SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 2, pp. 301–311, April 1990

ON RUTISHAUSER'S APPROACH TO SELF-SIMILAR FLOWS *

D. S. WATKINS † AND L. ELSNER ‡

Abstract. Certain variants of the Toda flow are continuous analogues of the QR algorithm and other algorithms for calculating eigenvalues of matrices. This was a remarkable discovery of the early eighties. Until very recently contemporary researchers studying this circle of ideas have been unaware that continuous analogues of the quotient-difference and LR algorithms were already known to Rutishauser in the fifties. Rutishauser's continuous analogue of the quotient-difference algorithm contains the finite, nonperiodic Toda flow as a special case. A nice feature of Rutishauser's approach is that it leads from the (discrete) eigenvalue algorithm to the (continuous) flow by a limiting process. Thus the connection between the algorithm and the flow does not come as a surprise. In this paper it is shown how Rutishauser's approach can be generalized to yield large families of flows in a natural manner. The flows derived include continuous analogues of the LR, QR, SR, and HR algorithms.

Key words. Toda flow, self-similar flow, quotient-difference algorithm, LR algorithm, QR algorithm

AMS(MOS) subject classifications. 15A18, 15A23, 58F19, 58F25, 65F15

C.R. classification. G.1.3

1. The Toda flow and the quotient-difference algorithm. In recent years there has been considerable interest in flows that are continuous analogues of the QR algorithm and other algorithms for calculating the eigenvalues of a matrix [2], [16], [18]. The present interest dates from Toda's study [17] of a dynamical system that came to be known as the Toda lattice. This is a system of infinitely many points of unit mass constrained to lie on a line, such that each point exerts an exponential repelling force on its two nearest neighbors. If the ith point has position q_i and momentum p_i, then

$$\dot{q}_i = p_i, \qquad \dot{p}_i = \exp(q_{i-1} - q_i) - \exp(q_i - q_{i+1}). \tag{1}$$

In addition Toda applied a periodicity condition $q_{n+i} = q_i + 2\pi l$, for all i. Here n and l are fixed positive numbers, n an integer. Toda's work was published in 1970. Subsequently many workers in dynamical system theory studied the Toda flow and numerous variants and generalizations. See, for example, [3], [5], [7], [8], [15], and the works cited above. (Additional works are cited in the bibliography of [19].) We will focus on a few of these. Moser [8] studied a variant with finitely many points and no periodicity condition. This finite, nonperiodic Toda lattice satisfies (1) for $i = 1, \cdots, n$ with $q_0 = -\infty$ and $q_{n+1} = \infty$. We will restrict our attention to this version of the Toda lattice. Flaschka [3] noticed that the change of variables

$$a_i = -\tfrac{1}{2}p_i, \qquad b_i = \tfrac{1}{2}\exp\left(\tfrac{1}{2}(q_i - q_{i+1})\right)$$

leads to the system

$$\dot{a}_i = 2(b_i^2 - b_{i-1}^2), \qquad i = 1, \cdots, n,$$

* Received by the editors November 14, 1988; accepted for publication (in revised form) June 1, 1989.

† Department of Pure and Applied Mathematics, Washington State University, Pullman, Washington 99164-2930 (watkins@wsumath.bitnet).

‡ Fakultät für Mathematik, Universität Bielefeld, Postfach 8640, D-4800 Bielefeld 1, Federal Republic of Germany (umatf105@dbuni11.bitnet).

$$\dot{b}_i = b_i(a_{i+1} - a_i), \qquad i = 1, \cdots, n-1, \tag{2}$$

$$b_0 = b_n = 0,$$

which can be expressed as a matrix differential equation

$$\dot{B} = B\rho(B) - \rho(B)B, \tag{3}$$

where B and $\rho(B)$ are the tridiagonal matrices

$$B = \begin{bmatrix} a_1 & b_1 & & & \\ b_1 & a_2 & & & \\ & \ddots & \ddots & \ddots & \\ & & & a_{n-1} & b_{n-1} \\ & & & b_{n-1} & a_n \end{bmatrix}, \qquad \rho(B) = \begin{bmatrix} 0 & -b_1 & & & \\ b_1 & 0 & & & \\ & \ddots & \ddots & \ddots & \\ & & & 0 & -b_{n-1} \\ & & & b_{n-1} & 0 \end{bmatrix}.$$

