4 we consider what happens during a singular chasing step. We show that if ν of the shifts are eigenvalues, a $\nu \times \nu$ block can, in principle, be deflated from the matrix after the chasing step. As far as the rest of the matrix is concerned, normal progress is made during the chasing step. That is, a step of nested subspace iteration, combined with a change of coordinate system, takes place. Also considered in §4 is the connection between our approach and the traditional approach in the singular case. Actually the results of §4 include those of §3 as a special case. We have chosen to present the nonsingular case separately because (i) it is generic, and (ii) it is much simpler.

Finally, we wish to emphasize that the theorems in this paper are not at all difficult, nor are the tools used to prove them by any means novel. This paper's main contribution is a new, more flexible, way of looking at chasing algorithms that allows for greater generality than has been attained previously.

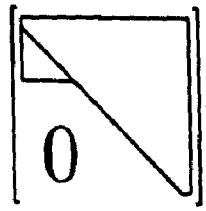
2. The generic GR and chasing algorithms. We described the generic GR algorithm in [14]. For completeness, we briefly repeat the description here. Each GR

algorithm is based on a GR decomposition, which is a rule that specifies a unique way of decomposing any matrix C in some large class of matrices $\mathcal{C}$ into a product $C = GR$, where G is nonsingular, and R is upper triangular. Well-known examples of GR decompositions are the QR and LR decompositions. Corresponding to each GR decomposition is a GR algorithm, an iterative algorithm for finding the eigenvalues of matrices. Given a matrix $A \in \mathbb{C}^{n\times n}$, whose eigenvalues we would like to know, the GR algorithm produces a sequence of similar matrices that, hopefully, converges to upper triangular or block triangular form. A GR step on A is performed as follows. Choose a positive integer m, the *multiplicity* of the step. Choose m *shifts* $\sigma_1, \cdots, \sigma_m$, complex numbers that approximate eigenvalues of A. Let $p(A) = (A-\sigma_1)\cdots(A-\sigma_m)$. Find the GR decomposition of $p(A)$: $p(A) = GR$. Finally, replace A by the similar matrix $B = G^{-1}AG$.

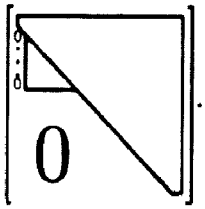
The generic chasing algorithm. A matrix $A = (a_{ij}) \in \mathbb{C}^{n\times n}$ is said to be in *upper Hessenberg* form if $a_{ij} = 0$ for all $i > j+1$. It is in *irreducible* upper Hessenberg form if it satisfies the additional condition $a_{ij} \neq 0$ for all $i = j+1$.

Let $A \in \mathbb{C}^{n\times n}$ be in irreducible upper Hessenberg form, let m be a positive integer, let $\sigma_1, \cdots, \sigma_m$ be approximations to eigenvalues of A, and let $p(A) = (A-\sigma_1)\cdots(A-\sigma_m)$. A step of the generic chasing algorithm of multiplicity m replaces A by $B = G^{-1}AG$, where B is upper Hessenberg, and the first column of G is proportional to the first column of $p(A)$.

The algorithm begins by computing the first column of $p(A)$. This is given by $x = p(A)e_1$, where $e_1 = [1, 0, \cdots, 0]^T$. To avoid potential problems with overflow or underflow, we actually compute $z = x/\|x\|$ using any convenient norm. This can be done efficiently, without actually forming $p(A)$, by the recursion $z \leftarrow (A-\sigma_i)z$, $z \leftarrow z/\|z\|$, $i = 1, \cdots, m$, starting from $z \leftarrow e_1$. Because A is irreducible upper Hessenberg and p has degree m, z satisfies $z_{m+1} \neq 0$, and $z_i = 0$ for $i > m+1$. The next step is to determine a nonsingular matrix $G_0 = \mathrm{diag}\{\tilde{G}_0, I_{n-m-1}\}$, with $\tilde{G}_0 \in \mathbb{C}^{(m+1)\times(m+1)}$, whose first column is proportional to z. Obviously $\tilde{G}_0$ is not uniquely determined. Some common ways of constructing $\tilde{G}_0$ will be discussed below. Once we have G_0, we use it to transform A to $A_0 = G_0^{-1}AG_0$. It is a simple matter to check that A_0 has the form



That is, it is almost in upper Hessenberg form, except that it has a triangular *bulge* with height m rows, base m columns, and vertex at position $(m+2, 1)$. The rest of the algorithm consists of returning A_0 to upper Hessenberg form. The first step in this direction is to transform A_0 to $A_1 = G_1^{-1}A_0G_1$, where G_1 has the following form: $G_1 = \mathrm{diag}\{1, \tilde{G}_1, I_{n-m-2}\}$, $\tilde{G}_1 \in \mathbb{C}^{(m+1)\times(m+1)}$, and the first column of $\tilde{G}_1$ is proportional to the vector $y_1 \in \mathbb{C}^{m+1}$ consisting of the $(2,1)$ through $(m+2,1)$ entries of A_0. It is clear that $\tilde{G}_1$ is not uniquely determined and that any of the techniques for constructing $\tilde{G}_0$ can also be used to construct $\tilde{G}_1$. Since $\tilde{G}_1^{-1}y_1 = ce_1$ for some nonzero constant c, left multiplication of A_0 by G_1^{-1} will create zeros in the first column of the product, below the $(2,1)$ entry. The subsequent right multiplication of $G_1^{-1}A_0$ by G_1 leaves the first column unaltered, but it does create new nonzero entries in row $m+3$, in positions $(m+3, 2)$ through $(m+3, m+1)$. Thus A_1 has the form



