The Bidiagonal Singular Value Decomposition and Hamiltonian Mechanics

Percy Deift
James Demmel[1]
Courant Institute
New York, NY 10012

Chau Li
Mathematics Department
The Pennsylvania State University
University Park, PA 16802

Carlos Tomei
PUC
Rio de Janeiro, Brazil
(visiting Yale University)

Abstract

We consider computing the singular value decomposition of a bidiagonal matrix B. This problem arises in the singular value decomposition of a general matrix, and in the eigenproblem for a symmetric positive definite tridiagonal matrix. We show that if the entries of B are known with high relative accuracy, the singular values and singular vectors of B will be determined to much higher accuracy than the standard perturbation theory suggests. We also show that the algorithm in [Demmel and Kahan] computes the singular vectors as well as the singular values to this accuracy. We also give a Hamiltonian interpretation of the algorithm and use differential equation methods to prove many of the basic facts. The Hamiltonian approach suggests a way to use flows to predict the accumulation of error in other eigenvalue algorithms as well.

(This paper appeared in the SIAM J. Numer. Anal., v. 18, n. 5, pp. 1463-1516, 1991)

1 Introduction

The singular value decomposition (SVD) of a real general n by n matrix A is the factorization $A = U\Sigma V^T$, where U and V are orthogonal, $\Sigma = \text{diag}\ \{\sigma_1, \ldots, \sigma_n\}$, and $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$. The σ_i's are the *singular values* of A, the columns v_i of V the *right singular vectors* of A, and the columns u_i of U the *left singular vectors* of A.

In this paper we will consider the SVD of a bidiagonal matrix B

$$B = \begin{bmatrix} a_1 & b_1 & & \\ & \ddots & \ddots & \\ & & \ddots & b_{n-1} \\ & & & a_n \end{bmatrix}, \tag{1.1}$$

where we may assume without loss of generality that the a_i and b_i are positive. (Recall that this assumption implies that all the σ_i are positive and distinct [Par80].) This problem arises as the final stage of the SVD of a general matrix A [GK65, GVL83], as well as in computing the eigendecomposition of a symmetric positive-definite tridiagonal matrix T [BD88]. Both computations arise frequently in a wide variety of applications. Our goal in this paper is threefold: to show that the SVD of a bidiagonal B can be computed much more accurately than the SVD of a general matrix A, to explain this with the aid of a Hamiltonian differential equation underlying the SVD algorithm used for B, and to suggest using similar differential equations to find high accuracy algorithms for other eigenvalue and singular value problems.

How accurately can the SVD of a general matrix A be computed? To answer this question, we must consider both the effects of uncertainty in the initial data A, as well as errors introduced by an algorithm (roundoff errors and the effects of the stopping criterion); a good algorithm introduces errors no worse than inherent errors caused by uncertainty in the data. The standard approach is to bound the uncertainty δA in the initial data by its two-norm $\|\delta A\|$: we say that δA is an *absolute error of scale* η *in* A if $\|\delta A\| \,/\, \|A\| \leq \eta$. With this definition of η we have:

(A) - The singular values σ_i of A and σ_i' of $A + \delta A$ can differ by at most $\eta\,\|A\|$ for all i [GVL83]:

$$|\sigma_i \Leftrightarrow \sigma_i'| \leq \eta\,\|A\| \tag{1.2}$$

(B) - Let u_i and u_i' be the i-th left unit singular vectors of A and $A + \delta A$ respectively, and v_i and v_i' be the right unit singular vectors. Let the *absolute gap* for σ_i be defined as the distance from σ_i to the nearest different singular value, normalized by σ_1:

$$absgap_i \equiv min_{j \neq i} |\sigma_i \Leftrightarrow \sigma_j| / \sigma_1 \tag{1.3}$$

If $\eta < absgap_i/2$, then the sines of the angles between u_i and u_i' ($\sin\theta(u_i, u_i')$), and between v_i and v_i' ($\sin\theta(v_i, v_i')$), are bounded as follows[DK70]:

$$\max(\sin\theta(u_i, u_i'), \sin\theta(v_i, v_i')) \leq \frac{\eta}{absgap_i \Leftrightarrow \eta} \tag{1.4}$$

The reason for limiting $\eta < absgap_i/2$ is that for larger perturbations the bound (1.4) is vacuous. This reflects the fact that larger perturbations by a general matrix can cause σ_i to merge with another singular value, so that the singular vectors fail to be uniquely defined.

Now we consider the errors introduced by the standard algorithm [GK65] for the SVD of a general matrix. We assume a standard model of floating point arithmetic, with relative errors of size at most ε, the machine precision, in each basic operation, and assume neither overflow nor underflow occur. Under these assumptions it is well known [GVL83] that the error bound of the standard algorithm is equivalent to an uncertainty δA in the initial data with absolute error scale $\|\delta A\| \,/\, \|A\| \le p(n)\varepsilon$, where $p(n)$ depends on details of the arithmetic and algorithm, and is a modestly growing function of the dimension n of A (a cubic polynomial in n with modest coefficients). In other words, the error introduced by the algorithm is no worse than $p(n)\varepsilon \, \|A\| = p(n)\varepsilon\sigma_1$ in the singular values and no worse than $p(n)\varepsilon / absgap_i$ in the singular vectors.

In particular, suppose A has one or more tiny singular values σ_j, where by tiny we mean $\sigma_j \ll \sigma_1 = \|A\|$. The error bound $p(n)\varepsilon\sigma_1$ for the singular values implies that while large singular values may be computed with high relative accuracy, tiny ones will in general not be, since the error bound may greatly exceed their value. Also, if there are two or more tiny singular values, their absolute gaps will necessarily be small compared to σ_1, and their singular vectors correspondingly uncertain. Indeed, these uncertainties are unavoidable as long as one considers general perturbations δA bounded only in norm, because the bounds in (**A**) and (**B**) above are attainable [Wil65].

Now we return to the case of a bidiagonal matrix B. It turns out that the effects of both initial data uncertainties and roundoff errors are significantly smaller than for general matrices, and are controlled by the *relative error* of δB

$$\eta_r \equiv (2n \Leftrightarrow 1) \cdot \max_{ij} |\log \frac{B_{ij} + \delta B_{ij}}{B_{ij}}| \tag{1.5}$$

instead of by its norm $\|\delta B\|$. When $\eta_r \ll 1$, this means that the sum of the componentwise relative errors $\sum_{ij} |\frac{\delta B_{ij}}{B_{ij}}|$ is approximately bounded by η_r. In other words, the zero entries of B must remain zero, and we only permit relative perturbations in the remaining entries, rather than norm bounded perturbations. Using η_r as measure of uncertainty in B, one can prove:

(**A′**) - Let the σ_i be the singular values of B and σ_i' be the singular values of $B + \delta B$. Then

$$e^{-\eta_r} \le \frac{\sigma_i'}{\sigma_i} \le e^{\eta_r} \tag{1.6}$$

This bound holds for all $\eta_r \ge 0$, just as (1.2) holds for all $\eta \ge 0$. When $\eta_r \ll 1$, these upper and lower bounds on $\frac{\sigma_i'}{\sigma_i}$ are approximately $1 \pm \eta_r$, meaning that small relative perturbations in the entries of B only cause small relative perturbations in the σ_i, independent of their magnitudes.

(B′) - For simplicity of notation write $e^{\eta_r} = 1 + \eta_r'$; when $\eta_r \ll 1$, $\eta_r \approx \eta_r'$. Let u_i and v_i be the singular vectors of B, and let u_i' and v_i' be the singular vectors of $B + \delta B$. Let the *relative gap* for σ_i be defined as the *relative* distance from σ_i to the nearest different singular value:

$$relgap_i \equiv min_{j \neq i} |\sigma_i \Leftrightarrow \sigma_j| / |\sigma_i + \sigma_j| \tag{1.7}$$

If $\eta_r' < relgap_i$, then the sines of the angles between u_i and u_i', and between v_i and v_i', are bounded by

$$\max(\sin\theta(u_i, u_i'), \sin\theta(v_i, v_i') \leq \frac{2^{1/2}\eta_r'(1 + \eta_r')}{relgap_i \Leftrightarrow \eta_r'} \tag{1.8}$$

Result (**A′**) was proven in [Kah68, DK88, BD88]; we include a (new) proof for the convenience of the reader. Result (**B′**) is proved here for the first time; it was conjectured in [DK88] with a proof of a weaker result in [BD88].

One can easily see that results (**A′**) and (**B′**) are always at least about as strong as their counterparts (**A**) and (**B**). To show how much stronger they may be, consider making relative perturbations of size 10^{-10} in a 3 by 3 bidiagonal matrix with singular values $\sigma_1 = 1$, $\sigma_2 = 2 \cdot 10^{-20}$, and $\sigma_3 = 10^{-20}$. Note that $absgap_3 = absgap_2 = 10^{-20}$, and that $relgap_3 = relgap_2 = 1/3$. Since the norm of this perturbation is about 10^{-10}, we may apply (**A**) and (**B**) to get the absolute error bound $10^{-10} \gg \sigma_3$ for σ_3, and, since $10^{-10} \gg absgap_3$, no error bound for the singular vectors at all. Applying (**A′**), we get a relative error bound of about $5 \cdot 10^{-10}$ in σ_3. Thus, we have at least 9 decimal digits of accuracy in σ_3, whereas (**A**) predicts changes 10^{20} times larger. Applying (**B′**), we get an error bound of about $2.1 \cdot 10^{-9}$ in the direction of the singular vectors, whereas (**B**) provides no error bound at all. The same results hold for σ_2 and its singular vectors.

In summary, *absolute* uncertainties in the entries of a general matrix A yield *absolute* error bounds on its singular values, and error bounds depending on the *absolute gap* for its singular vectors. In contrast, *relative* uncertainties in the entries of a bidiagonal matrix B yield *relative* error bounds on its singular values, and error bounds depending on the *relative gap* for its singular vectors.

Given the much greater accuracy to which singular values and singular vectors of bidiagonal matrices are determined by the data, it is desirable to have an algorithm which computes them to their inherent accuracy. In [DK88] such an algorithm was provided for computing the singular values to high relative accuracy. This new algorithm is a hybrid of the standard, shifted QR algorithm in [GK65, BDMS79], and a new, stable implementation of QR with a zero shift. It was also demonstrated empirically that this new algorithm was about as fast, and often much faster, than the standard algorithm [BDMS79], which can only provide absolute error bounds on the singular values.

In this paper we will prove that the algorithm in [DK88] also computes the singular vectors of bidiagonal B with an error bound depending on the relative gap as in (**B′**). More precisely, it will compute singular vectors u_i and v_i with an error bound $q(n)\varepsilon / relgap_i$, where $q(n)$ is a modest function of the dimension n of B, and ε is the machine precision as above. Thus, this algorithm computes all features of the SVD of a bidiagonal matrix to their inherent uncertainties. (Actually, we will need to change one line in the algorithm of [DK88], but this change will effect none of the numerical or timing results reported in [DK88]. We will discuss this change further in sections 3 and 6.)

If the bidiagonal matrix B is the result of reducing a dense matrix A to bidiagonal form, then in general the entries of B will have uncertainties of order $\varepsilon\|A\|$, thus limiting the accuracy of the overall computation to bounds (**A**) and (**B**). A high accuracy SVD algorithm for dense matrices can be found in [DV90]. On situation where the reduction to bidiagonal form is accurate is the symmetric positive definite eigenvalue problem [BD88], where $B^TB = A$ is the Cholesky decomposition of A, so the eigenvalues of A are the squares of the singular values of B.

The proof that the algorithm computes singular vectors as accurately as claimed has two new parts: bounding the error due to the stopping criterion, and bounding the error due to roundoff in the zero-shift QR iteration. The stopping criterion bounds are similar to the bounds in ($\mathbf{A'}$) and ($\mathbf{B'}$), and are obtained via a specialized perturbation argument in which the SVD problem for a bidiagonal matrix is converted into an eigenvalue problem for a tridiagonal matrix with zero diagonal, following [GVL83] (see section 4). On the other hand, bounds for the roundoff errors due to repeated iterations of the algorithm are conveniently analyzed in terms of the long time behavior of a Hamiltonian differential equation on the space of matrices naturally associated with the algorithm ([DLT89, Sym82, Chu86]).