Note that B is symmetric and $\rho(B)$ is skew-symmetric. Given any symmetric, tridiagonal initial matrix $\hat{B}$, let $B(t)$ be the unique solution of (3) satisfying $B(0) = \hat{B}$. Then it is not hard to show that $B(t)$ is orthogonally similar to $\hat{B}$ for all t. Hence we say that the flow is *self-similar*. It is also called *isospectral* because the eigenvalues of $B(t)$ are invariant. Since the points of the lattice repel one another, we must eventually have $q_i - q_{i+1} \to -\infty$. Thus $b_i \to 0$ for $i = 1, \cdots, n-1$, and the a_i converge to the eigenvalues of $\hat{B}$. In a paper published in 1982, Symes [16] made the remarkable observation that the finite, nonperiodic Toda flow is a continuous analogue of the QR algorithm [22] for calculating the eigenvalues of a matrix. Starting from some initial matrix A_0, the QR algorithm produces a sequence (A_k) of matrices similar to A_0. Symes showed that the unshifted QR algorithm with initial matrix $A_0 = \exp(\hat{B})$ produces the sequence $(A_k) = (\exp(B(k)))$. This observation was generalized in various directions. Deift, Nanda, and Tomei [2] considered more general flows of the form

$$\dot{B} = B\rho(f(B)) - \rho(f(B))B$$

for suitable functions f. For a fixed f, the more general flow produces $B(t)$ such that the QR algorithm with starting matrix $A_0 = \exp(f(\hat{B}))$ produces the sequence

$$(A_k) = (\exp\{f(B(k))\}).$$

In particular, the choice $f(x) = \log x$ yields a flow for which $B(0), B(1), B(2), \cdots$ is exactly the sequence produced by the unshifted QR algorithm with starting matrix $A_0 = \hat{B}$. In other words, this flow interpolates the QR algorithm. Chu [1] extended the family of flows to include nonsymmetric, nontridiagonal B. We refer to this family of flows collectively as *QR flows*. In [18] Watkins introduced a family of LR flows (called LU flows in [18]) that are related to the unshifted LR algorithm [22] in exactly the same way.

All of this work was published after 1970, and all of it was done in ignorance of earlier work of Rutishauser [11], [14]. It is well known that Rutishauser invented the LR algorithm in the fifties [13], [14]. In 1958, in one of his early papers on the subject [14], he included a section entitled "A continuous analogue to the LR transformation," in which he developed the LR analogue of the Toda flow. It turns out that Rutishauser's flow is a member of the family of LR flows introduced by Watkins [18] much later.

A pleasing feature of Rutishauser's derivation is that it proceeds from the LR algorithm to the flow in a natural way, i.e., by taking a limit. Thus the connection does not come as a surprise, as it did in the case of Symes's discovery of the connection between the Toda flow and the QR algorithm. One might well wonder what led Rutishauser to this natural approach. The answer lies in the historical roots of the LR algorithm. The LR algorithm evolved from the quotient-difference (q-d) algorithm, which was also developed by Rutishauser [9], [10], [12]. The q-d algorithm started out as a method for finding the poles of a meromorphic function. For almost all choices of $x, y \in \mathbb{C}^n$, the function $f(\lambda) = y^T(\lambda I - A)^{-1}x$ has the eigenvalues of A as its poles, so the algorithm can also be used to find the eigenvalues of a matrix. As it was originally formulated, the q-d algorithm consisted of filling out a so-called q-d table, which resembles a table of differences, except that the rules for forming a q-d table are more complicated. For details see the original work of Rutishauser or Henrici's book [6]. The zeroth column of an ordinary difference table consists of the values of a smooth function at equally spaced points. As the spacing tends to zero, the first and higher order differences tend to zero as well. However, if the table is modified so that it contains divided differences instead of simple differences, the column of kth differences will tend to the kth derivative as the spacing tends to zero. Notice that if we let $g_k(t)$ denote the limit of the kth column, then $\dot{g}_k = g_{k+1}$ for $k = 0, 1, 2, \cdots$. Thus the columns are related by a simple system of differential equations. The entries in the zeroth column of a q-d table can also be viewed as values of a certain function at equally spaced points. It is therefore quite natural to ask what happens as the spacing converges to zero. It turns out that the limit is not very interesting. Certain columns (the quotients) tend to 1, while others (the differences) tend to zero. However, we would hope to be able to modify the table in the spirit of divided differences, so that an interesting limit is obtained. This turns out to be possible, but since the formation rules for a q-d table are more complicated than for a simple difference table, the columns (of the modified table) do not converge to simple derivatives of the original function. Instead, the limit satisfies a more complicated system of differential equations:

$$\begin{aligned} \dot{Q}_i &= E_i - E_{i-1}, & i &= 1, \cdots, n, \\ \dot{E}_i &= E_i(Q_{i+1} - Q_i), & i &= 1, \cdots, n-1, \\ E_0 &= 0 = E_n. \end{aligned} \tag{4}$$

$Q_i(t)$ is the limit of the ith column of (modified) quotients and $E_i(t)$ is the limit of the ith column of (modified) differences. This continuous analogue of the q-d algorithm was published by Rutishauser [11] in 1954. The equations (4) resemble Flaschka's form (2) of the finite, nonperiodic Toda equations. In fact, the change of variables

$$Q_i = 2a_i, \qquad E_i = 4b_i^2$$

transforms (4) into (2). Thus Rutishauser published a form of the Toda flow 16 years before Toda. The system (4) is actually more general than the Toda flow, since the Toda flow corresponds to the special case $E_i > 0$, $i = 1, \cdots, n-1$.