The bulge has been "chased" one position down and to the right, so that its vertex now lies at position $(m+3,2)$. The next step produces $A_2 = G_2^{-1}A_1G_2$, where $G_2 = \text{diag}\{1, 1, \tilde{G}_2, I_{n-m-3}\}$, $\tilde{G}_2$ being defined analogously to $\tilde{G}_1$. A_2 has a bulge that is down and to the right one position from that of A_1. After $n-m-2$ such steps, the bulge will have been chased to the lower right-hand corner of the matrix. An additional m steps shrink the bulge until it disappears completely. Another viewpoint is that the bulge is pushed off of the edge of the matrix. The similarity transformations used in this phase have the form $G_k = \text{diag}\{I_k, \tilde{G}_k\}$, where $\tilde{G}_k \in \mathbb{C}^{(n-k)\times(n-k)}$, $k = n-(m+1), \cdots, n-2$. After a total of $n-2$ steps we are done; we let $B = A_{n-2} = G^{-1}AG$, where $G = G_0G_1G_2\cdots G_{n-2}$. Each of the matrices $G_1, \cdots, G_{n-2}$ has e_1 as its first column, so the first column of G is the same as the first column of G_0, which is proportional to the first column of $p(A)$.

In the past it has been customary to take m to be a small number, say one or two. The advantage of taking larger values of m is that it improves the vectorizability of the code. The main operations can be expressed as matrix-vector products, and level 2 BLAS [11] can be used. It is also possible to organize the algorithm so that several columns are chased at a time, using tools such as the WY representation of reflectors [4]. This allows the main operations to be expressed as matrix-matrix products, and the algorithm can be coded using level 3 BLAS [10]. This increases the scope for parallelization and efficient use of hierarchical memory. Bai and Demmel [3] have implemented such a version of the QR algorithm and have experimented with values of m as high as 20.

The main point of this paper is that the chasing algorithm effectively performs a step of the generic GR algorithm. The proof follows from three lemmas, whose proofs are easy exercises. We begin with some terminology. Given $x \in \mathbb{C}^n$ and $A \in \mathbb{C}^{n\times n}$, the *Krylov matrix* $K(A,x) \in \mathbb{C}^{n\times n}$ is defined by $K(A,x) = [x, Ax, A^2x, \cdots, A^{n-1}x]$. Clearly $K(A,\beta x) = \beta K(A,x)$ for every scalar β.

LEMMA 2.1. *If $x = p(A)e_1$, then $K(A,x) = p(A)K(A,e_1)$.*

LEMMA 2.2. *For every nonsingular $G \in \mathbb{C}^{n\times n}$, $G^{-1}K(A,x) = K(G^{-1}AG, G^{-1}x)$.*

Lemmas 2.1 and 2.2 are closely related. In fact, Lemma 2.1 for nonsingular $p(A)$ is a special case of Lemma 2.2. However, Lemma 2.1 is valid regardless of whether or not $p(A)$ has an inverse, the key point being that $p(A)$ commutes with A.

LEMMA 2.3. *If A is upper Hessenberg, then $K(A,e_1)$ is upper triangular. Furthermore A is an irreducible upper Hessenberg matrix if and only if $K(A,e_1)$ is upper triangular and nonsingular.*

We are now set to prove the main result.

THEOREM 2.4. *Let $A \in \mathbb{C}^{n\times n}$ be an irreducible upper Hessenberg matrix, and let p be a polynomial. Let G be a nonsingular matrix whose first column is proportional to $x = p(A)e_1$, such that $B = G^{-1}AG$ is upper Hessenberg. Then there exists an upper triangular matrix R such that $p(A) = GR$.*

Proof. By hypothesis, $Ge_1 = \alpha x$ for some nonzero α. Applying Lemmas 2.1 and 2.2 we find that $G^{-1}p(A)K(A,e_1) = G^{-1}K(A,x) = \alpha^{-1}K(B,e_1)$. By Lemma 2.3, $K(A,e_1)$ is both nonsingular and upper triangular, and $K(B,e_1)$ is upper triangular. Therefore $p(A) = G\alpha^{-1}K(B,e_1)K(A,e_1)^{-1} = GR$, where R is the upper triangular

matrix $\alpha^{-1}K(B,e_1)K(A,e_1)^{-1}$. □

Theorem 2.4 shows that the generic chasing algorithm performs a generic GR step implicitly. In this case the rule for calculating the GR decomposition (i.e., constructing G from $p(A)$) is given by the chasing algorithm itself.