To proceed, we need to introduce some notation. QR iteration with a zero shift applied to a general invertible matrix A_0 produces a sequence of orthogonally similar matrices A_i as follows. Given A_i, compute its QR decomposition $A_i = QR$, where Q is orthogonal and R is upper triangular with positive diagonal. Then $A_{i+1} = RQ = Q^TA_iQ$. It is well known that if A_0 has eigenvalues with distinct moduli, then A_i converges to a triangular matrix with the eigenvalues on the diagonal as $i \rightarrow \infty$. This algorithm may be applied to the bidiagonal singular value problem as follows [GVL83]. Let B_0 be our initial bidiagonal matrix. Given B_i, compute the QR decompositions $B_iB_i^T = Q_1R_1$ and $B_i^TB_i = Q_2R_2$. Then let $B_{i+1} = Q_1^TB_iQ_2$. Then B_i is bidiagonal for all i and converges as $i \rightarrow \infty$ to a diagonal matrix with the singular values on the diagonal. Observe that $B_{i+1}B_{i+1}^T = R_1Q_1$ and $B_{i+1}^TB_{i+1} = R_2Q_2$, so that the above *zero-shift SVD algorithm* implicitly applies the usual QR iteration to $B_iB_i^T$ and $B_i^TB_i$ simultaneously.

We think of the SVD algorithm as a mapping from R^{2n-1} (the entries of B_i) to R^{2n-1} (the entries of B_{i+1}). To understand how errors propagate through iterations of the SVD algorithm, it is natural to look at the Jacobian of this map, since the Jacobian describes how small perturbations in B_i affect B_{i+1}. However, since we are interested in the propagation of *relative* errors, we will look instead at a Jacobian which maps small relative perturbations in B_i to relative perturbations in B_{i+1}. To this end, we will work with the logarithms of the entries B_i and B_{i+1}, since small perturbations in the logarithms of the matrix entries are equivalent to small relative perturbations in the matrix entries themselves. Thus, we will think of a bidiagonal B as a point in R^{2n-1} through the identification (recall that the nontrivial entries of B are positive)

$$\begin{bmatrix} a_1 & b_1 & & \\ & \ddots & \ddots & \\ & & \ddots & b_{n-1} \\ & & & a_n \end{bmatrix} \Longleftrightarrow \begin{bmatrix} \log b_1 \\ \vdots \\ \log b_{n-1} \\ \log a_1 \\ \vdots \\ \log a_n \end{bmatrix}$$

and think of one step of the SVD algorithm as a map F which maps vectors of logarithms of entries of B_i to vectors of logarithms of entries of B_{i+1}, $i = 0, 1, 2, \ldots$. Thus for $j > i$

$$F^{(j-i)}(B_i) = \underbrace{F \circ \cdots \circ F}_{j-i \text{ times}}(B_i) = B_j$$

We will call its Jacobian $M(j, i)$, which by the chain rule is the product of the one step Jacobians $M(j, i) = M(j, j \Leftrightarrow 1) \cdots M(i + 1, i)$. It is $M(j, i)$ which describes how initial relative errors in B and roundoff errors committed during prior SVD iterations propagate during later SVD iterations.

The following four facts were observed during initial numerical experiments:

Fact 1: The eigenvalues of $M(j, i)$ appear in reciprocal pairs. In other words, if λ is an eigenvalue, so is $1/\lambda$.

Fact 2: Near convergence (i.e. for i large enough), the eigenvalues of $M(i+1, i)$ are simple, approach 1 and all lie on the unit circle.

Fact 3: As $i \rightarrow \infty$, $M(i + 1, i)$ converges to the constant matrix

$$M_\infty = \begin{bmatrix} I_{n-1} & ,_n \\ 0 & I_n \end{bmatrix} \text{ where } ,_n = \begin{bmatrix} \Leftrightarrow 2 & 2 & & \\ & \ddots & \ddots & \\ & & \Leftrightarrow 2 & 2 \end{bmatrix} \qquad (1.9)$$

independent of initial data.

Fact 4: $\|M(j, i)\|$ grows linearly in the number of SVD steps $j \Leftrightarrow i$.

More precisely, we observed numerically for a large class of problems that $\|M(j, i)\|_\infty \leq 5.06 \cdot n \cdot (j \Leftrightarrow i)$ (n is the matrix dimension). In section 9, using O.D.E. methods, we will prove $\|M(j, i)\|_\infty \leq (8n \Leftrightarrow 4)(j \Leftrightarrow i) + O(1)$. This is the essential property of roundoff error propagation which lets us prove that the algorithm computes singular vectors as accurately as claimed. This linear growth is to be expected because near convergence we have

$$M(j, i) \approx M_\infty^{j-i} = \begin{bmatrix} I_{n-1} & (j \Leftrightarrow i),_n \\ 0 & I_n \end{bmatrix}$$

which grows linearly in norm. In section 10 we sketch an alternate proof of **Fact 4** based on this intuition.

In contrast, a straightforward error analysis without the machinery developed in this paper would yield an error bound growing exponentially in n.

As we will see, **Facts 1** and **2** follow from the observation that linear combinations of the variables $\log b_1, \ldots, \log a_n$ satisfy a Hamiltonian differential equation with Hamiltonian $\Leftrightarrow \mathrm{tr}(\log(BB^T)^2)/4$. The relationship between the flow and the algorithm $\cdots \rightarrow B_i \rightarrow B_{i+1} \rightarrow \cdots$ is as follows: if $\log b_1(t), \ldots, \log a_n(t)$ solve the Hamiltonian flow with initial conditions $\log b_1^{(0)}, \ldots, \log a_n^{(0)}$, then

$$\begin{bmatrix} \log b_1(i) \\ \vdots \\ \log a_n(i) \end{bmatrix} = \begin{bmatrix} \log b_1^{(i)} \\ \vdots \\ \log a_n^{(i)} \end{bmatrix}$$

where

$$B_i = \begin{bmatrix} a_1^{(i)} & b_1^{(i)} & & \\ & \ddots & \ddots & \\ & & \ddots & b_{n-1}^{(i)} \\ & & & a_n^{(i)} \end{bmatrix}$$

gives the i-th step in the SVD algorithm, starting from

$$B_0 = \begin{bmatrix} a_1^{(0)} & b_1^{(0)} & & \\ & \ddots & \ddots & \\ & & \ddots & b_{n-1}^{(0)} \\ & & & a_n^{(0)} \end{bmatrix}$$

(see sections 7 and 8 below). In contrast to many eigenvalue algorithms (see [Sym82, DLT89]), where the underlying symplectic structures are Lie-Poisson structures, here the underlying structure is a so called *Sklyanin* structure [Sem84]. The variables $\log a_1$, $\log a_1 a_2 = \log a_1 + \log a_2, \ldots$, $\log a_1 \cdots a_{n-1} = \log a_1 + \cdots + \log a_{n-1}$ will turn out to be the *momentum* variables (note that $\log a_n$ does not appear because it is determined by the other variables through the relation $a_1 \cdots a_n = \det(B) = \text{constant}$), and the $\log b_i$ will be the *position* variables. In the limit, the momenta converge to constants (the sums of the logarithms of the singular values), and the positions move at constant speed toward $\Leftrightarrow\infty$ (i.e. the offdiagonals b_i decay to zero geometrically).

Fact 1 will follow from the fact that the Jacobian with respect to the canonical Hamiltonian variables is symplectic; symplectic matrices have eigenvalues appearing in reciprocal pairs. **Facts 2, 3** and **4** will follow from the asymptotics of the Hamiltonian system, **Fact 1**, and Krein's theory of strongly stable symplectic matrices [Kre50, Kre55].

The use of the differential equation as outlined above suggests a paradigm for seeking algorithms to solve other eigenvalue problems to high relative accuracy. The symplectic interpretation of the (fortuitously chosen) relative errors is that they correspond to perturbations in the canonical variables for the symplectic structure in which the SVD algorithm is Hamiltonian. The general paradigm we suggest is the following: given a Poisson structure in which a given eigenvalue algorithm is Hamiltonian, one should try to construct natural global canonical variables (i.e. a global Darboux coordinate system). Such variables would indicate which functions of the eigenvalues (in our case, their logarithms) are relatively insensitive to appropriate perturbations in the matrix (in our case, relative perturbations in the entries) and hence are computable to high accuracy.

The rest of the paper is organized as follows. Section 2 proves the perturbation results ($\mathbf{A}'$) and ($\mathbf{B}'$). Section 3 describes the algorithm, and section 4 bounds the error in the singular values and vectors introduced by its stopping criterion. Section 5 uses the bound on $\|M(j,i)\|$ (**Fact 4**) to prove the error bounds for the zero-shift QR algorithm. Section 6 proves the main numerical result of the paper, an error bound for the singular vectors computed by the overall algorithm. Section 7 describes flows and the SVD algorithm. It also provides an independent proof of the convergence of the zero-shift SVD algorithm with detailed (and we believe new) asymptotic expressions for the matrix entries. Section 8 discusses the Hamiltonian structure of the flow and proves **Fact 1** above. Section 9 analyzes the asymptotics of $\|M(j,i)\|$ and proves **Fact 4**. Section 10 discusses the spectrum of the one-step Jacobian of the SVD, and proves **Facts 2** and **3**. Section 11 presents numerical experiments, and section 12 draws our conclusions.

We note that an alternative approach to computing singular values using gradient flows is presented in [Dri87].

Caveat: We will abuse the word "algorithm" in several differents ways. Sometimes it will refer to one step of the QR iteration (with or without shift) and sometimes it will refer to the full implementation with stopping criteria (the conventional [BDMS79] or the new [DK88] one). Which one is meant will be clear from context.

2 Perturbation Theory for Singular Vectors

In this section we prove the perturbation bound (1.8), which says that small relative perturbations in the entries of a bidiagonal matrix perturb the singular vectors by an amount proportional to the reciprocal of the relative gap (1.7). This result was conjectured in [DK88], and a weaker result proven in [BD88]. For the reader's convenience we also include a (new) proof of the eigenvalue bound (1.6).

The proofs depend on the following standard transformation [GK65]. Suppose the bidiagonal matrix B has entries

$$\begin{pmatrix} s_1 & s_2 & & & \bigcirc \\ & s_3 & s_4 & & \\ & & \ddots & & \\ & & & s_{2n-3} & s_{2n-2} \\ & \bigcirc & & & s_{2n-1} \end{pmatrix} \tag{2.1}$$

and SVD $B = U\Sigma V^T$, with $\Sigma = \mathrm{diag}\{\sigma_1, \ldots, \sigma_n\}$, $V = [v_1, \ldots, v_n]$ and $U = [u_1, \ldots, u_n]$. Then the symmetric matrix

$$S = \begin{pmatrix} 0 & s_1 & & & \bigcirc \\ s_1 & 0 & s_2 & & \\ & s_2 & 0 & & \\ & & \ddots & & \ddots \\ & & & & s_{2n-1} \\ \bigcirc & & & s_{2n-1} & 0 \end{pmatrix} \tag{2.2}$$

has eigenvalues $\pm\sigma_i$ with normalized, associated eigenvectors

$$h_i^{\pm} \equiv 2^{-1/2}(v_{i1}, \pm u_{i1}, v_{i2}, \pm u_{i2}, \ldots, v_{in}, \pm u_{in})^T \ .$$

(Note for future reference that the components of $h_i^{\pm}$ are bounded by $1/\sqrt{2}$.) Thus, the eigendecomposition for S also yields the SVD for B, and so perturbation theory for the eigenproblem for S also computes perturbation theory for the SVD of B.

As described in [GvL], the transformation $B \mapsto S$ should be viewed as the result of composing the SVD $\rightarrow$ eigenproblem map,

$$B \mapsto \begin{pmatrix} 0 & B^T \\ B & 0 \end{pmatrix} ,$$

with a perfect shuffle of the rows and columns, $\{1, 2, 3, \ldots, 2n\} \rightarrow \{1, n{+}1, 2, n{+}2, \ldots, n, 2n\}$, taking

$$\begin{pmatrix} 0 & B^T \\ B & 0 \end{pmatrix} \mapsto S \ .$$

Our first result bounds the effect of infinitesimal relative perturbations in the entries of S; the second result generalizes to finite perturbations.