The original formulation of the quotient-difference algorithm was found to be unstable. A better approach is to fill in the q-d table from top to bottom, rather than from left to right. Rutishauser quickly recognized that the top-to-bottom procedure could be interpreted as a process of matrix factorization and recombination, and the LR algorithm was born. The q-d algorithm is just the LR algorithm applied to a tridiagonal matrix with 1's on the superdiagonal. Once the algorithm assumed this new guise, it became easy to forget the q-d table and its infinitesimal limit. But Rutishauser did not forget. Generalizing from the q-d algorithm, he obtained a continuous analogue of the LR algorithm [14], which he published in 1958.

Given that the Toda flow is a continuous analogue of the QR algorithm, whereas Rutishauser's flow (4) is associated with the LR algorithm, it might seem surprising that (4) should include the Toda flow as a special case. Actually, this need not be such a surprise. Suppose the LR algorithm, or, equivalently, the q-d algorithm, is applied to a symmetric, positive definite, tridiagonal matrix. The symmetry is not preserved by the algorithm, but a trivial rescaling transforms the LR algorithm to the Cholesky LR algorithm [22], which does preserve symmetry. The outputs of the two algorithms differ by diagonal similarity transformations, so we can think of them as the same, at least in principle. It is well known [22] that two steps of the Cholesky LR algorithm are equivalent to one step of the (symmetric, unshifted) QR algorithm. Thus, in a sense, the q-d algorithm includes as a special case the QR algorithm for symmetric, positive definite, tridiagonal matrices. The same must be true of the continuous analogues.

In the remainder of the paper we will show how to construct flows by Rutishauser's method. Our construction will be based on Rutishauser's LR flow, not the q-d flow; the former is more general than the latter. We will present a generalization of Rutishauser's construction that produces QR, SR, HR, and other flows as well. We begin by introducing a generic eigenvalue algorithm, the FG *algorithm*. We then derive a continuous analogue, a generic FG *flow*. In §3 the construction is generalized to yield a whole family of FG flows associated with each FG algorithm. This is exactly the family of autonomous FG flows discussed in [19]. The contribution of the present paper is not to develop new flows, but to show how Rutishauser's construction can be generalized to produce known flows in a very natural manner. An additional contribution is that our development is rigorous. By contrast, Rutishauser's development was sketchy and omitted numerous details.

The approach developed here can also be used to derive families of flows associated with algorithms for the generalized eigenvalue problem $\hat{A}x = \lambda \hat{B}x$. These are exactly the autonomous FGZ flows of [20]. The same approach can also be used to derive the autonomous flows associated with the singular value decomposition discussed in [21]. The constructions are straightforward, and we omit them.

2. Construction of flows by Rutishauser's approach. In order to achieve the desired level of generality, we will make use of some notions from elementary Lie theory. The reader who would rather not learn about Lie theory at this time should skim the next two paragraphs lightly, then have a close look at Examples 2.1L and 2.1Q. The reader can then read the rest of the paper easily by substituting either QR or LR for FG.

Let $\mathbb{F} = \mathbb{R}$ or $\mathbb{C}$, and let $GL_n(\mathbb{F})$ denote the general linear group of nonsingular $n \times n$ matrices over $\mathbb{F}$. Given a closed subgroup $\mathcal{G}$ of $GL_n(\mathbb{F})$, let $\Lambda(\mathcal{G}) \subset \mathbb{F}^{n\times n}$ denote the Lie algebra associated with $\mathcal{G}$. The basic facts about Lie algebras of matrices are stated in [19]. For more complete information about Lie groups and algebras see [4], for example. The Lie algebra $\Lambda(\mathcal{G})$ is most easily viewed as the tangent space of the manifold $\mathcal{G}$ at the identity element. Thus it can be thought of as a subspace of $\mathbb{F}^{n\times n}$. Let $\mathcal{F}$ and $\mathcal{G}$ be two closed subgroups of $GL_n(\mathbb{F})$ such that

$$\Lambda(\mathcal{F}) \oplus \Lambda(\mathcal{G}) = \mathbb{F}^{n\times n}, \tag{5}$$

and $\Lambda(\mathcal{G})$ contains the identity matrix. This last assumption implies that $\Lambda(\mathcal{G})$ contains the Lie algebra of all real multiples of I, which is equivalent to the condition that $\mathcal{G}$ contains the Lie group of all positive multiples of I. We could equally well require that $\Lambda(\mathcal{F})$, rather than $\Lambda(\mathcal{G})$, contain the identity matrix. However, as we shall later see, neither of these assumptions is really necessary. The assumption (5) means that every $X \in \mathbb{F}^{n\times n}$ can be decomposed in exactly one way as

$$X = \rho(X) + \sigma(X), \qquad \rho(X) \in \Lambda(\mathcal{F}),\ \sigma(X) \in \Lambda(\mathcal{G}). \tag{6}$$

This equation defines linear transformations ρ and σ, which are complementary projectors of $\mathbb{F}^{n\times n}$ onto $\Lambda(\mathcal{F})$ and $\Lambda(\mathcal{G})$, respectively. The existence of the additive decomposition (5) implies the existence of a multiplicative decomposition: There is a neighborhood $\mathcal{V}$ of I in $GL_n(\mathbb{F})$ such that every $A \in \mathcal{V}$ has a unique *FG decomposition*; that is, A can be expressed uniquely as a product $A = FG$, where $F \in \mathcal{F}$ and $G \in \mathcal{G}$ [4], [19].