There are numerous ways of constructing the $\tilde{G}_i$ in the chasing algorithm. Each $\tilde{G}$ satisfies $\tilde{G}e_1 = \beta y$, or equivalently $\tilde{G}^{-1}y = \beta^{-1}e_1$, for some y. One way to do this, which works if $y_1 \neq 0$, is to define $\tilde{G}$ to be a Gauss transformation:

$$\tilde{G} = \begin{bmatrix} 1 & & & \\ \ell_2 & 1 & & \\ \vdots & & \ddots & \\ \ell_{m+1} & & & 1 \end{bmatrix}, \tag{1}$$

where $\ell_i = y_i/y_1$. If this choice is used, the chasing algorithm amounts to the LR algorithm without pivoting (cf. Example L below). We can eliminate the requirement that $y_1 \neq 0$ by defining $\tilde{G} = PL$, where P is either the identity matrix or a permutation matrix, and L has the form (1). If $|y_1| = \max\{|y_1|, \cdots, |y_{m+1}|\}$, we define P to be the identity matrix. Otherwise we take P to be the transposition matrix whose action on a column vector is to interchange its first and ith entries, where i is the first index for which $|y_i| = \max\{|y_1|, \cdots, |y_{m+1}|\}$. Defining a new vector z by $Py = z$, we then take L to have the form (1), where $\ell_i = z_i/z_1$. This choice yields the LR algorithm with partial pivoting. It is also possible to take the $\tilde{G}$ to be unitary matrices, for example, reflectors (Householder transformations) [12]. In this case the chasing algorithm amounts to the QR algorithm (cf. Example Q below).

For some problems the choice of transformation type is dictated by the structure of the matrix. For example, if A_0 is normal, and we wish to preserve that property, we should use only unitary transformations. If A_0 is Hamiltonian, and we wish to preserve that property, we should use symplectic transformations [6]. This gives the SR algorithm (cf. Example S below).

Other problems have no special structure to exploit. For these problems the transformation type is chosen on the basis of efficiency and stability. From [14] we know that no matter how we choose the G_i, we are doing subspace iteration. The theory developed in [14] suggests that our only consideration is to make the transforming matrices as well conditioned as possible. We might then conclude that we should use only unitary transformations, which are optimally conditioned; that is, we should use the QR algorithm. This conclusion ignores the question of cost. A chasing step using reflectors has about double the flop count of a chasing step using Gauss transformations. Thus LR steps are about half as expensive as QR steps. Of course, the use of Gauss transformations without pivoting is risky; matrices of the form (1) can be made arbitrarily ill conditioned by making the multipliers ℓ_i large. On the other hand, Gauss transformations with partial pivoting, $\tilde{G} = PL$, tend to be well conditioned, as the multipliers never exceed one in modulus. Furthermore, as the iterates approach triangular or block triangular form, the multipliers in the transformations that are generated tend toward zero. As the multipliers approach zero, the condition numbers approach one. Of course, this does not guarantee that the condition numbers of products of many such transformations will remain small. Another possibility, which we already mentioned in the Introduction, is to mix Gauss transformations with unitary transformations. Our theory allows us to do this. All that matters is that the condition numbers of the transforming matrices be kept under control. A hybrid algorithm of this type might well possess a superior combination of speed and robustness.

3. Connection with the traditional approach. In this section we show how our approach can be used to establish the implicit

Q theorem and other results associated with the traditional approach to chasing algorithms. For clarity we restrict our attention to the nonsingular case. That is, we assume that $p(A)$ is nonsingular, which is the same as to say that none of the shifts $\sigma_1, \cdots, \sigma_m$ are eigenvalues of A. We begin by noting that in this case, the matrix B produced by the chasing algorithm is in irreducible upper Hessenberg form.

THEOREM 3.1. *Let $A \in \mathbb{C}^{n\times n}$ be an irreducible upper Hessenberg matrix, and let p be a polynomial for which $p(A)$ is nonsingular. Let G be a nonsingular matrix whose first column is proportional to $x = p(A)e_1$, such that $B = G^{-1}AG$ is upper Hessenberg. Then B has irreducible upper Hessenberg form.*

Proof. The hypotheses are the same as in Theorem 2.4, except that now we are assuming that $p(A)$ is nonsingular. As in the proof of Theorem 2.4, we have $K(B, e_1) = \alpha G^{-1}p(A)K(A, e_1)$. Since G^{-1}, $p(A)$ and $K(A, e_1)$ are all nonsingular, $K(B, e_1)$ must also be nonsingular, in addition to being upper triangular. Therefore, by Lemma 2.3, B has irreducible upper Hessenberg form. □

As we have already mentioned in the Introduction, the transforming matrix G is not uniquely determined by its first column. However, G does have some structure that is specified uniquely, namely, its flag. This useful concept comes from geometry [1], [2]. A *flag* in $\mathbb{C}^n$ is just a nested sequence of subspaces of dimensions $1, 2, \cdots, n$. Given a nonsingular matrix $S \in \mathbb{C}^{n\times n}$ with columns $s_1, \cdots, s_n$, we define the *flag* of S, denoted flag(S), to be the sequence $\{\langle s_1\rangle, \langle s_1, s_2\rangle, \langle s_1, s_2, s_3\rangle, \cdots, \langle s_1, s_2, \cdots, s_n\rangle\}$ determined by the columns of S. It is a simple matter to prove the following lemma.