Recall the standard fact that the eigenvalues of a tridiagonal matrix with nonzero off-diagonal entries are simple, and hence that the eigenvalues and eigenvectors depend smoothly on the entries of the matrix.

Theorem 2.3 *Let $S(t)$ be a matrix of the form (2.2), but with entries $s_i(t)$ which are smooth, positive functions of t. Let $\pm\sigma_i(t)$ and $h_i^{\pm}(t)$ denote the eigenvalues and eigenvectors of $S(t)$ respectively. Then*

$$|\frac{\dot{\sigma}_i(0)}{\sigma_i(0)}| \leq (2n \Leftrightarrow 1)(\max_m |\frac{\dot{s}_m(0)}{s_m(0)}|) \tag{2.4}$$

and

$$\|\dot{h}_i^{\pm}(0)\| \leq (2n \Leftrightarrow 1)(\max_m |\frac{\dot{s}_m(0)}{s_m(0)}|)(\frac{1}{relgap_i}) \; . \tag{2.5}$$

Proof. Let

$$\begin{aligned} \{\mu_1, \mu_2, \mu_3, \mu_4, \ldots\} &= \{+\sigma_1, \Leftrightarrow\sigma_1, +\sigma_2, \Leftrightarrow\sigma_2, \ldots\} \\ \{\omega_1, \omega_2, \omega_3, \omega_4, \ldots\} &= \{h_1^+, h_1^-, h_2^+, h_2^-, \ldots\} \end{aligned} \tag{2.6}$$

denote the eigenvalues and associated eigenvectors of S. Then, by regular perturbation theory, and repeated use of the eigenvalue equation,

$$\begin{aligned} \dot{\mu}_i &= (\omega_i, \dot{S}\omega_i) = \sum_{j=1}^{2n-1} (\frac{2\dot{s}_j}{s_j})(\omega_i(j) \; s_j \omega_i(j+1)) \\ &= 2 \sum_{j=1}^{2n-1} \frac{\dot{s}_j}{s_j} \omega_i(j)(\mu_i \omega_i(j) \Leftrightarrow s_{j-1}\omega_i(j \Leftrightarrow 1)) \\ &= \cdots = 2\mu_i \sum_{j=1}^{2n-1} \frac{\dot{s}_j}{s_j}(\sum_{m=1}^{j} \omega_i^2(m)(\Leftrightarrow 1)^{j-m}) \; . \end{aligned} \tag{2.7}$$

Thus

$$|\frac{\dot{\mu}_i}{\mu_i}| \leq 2(\max_k |\frac{\dot{s}_k}{s_k}|) \sum_{j=1}^{2n-1} | \sum_{m=1}^{j} \omega_i^2(m)(\Leftrightarrow 1)^{j-m}| \; .$$

On the other hand,it follows from (2.6) and the perpendicularity of h_q^+ and h_q^-, that

$$\sum_{m=1}^{n} \omega_i^2(2m) = \sum_{m=1}^{n} \omega_i^2(2m \Leftrightarrow 1) = \frac{1}{2}(\sum_{m=1}^{2n} \omega_i^2(m)) = \frac{1}{2} \; . \tag{2.8}$$

But then

$$| \sum_{m=1}^{j} \omega_i^2(m)(\Leftrightarrow 1)^{j-m}| \leq \frac{1}{2} \; , \tag{2.9}$$

which proves (2.4).

Again by regular perturbation theory and repeated use of the eigenvalue equation,

$$\dot{\omega}_i \;\; = \;\; \sum_{k \neq i} (\omega_k, \dot{S}\,\omega_i) \frac{\omega_k}{\mu_i \Leftrightarrow \mu_k}$$

$$
\begin{aligned}
&= \sum_{m=1}^{2n-1} (\frac{\dot{s}_m}{s_m}) \sum_{k\neq i} (\omega_k(m)\, s_m \omega_i(m+1) + \omega_i(m) s_m \omega_k(m+1)) \frac{\omega_k}{\mu_i \Leftrightarrow \mu_k} \\
&= \sum_{m=1}^{2n-1} (\frac{\dot{s}_m}{s_m}) \sum_{k\neq i} (\omega_k(m)\, \omega_i(m)(\mu_i + \mu_k) \Leftrightarrow \omega_i(m \Leftrightarrow 1) s_{m-1} \omega_k(m) \\
&\qquad \Leftrightarrow \omega_k(m \Leftrightarrow 1) s_{m-1} \omega_i(m)) \frac{\omega_k}{\mu_i \Leftrightarrow \mu_k} \\
&= \cdots = \sum_{m=1}^{2n-1} (\frac{\dot{s}_m}{s_m}) \sum_{k\neq i} (\frac{\mu_i + \mu_k}{\mu_i \Leftrightarrow \mu_k}) (\sum_{\ell=1}^{m} \omega_k(\ell) \omega_i(\ell) (\Leftrightarrow 1)^{m-\ell}) \omega_k \ . \qquad (2.10)
\end{aligned}
$$

Using the orthonormality of the eigenvectors ω_k, we obtain

$$
\begin{aligned}
\|\dot{\omega}_i\| &\leq (\max_m |\frac{\dot{s}_m}{s_m}|) \sum_{m=1}^{2n-1} \| \sum_{k\neq i} (\frac{\mu_i + \mu_k}{\mu_i \Leftrightarrow \mu_k}) (\sum_{\ell=1}^{m} \omega_k(\ell)\, \omega_i(\ell) (\Leftrightarrow 1)^{m-\ell}) \omega_k \| \\
&= (\max_m |\frac{\dot{s}_m}{s_m}|) \sum_{m=1}^{2n-1} (\sum_{k\neq i} (\frac{\mu_i + \mu_k}{\mu_i \Leftrightarrow \mu_k})^2 (\sum_{\ell=1}^{m} \omega_k(k) \omega_i(\ell) (\Leftrightarrow 1)^{m-\ell})^2)^{1/2} \\
&\leq (\max_m |\frac{\dot{s}_m}{s_m}|) (\max_{k\neq i} |\frac{\mu_i + \mu_k}{\mu_i \Leftrightarrow \mu_k}|) \sum_{m=1}^{2n-1} (\sum_{k=1}^{2n} (\sum_{\ell=1}^{m} \omega_k(\ell) \omega_i(\ell) (\Leftrightarrow 1)^{m-\ell})^2)^{1/2}
\end{aligned}
$$

But again, by orthonormality,

$$
\begin{aligned}
\sum_{k=1}^{2n} (\sum_{\ell=1}^{m} \omega_k(\ell) \omega_i(\ell) (\Leftrightarrow 1)^{m-\ell})^2 &= \sum_{1\leq \ell, q \leq m} (\sum_{k=1}^{2n} \omega_k(\ell) \omega_k(q) \omega_i(\ell) \omega_i(q) (\Leftrightarrow 1)^{m-\ell} (\Leftrightarrow 1)^{m-q}) \\
&= \sum_{\ell=1}^{m} \omega_i^2(\ell) \leq 1 \ .
\end{aligned}
$$

A simple computation shows that if $\mu_i = \pm \sigma_{i'}$, then $\max_{k\neq i} |\frac{\mu_i + \mu_k}{\mu_i - \mu_k}| = (\text{rel gap}_{i'})^{-1}$. This proves (2.5), and the Theorem. □

Remark. From (2.3), $\|\dot{h}_i^{\pm}\|^2 = \frac{1}{2}(\|\dot{u}_i\|^2 + \|\dot{v}_i\|^2)$. Hence (2.5) yields

$$
\max(\|\dot{u}_i\|, \|\dot{v}_i\|) \leq \sqrt{2}(2n \Leftrightarrow 1)(\max_m |\frac{\dot{s}_m(0)}{s_m(0)}|)(\frac{1}{relgap_i}) \ . \qquad (2.11)
$$

We also have the following global error bound.

Theorem 2.12 *Let B and B' be bidiagonal matrices with positive entries B_{ik} and B'_{ik}. Let*

$$
\eta_r \equiv (2n \Leftrightarrow 1) \max_{j,k} |\log \frac{B'_{jk}}{B_{jk}}| \qquad (2.13)
$$

be the relative error in B as defined in (1.5), and $\eta'_r \equiv e^{\eta_r} \Leftrightarrow 1 > \eta_r > 0$. Then

$$
1 \Leftrightarrow \eta'_r < e^{-\eta_r} < \frac{\sigma'_i}{\sigma_i} < e^{\eta_r} = 1 + \eta'_r \qquad (2.14)
$$

which implies

$$\Leftrightarrow\eta_r' < \frac{\sigma_i' \Leftrightarrow\sigma_i}{\sigma_i} < \eta_r' \ . \qquad (2.15)$$

Furthermore, if $\eta_r' < relgap_i$, the sines of the angles $\theta(u_i, u_i'), \theta(v_i, v_i')$ between the unperturbed singular vectors u_i, v_i and the perturbed singular vectors u_i', v_i', are bounded by

$$\max(\sin\theta(u_i, u_i'), \sin\theta(v_i, v_i')) \leq \frac{\sqrt{2}\eta_r'(1+\eta_r')}{relgap_i \Leftrightarrow\eta_r'} \qquad (2.16)$$

Proof. Let S and S' be matrices of the form (2.2) derived from B and B' as before. Set $S(t) = S + t(S' \Leftrightarrow S)$, $0 \leq t \leq 1$. With the notation of Theorem 2.3, we have from (2.7) and (2.9),

$$\begin{aligned} |\log\frac{\mu_i'}{\mu_i}| &\leq \int_0^t |\frac{d}{dt}\log\mu_i(t)|\,dt \\ &\leq \sum_{j=1}^{2n-1} (\int_0^1 |\frac{\dot{s}_j}{s_j}|\,dt) \\ &= \sum_{j=1}^{2n-1} |\log\frac{s_j'}{s_j}| \\ &\leq \eta_r \ , \end{aligned}$$

which proves (2.14). Note that the eigenvalues of $s(t)$ cannot pass through zero as t varies from zero to one.

In a similar way,from (2.10) and the calculation that follows in Theorem 2.3,

$$\|\dot{\omega}_i\| \leq (\max_{0\leq t\leq 1}\max_{k\neq i} |\frac{\mu_i(t)+\mu_k(t)}{\mu_i(t)\Leftrightarrow\mu_k(t)}|) \sum_{m=1}^{2n-1} |\frac{\dot{s}_m(t)}{s_m(t)}| \ ,$$

which yields

$$\|w_i' \Leftrightarrow w_i\| \leq (\max_{0\leq t\leq 1}\max_{k\neq i} |\frac{\mu_i(t)+\mu_k(t)}{\mu_i(t)\Leftrightarrow\mu_k(t)}|)\ \eta_r \ ,$$

and hence

$$\|(h_i^{\pm})' \Leftrightarrow h_i^{\pm}\| \leq (\max_{0\leq t\leq 1}\max_{k\neq i} |\frac{\sigma_i(t)+\sigma_k(t)}{\sigma_i(t)\Leftrightarrow\sigma_k(t)}|)\ \eta_r \ ,$$

as noted at the end of the proof of Theorem 2.3.