Let $A(\epsilon)$ be an analytic function of ϵ with $A(0) = I$. Then for sufficiently small ϵ, $A(\epsilon)$ has an FG decomposition $A(\epsilon) = F(\epsilon)G(\epsilon)$, and the factors $F(\epsilon) \in \mathcal{F}$ and $G(\epsilon) \in \mathcal{G}$ are also analytic functions satisfying $F(0) = G(0) = I$. Expanding each in a Taylor series we have

$$F(\epsilon) = I + \epsilon X + \epsilon^2 M + O(\epsilon^3), \qquad G(\epsilon) = I + \epsilon Y + \epsilon^2 N + O(\epsilon^3),$$

where $X = F'(0) \in \Lambda(\mathcal{F})$ and $Y = G'(0) \in \Lambda(\mathcal{G})$. We will need to use expansions of this type to derive the FG flows.

Example 2.1L. Rutishauser considered the special case in which the FG decomposition is the LR decomposition. In this case $\mathcal{F}$ is the group of unit lower triangular matrices, and $\mathcal{G}$ is the group of nonsingular upper triangular matrices. ($\mathbb{F}$ can be either $\mathbb{R}$ or $\mathbb{C}$.) Thus $\Lambda(\mathcal{F})$ and $\Lambda(\mathcal{G})$ are the Lie algebras of strictly lower triangular and upper triangular matrices, respectively. Clearly (5) holds, and $\Lambda(\mathcal{G})$ contains I. Given $X \in \mathbb{F}^{n\times n}$, we obtain $\sigma(X)$ by setting the lower triangular entries of X to zero. Then $\rho(X) = X - \sigma(X)$.

Example 2.1Q. Let $\mathbb{F} = \mathbb{C}$. If we take $\mathcal{F}$ to be the unitary group and $\mathcal{G}$ the group of upper triangular matrices with real, positive, main diagonal entries, then the FG decomposition is just the QR decomposition. The Lie algebras $\Lambda(\mathcal{F})$ and $\Lambda(\mathcal{G})$ are just the skew-Hermitian matrices and the upper triangular matrices with real main diagonal entries, respectively. Obviously $\Lambda(\mathcal{G})$ contains I. It is easy to show that (5) holds. Every $X \in \mathbb{F}^{n\times n}$ can be expressed uniquely as a sum $X = L + D_r + D_i + U$, where L is strictly lower triangular, D_r is diagonal and real, D_i is diagonal and imaginary, and U is strictly upper triangular. We have $\rho(X) = L + D_i - L^*$ and $\sigma(X) = D_r + U + L^*$. There is also a real QR decomposition, which we obtain by taking $\mathcal{F}$ to be the group of real, orthogonal matrices and $\mathcal{G}$ the group of real, upper triangular matrices with positive entries on the main diagonal.

Two other examples, the SR and HR decompositions, are discussed in [19].

Associated with each FG decomposition is an *FG algorithm* for calculating eigenvalues of matrices. The *shifted FG* algorithm associated with $\mathcal{F}$ and $\mathcal{G}$ begins with a matrix $\hat{B} \in GL_n(\mathbb{F})$ and produces a sequence (B_k) by setting $B_0 = \hat{B}$, and then defining B_k, for $k = 1, 2, 3, \cdots$, by the equations

$$B_{k-1} - \sigma_k I = \bar{F}_k \bar{G}_k, \qquad \bar{G}_k \bar{F}_k + \sigma_k I = B_k, \tag{7}$$

where $\bar{F}_k \in \mathcal{F}$, $\bar{G}_k \in \mathcal{G}$, and the shift σ_k is chosen so that $B_{k-1} - \sigma_k I$ has an FG decomposition. The meaning of (7) is that a shift is subtracted from B_{k-1}, an FG

decomposition of the shifted matrix is performed, the factors of the decomposition are multiplied back together in reverse order, then the shift is added back on, giving B_k. It is easy to show that the B_k so produced are all similar to $\hat{B}$, so they have the same eigenvalues. Under certain conditions on $\hat{B}$, $\mathcal{F}$, $\mathcal{G}$, and (σ_k), the sequence (B_k) can be shown to converge to triangular or quasi-triangular form, yielding the eigenvalues of $\hat{B}$. The shifts are generally chosen with an eye to accelerating convergence. Rutishauser used shifts for a different purpose, namely, to pass to a continuous limit. Following Rutishauser we consider a constant shift $\sigma_k = -\mu$, where μ is positive and large. (We plan to take a limit in which $\mu \to \infty$.) Then the sequence (B_k) is generated by

$$B_{k-1} + \mu I = \bar{F}_k(\mu \bar{G}_k), \qquad (\mu \bar{G}_k)\bar{F}_k - \mu I = B_k. \tag{8}$$