LEMMA 3.2. *Two nonsingular matrices $S, G \in \mathbb{C}^{n\times n}$ have the same flag if and only if there is a nonsingular upper triangular matrix R such that $S = GR$.*

Theorem 2.4 shows that $p(A) = GR$ for some upper triangular R. Since we are now assuming that $p(A)$ is nonsingular, R must also be nonsingular. Therefore flag(G) = flag$(p(A))$. This holds regardless of the type of transformations that are used to build G. We did not use the term "flag" in [14]. However, the nested subspace iterations that are effected by GR steps can be seen to be a consequence of this equality of flags.

From Lemma 2.3 we know that $K(A, e_1)$ is upper triangular and nonsingular. Therefore, by Lemma 2.1, flag$(p(A))$ = flag$(K(A, x))$. Consequently flag(G) = flag$(K(A, x))$. This is actually a special case of a known result that characterizes transformations that reduce a matrix to upper Hessenberg form: Let $A \in \mathbb{C}^{n\times n}$, and suppose there is a vector $x \in \mathbb{C}^n$ for which $K(A, x)$ is nonsingular. Let G be a nonsingular matrix whose first column is proportional to x. Then $B = G^{-1}AG$ is upper Hessenberg if and only if flag(G) = flag$(K(A, x))$. If B is upper Hessenberg, then it is irreducible. See especially [6, Satz 4.4.1], but also [12, Thm. 7.4.3]. The case of singular $K(A, x)$ is also covered in [6], but we will give a more general formulation of that case in §4. For now we will state a portion of this result as a lemma for immediate use.

LEMMA 3.3. *Let $A \in \mathbb{C}^{n\times n}$ and let G be a nonsingular matrix such that $B = G^{-1}AG$ is in irreducible upper Hessenberg form. Let $x \in \mathbb{C}^n$ be a vector proportional to the first column of G. Then $K(A, x)$ is nonsingular, and* flag(G) = flag$(K(A, x))$.

Proof. By Lemma 2.2, $K(A, x) = G\alpha^{-1}K(B, e_1)$. Since B is irreducible upper Hessenberg, $K(B, e_1)$ is upper triangular and nonsingular. Thus flag(G) = flag$(K(A, x))$. □

The transforming matrices G utilized by GR algorithms always lie in $GL_n(\mathbb{C})$,

the group of nonsingular matrices in $\mathbb{C}^{n\times n}$. Nothing more than that is said, in general. However, certain GR algorithms (e.g., the QR algorithm) use only transforming matrices that lie in some proper subgroup $\mathcal{G}$ (e.g., the unitary group). Similarly, one may be able to implement the chasing algorithm in such a way that the transforming matrices all lie in $\mathcal{G}$ (e.g., implicit QR algorithm). The next theorem shows that in such cases the transforming matrices produced by the two algorithms are the same up to right multiplication by a matrix in a certain subgroup $\mathcal{T}$, which we call the *trivial* group. If $\mathcal{G}$ is not too large, then $\mathcal{T}$ really is trivial, and we can conclude that the chasing algorithm and the GR algorithm produce essentially the same result. Examples are given below.

THEOREM 3.4. *Let $\mathcal{G}$ be a subgroup of $GL_n(\mathbb{C})$. Define the* trivial *group $\mathcal{T}$ associated with $\mathcal{G}$ by $\mathcal{T} = \mathcal{G} \cap \mathcal{U}$, where $\mathcal{U}$ denotes the subgroup of $GL_n(\mathbb{C})$ consisting of upper triangular matrices. Let $A \in \mathbb{C}^{n\times n}$, let $G, \tilde{G} \in \mathcal{G}$ have proportional first columns, let $B = G^{-1}AG$ and $\tilde{B} = \tilde{G}^{-1}A\tilde{G}$, and suppose B and $\tilde{B}$ are both irreducible upper Hessenberg. Then there exists $T \in \mathcal{T}$ such that $\tilde{G} = GT$ and $\tilde{B} = T^{-1}BT$.*

Proof. By Lemma 3.3, $\mathrm{flag}(\tilde{G}) = \mathrm{flag}(K(A,x)) = \mathrm{flag}(G)$, where x is a vector proportional to the first columns of G and $\tilde{G}$. Therefore there exists $T \in \mathcal{U}$ such that $\tilde{G} = GT$. But $T = G^{-1}\tilde{G}$, so $T \in \mathcal{G}$. Thus $T \in \mathcal{G} \cap \mathcal{U} = \mathcal{T}$. □