Now observe that

$$\eta_r(t) = (2n \Leftrightarrow 1)\max_j |\log\frac{s_j(t)}{s_j(0)}| \leq \eta_r$$

for all $0 \leq t \leq 1$. Thus from (2.14)

$$1 \Leftrightarrow\eta_r' < e^{-\eta_r} \leq \frac{\sigma_i(t)}{\sigma_i(0)} \leq e^{\eta_r} = 1 + \eta_r' \ .$$

Suppose $\sigma_i(t) > \sigma_k(t) > 0$. Then

$$\sigma_i(t) + \sigma_k(t) \leq e^{\eta_r}(\sigma_i(0) + \sigma_k(0)\,) \ ,$$

and

$$\begin{aligned}
\sigma_i(t) \Leftrightarrow \sigma_k(t) &\geq \sigma_i(0)(1 \Leftrightarrow \eta_r') \Leftrightarrow \sigma_k(0)(1 + \eta_r') \\
&= [(\frac{\sigma_i(0) \Leftrightarrow \sigma_k(0)}{\sigma_i(0) + \sigma_k(0)}) \Leftrightarrow \eta_r'](\sigma_i(0) + \sigma_k(0)) \\
&\geq (relgap_i \Leftrightarrow \eta_r')(\sigma_i(0) + \sigma_k(0)) ,
\end{aligned}$$

so that for $\eta_r' < relgap_i$,

$$|\frac{\sigma_i(t) + \sigma_k(t)}{\sigma_i(t) \Leftrightarrow \sigma_k(t)}| \leq \frac{e^{\eta_r}}{relgap_i \Leftrightarrow \eta_r'} = \frac{1 + \eta_r'}{relgap_i \Leftrightarrow \eta_r'} .$$

The same inequality holds if $\sigma_k(t) > \sigma_i(t)$, and we obtain

$$\|(h_i^{\pm})' \Leftrightarrow (h_i^{\pm})\| \leq \frac{\eta_r(1 + \eta_r')}{relgap_i \Leftrightarrow \eta_r'} .$$

As

$$\begin{aligned}
\sin\theta(u_i, u_i') \leq \|u_i' \Leftrightarrow u_i\| \leq \sqrt{2}\|(h_i^+)' \Leftrightarrow h_i^+\| \\
\sin\theta(v_i, v_i') \leq \|v_i' \Leftrightarrow v_i\| \leq \sqrt{2}\|(h_i^+)' \Leftrightarrow h_i^+\| ,
\end{aligned}$$

and as $\eta_r < \eta_r'$, the result follows. $\Box$

3 The SVD Algorithm

In this section we describe the algorithm for the bidiagonal SVD. This algorithm was introduced in [DK88], and discussed in detail there. Here, we present a simplified version of the algorithm which is adequate to prove our error bounds. The practical enhancements we omit here can greatly improve performance in some cases, but will not invalidate our error analysis. It turns out our eventual error bounds will depend on the number of QR steps necessary for convergence; the practical enhancements often reduce this number dramatically, and we summarize our numerical experience with the number of QR steps required in section 11.

Briefly, the algorithm is a hybrid of the standard shifted QR algorithm and the *implicit zero-shift QR algorithm.* The standard shifted QR is used on matrices which are well-conditioned (σ_n is not much smaller than σ_1), and implicit zero-shift QR is used on ill-conditioned submatrices ($\sigma_n \ll \sigma_1$). Implicit zero-shift QR is much more accurate than shifted QR, but much slower if the matrix is well-conditioned. Fortunately, shifted QR is adequately accurate on well-conditioned matrices, so we only need to exploit the more accurate implicit zero-shift QR when it is also fast. Thus, as the algorithm runs, the offdiagonal entries decrease and are eventually set to zero, deflating the matrix. On each newly deflated submatrix, σ_n and σ_1 are cheaply but reliably estimated and either shifted QR or implicit zero-shift QR used depending on the ratio σ_n/σ_1.

We will need to change one line in the algorithm presented in [DK88] in order to prove our error bounds in section 6. This change will not alter any of the numerical or timing results reported in [DK88].

We will present the implicit zero-shift QR algorithm, the stopping criterion for setting tiny offdiagonal entries to zero, and finally the overall algorithm. Implicit zero-shift QR calls the subroutine $ROT(f, g, cs, sn, r)$ [GVL83], which takes f and g as inputs and returns r, $cs = \cos\theta$ and $sn = \sin\theta$ such that

$$\begin{bmatrix} cs & sn \\ \Leftrightarrow sn & cs \end{bmatrix} \cdot \begin{bmatrix} f \\ g \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix} \tag{3.1}$$

$\mathbf{ROT(f, g, cs, sn, r)}$: take f and g as input and returns cs, sn and r satisfying (3.1).

```
if     ($f = 0$) then
       $cs = 0$; $sn = 1$; $r = g$;
elseif ($|f| > |g|$) then
       $t = g/f$; $tt = \sqrt{1 + t^2}$
       $cs = 1/tt$; $sn = t * cs$; $r = f * tt$
else
       $t = f/g$; $tt = \sqrt{1 + t^2}$
       $sn = 1/tt$; $cs = t * sn$; $r = g * tt$
endif
```

The algorithm also calls subroutine $UPDATE(cs, sn, v_1, v_2)$:

UPDATE$(\mathbf{cs}, \mathbf{sn}, \mathbf{v_1}, \mathbf{v_2})$: replace n-vectors v_1 and v_2 by $cs \cdot v_1 + sn \cdot v_2$ and $-sn \cdot v_1 + cs \cdot v_2$.

```
for   $i = 1$ to $n$
      $t = v_1(i)$
      $v_1(i) = cs * t + sn * v_2(i)$
      $v_2(i) = -sn * t + cs * v_2(i)$
endfor
```

Implicit Zero-Shift QR Algorithm: Let B be an n by n bidiagonal matrix with diagonal entries $a_1, \ldots, a_n$ and superdiagonal entries $b_1, \ldots, b_{n-1}$. The following algorithm replaces a_i and b_i by new values corresponding to one step of the QR iteration with zero shift. It also updates the right unit singular vectors v_i and left unit singular vectors u_i (at the start of the algorithm, these should be initialized to the columns of the identity matrix: $v_i(j) = u_i(j) = \delta_{ij}$.

```
$oldcs = 1$
$f = a_1$
$g = b_1$
for $i = 1,\ n-1$
      call $ROT(f, g, cs, sn, r)$
      call $UDPATE(cs, sn, v_i, v_{i+1})$
      if $(i \neq 1)$, $b_{i-1} = oldsn * r$
      $f = oldcs * r$
      $g = a_{i+1} * sn$
      $h = a_{i+1} * cs$
      call $ROT(f, g, cs, sn, r)$
      call $UPDATE(cs, sn, u_i, u_{i+1})$
      $a_i = r$
      $f = h$
      $g = b_{i+1}$
      $oldcs = cs$
      $oldsn = sn$
endfor
$b_{n-1} = h * sn$
$a_n = h * cs$
```

This algorithm may also be expressed in the following terser but equivalent form (we will need the expanded form above for the analysis):

```
$oldcs = 1$
$cs = 1$
for $i = 1,\ n \Leftrightarrow 1$
    call $ROT(a_i * cs, b_i, cs, sn, r)$
    call $UDPATE(cs, sn, v_i, v_{i+1})$
    if $(i \neq 1)$, $b_{i-1} = oldsn * r$
    call $ROT(oldcs * r, a_{i+1} * sn, oldcs, oldsn, a_i)$
    call $UPDATE(cs, sn, u_i, u_{i+1})$
endfor
$h = a_n * cs$
$b_{n-1} = h * oldsn$
$a_n = h * oldcs$
```

Next we discuss the stopping criterion, i.e. how to decide when to set an offdiagonal b_i to zero and so converge. It is important that the stopping criterion introduce error in the singular vectors not much worse than the bound of Theorem 2.2, since otherwise we will not compute the singular vectors to their inherent accuracy. The criterion described in [DK88] is the following.

Stopping Criterion: This algorithm decides when an offdiagonal entry b_i can be set to zero. $0 < tol < 1$ is a relative error tolerance.

```
$\mu_1 = a_1$
for $j = 1,\ n \Leftrightarrow 1$
    if $|b_j| < tol * \mu_j$, set $b_j = 0$
    $\mu_{j+1} = |a_{j+1}| * (\mu_j / (\mu_j + |b_j|))$
endfor
```

It was shown in [DK88] that this criterion perturbs the singular values of B by no more than about $n \cdot tol/2^{1/2}$ when $tol \ll 1$. In sections 4, 5 and 6 we will show that it also has a small effect on the singular vectors, in particular that it does not change them in direction by more than about $O(tol/relgap_i)$, which is the same level as their inherent uncertainty.

Thus, the overall algorithm can be summarized as follows.

Bidiagonal SVD Algorithm (simplified)

```
Loop:

    Find the bottommost unreduced submatrix of $B$; call it $\hat{B}$.
        (Let $s$ and $e$ be the starting and ending indices of $\hat{B}$ within $B$.
        Then $b_e = 0$ if $e < n$, $b_{s-1} = 0$ if $s > 1$ and $b_i \neq 0$ for $s \leq i \leq e \Leftrightarrow 1$.)
    If $\hat{B}$ is 1 by 1 ($s = e$), we are done.

    Apply the stopping criterion to $\hat{B}$; if any $b_i$ are set to 0, return to Loop

    Estimate the smallest singular value $\underline{\sigma}$ and the largest singular value $\overline{\sigma}$ of $\hat{B}$.

    if $n * \underline{\sigma}/\overline{\sigma} < \max(\varepsilon/tol, .01)$ then
        Use implicit zero-shift QR on $\hat{B}$
    else
        Use standard shifted QR on $\hat{B}$
    endif

Goto Loop
```

The difference between this algorithm and the one in [DK88] (besides some insignificant simplifications) is the use of the test $(n * \underline{\sigma}/\overline{\sigma} < \max(\varepsilon/tol, .01))$ in place of $(n * \underline{\sigma}/\overline{\sigma} < \varepsilon/tol)$ to determine whether or to use zero-shift QR or shifted QR. The reason for this change will become apparent in section 6. In [DK88] the value of tol used in the numerical tests was $tol = 100 * \varepsilon$, so this change has no effect on the results reported there. The value 100 was chosen empirically to make the algorithm fast, but could easily be made as large as 1000 or as small as 10 without greatly impacting performance.

We summarize here the ways in which the above description simplifies the actual algorithm of [DK88], argue that we have not omitted any features which could greatly increase the error, and summarize the properties of the practical algorithm we do need for the error analysis.

Bidirectional QR: If B is as given in (1.1), we define $rev(B)$ as the matrix

$$rev(B) = \begin{bmatrix} a_n & b_{n-1} & & \\ & \ddots & \ddots & \\ & & \ddots & b_1 \\ & & & a_1 \end{bmatrix}, \tag{3.2}$$

i.e. B with the diagonals reversed. The SVDs of B and $rev(B)$ are simply related since $B = Prev(B)^T P^T$, where P is the permutation matrix with ones going from the bottom left to the upper right: $B = U\Sigma V^T$ implies $rev(B) = (PV)\Sigma(PU)^T$. It turns out that

QR iteration may converge much faster applied to $rev(B)$ than B, and so the practical algorithm tries to exploit this and perform whichever one is faster. The reason for the speed difference is as follows. As zero-shift QR converges, the singular values appear on the diagonal in decreasing order from upper left to lower right. If the entries of B are already graded in this way, the algorithm will converge more quickly than if they are not. The algorithm tests for this grading in a very simple way: if $|a_1| > |a_n|$, QR is applied to B, and otherwise to $rev(B)$. This is not a foolproof scheme, but can quadruple the speed for strongly graded matrices.

Since the SVDs of B and $rev(B)$ are essentially permutations of one another, it suffices to perform an error analysis either of QR applied to B or QR applied to $rev(B)$. Our error bound will however depend on the total number of QR steps taken, and so benefit from the practical enhancements; we summarize our numerical experience with the number of QR steps required for convergence in the section 11. Of course, in practice the number of QR steps is known after the algorithm terminates, and this value could be used in our later error bounds.

Bidirectional stopping criterion: Just as QR can be applied either to B or $rev(B)$, so can the stopping criterion. In fact we apply it to both B and $rev(B)$, no matter to which of the two we apply QR.

2 by 2 submatrices: When the bottommost unreduced submatrix $\hat{B}$ is 2 by 2, we can apply the quadratic formula to directly compute its SVD. In practice, we implement it quite carefully so that when addition and subtraction are implemented with a guard digit ($fl(a \pm b) = (a \pm b)(1 + \varepsilon_1)$, $|\varepsilon_1| \leq \varepsilon$) we compute both the singular values and singular vectors to nearly full machine precision, even if the relative gap is small so that the singular vectors are ill-conditioned. In the absence of a guard digit (the model of arithmetic in (5.1)) we only compute nearly the exact singular values and vectors of a matrix which differs from the input in a few bits in each entry. Thus, the global perturbation bounds of section 2 are respected.