We have factored the scalar μ out of $\bar{G}_k$ for convenience. Because of the assumption that $\mathcal{G}$ contains all positive multiples of the identity matrix, we have $\bar{G}_k \in \mathcal{G}$ if and only if $\mu \bar{G}_k \in \mathcal{G}$.

Notice that this choice of shifts actually slows convergence. This is so because the rate of convergence (when it occurs at all) is determined, at least in part, by ratios of eigenvalues of $\hat{B} + \mu I$. As μ is made larger, the ratios of the eigenvalues approach one, indicating progressively slower convergence.

It is easy to show that

$$B_k = \bar{F}_k^{-1} B_{k-1} \bar{F}_k = \bar{G}_k B_{k-1} \bar{G}_k^{-1}.$$

Letting

$$F_k = \bar{F}_1 \bar{F}_2 \cdots \bar{F}_k, \qquad G_k = \bar{G}_k \cdots \bar{G}_2 \bar{G}_1,$$

we have

$$B_k = F_k^{-1} \hat{B} F_k = G_k \hat{B} G_k^{-1}. \tag{9}$$

We can also show easily by induction that

$$(\hat{B} + \mu I)^k = \mu^k F_k G_k. \tag{10}$$

We prefer to work with a small parameter rather than the large parameter μ, so let $\epsilon = 1/\mu$. Then (8) and (10) can be rewritten as

$$I + \epsilon B_{k-1} = \bar{F}_k \bar{G}_k, \qquad \bar{G}_k \bar{F}_k = I + \epsilon B_k. \tag{11}$$

$$(I + \epsilon \hat{B})^k = F_k G_k. \tag{12}$$

We used the assumption that $\Lambda(\mathcal{G})$ contains the identity matrix to write the shifted FG algorithm in the form (8), which we then rewrote in the equivalent form (11). This assumption will not be used anywhere else. If we use (11) as our point of departure instead of the shifted FG algorithm, we can drop the assumption.

It is useful to view the FG algorithm (11) as a discrete-time dynamical system governed by the difference equation

$$B_k = B_{k-1} + \frac{1}{\epsilon}(\bar{G}_k \bar{F}_k - \bar{F}_k \bar{G}_k). \tag{13}$$

We will view each step forward as a time step of length ϵ. Thus the elapsed time after k steps is $k\epsilon$. If we let $\epsilon \to 0$ and $k \to \infty$, holding $t = k\epsilon$ fixed, the difference equation (13) is transformed into a differential equation, a continuous analogue of the FG algorithm.

In order to carry out the limiting process rigorously, we need to know that certain limits exist. The matrices B_k, F_k, G_k, $\bar{F}_k$, and $\bar{G}_k$ are all functions of ϵ as well as k, and we will write $B_k = B(k,\epsilon)$, for example, when we want to emphasize this fact. From (12) it is clear that $F(k,\epsilon)$ and $G(k,\epsilon)$ are well defined for all complex k and sufficiently small complex ϵ, and they are analytic in both variables. From (9) and (11) we see that the same is true of $B(k,\epsilon)$, $\bar{F}(k,\epsilon)$, and $\bar{G}(k,\epsilon)$ as well. Since we intend to hold $t = k\epsilon$ fixed as we pass to the limit, it is useful to write $F(k,\epsilon) = F(t/\epsilon,\epsilon)$, for example. With this notation we can rewrite (12) as

$$(I + \tfrac{\epsilon}{t}(t\hat{B}))^{t/\epsilon} = F(t/\epsilon,\epsilon)G(t/\epsilon,\epsilon). \tag{14}$$

The limit of the left-hand side as $\epsilon \to 0$ is $\exp(t\hat{B})$. Suppose $\exp(t\hat{B})$ has an FG decomposition. (This will certainly be the case if t is sufficiently small.) Define $F(t) \in \mathcal{F}$ and $G(t) \in \mathcal{G}$ to be the FG factors of $\exp(t\hat{B})$; that is,

$$\exp(t\hat{B}) = F(t)G(t). \tag{15}$$

Since the FG decomposition is analytic, it is certainly continuous. Thus (14) and (15) imply that

$$\lim_{\epsilon\to 0} F(t/\epsilon,\epsilon) = F(t), \qquad \lim_{\epsilon\to 0} G(t/\epsilon,\epsilon) = G(t).$$