Example Q. The m-step QR algorithm performs a similarity transformation $\tilde{B} = \tilde{Q}^{-1}A\tilde{Q}$, where $\tilde{Q}$ is unitary and $p(A) = \tilde{Q}\tilde{R}$. Similarly, if the chasing algorithm is carried out using unitary transformations exclusively, it performs a similarity transformation $B = Q^{-1}AQ$, where Q is unitary and $p(A) = QR$. Since $\tilde{Q}$ and Q must have proportional first columns, Theorem 3.4 can be applied, with the role of $\mathcal{G}$ played by the unitary group. The group $\mathcal{T}$ associated with this choice of $\mathcal{G}$ is the group of diagonal matrices whose main diagonal entries have unit modulus. Thus $\tilde{Q} = QD$, where D is diagonal with $|d_{ii}| = 1$ for $i = 1, \cdots, n$. This is the complex version of the implicit Q theorem. Thus the chasing algorithm using unitary transformations produces essentially the same result as the QR algorithm.

Example L. The m-step LR algorithm without pivoting performs a similarity transformation $\tilde{B} = \tilde{L}^{-1}A\tilde{L}$, where $\tilde{L}$ is unit lower triangular, and $p(A) = \tilde{L}\tilde{R}$. Similarly, if the chasing algorithm is carried out using Gauss transformations (1) exclusively, it performs a similarity transformation $B = L^{-1}AL$, where L is unit lower triangular, and $p(A) = LR$. Thus Theorem 3.4 applies with $\mathcal{G}$ taken to be the group of unit lower triangular matrices. Then $\mathcal{T} = \{I\}$, so $\tilde{G} = G$ and $\tilde{B} = B$. We conclude that the chasing algorithm using Gauss transformations without pivoting produces exactly the same result as the LR algorithm without pivoting.

Example S. Let n and m be even. The m-step SR algorithm performs a similarity transformation by a symplectic matrix. Taking $\mathcal{G}$ to be the symplectic group (in shuffled form (cf. [14])), we find that $\mathcal{T}$ is the group of all block diagonal matrices $T = \mathrm{diag}\{T_1, \cdots, T_k\}$ (where $k = n/2$), for which each block has the form

$$T_i = \begin{bmatrix} a_i & b_i \\ 0 & a_i^{-1} \end{bmatrix}.$$

Thus the chasing algorithm using symplectic transformations produces a result that differs from the output of the SR algorithm only by a similarity transformation of this simple form.

Remark. In the case $\mathcal{G} = GL_n(\mathbb{C})$, Theorem 3.4 reduces to part (iii) of [6, Satz 4.4.1].

4. The singular case. Before considering the singular case, we introduce some new terminology and present some preliminary results. Given $B \in \mathbb{C}^{n\times n}$ and $j \in \{1,\cdots,n-1\}$, we will say that B is *j-Hessenberg* if its first j columns are in upper Hessenberg form; that is, if B has the form

$$B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix},$$

where $B_{11} \in \mathbb{C}^{j\times j}$ is upper Hessenberg, and $B_{21} \in \mathbb{C}^{(n-j)\times j}$ consists entirely of zeros, with the possible exception of the single entry $b_{j+1,j}$ in the upper right-hand corner. We will call B *j-reducible* j-Hessenberg if it is j-Hessenberg, B_{11} is irreducible upper Hessenberg, and $B_{21} = 0$. For the sake of completeness we also include the case $j = n$; the term n-reducible n-Hessenberg will be a synonym for irreducible upper Hessenberg. The first lemma generalizes Lemma 2.3. Again, we leave the proof as an exercise.

LEMMA 4.1. *Let $B \in \mathbb{C}^{n\times n}$ and suppose $K(B,e_1)$ has rank j. Then the following four conditions are equivalent:*

(i) *$K(B,e_1)$ is upper triangular.*

(ii) *$K(B,e_1)$ has the form*

$$K(B,e_1) = \begin{bmatrix} S_{11} & S_{12} \\ 0 & 0 \end{bmatrix}, \tag{2}$$

where $S_{11} \in \mathbb{C}^{j\times j}$ is upper triangular and nonsingular.

(iii) *B is j-Hessenberg.*

(iv) *B is j-reducible j-Hessenberg.*

Extending the definition of the Krylov matrix $K(A,x)$, we define the $n \times j$ *Krylov matrix* $K(A,x,j)$ by $K(A,x,j) = [x, Ax, A^2x, \cdots, A^{j-1}x]$. We also need to extend the definition of the flag of a matrix. Let $S \in \mathbb{C}^{n\times j}$ have linearly independent columns $s_1,\cdots,s_j$. We define the *flag* of S to be the nested seqence of j subspaces $\{\langle s_1\rangle, \langle s_1,s_2\rangle, \langle s_1,s_2,s_3\rangle, \cdots, \langle s_1,s_2,\cdots,s_j\rangle\}$. Generalizing Lemma 3.2, we have Lemma 4.2.

LEMMA 4.2. *Two full-rank matrices $S,G \in \mathbb{C}^{n\times j}$ have the same flag if and only if there is a nonsingular upper triangular matrix $R \in \mathbb{C}^{j\times j}$ such that $S = GR$.*

The next theorem extends parts (i) and (ii) of [6, Satz 4.4.1].