Deflation when $a_i = 0$: When some $a_i = 0$, then the code will automatically set b_{i-1}, b_{n-1} and a_n to zero; this is because $f = h = 0$ at the end of each loop iteration. Thus, this case needs no special consideration (in contrast to the standard SVD [BDMS79] which treats this case specially).

Choosing implicit zero-shift QR versus standard shifted QR: We need to compute cheap and reliable estimates of the smallest singular value $\underline{\sigma}$ and largest singular value $\overline{\sigma}$ in order to decide whether to use standard or implicit zero-shift QR. The largest singular value $\overline{\sigma}$ can clearly be estimated to within a factor of 2 by the largest absolute entry of B. The smallest singular value $\underline{\sigma}$ turns out to be estimated to within a factor of $n^{\pm 1/2}$ by $\min_k \mu_k$, where the μ_k are computed by the recurrence in the Stopping Criterion above [DK88]. This uncertainty in $\underline{\sigma}$ and $\overline{\sigma}$ will contribute a factor of $n^{1/2}$ to our final error bound.

4 Error Bounds for the Stopping Criterion

Let B be a bidiagonal matrix with positive entries a_p, b_q, and let B' be the bidiagonal matrix obtained from B by setting $b_j = 0$ for some j, $1 \leq j \leq n \Leftrightarrow 1$. Let S and S' be the associated tridiagonal matrices of form (2.2).

Define vectors z_{2j}, z_{2j+1} through

$$Sz_{2j} = e_{2j} \tag{4.1}$$

$$Sz_{2j+1} = e_{2j+1} \ , \tag{4.2}$$

where $\{e_i\}$ give the standard basis in $\mathbb{R}^{2n}$. A simple calculation, using the explicit form of S, shows that

$$z_{2j}(\ell) = 0 \quad \text{if} \quad \ell \text{ is even or } \ell > 2j+1 \tag{4.3}$$

$$z_{2j+1}(\ell) = 0 \text{ if } \ell \text{ is odd} \quad \text{ or } \ell < 2j \tag{4.4}$$

Set

$$\begin{aligned} m_1 &\equiv \min(s_{2j}\|z_{2j}\|_1, \ s_{2j}\|z_{2j+1}\|_1) \\ m_1' &\equiv e^{m_1} \Leftrightarrow 1 > m_1 \end{aligned}$$

$$\begin{aligned} m_2 &\equiv \min\left(s_{2j}\|z_{2j}\|_2 \ , \ s_{2j}\|z_{2j+1}\|_2\right) \\ m_2' &= e^{m_2} \Leftrightarrow 1 > m_2 \ , \end{aligned}$$

where $\|\cdot\|_1, \|\cdot\|_2$ denote the L^1 and L^2 norms respectively. It is easy to see that for a given j,

$$m_1 = b_j \mu_j \tag{4.5}$$

where μ_j appears in the Stopping Criterion of §3. Similarly,

$$m_2 = b_j \eta_j \tag{4.6}$$

corresponds to μ_j, but for rev (B) as in (3.2). (See the discussion of the bidirectional stopping criterion in §3.)

Theorem 4.1 *Let B, B' etc. be as above. Let σ_i, σ_i' be the singular values of B, B' respectively, and let u_i, v_i and u_i', v_i' be the associated singular vectors, making angles $\theta(u_i, u_i'), \theta(v_i, v_i')$ respectively. Then for $1 \leq i \leq n$,*

$$1 \Leftrightarrow m_\ell' < e^{-m_\ell} \leq \frac{\sigma_i'}{\sigma_i} \leq e^{m_\ell} = 1 + m_\ell' \ , \tag*{(4.7)$_\ell$}$$

where $\ell = 1, 2$.

Furthermore, if $m_1' < relgap_i$,

$$\max(\sin\theta(u_i, u_i'), \sin\theta(v_i, v_i')) \leq m_1'\Big(\frac{1}{\sqrt{2}}\Big(1 + \sqrt{n \Leftrightarrow \frac{1}{2}}\Big) + \Big(\frac{1 + m_1'}{relgap_i \Leftrightarrow m_1'}\Big)\sqrt{\frac{2n+5}{4}}\Big) \ , \tag*{(4.8)$_1$}$$

and if $m_2' < relgap$,

$$\max(\sin\theta(u_i, u_i'), \sin\theta(v_i, v_i')) \leq m_2'\Big(1 + \frac{(1 + m_2')}{relgap_i \Leftrightarrow m_2'}\Big). \tag*{(4.8)$_2$}$$

Proof. Let $S(t)$ be the matrix obtained from S by replacing s_{2j} by ts_{2j}, $0 \le t \le 1$. Note that $S(1) = S$, $S(0) = S'$. By (4.1) and (4.2), note that

$$\begin{aligned} S(t)z_{2j} &= Sz_{2j} = e_{2j} \\ S(t)z_{2j+1} &= Sz_{2j+1} = e_{2j+1} \end{aligned}$$

for all t in the interval.

With the notation of Theorem 2.3,

$$\begin{aligned} \dot{\mu}_i(t) &= 2s_{ij}(e_{2j}, w_i(t))\,(e_{2j+1}, w_i(t)) \\ &= 2s_{ij}(S(t)z_{2j}, w_i(t))\;(e_{2j+1}, w_i(t)) \\ &= 2s_{ij}\;\mu_i\;(z_{2j}, w_i(t))\;(e_{2j+1}, w_i(t))\;. \end{aligned}$$

But $|(e_{2j+1}, w_i(t))| = 2^{-1/2}$ and $|(z_{2j}, w_i(t))| \le \|z_{2j}\|_1 \|w_i(t)\|_\infty \le \|z_{2j}\|_1 2^{-1/2}$. On the other hand from (4.3),

$$\begin{aligned} |(z_{2j}, w_i(t))| &\le 2^{-1/2}\|z_{2j}\|_2 \|v_{i'}(t)\|_2\;,\ \text{for suitable } i', \\ &= 2^{-1/2}\|z_{2j}\|_2\;. \end{aligned} \tag{4.9}$$

Thus

$$\left|\frac{d\log\mu_i(t)}{dt}\right| \le s_{2j}\|z_{2j}\|_\ell = m_\ell\;, \qquad \ell = 1, 2$$

and integration gives the desired eigenvalue bound $(4.7)_\ell$, $\ell = 1, 2$.

We now prove $(4.8)_2$. Perturbation theory gives

$$\begin{aligned} \frac{dw_i}{dt} &= \sum_{k\neq i}(w_i, \dot{S}w_k)\frac{w_k}{\mu_i \Leftrightarrow \mu_k} \\ &= s_{2j}\sum_{k\neq i}\left((e_{2j}, w_i)\,(e_{2j+1}, w_k) + (e_{2j}, w_k)(e_{2j+1}, w_i)\right)\frac{w_k}{\mu_i \Leftrightarrow \mu_k} \\ &= s_{2j}\sum_{k\neq i}\left(\mu_i(z_{2j}, w_i)(e_{2j+1}, w_k) + \mu_k(z_{2j}, w_k)(e_{2j+1}, w_i)\right)\frac{w_k}{\mu_i \Leftrightarrow \mu_k}\;, \qquad \text{as } S(t)z_{2j} = e_{2j}\;, \\ &= \frac{s_{2j}}{2}\sum_{k\neq i}\left[\left(\frac{\mu_i + \mu_k}{\mu_i \Leftrightarrow \mu_k}\right)(z_{2j}, w_i)(e_{2j+1}, w_k) + \left(\frac{\mu_k + \mu_i}{\mu_i \Leftrightarrow \mu_k}\right)(z_{2j}, w_k)(e_{2j+1}, w_i)\right]w_k \\ &\quad + \frac{s_{2j}}{2}\sum_{k\neq i}\left[(z_{2j}, w_i)(e_{2j+1}, w_k) \Leftrightarrow (z_{2j}, w_k)(e_{2j+1}, w_i)\right]w_k\;. \end{aligned}$$

The second sum is bounded in norm by

$$|(z_{2j}, w_i)|\;\|e_{2j+1}\|_2 + |(e_{2j+1}, w_i)|\;\|z_{2j}\|_2 \le \sqrt{2}\|z_{2j}\|_2\;,\ \text{by (4.9)}.$$

The first sum is bounded in norm by

$$\max_{k\neq i}\left|\frac{\mu_i + \mu_k}{\mu_i \Leftrightarrow \mu_k}\right|\left\{\sum_{k=1}^{2n}\left[(z_{2j}, w_i)^2(e_{2j+1}, w_k)^2 + (z_{2j}, w_k)^2(e_{2j+1}, w_i)^2\right.\right.$$
$$\left.\left. + 2(z_{2j}, w_i)(e_{2j+1}, w_i)(e_{2j+1}, w_k)(z_{2j}, w_k)\right]\right\}^{1/2}$$

$$
\begin{aligned}
&= \max_{k\neq i}\left|\frac{\mu_i+\mu_k}{\mu_i \Leftrightarrow \mu_k}\right|\left\{\frac{\|z_{2j}\|_2^2}{2}+\frac{\|z_{2j}\|_2^2}{2}+2\cdot(2^{-1/2}\|z_{2j}\|_2)\cdot 2^{-1/2}\cdot\|z_{2j}\|_2\right\}^{1/2}\\
&= \sqrt{2}\left(\max_{k\neq i}\left|\frac{\mu_i+\mu_k}{\mu_i \Leftrightarrow \mu_k}\right|\right)\|z_{2j}\|_2 \ .
\end{aligned}
$$

Combining terms,

$$
\|\frac{d\omega_i(t)}{dt}\|_2 \le \frac{s_{2j}\|z_{2j}\|_2}{\sqrt{2}}(1+\max_{k\neq i}|\frac{\mu_i(t)+\mu_k(t)}{\mu_i(t)\Leftrightarrow\mu_k(t)}|) \ .
$$

A similar computation shows that the same inequality holds with $\|z_{2j}\|_2$ replaced by $\|z_{2j+1}\|_2$. Furthermore, note that the proof of $(4.7)_\ell$ shows that

$$
1 \Leftrightarrow m'_\ell < e^{-m_\ell} \le e^{-(1-t)m_\ell} \le \frac{\mu_i(t)}{\mu_i} \le e^{(1-t)m_\ell} \le e^{m_\ell} = 1+m'_\ell \ ,
$$

and hence, arguing as in the proof of Theorem 2.11,

$$
\|\frac{dh_i^{\pm}(t)}{dt}\|_2 \le \frac{m_2}{\sqrt{2}}(1+\frac{1+m'_2}{relgap_i \Leftrightarrow m'_2}) \ ,
$$

which yields $(4.8)_\ell$, upon integration.

The inequality $(4.8)_1$ is proved in the same way. Factors of order $\sqrt{n}$ appear, for example, in estimating the term

$$
\sum_{k\neq i}(z_{2j},w_k)^2 \le (2n\Leftrightarrow 1)(2^{-1/2}\|z_{2j}\|_1 \ .
$$

□

Remark. The inequality $(4.7)_\ell$ was proved in [DK88] by a different method.

5 Error Bounds for the Implicit Zero-Shift QR Algorithm

In this section we will derive error bounds for the quantities computed by m steps of the implicit zero-shift QR algorithm of section 3. We will use the bound on $\|M(i,j)\|$ to be derived in section 9 to bound the round off error propagated from step to step. Our main results will be Lemma 5.11, which bounds the relative errors in the bidiagonal matrix entries after m zero-shift QR steps, and Lemma 5.14, which bounds the absolute error in the computed orthogonal matrix of the m zero-shift QR steps.

In the error analysis, we will use the fact that an absolute perturbation ϵ in $\log x$ is to first order equivalent to a relative perturbation ϵ in x:

$$\log x(1+\epsilon) = \log x + \log(1+\epsilon) \approx \log x + \epsilon$$

Therefore, the Jacobian map $M(j,i)$ which propagates absolute perturbations in the logarithms of entries of B_i to B_j also propagates relative perturbations of entries of B_i to B_j.

As is traditional in numerical analysis, we will bound quantities of the form $\prod(1+\epsilon_i)$ by instead bounding $s \equiv \sum |\epsilon_i|$. When $s \ll 1$ (the case of interest), we then have approximately that $1 \Leftrightarrow s \leq \prod(1+\epsilon_i) \leq 1+s$. If more rigor is desired, we can use the fact that for all $s < 1$ we have $1 \Leftrightarrow s \leq \prod(1+\epsilon_i) \leq e^s$.