For fixed t the left-hand side of (14) is an analytic function of ϵ in a deleted neighborhood of zero, with a removable singularity at $\epsilon = 0$. Therefore $F(t/\epsilon,\epsilon)$ and $G(t/\epsilon,\epsilon)$ are also analytic functions of ϵ with removable singularities at zero, provided $\exp(t\hat{B})$ has an FG decomposition. Define another analytic function $B(t)$ by

$$B(t) = F(t)^{-1}\hat{B}F(t) = G(t)\hat{B}G(t)^{-1}. \tag{16}$$

The equations (9) can be rewritten as

$$B(t/\epsilon,\epsilon) = F(t/\epsilon,\epsilon)^{-1}\hat{B}F(t/\epsilon,\epsilon) = G(t/\epsilon,\epsilon)\hat{B}G(t/\epsilon,\epsilon)^{-1}.$$

Therefore $B(t/\epsilon,\epsilon)$ is an analytic function of ϵ in a neighborhood of zero. Taking the limit as $\epsilon \to 0$, we find that

$$\lim_{\epsilon\to 0} B(t/\epsilon,\epsilon) = B(t).$$

The function $B(t)$ is, in fact, our continuous analogue of the sequence (B_k).

We now have in hand the tools to prove the following interpolation result: Let (A_k) be the output of the FG algorithm with zero shifts, starting with $A_0 = \hat{A} = \exp(\hat{B})$. Then

$$A_k = \exp(B(k)), \qquad k = 0,1,2,3,\cdots.$$

The main tools for proving this are (15) and its discrete analogue $\hat{A}^k = F_kG_k$, which holds for the unshifted FG algorithm. See [19] for a proof. Rutishauser stated the

LR case of (15), but he did not arrive at it in the same manner as we have here. He may have been unaware of the interpolation result, as he did not mention it in [14].

We will now derive the continuous analogue of the FG algorithm, i.e., the differential equation that $B(t)$ satisfies. The usual approach is just to differentiate (15) and (16). This yields differential equations for $F(t)$ and $G(t)$, as well as $B(t)$. Now let us see how Rutishauser obtained them by passing to a limit. For this we need Taylor expansions of the quantities $\bar{F}_k = \bar{F}(t/\epsilon, \epsilon)$ and $\bar{G}_k = \bar{G}(t/\epsilon, \epsilon)$, which appear in (13). The first equation in (11) can be written as

$$I + \epsilon B((t-\epsilon)/\epsilon, \epsilon) = \bar{F}(t/\epsilon, \epsilon)\bar{G}(t/\epsilon, \epsilon). \tag{17}$$

Letting $A(\epsilon) = I + \epsilon B((t-\epsilon)/\epsilon, \epsilon)$ we see that $A(\epsilon)$ is analytic, and $\lim_{\epsilon \to 0} A(\epsilon) = I$. Thus $\bar{F}(t/\epsilon, \epsilon)$ and $\bar{G}(t/\epsilon, \epsilon)$ have Taylor expansions

$$\begin{aligned} \bar{F}(t/\epsilon, \epsilon) &= I + \epsilon X(t) + \epsilon^2 M(t) + O(\epsilon^3), \\ \bar{G}(t/\epsilon, \epsilon) &= I + \epsilon Y(t) + \epsilon^2 N(t) + O(\epsilon^3), \end{aligned} \tag{18}$$

where $X(t) \in \Lambda(\mathcal{F})$ and $Y(t) \in \Lambda(\mathcal{G})$. Substituting the expansions (18) into (17), we find that

$$B((t-\epsilon)/\epsilon, \epsilon) = X(t) + Y(t) + O(\epsilon).$$

Letting $\epsilon \to 0$, we obtain

$$B(t) = X(t) + Y(t).$$

Since $X(t) \in \Lambda(\mathcal{F})$ and $Y(t) \in \Lambda(\mathcal{G})$, it follows that

$$X(t) = \rho(B(t)) \quad \text{and} \quad Y(t) = \sigma(B(t)),$$

where ρ and σ are defined by (6). We are finally ready to pass to the limit. Following Rutishauser we substitute the expansions (18) into (13), which can then be rewritten as

$$\frac{B(t/\epsilon, \epsilon) - B((t-\epsilon)/\epsilon, \epsilon)}{\epsilon} = [Y(t), X(t)] + O(\epsilon),$$

where $[Y, X] = YX - XY$. Taking the limit as $\epsilon \to 0$, we obtain

$$\dot{B}(t) = [\sigma(B(t)), \rho(B(t))]. \tag{19}$$

This is our continuous analogue of the FG algorithm. Since $[\sigma(B), \rho(B)] = [B, \rho(B)] = [\sigma(B), B]$, (19) also has the forms

$$\dot{B} = [B, \rho(B)] \quad \text{and} \quad \dot{B} = [\sigma(B), B]. \tag{20}$$

This shows that this flow is a member of the family of FG flows introduced in [19].