THEOREM 4.3. *Let $A \in \mathbb{C}^{n\times n}$ and $x \in \mathbb{C}^n$, $x \neq 0$, with $\operatorname{rank}(K(A,x)) = j$. Let $G \in \mathbb{C}^{n\times n}$ be a nonsingular matrix whose first column is proportional to x, and let $B = G^{-1}AG$. Define submatrices $G_1 \in \mathbb{C}^{n\times j}$ and $G_2 \in \mathbb{C}^{n\times(n-j)}$ by $G = [G_1, G_2]$. Then the following three conditions are equivalent:*

(i) *B is j-Hessenberg.*

(ii) *B is j-reducible j-Hessenberg.*

(iii) $\operatorname{flag}(K(A,x,j)) = \operatorname{flag}(G_1)$.

Proof. Since $x = \alpha G e_1$ for some nonzero α, we have

$$K(A,x) = \alpha G K(B,e_1) \tag{3}$$

by Lemma 2.2. Thus the hypothesis of Lemma 4.1, $\operatorname{rank}(K(B,e_1)) = j$, holds. Therefore (i) and (ii) are equivalent. We now show that (iii) is equivalent to (i) and (ii). Suppose B is j-Hessenberg. Then $K(B,e_1)$ is upper triangular and has the special form (2). Writing (3) in block form, we find that it implies $K(A,x,j) = G_1(\alpha S_{11})$, where αS_{11} is upper triangular and nonsingular. Thus $\operatorname{flag}(K(A,x,j)) = \operatorname{flag}(G_1)$.

Conversely, suppose $\text{flag}(K(A,x,j)) = \text{flag}(G_1)$. Then $K(A,x,j) = G_1S$, where S is upper triangular and nonsingular. We find by inspection that $AK(A,x,j) = K(A,x,j)C$, where $C \in \mathbb{C}^{j\times j}$ is a companion matrix:

$$C = \begin{bmatrix} 0 & \cdots & 0 & * \\ 1 & & 0 & * \\ & \ddots & & \vdots \\ & & 1 & * \end{bmatrix}.$$

In particular, C is irreducible upper Hessenberg. Combining the equations $AK(A,x,j) = K(A,x,j)C$ and $K(A,x,j) = G_1S$, we find that $AG_1S = G_1SC$, or $AG_1 = G_1H$, where $H = SCS^{-1}$ is irreducible upper Hessenberg. Define $F \in \mathbb{C}^{n\times n}$ by $F^* = G^{-1}$, and make the partition $F = [F_1, F_2]$, where $F_1 \in \mathbb{C}^{n\times j}$. Then $F_1^*G_1 = I \in \mathbb{C}^{j\times j}$ and $F_2^*G_1 = 0 \in \mathbb{C}^{(n-j)\times j}$. Also

$$B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} F_1^*AG_1 & F_1^*AG_2 \\ F_2^*AG_1 & F_2^*AG_2 \end{bmatrix}.$$

Thus $B_{11} = F_1^*AG_1 = F_1^*G_1H = H$ and $B_{21} = F_2^*AG_1 = F_2^*G_1H = 0$. Therefore B has j-reducible j-Hessenberg form. □

We are now ready to consider a step of the chasing algorithm for which $p(A)$ is singular. Write $p(A)$ in the factored form

$$p(A) = (A - \sigma_1)(A - \sigma_2)\cdots(A - \sigma_m).$$

$p(A)$ is singular if and only if at least one of the shifts σ_i is an eigenvalue of A. Let ν denote the number of shifts that are equal to eigenvalues of A. Here we count a repeated shift according to its multiplicity as a zero of p, except that the number of times we count it must not exceed its multiplicity as a zero of the characteristic polynomial of A (algebraic multiplicity).

LEMMA 4.4. *The rank of $p(A)$ is $n - \nu$.*

Proof. Since A has irreducible upper Hessenberg form, its eigenspaces are one-dimensional. Thus A has just one Jordan block [13] associated with each eigenvalue; that is, A is nonderogatory. Let $J = \text{diag}\{J_1, \cdots, J_k\}$ be the Jordan canonical form of A. Since the Jordan blocks correspond to distinct eigenvalues, each shift can be an eigenvalue of at most one block. For $i = 1, \cdots, k$, let λ_i be the eigenvalue associated with the block J_i, and let n_i be the dimension of the block. Let $\tilde{\nu}_i$ be the number of shifts that are equal to λ_i, and let $\nu_i = \min\{\tilde{\nu}_i, n_i\}$. Then $\nu = \sum_{i=1}^{k} \nu_i$. For each i, consider the factored form $p(J_i) = (J_i - \sigma_1)\cdots(J_i - \sigma_m)$. The factor $J_i - \sigma_l$ is nonsingular if and only if $\sigma_l \neq \lambda_i$. If $\sigma_l = \lambda_i$, then $J_i - \sigma_l = N$, where N is the nilpotent matrix with ones on the superdiagonal and zeros elsewhere. Since $\tilde{\nu}_i$ of the factors are equal to N, $p(J_i)$ has the form $p(J_i) = MN^{\tilde{\nu}_i} = MN^{\nu_i}$, where M is nonsingular. The nullity of N^{ν_i}, hence also of $p(J_i)$, is ν_i. The nullity of $p(J)$ is the sum of the nullities of the blocks, which is ν. Thus $\text{rank}(p(A)) = \text{rank}(p(J)) = n - \nu$. □