Our model of arithmetic is a variation on the standard one: the floating point result $fl(\cdot)$ of the operation $(\cdot)$ is given by

$$\begin{aligned} fl(a \pm b) &= a(1+\varepsilon_1) \pm b(1+\varepsilon_2) \\ fl(a \times b) &= (a \times b)(1+\varepsilon_3) \\ fl(a/b) &= (a/b)(1+\varepsilon_4) \\ fl(\sqrt{a}) &= \sqrt{a}(1+\varepsilon_5) \end{aligned} \tag{5.1}$$

where $|\varepsilon_i| \leq \varepsilon$, and $\varepsilon \ll 1$ is the machine precision. This is somewhat more general than the usual model which uses $fl(a \pm b) = (a \pm b)(1+\varepsilon_1)$ and includes machines like the Cray which do not have a guard digit. We do not consider over/underflow; methods for extending error analysis to include underflow are presented in [Dem84].

Our analysis proceeds by five lemmas. Lemma 5.2 [DK88] analyzes the roundoff errors in the subroutine *ROT* of section 3. Lemma 5.4 uses Lemma 5.2 to bound the errors in the bidiagonal matrix B after one step of the implicit zero-shift QR algorithm. Lemma 5.11 uses Lemma 5.4 and the bound on $\|M(j,i)\|$ of section 9 to bound the errors in the bidiagonal matrix after m steps of the implicit zero-shift QR algorithm. Lemma 5.13 shows that small errors in the sines and cosines computed by one step of the algorithm only cause small errors in the computed orthogonal matrices containing the singular vectors. Finally Lemma 5.14 bounds the absolute error in the computed orthogonal matrices containing the singular vectors after m steps of the algorithm.

Lemma 5.2 *Let $\cos\theta$, $\sin\theta$ and ρ denote the exact outputs of ROT for inputs f and g and exact arithmetic. Let $cs = (1+\epsilon_{cs})\cos\theta$, $sn = (1+\epsilon_{sn})\sin\theta$ and $r = (1+\epsilon_r)\rho$ denote the floating point results of ROT applied to the perturbed inputs $\hat{f} = (1+\epsilon_f)f$ and $\hat{g} = (1+\epsilon_g)g$.*

Then to first order we may bound the relative errors ϵ_{cs}, ϵ_{sn} and ϵ_r in terms of ϵ_f, ϵ_g and the machine precision ε as follows:

$$\begin{aligned} |\epsilon_{cs}| &\leq (|\epsilon_f| + |\epsilon_g|)\sin^2\theta + \frac{21\varepsilon}{4} \\ |\epsilon_{sn}| &\leq (|\epsilon_f| + |\epsilon_g|)\cos^2\theta + \frac{21\varepsilon}{4} \\ |\epsilon_r| &\leq |\epsilon_f|\cos^2\theta + |\epsilon_g|\sin^2\theta + \frac{13\varepsilon}{4} \end{aligned} \tag{5.3}$$

Proof. See the proof of Lemma 5 in [DK88]. The slightly different model $fl(a \pm b) = (a \pm b)(1 + \varepsilon_1)$ used in [DK88] does not affect the final result because only positive quantities are added in *ROT*; this makes the two models equivalent. □

To state the next lemma, we need to be able to distinguish the different values the variables in the implicit zero-shift QR algorithm take on at different times. To this end, we state the following equivalent algorithm, where the variables are labeled by the loop counter i:

Labeled Implicit Zero-Shift QR Algorithm:

```
$oldcs_1 = 1$
$f_1 = a_1$
$g_1 = b_1$
for $i = 1, n \Leftrightarrow 1$
    call $ROT(f_i, g_i, cs_{i1}, sn_{i1}, r_{i1})$
    call $UDPATE(cs_{i1}, sn_{i1}, v_i, v_{i+1})$
    if $(i \neq 1)$, $b_{i-1} = oldsn_i * r_{i1}$
    $f_{i1} = oldcs_i * r_{i1}$
    $g_{i1} = a_{i+1} * sn_{i1}$
    $h_i = a_{i+1} * cs_{i1}$
    call $ROT(f_{i1}, g_{i1}, cs_{i2}, sn_{i2}, r_{i2})$
    call $UPDATE(cs_{i2}, sn_{i2}, u_i, u_{i+1})$
    $a_i = r_{i2}$
    $f_{i+1} = h_i$
    $g_{i+1} = b_{i+1}$
    $oldcs_{i+1} = cs_{i2}$
    $oldsn_{i+1} = sn_{i2}$
endfor
$b_{n-1} = h_{n-1} * sn_{n-1,2}$
$a_n = h_{n-1} * cs_{n-1,2}$
```

Let the n by n bidiagonal matrix B have diagonal entries a_i and offdiagonal entries b_i. Let the matrix B' with entries a_i' and b_i' be the the result of applying the implicit zero-shift QR algorithm to B once in exact arithmetic, and let the variables in the labeled implicit zero-shift QR algorithm above denote the corresponding intermediate values.

Now let $\hat{B}$ be the slightly perturbed matrix with entries $\hat{a}_i = a_i(1 + \epsilon_{a_i})$ and $\hat{b}_i = b_i(1 + \epsilon_{b_i})$, and let $\hat{B}'$ (with entries $\hat{a}_i' = a_i'(1 + \epsilon_{a_i'})$ and $\hat{b}_i' = b_i'(1 + \epsilon_{b_i'})$) be the result of applying the implicit zero-shift QR algorithm in floating point arithmetic to $\hat{B}$. Let hatted variable denote the corresponding intermediate floating point variables (e.g. $\hat{f}_i = f_i(1+\epsilon_{f_i})$). We wish to bound the final relative errors $\epsilon_{a'_i}$ and $\epsilon_{b'_i}$ in the entries of $\hat{B}'$ and the relative errors $\epsilon_{\hat{cs}_{i1}}$ $\epsilon_{\hat{sn}_{i1}}$, $\epsilon_{\hat{cs}_{i2}}$, and $\epsilon_{\hat{sn}_{i2}}$ in the sines and cosines in terms of the initial relative errors ϵ_{a_i} and ϵ_{b_i} and the machine precision ε.

Lemma 5.4 *In terms of the notation just introduced,*

$$\max_i(|\epsilon_{a'_i}|, |\epsilon_{b'_i}|) \le 6(2n \Leftrightarrow 1)\max_i(|\epsilon_{a_i}|, |\epsilon_{b_i}|) + (47n \Leftrightarrow 27)\varepsilon \tag{5.5}$$

and

$$\epsilon_{cs,sn} \equiv \max_i(|\epsilon_{\hat{cs}_{i1}}|, |\epsilon_{\hat{sn}_{i1}}|, |\epsilon_{\hat{cs}_{i2}}|, |\epsilon_{\hat{sn}_{i2}}|) \le 5(2n \Leftrightarrow 1)\max_i(|\epsilon_{a_i}|, |\epsilon_{b_i}|) + (41n \Leftrightarrow 66)\varepsilon \tag{5.6}$$

Remark 5.7 Lemma 5.4 is a variation on Lemma 7 in [DK88].

Proof. We begin systematically applying (5.1) and Lemma 5.2 to all the operations in the labeled implicit zero-shift QR algorithm in order to derive a recurrence relation for $|\epsilon_{f_i}|$ and $|\epsilon_{oldcs_i}|$. Initially

$$\hat{f}_1 \equiv f_1(1+\epsilon_{f_1}) = a_1(1+\epsilon_{a_1}) \quad \text{and} \quad \hat{g}_1 \equiv g_1(1+\epsilon_{g_1}) = b_1(1+\epsilon_{b_1})$$

so that $\epsilon_{f_1} = \epsilon_{a_1}$ and $\epsilon_{g_1} = \epsilon_{b_1}$. At the top of the loop we always have $\hat{g}_i = \hat{b}_i$ and so $\epsilon_{g_i} = \epsilon_{b_i}$. After the first call to *ROT* we have

$$\begin{aligned}
\hat{cs}_{i1} &\equiv cs_{i1}(1+\epsilon_{cs_{i1}}) &&\text{where} && |\epsilon_{cs_{i1}}| \le (|\epsilon_{f_i}| + |\epsilon_{b_i}|)sn_{i1}^2 + \frac{21\varepsilon}{4} \\
\hat{sn}_{i1} &\equiv sn_{i1}(1+\epsilon_{sn_{i1}}) &&\text{where} && |\epsilon_{sn_{i1}}| \le (|\epsilon_{f_i}| + |\epsilon_{b_i}|)cs_{i1}^2 + \frac{21\varepsilon}{4} \\
\hat{r}_{i1} &\equiv r_{i1}(1+\epsilon_{r_{i1}}) &&\text{where} && |\epsilon_{r_{i1}}| \le |\epsilon_{f_i}|cs_{i1}^2 + |\epsilon_{b_i}|sn_{i1}^2 + \frac{13\varepsilon}{4}
\end{aligned}$$

Next, we get

$$\begin{aligned}
\hat{f}_{i1} &\equiv f_{i1}(1+\epsilon_{f_{i1}}) &&\text{where} && |\epsilon_{f_{i1}}| \le |\epsilon_{oldcs_i}| + |\epsilon_{r_{i1}}| + \varepsilon \le |\epsilon_{oldcs_i}| + |\epsilon_{f_i}|cs_{i1}^2 + |\epsilon_{b_i}|sn_{i1}^2 + \frac{17\varepsilon}{4} \\
\hat{g}_{i1} &\equiv g_{i1}(1+\epsilon_{g_{i1}}) &&\text{where} && |\epsilon_{g_{i1}}| \le |\epsilon_{a_{i+1}}| + |\epsilon_{sn_{i1}}| + \varepsilon \le |\epsilon_{a_{i+1}}| + (|\epsilon_{f_i}| + |\epsilon_{b_i}|)cs_{i1}^2 + \frac{25\varepsilon}{4} \\
\hat{h}_i &\equiv h_i(1+\epsilon_{h_i}) &&\text{where} && |\epsilon_{h_i}| \le |\epsilon_{a_{i+1}}| + |\epsilon_{cs_{i1}}| + \varepsilon \le |\epsilon_{a_{i+1}}| + (|\epsilon_{f_i}| + |\epsilon_{b_i}|)sn_{i1}^2 + \frac{25\varepsilon}{4}
\end{aligned}$$

After the second call to *ROT* we have

$$\begin{aligned}
\hat{cs}_{i2} &\equiv cs_{i2}(1+\epsilon_{cs_{i2}}) &&\text{where} && |\epsilon_{cs_{i2}}| &\le (|\epsilon_{oldcs_i}| + |\epsilon_{a_{i+1}}| + 2|\epsilon_{f_i}|cs_{i1}^2 + |\epsilon_{b_i}|)sn_{i2}^2 + \frac{63\varepsilon}{4} \\
\hat{sn}_{i2} &\equiv sn_{i2}(1+\epsilon_{sn_{i2}}) &&\text{where} && |\epsilon_{sn_{i2}}| &\le (|\epsilon_{oldcs_i}| + |\epsilon_{a_{i+1}}| + 2|\epsilon_{f_i}|cs_{i1}^2 + |\epsilon_{b_i}|)cs_{i2}^2 + \frac{63\varepsilon}{4} \\
\hat{r}_{i2} &\equiv r_{i2}(1+\epsilon_{r_{i2}}) &&\text{where} && |\epsilon_{r_{i2}}| &\le |\epsilon_{oldcs_i}|cs_{i2}^2 + |\epsilon_{a_{i+1}}|sn_{i2}^2 + |\epsilon_{f_i}|cs_{i1}^2 \\
& && && &\quad +|\epsilon_{b_i}|(cs_{i2}^2 sn_{i1}^2 + sn_{i2}^2 cs_{i1}^2) + \frac{55\varepsilon}{4}
\end{aligned}$$