The differential equations for F and G are also easily obtained.

$$\frac{F_k - F_{k-1}}{\epsilon} = F_{k-1}\frac{(\bar{F}_k - I)}{\epsilon} = F_{k-1}\{\rho(B(t)) + O(\epsilon)\}.$$

Taking the limit, we have

$$\dot{F} = F\rho(B) = F\rho(F^{-1}\hat{B}F). \tag{21}$$

Similarly,

$$\dot{G} = \sigma(B)G = \sigma(G\hat{B}G^{-1})G. \tag{22}$$

These equations are familiar from [19]. They were also stated by Rutishauser [14] for the LR case.

A second way to obtain the differential equation for $B(t)$ is to use the equation

$$B_k = \bar{F}_k^{-1} B_{k-1} \bar{F}_k. \tag{23}$$

From the first expansion in (18) it is obvious that

$$\bar{F}_k^{-1} = \bar{F}(t/\epsilon, \epsilon)^{-1} = I - \epsilon X(t) + O(\epsilon^2).$$

Substituting this expansion and the first expansion of (18) into (23), we find that

$$B_k = B_{k-1} + \epsilon[B_{k-1}, X(t)] + O(\epsilon^2). \tag{24}$$

Thus

$$\frac{B(t/\epsilon,\epsilon) - B((t-\epsilon)/\epsilon, \epsilon)}{\epsilon} = [B((t-\epsilon)/\epsilon, \epsilon), X(t)] + O(\epsilon). \tag{25}$$

Taking the limit as $\epsilon \to 0$ we obtain

$$\dot{B}(t) = [B(t), \rho(B(t))],$$

the first equation of (20). We could equally well have started with the equation

$$B_k = \bar{G}_k B_{k-1} \bar{G}_k^{-1}. \tag{26}$$

This gives

$$B_k = B_{k-1} + \epsilon[Y(t), B_{k-1}] + O(\epsilon^2),$$

which leads to $\dot{B} = [\sigma(B), B]$, the second equation of (20). The nicest feature of this approach is that it can be generalized. We will carry out the generalization in the next section.

In order to carry out the construction, we have had to assume that t is such that $\exp(t\hat{B})$ has an FG decomposition. We have already shown in [19] that the points at which $\exp(t\hat{B})$ does not have an FG decomposition are exactly the points at which the flow has singularities.

3. Carrying the generalization further. So far we have derived FG flows of the form $\dot{B} = [B, \rho(B)]$. This is a special case of a more general family of autonomous FG flows of the form $\dot{B} = [B, \rho(f(B))]$, which we studied in [19]. Here f is any locally analytic function defined on the spectrum of $\hat{B}$. In the present section we will show how to derive this entire family of flows by taking limits.

We will make use of the following generalization of the FG algorithm. Instead of choosing a sequence of shifts (σ_k), we choose a sequence (p_k) of analytic functions defined on the spectrum of $\hat{B}$. Then, starting with $B_0 = \hat{B}$, we define, for $k = 1, 2, 3, \cdots$

$$\begin{cases} B_k = \bar{F}_k^{-1} B_{k-1} \bar{F}_k = \bar{G}_k B_{k-1} \bar{G}_k^{-1}, \\ \text{where} \quad p_k(B_{k-1}) = \bar{F}_k \bar{G}_k, \quad \bar{F}_k \in \mathcal{F}, \ \bar{G}_k \in \mathcal{G}. \end{cases} \tag{27}$$

If we choose $p_k(x) = x - \sigma_k$, (27) reduces to the shifted FG algorithm introduced in the previous section. The choice $p_k(x) = (x - \sigma_k)(x - \tau_k)$ gives the *double-step* FG algorithm. In actual implementations the p_k would be chosen with the intent of accelerating convergence, but for our purposes we will choose $p_k(x) = 1 + \epsilon f(x)$, $k = 1, 2, 3, \cdots$, where f is a fixed analytic function defined on the spectrum of $\hat{B}$. Then

(28) $$I + \epsilon f(B_{k-1}) = \bar{F}_k \bar{G}_k.$$

Defining $F_k = \bar{F}_1 \cdots \bar{F}_k$ and $G_k = \bar{G}_k \cdots \bar{G}_1$, we have

$$B_k = F_k^{-1} \hat{B} F_k = G_k \hat{B} G_k^{-1}$$

and

(29) $$(I + \epsilon f(\hat{B}))^k = F_k G_k.$$

In the case $f(x) = x$, (28) and (29) reduce to (11, first equation) and (12), respectively. Letting $t = k\epsilon$ and using the same notational conventions as before, we can rewrite (29) as

$$(I + \tfrac{\epsilon}{t}(tf(\hat{B})))^{t/\epsilon} = F(t/\epsilon, \epsilon) G(t/\epsilon, \epsilon),$$