THEOREM 4.5. *Let $B = G^{-1}AG$ be the outcome of one step of the generic chasing algorithm in which $\text{rank}(p(A)) = n - \nu = j$. Then $p(A) = GR$, where R is an upper triangular matrix of the form*

$$R = \begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \end{bmatrix},$$

with $R_{11} \in \mathbb{C}^{j\times j}$ upper triangular and nonsingular. Furthermore B has j-reducible j-Hessenberg form:

$$B = \begin{bmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{bmatrix}.$$

The eigenvalues of $B_{22} \in \mathbb{C}^{\nu\times\nu}$ are just the ν shifts σ_i that are eigenvalues of A.

Proof. From the proof of Theorem 2.4 we know that $p(A) = GR$, where $R = \alpha^{-1}K(B,e_1)K(A,e_1)^{-1}$. Since R must have rank j, so must $K(B,e_1)$. Therefore, by Lemma 4.1, $K(B,e_1)$ must have the form (2). It follows immediately that R has the same form. The form of B also follows from Lemma 4.1.

It would appear to be an easy result that the eigenvalues of B_{22} are exactly the shifts that are eigenvalues of A. Consider first the case in which the eigenvalues of A are distinct. The eigenvalues of B_{11} are just those of $A|_{R(p(A))}$. The range of $p(A)$ is exactly the invariant subspace of A associated with the eigenvalues that are not among the shifts. The eigenvalues of B_{22} are the remaining eigenvalues of A, namely, those that are shifts. If A has multiple eigenvalues, this argument is clouded by the multiplicity question: it is possible that B_{11} and B_{22} have common eigenvalues. However, a careful inspection of $p(J)$, where J is the (nonderogatory) Jordan form of A, reveals that the argument can be extended to the general situation: If λ_i is an eigenvalue of J of multiplicity n_i and is used as a shift of multiplicity ν_i, with $\nu_i < n_i$, then λ_i is an eigenvalue of $J|_{R(p(J))}$ (hence of B_{11}) with multiplicity $n_i - \nu_i$. Therefore λ_i must be an eigenvalue of B_{22} of multiplicity ν_i. □

Theorem 4.5 shows that singular $p(A)$ are desirable, as they allow the problem to be deflated after one step. Of course this result ignores the effect of roundoff errors, which will cause $b_{j+1,j}$ to be nonzero in practice. Experience suggests that the computed $b_{j+1,j}$ will usually be large enough to prevent deflation.

Our final task is to extend Theorem 3.4 and its corollaries (Examples Q, R, and S). Let $\mathcal{G}$ be a subgroup of $GL_n(\mathbb{C})$, and for $j = 1,\cdots,n$, let $\mathcal{G}_j$ denote the subset of $GL_j(\mathbb{C})$ consisting of all $G \in \mathbb{C}^{j\times j}$ for which there exist $X \in \mathbb{C}^{j\times(n-j)}$ and $Y \in \mathbb{C}^{(n-j)\times(n-j)}$ such that $\left[\begin{smallmatrix} G & X \\ 0 & Y \end{smallmatrix}\right] \in \mathcal{G}$. It is easy to show that $\mathcal{G}_j$ is a subgroup of $GL_j(\mathbb{C})$. Let $\mathcal{U}_j$ denote the upper triangular subgroup of $GL_j(\mathbb{C})$, and let $\mathcal{T}_j = \mathcal{G}_j \cap \mathcal{U}_j$.

Example Q′. If $\mathcal{G}$ is the unitary group, then $\mathcal{G}_j$ is the unitary group in $GL_j(\mathbb{C})$, so $\mathcal{T}_j$ is the group of $j \times j$ diagonal matrices with main diagonal elements of unit modulus.

Example L′. If $\mathcal{G}$ is the group of unit lower triangular matrices in $GL_n(\mathbb{C})$, then $\mathcal{G}_j$ is the group of unit lower triangular matrices in $GL_j(\mathbb{C})$, so $\mathcal{T}_j$ is the subgroup of $GL_j(\mathbb{C})$ consisting of the single element I.

Example S′. If $\mathcal{G}$ is the symplectic group, and j is even, then $\mathcal{G}_j$ is the symplectic group in $GL_j(\mathbb{C})$, so $\mathcal{T}_j$ is the group of block diagonal matrices in $GL_j(\mathbb{C})$ with 2×2 blocks of the form given in Example S.