Since $oldcs_{i+1} = cs_{i2}$ and $f_{i+1} = h_i$, we have $\epsilon_{oldcs_{i+1}} = \epsilon_{cs_{i2}}$ and $\epsilon_{f_{i+1}} = \epsilon_{h_i}$, so

$$\begin{bmatrix} |\epsilon_{f_{i+1}}| \\ |\epsilon_{oldcs_{i+1}}| \end{bmatrix} \leq \begin{bmatrix} sn_{i1}^2 & 0 \\ 2cs_{i1}^2 sn_{i2}^2 & sn_{i2}^2 \end{bmatrix} \cdot \begin{bmatrix} |\epsilon_{f_i}| \\ |\epsilon_{oldcs_i}| \end{bmatrix} + \begin{bmatrix} sn_{i1}^2 & 1 \\ sn_{i2}^2 & sn_{i2}^2 \end{bmatrix} \cdot \begin{bmatrix} |\epsilon_{b_i}| \\ |\epsilon_{a_{i+1}}| \end{bmatrix} + \begin{bmatrix} 25 \\ 63 \end{bmatrix} \frac{\varepsilon}{4}$$

We may write this as

$$E_{i+1} \leq A_i \cdot E_i + F_i + G_i$$

where

$$E_i = \begin{bmatrix} |\epsilon_{f_i}| \\ |\epsilon_{oldcs_i}| \end{bmatrix} \ , \ F_i = \begin{bmatrix} 1 \\ 1 \end{bmatrix} (|\epsilon_{b_i}| + |\epsilon_{a_{i+1}}|) \quad \text{and} \quad G_i = \begin{bmatrix} 25 \\ 63 \end{bmatrix} \frac{\varepsilon}{4}$$

This implies

$$E_{i+1} \leq \sum_{j=1}^{i} (A_i \cdots A_{j+1})(F_j + G_j) + (A_i \cdots A_1) E_1$$

Since componentwise

$$A_i \cdots A_{j+1} \leq \begin{bmatrix} sn_{i1}^2 & 0 \\ 2cs_{i1}^2 & 1 \end{bmatrix} \cdots \begin{bmatrix} sn_{j+1,1}^2 & 0 \\ 2cs_{j+1,1}^2 & 1 \end{bmatrix} = \begin{bmatrix} sn_{i1}^2 \cdots sn_{j+1,1}^2 & 0 \\ 2(1 \Leftrightarrow sn_{i1}^2 \cdots sn_{j+1,1}^2) & 1 \end{bmatrix} \leq \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}$$

we finally get that

$$E_{i+1} = \begin{bmatrix} |\epsilon_{f_{i+1}}| \\ |\epsilon_{oldcs_{i+1}}| \end{bmatrix} \leq \begin{bmatrix} 1 \\ 3 \end{bmatrix} (\sum_{j=1}^{i} |\epsilon_{b_j}| + \sum_{j=1}^{i+1} |\epsilon_{a_j}|) + \begin{bmatrix} 25 \\ 113 \end{bmatrix} \frac{i\varepsilon}{4} \tag{5.8}$$

Now we can bound $|\epsilon_{a'_i}|$ and $|\epsilon_{b'_i}|$. From the algorithm we see for $i \leq n \Leftrightarrow 1$ we have $a'_i = r_{i2}$ so that $\epsilon_{a'_i} = \epsilon_{r_{i2}}$. Substituting the bounds for $|\epsilon_{oldcs_i}|$ and $|\epsilon_{f_i}|$ from (5.8) into the bound for $|\epsilon_{r_{i2}}|$ and simplifying yields

$$|\epsilon_{a'_i}| \leq 4(\sum_{j=1}^{i} |\epsilon_{b_j}| + \sum_{j=1}^{i+1} |\epsilon_{a_j}|) + (35i \Leftrightarrow 20)\varepsilon \quad \text{where} \quad i \leq n \Leftrightarrow 1$$

We also see from the algorithm that $a_n = h_{n-1} * cs_{n-1,2}$, which implies $|\epsilon_{a'_n}| \leq \varepsilon + |\epsilon_{h_{n-1}}| + |\epsilon_{cs_{n-1,2}}|$. Substituting the bounds for $|\epsilon_{oldcs_i}|$ and $|\epsilon_{f_i}|$ from (5.8) into the bounds for $|\epsilon_{h_{n-1}}|$ and $|\epsilon_{cs_{n-1,2}}|$, adding and simplifying, yields

$$|\epsilon_{a'_n}| \leq 5(\sum_{j=1}^{n-1} |\epsilon_{b_j}| + \sum_{j=1}^{n} |\epsilon_{a_j}|) + (41n \Leftrightarrow 58)\varepsilon$$

Next we see from the algorithm that $b_i = oldsn_{i+1} * r_{i+1,1}$ for $i \leq n \Leftrightarrow 2$. Since $oldsn_{i+1} = sn_{i2}$, this implies $|\epsilon_{b'_i}| \leq \varepsilon + |\epsilon_{sn_{i2}}| + |\epsilon_{r_{i+1}}|$. Substituting the bounds for $|\epsilon_{oldcs_i}|$ and $|\epsilon_{f_i}|$ from (5.8) into the bounds for $|\epsilon_{sn_{i2}}|$ and $|\epsilon_{r_{i+1}}|$, simplifying and adding yields

$$|\epsilon_{b'_i}| \leq 6(\sum_{j=1}^{i+1} |\epsilon_{b_j}| + \sum_{j=1}^{i+1} |\epsilon_{a_j}|) + (47n \Leftrightarrow 27)\varepsilon \quad \text{where} \quad i \leq n \Leftrightarrow 2$$

We also see from the algorithm that $b_{n-1} = h_{n-1} * sn_{n-1,2}$, which implies $|\epsilon_{b'_{n-1}}| \leq \varepsilon + |\epsilon_{h_{n-1}}| + |\epsilon_{sn_{n-1,2}}|$. Substituting the bounds for $|\epsilon_{oldcs_i}|$ and $|\epsilon_{f_i}|$ from (5.8) into the bounds for $|\epsilon_{h_{n-1}}|$ and $|\epsilon_{sn_{n-1,2}}|$, simplifying and adding yields

$$|\epsilon_{b'_{n-1}}| \leq 5(\sum_{j=1}^{n-1} |\epsilon_{b_j}| + \sum_{j=1}^{n} |\epsilon_{a_j}|) + (41n \Leftrightarrow 58)\varepsilon$$

Combining the last four displayed inequalities yields claim (5.5) of the lemma.

Next we bound the relative errors in the computed sines and cosines $\epsilon_{sn_{i1}}$, $\epsilon_{sn_{i2}}$, $\epsilon_{cs_{i1}}$ and $\epsilon_{cs_{i2}}$. Substituting the bounds for $|\epsilon_{oldcs_i}|$ and $|\epsilon_{f_i}|$ from (5.8) into the earlier bounds on the $|\epsilon_{sn_{i1}}|$, $|\epsilon_{sn_{i2}}|$, $|\epsilon_{cs_{i1}}|$ and $|\epsilon_{cs_{i2}}|$ and simplifying yields

$$|\epsilon_{sn_{i1}}| \leq \sum_{j=1}^{i} |\epsilon_{b_j}| + \sum_{j=1}^{i} |\epsilon_{a_j}| + (7i \Leftrightarrow 1)\varepsilon$$

$$|\epsilon_{sn_{i2}}| \leq 5(\sum_{j=1}^{i} |\epsilon_{b_j}| + \sum_{j=1}^{i+1} |\epsilon_{a_j}|) + (41i \Leftrightarrow 25)\varepsilon$$

$$|\epsilon_{cs_{i1}}| \leq \sum_{j=1}^{i} |\epsilon_{b_j}| + \sum_{j=1}^{i} |\epsilon_{a_j}| + (7i \Leftrightarrow 1)\varepsilon$$

$$|\epsilon_{cs_{i2}}| \leq 5(\sum_{j=1}^{i} |\epsilon_{b_j}| + \sum_{j=1}^{i+1} |\epsilon_{a_j}|) + (41i \Leftrightarrow 25)\varepsilon$$

Combining the last four inequalities yields

$$\epsilon_{cs,sn} \equiv \max_i(|\epsilon_{sn_{i1}}|, |\epsilon_{sn_{i2}}|, |\epsilon_{cs_{i1}}|, |\epsilon_{cs_{i2}}|) \leq 5(\sum_{j=1}^{n-1} |\epsilon_{b_j}| + \sum_{j=1}^{n} |\epsilon_{a_j}|) + (41n \Leftrightarrow 66)\varepsilon$$

implying (5.6). □

Now we bound the relative error in the computed bidiagonal matrix after m steps of the implicit zero-shift QR algorithm. For reasons which will become clear shortly, we will denote our initial bidiagonal matrix by $B^{(00)}$, the matrix after m steps of the algorithm in exact arithmetic by $B^{(m0)}$, and the matrix after m steps of the algorithm in floating point arithmetic by $B^{(mm)}$. We wish to bound the maximum componentwise relative difference between $B^{(m0)}$ and $B^{(mm)}$, which we measure by

$$reldif(B^{(m0)}, B^{(mm)}) \equiv \max_{ij} |\log \frac{B_{ij}^{(m0)}}{B_{ij}^{(m0)}}|$$

the subscripts i, j varying over the bidiagonal entries.

The proof will use the matrix $M(m_1, m_2)$, which is the Jacobian matrix of the transformation induced by the algorithm in going from the m_1-st to the m_2-nd bidiagonal matrix, but in the variables $\log a_i$ and $\log b_i$. Because small absolute perturbations of the logarithms $\log a_i + \epsilon$ are the same as small relative perturbations of the matrix entries $\log a_i(1 + \epsilon)$,

$M(m_1, m_2)$ measures how relative errors in the m_1-st bidiagonal matrix are propagated to the m_2-nd.

$M(m_1, m_2)$ will be analyzed in detail in section 9, but we need just one result (Theorem 9.23) from that section:

$$\|M(m_1, m_2)\|_\infty \leq (8n \Leftrightarrow 4)(m_2 \Leftrightarrow m_1) + O(1) \tag{5.9}$$

where n is the dimension of the bidiagonal matrix. The numerical simulations in section 11 show that the $O(1)$ can indeed be replaced by zero.

Since the bound (5.9) does not depend on the bidiagonal matrix, it can be used to get global error bounds: if B_1 and B_2 are two bidiagonal matrices, and $B_1(m)$ and $B_2(m)$ are the matrices after m applications of the algorithm in exact arithmetic, then

$$\mathit{reldif}(B_1(m), B_2(m)) \leq ((8n \Leftrightarrow 4)m + O(1)) \cdot \mathit{reldif}(B_1, B_2) \tag{5.10}$$

Lemma 5.11 *In terms of the above notation*

$$\mathit{reldif}(B^{(m0)}, B^{(mm)}) \leq 188n^2m^2\varepsilon + O(nm\varepsilon)$$

Remark 5.12 In practice, as illustrated by the numerical experiments of section 11, the $O(nm\varepsilon)$ term may be replaced by 0. By assuming that the angles encountered in the course of the algorithm are bounded away from $\pi/2$ (which is reasonable, since they approach zero in the limit), the n^2 dependence may be replaced by something proportional to n (see Lemma 8 in [DK88]).

PROOF. Let $B^{(kk)}$ denote the computed bidiagonal matrix after k steps of the algorithm in floating point arithmetic. Starting from $B^{(kk)}$, consider the sequence $B^{(k+1,k)}$, $B^{(k+2,k)}$, ..., $B^{(mk)}$ which would be computed by applying the algorithm in *exact* arithmetic to $B^{(kk)}$. We will bound $\mathit{reldif}(B^{(m0)}, B^{(mm)})$ by

$$\mathit{reldif}(B^{(m0)}, B^{(mm)}) \leq \sum_{k=1}^{m} \mathit{reldif}(B^{(mk)}, B^{(m,k-1)})$$

Now $B^{(mk)}$ and $B^{(m,k-1)}$ arise from applying the algorithm in exact arithmetic to $B^{(kk)}$ and $B^{(k,k-1)}$, respectively. Therefore, by (5.10) the relative difference between them will be bounded by

$$\mathit{reldif}(B^{(mk)}, B^{(m,k-1)}) \leq ((8n \Leftrightarrow 4)(m \Leftrightarrow k) + O(1))\mathit{reldif}(B^{(kk)}, B^{(k,k-1)})$$

But $B^{(kk)}$ is obtained from $B^{(k-1,k-1)}$ by one step of the algorithm in floating point arithmetic, and $B^{(k,k-1)}$ is obtained from $B^{(k-1,k-1)}$ by one step of the algorithm in exact arithmetic. Therefore, $\mathit{reldif}(B^{(kk)}, B^{(k,k-1)})$ is bounded by (5.5) in Lemma 5.4:

$$\mathit{reldif}(B^{(kk)}, B^{(k,k-1)}) \leq (47n \Leftrightarrow 27)\varepsilon$$

Combining the last three displayed equations yields the claimed result. □

We note that the method of proof is analogous to the way error bounds are derived for computed solutions of initial value problems for differential equations: The truncation error at each step is analogous to our one step error bounded in Lemma 5.4. Then the differential equation being solved is used to propagate the error bound for the truncation error forward; this is exactly what we are doing.