which is analogous to (14). Obviously the limit of the left-hand side as $\epsilon \to 0$ is $\exp(tf(\hat{B}))$. The entire development of the previous section can be generalized in a straightforward manner. Now $F(t)$ and $G(t)$ are defined by the FG decomposition

$$\exp(tf(\hat{B})) = F(t)G(t).$$

The Taylor expansions

$$\bar{F}(t/\epsilon, \epsilon) = I + \epsilon X(t) + O(\epsilon^2),$$

$$\bar{G}(t/\epsilon, \epsilon) = I + \epsilon Y(t) + O(\epsilon^2)$$

continue to be valid, but now

$$X(t) + Y(t) = f(B(t)),$$

so

$$X(t) = \rho(f(B(t))) \quad \text{and} \quad Y(t) = \sigma(f(B(t))).$$

Equations (23), (24), and (25) continue to hold, except that now $X(t) = \rho(f(B(t)))$. Taking the limit as $\epsilon \to 0$ in (25), we obtain

$$\dot{B} = [B, \rho(f(B))],$$

as desired. Alternatively we can start from (26) and obtain the form $\dot{B} = [\sigma(f(B)), B]$. Finally, in analogy with (21) and (22) we find that

$$\dot{F} = F\rho(F^{-1} f(\hat{B}) F), \qquad \dot{G} = \sigma(G f(\hat{B}) G^{-1}) G.$$

This flow has the interpolation property $\exp\{f(B(k))\} = A_k$, where (A_k) is the output of the FG algorithm with zero shifts, starting with $A_0 = \exp\{f(\hat{B})\}$. In particular, the choice $f(x) = \log x$ yields a flow that interpolates the FG algorithm.

REFERENCES

[1] M. Chu, *The generalized Toda flow, the QR algorithm, and the centre manifold theory*, SIAM J. Algebraic Discrete Methods, 5 (1984), pp. 187–201.

[2] P. Deift, T. Nanda, and C. Tomei, *Differential equations for the symmetric eigenvalue problem*, SIAM J. Numer. Anal., 20 (1983), pp. 1–22.

[3] H. Flaschka, *The Toda lattice, II, existence of integrals*, Phys. Rev. B, 9 (1974), pp. 1924–1925.

[4] S. Helgason, *Differential Geometry, Lie Groups and Symmetric Spaces*, Academic Press, New York, 1978.

[5] M. Hénon, *Integrals of the Toda lattice*, Phys. Rev. B, 9 (1974), pp. 1921–1923.

[6] P. Henrici, *Applied and Computational Complex Analysis, Vol.* I, John Wiley, New York, 1974.

[7] J. Moser, *Dynamical Systems, Theory and Applications*, Springer-Verlag, Berlin, New York, 1975.

[8] ———, *Finitely many mass points on the line under the influence of an exponential potential—An integrable system*, in Dynamical Systems, Theory and Applications, Springer-Verlag, Berlin, New York, 1975, pp. 467–497 in [7].

[9] H. Rutishauser, *Der Quotienten-Differenzen-Algorithmus*, Z. Angew. Math. Phys., 5 (1954), pp. 233–251.

[10] ———, *Anwendungen des Quotienten-Differenzen-Algorithmus*, Z. Angew. Math. Phys., 5 (1954), pp. 496–508.

[11] ———, *Ein infinitesimales Analogon zum Quotienten-Differenzen-Algorithmus*, Arch. Math., 5 (1954), pp. 132–137.

[12] ———, *Der Quotienten-Differenzen-Algorithmus*, Mitt. Inst. Angew. Math., No. 7, ETH, Zürich, 1957. MR 19–686. (This report gathers the material of [9], [10], [11], and parts of [14] into a single volume.)

[13] ———, *Une méthode pour la détermination des valeurs propres d'une matrice*, Comptes Rendus Acad. Sci. Paris, 240 (1955), pp. 34–36.

[14] ———, *Solution of eigenvalue problems with the LR-transformation*, National Bureau of Standards Applied Mathematics Series, 49 (1958), pp. 47–81.

[15] M. Shub and A. T. Vasquez, *Some linearly induced Morse-Smale systems, the QR algorithm and the Toda lattice*, Contemp. Math., 64 (1987), pp. 181–194.

[16] W. W. Symes, *The QR algorithm and scattering for the finite nonperiodic Toda lattice*, Physica, 4D (1982), pp. 275–280.

[17] M. Toda, *Waves in nonlinear lattice*, Prog. Theoret. Phys. (Supp.), 45 (1970), pp. 174–200.

[18] D. S. Watkins, *Isospectral flows*, SIAM Rev., 26 (1984), pp. 379–391.

[19] D. S. Watkins and L. Elsner, *Self-similar flows*, Linear Algebra Appl., 110 (1988), pp. 213–242.

[20] ———, *Self-equivalent flows associated with the generalized eigenvalue problem*, Linear Algebra Appl., 118 (1989), pp. 107–127.

[21] ———, *Self-equivalent flows associated with the singular value decomposition*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 244–258.

[22] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.