THEOREM 4.6. *Let $x \in \mathbb{C}^n$ and $A \in \mathbb{C}^{n\times n}$, with* $\operatorname{rank}(K(A,x)) = j$. *Let $\mathcal{G}$ be a subgroup of $GL_n(\mathbb{C})$, and let $G, \tilde{G} \in \mathcal{G}$ be matrices whose first columns are proportional to x. Suppose $B = G^{-1}AG$ and $\tilde{B} = \tilde{G}^{-1}A\tilde{G}$ both have j-Hessenberg form. Then both are j-reducible, and there exists $T \in \mathcal{T}_j$ such that $\tilde{B}_{11} = T^{-1}B_{11}T$.*

Proof. By Theorem 4.3 we know that B and $\tilde{B}$ are both j-reducible. Furthermore, $\operatorname{flag}(G_1) = \operatorname{flag}(K(A,x,j)) = \operatorname{flag}(\tilde{G}_1)$, where G_1 is defined as in Theorem 4.3, and $\tilde{G}_1$ is defined analogously. Thus there is a $T \in \mathcal{U}_j$ such that $\tilde{G}_1 = G_1T$. We will show that $T \in \mathcal{G}_j$ also, so that in fact $T \in \mathcal{T}_j$. Obviously $G^{-1}\tilde{G} \in \mathcal{G}$. Defining

$F = [F_1, F_2]$ by $F^* = G^{-1}$, as in Theorem 4.3, we have $F_1^*\tilde{G}_1 = F_1^*G_1T = T$ and $F_2^*\tilde{G}_1 = F_2^*G_1T = 0$, so

$$G^{-1}\tilde{G} = \begin{bmatrix} F_1^*\tilde{G}_1 & F_1^*\tilde{G}_2 \\ F_2^*\tilde{G}_1 & F_2^*\tilde{G}_2 \end{bmatrix} = \begin{bmatrix} T & F_1^*\tilde{G}_2 \\ 0 & F_2^*\tilde{G}_2 \end{bmatrix}.$$

This proves that $T \in \mathcal{G}_j$, whence $T \in \mathcal{T}_j$. The equation $B = G^{-1}AG$ implies $AG = GB$. Since B is j-reducible, this implies in turn that $AG_1 = G_1B_{11}$. Similarly $A\tilde{G}_1 = \tilde{G}_1\tilde{B}_{11}$. Thus $\tilde{G}_1\tilde{B}_{11} = A\tilde{G}_1 = AG_1T = G_1B_{11}T = \tilde{G}_1(T^{-1}B_{11}T)$. Since $\tilde{G}_1$ has full rank, we can conclude that $\tilde{B}_{11} = T^{-1}B_{11}T$. □

REFERENCES

[1] G. AMMAR, *Geometric aspects of Hessenberg matrices*, Contemp. Math., 68 (1987), pp. 1–21.

[2] G. AMMAR AND C. MARTIN, *The geometry of matrix eigenvalue methods*, Acta Appl. Math., 5 (1986), pp. 239–278.

[3] Z. BAI AND J. DEMMEL, *On a block implementation of Hessenberg multishift iteration*, Internat. J. High Speed Comput., 1 (1989), pp. 97–121.

[4] C. BISCHOF AND C. VAN LOAN, *The WY representation for products of Householder matrices*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. s2–s13.

[5] M. A. BREBNER AND J. GRAD, *Eigenvalues of $Ax = \lambda Bx$ for real symmetric matrices A and B computed by reduction to a pseudosymmetric form and the HR process*, Linear Algebra Appl., 43 (1982), pp. 99–118.

[6] W. BUNSE AND A. BUNSE-GERSTNER, *Numerische lineare Algebra*, Teubner, Stuttgart, 1985.

[7] A. BUNSE-GERSTNER, *An analysis of the HR algorithm for computing the eigenvalues of a matrix*, Linear Algebra Appl., 35 (1981), pp. 155–178.

[8] A. BUNSE-GERSTNER AND V. MEHRMANN, *A symplectic QR-like algorithm for the solution of the real algebraic Riccati equation*, IEEE Trans. Automat. Control, 31 (1986), pp. 1104–1113.

[9] A. BUNSE-GERSTNER, V. MEHRMANN, AND D. WATKINS, *An SR algorithm for Hamiltonian matrices based on Gaussian elimination*, Methods Oper. Res., 58 (1989), pp. 339–358.

[10] J. DONGARRA, J. DU CROZ, S. HAMMARLING, AND I. DUFF, *A set of level 3 basic linear algebra subprograms*, ACM Trans. Math. Software, 16 (1990), pp. 1–17.

[11] J. DONGARRA, J. DU CROZ, S. HAMMARLING, AND R. HANSON, *An extended set of Fortran basic linear algebra subprograms*, ACM Trans. Math. Software, 14 (1988), pp. 1–17.

[12] G. GOLUB AND C. VAN LOAN, *Matrix Computations*, Second Edition, The Johns Hopkins University Press, Baltimore, MD, 1989.

[13] P. LANCASTER AND M. TISMINETSKY, *The Theory of Matrices*, Second Edition, Academic Press, Orlando, FL, 1985.

[14] D. S. WATKINS AND L. ELSNER, *Convergence of algorithms of decomposition type for the eigenvalue problem*, Linear Algebra Appl., 143 (1991), pp. 19–47.

[15] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Oxford University Press, Oxford, 1965.