The next lemma shows that if the maximum error $\epsilon_{cs,sn}$ in the computed sines and cosines is small, so will be the error in the singular vectors computed by $UPDATE$:

Lemma 5.13 *Let V be an n by n orthogonal matrix, and V' the updated orthogonal matrix obtained by running one step of the implicit zero-shift QR algorithm in exact arithmetic. Let δV be a perturbation of V, and let $V' + \delta V'$ be the matrix obtained by running the algorithm in floating point on $V + \delta V$, where we assume the relative errors in the computed sines and cosines are bounded by $\epsilon_{cs,sn}$. Then to first order in $\epsilon_{cs,sn}$, ε, and $\|\delta V\|_2$*

$$\|\delta V'\|_2 \le 2^{3/2}(n \Leftrightarrow 1)\varepsilon + 2^{1/2}(n \Leftrightarrow 1)\epsilon_{cs,sn} + \|\delta V\|_2$$

Proof. It suffices to analyze the errors from one call to $UPDATE$. Let cs and sn be the true values of the cosine and sine, and $cs(1+\epsilon_{cs})$ and $sn(1+\epsilon_{sn})$ the perturbed values. Let v_1 and v_2 denote the two columns of V being modified. Then their true new values are

$$[v_1', v_2'] = [v_1, v_2] \cdot \left[\begin{array}{cc} cs & \Leftrightarrow sn \\ sn & cs \end{array} \right]$$

Let $[\delta v_1, \delta v_2]$ be the perturbation of $[v_1, v_2]$ due to all previous contributions. Then the j-th components of the new perturbation after computing can be written (to first order in $\epsilon_{cs,sn}$, ε and $\|\delta V\|_2$) as

$$[\delta v_{1j}', \delta v_{2j}'] = [v_{1j}, v_{2j}] \cdot \left[\begin{array}{cc} cs(\epsilon_{cs} + \varepsilon_1 + \varepsilon_2) & \Leftrightarrow sn(\epsilon_{sn} + \varepsilon_3 + \varepsilon_4) \\ sn(\epsilon_{sn} + \varepsilon_5 + \varepsilon_6) & cs(\epsilon_{cs} + \varepsilon_7 + \varepsilon_8) \end{array} \right] + [\delta v_{1j}, \delta v_{2j}] \cdot \left[\begin{array}{cc} cs & \Leftrightarrow sn \\ sn & cs \end{array} \right]$$

where $|\varepsilon_i| \le \varepsilon$. Thus

$$\|[\delta v_1', \delta v_2']\|_2 \le 2^{1/2}(\epsilon_{cs,sn} + 2\varepsilon) + \|[\delta v_1, \delta v_2]\|_2$$

and applying this bound $n \Leftrightarrow 1$ times (for each call to $UPDATE$) we get the result claimed in the lemma. □

Lemma 5.14 *Let B be an n by n bidiagonal matrix, and let V and U be the orthogonal matrices obtained by running the implicit zero-shift QR algorithm m times in exact arithmetic. In other words, set V and U to n by n identity matrices initially, and let them be modified by the calls to $UPDATE$ in the algorithm. Now let $\hat{V}$ and $\hat{U}$ be the floating point matrices obtained by running the algorithm in arithmetic of precision ε. Then to first order in ε we have*

$$\max(\left\|\hat{V} \Leftrightarrow V\right\|_2, \left\|\hat{U} \Leftrightarrow U\right\|_2) \le 947 n^4 (m^3 + m)\varepsilon + O(\varepsilon n^3 m^2) \tag{5.15}$$

Remark 5.16 In the numerical experiments of section 11, the $O(\varepsilon n^3 m^2)$ term above may be replaced by 0. By assuming that the angles encountered in the course of the algorithm are bounded away from $\pi/2$ (which is reasonable, since they approach zero in the limit), the n^4 in the error bound may be replaced by something proportional to n^2 (see Lemma 8 in [DK88]).

Proof. By Lemma 5.13, after each application of the algorithm the error in V increases by

$$\sqrt{2}(n \Leftrightarrow 1)(\epsilon_{cs,sn} + 2\epsilon)$$

By Lemma 5.4, at the k-th stage $\epsilon_{cs,sn}$ is bounded by

$$|\epsilon_{cs,sn}| \leq 5(2n \Leftrightarrow 1) reldif(B^{(kk)}, B^{(k0)}) + (41n \Leftrightarrow 66)\varepsilon$$

By Lemma 5.11, $reldif(B^{(kk)}, B^{(k0)})$ is bounded by

$$reldif(B^{(kk)}, B^{(k0)}) \leq \varepsilon(188n^2 \Leftrightarrow 282n + 54)(k^2 \Leftrightarrow k + 1) + O(nk\varepsilon)$$

Combining the last three displayed expressions yields

$$\left\| V \Leftrightarrow \hat{V} \right\|_2 \leq \sum_{k=1}^{m} \sqrt{2}(n \Leftrightarrow 1) \cdot (5(2n \Leftrightarrow 1)[\varepsilon(188n^2 \Leftrightarrow 282n + 54)(k^2 \Leftrightarrow k + 1) + O(nk\varepsilon)] + (41n \Leftrightarrow 66)\varepsilon + 2\varepsilon)$$

which, when simplified, yields the desired result. □

6 Global Error Bounds on the Computed Singular Vectors

In this section we present the main practical contribution of the paper: an error bound for singular vectors computed by the overall bidiagonal SVD algorithm:

Theorem 6.1 *The error in the computed the i-th left and right singular vectors is*

$$p(n,m) tol/relgap_i$$

where tol is the desired relative precision input to the algorithm, $p(n,m)$ is a low degree polynomial in the matrix dimension n and number of QR steps m, and the relative gap $relgap_i$ was defined in 1.7.

Remark. If we were to apply straightforward error estimates without using the machinery of this paper, the resulting error bounds would grow exponentially in n and m instead of polynomially.

PROOF. To perform the error analysis we need to associate a tree with the execution of the algorithm. Nodes of the tree will correspond to unreduced submatrices B_i upon which the algorithm operates. The root node corresponds to the original matrix and the leaf nodes correspond to 1 by 1 and 2 by 2 submatrices where the algorithm has converged (recall that 2 by 2 matrices are handled specially). A directed edge from node B_i to node B_j will mean that B_j is obtained from B_i by performing some step of the algorithm. There are three kinds of edges: "stopping" edges, "zero-shift QR" edges, and "shifted QR" edges. Stopping edges correspond to the stopping criterion deciding to set one or more offdiagonal entries of B_i to zero. In this case B_j is a submatrix of B_i. If the stopping criterion sets p offdiagonal entries of B_i to zero at the same time, B_i will have $p+1$ child nodes, one for each resulting submatrix. A zero-shift QR edge corresponds to one or more applications of the zero-shift QR algorithm. A zero-shift QR edge connecting B_i to B_j represents all the zero-shift QR steps applied to B_i before the stopping criterion is satisfied. Finally, if the sequence of QR steps by which B_j is obtained from B_i includes at least one shifted QR step, then B_i is connected to B_j by a shifted QR edge. Normally a shifted QR edge will represent only shifted QR steps. However, our estimates $\underline{\sigma}(B_i)$ and $\overline{\sigma}(B_i)$ of the smallest and largest singular values of B_i, which are used to choose between shifted QR and zero-shift QR, are not perfect, so there is a chance the algorithm could apply both kinds of QR steps to the same submatrix. As we will see, this does not impact the error analysis.

Thus, any path from the root of the tree to a leaf node starts with a QR edge of either type, and then alternates between stopping edges and QR edges until finally hitting a leaf node at the end of a stopping edge. A node can have at most one entering edge (stopping or QR), and either one exiting edge (which must be a QR edge) or more than one exiting edge (which must be stopping edges).

The proof of the theorem proceeds by induction from the leaves of the tree toward the root. We will show that if the computed singular vectors of all the children of a parent node satisfy the error bound $O(tol/relgap_k)$, then so do the computed singular vectors of the parent node. First we explain the induction without computing detailed error estimates, and then we include the error estimates. In the proof the notation $O(\cdot)$ will absorb all dependence on dimension and number of QR steps.

First consider the leaf nodes, which are all 1 by 1 and 2 by 2. There is nothing to prove for the 1 by 1 nodes, and for 2 by 2 nodes the special subroutine discussed in section 3 computes the singular vectors with the desired error bounds.

Now Suppose B_i is the first node on the path from the root whose exiting edge is a shifted QR edge. In other words, only zero-shifted QR steps have been applied to B_i so far. Starting with B_i, the algorithm essentially reverts to the standard shifted SVD algorithm, which is backward stable in the usual normwise sense and so computes the k-th singular vectors of B_i with an error bound $O(tol/absgap_k)$, where $absgap_k = \min_l |\sigma_k(B_i) \Leftrightarrow \sigma_l(B_i)|/\sigma_1(B_i)$. From the algorithm of section 3, we see that shifted QR is used only when $\underline{\sigma}(B_i)/\overline{\sigma}(B_i) > n^{-1}\max(\varepsilon/tol, .01) > .01/n$, i.e. only when the smallest singular value of B_i is not more than about $.01/n$ times smaller than the largest singular value. This implies that the relative gap $relgap_k = \min_l |\sigma_k(B_i) \Leftrightarrow \sigma_l(B_i)|/|\sigma_k(B_i) + \sigma_l(B_i)|$ cannot be more than about $200n$ times larger than the absolute gap $absgap_k$. Thus the error bound for the k-th computed singular vectors of B_i are still $O(tol/relgap_k)$ as desired.

(This is where we use the modification of the original algorithm from [DK88]. If the threshold for using shifted QR had been $n^{-1}\varepsilon/tol$ instead of $n^{-1}\max(\varepsilon/tol, .01)$, the error bound would have been $O(tol^2/(\varepsilon \cdot relgap_k))$ instead of $O(tol/relgap_k)$. For tol just slightly larger than ε there is no difference, but for tol approaching $\varepsilon^{1/2}$ the bound $O(tol^2/(\varepsilon \cdot relgap_k))$ is significantly weaker than $O(tol/relgap_k)$.)

At this point in the induction we have shown that the error bounds for computed singular vectors are $O(\varepsilon/relgap_k)$ for all nodes which only have zero-shift QR edges and stopping edges between them and the root. First consider stopping edges. Suppose B_i is the parent node from which the stopping edges issue. By Theorem 4.1, the stopping criterion only changes the singular vectors by $O(tol/relgap_k)$, where $relgap_k$ is measured with respect to the singular values of B_i only. This $relgap_k$ may be larger than the $relgap_k$ measured with respect to the entire matrix (since B_i contains only a subset of the spectrum of the original matrix), but this only improves the error bound $O(tol/relgap_k)$. This lets us moves the induction toward the root along stopping edges.

Finally, consider a zero-shift QR edge from B_i to B_j. Let $true(B_j)$ denote the matrix that would have been computed from B_i in exact arithmetic, and U and V the orthogonal matrices that transform $true(B_j)$ to B_i: $U^T \cdot true(B_j) \cdot V = B_i$. We will consider right singular vectors only; the proof for left singular vectors is identical. Let the notation $true\_vector_k(B)$ denote the true k-th right singular vector of the matrix B, and let $comp\_vector_k(B)$ denote the computed k-th right singular vector of B. We want to show that $\|true\_vector_k(B_i) \Leftrightarrow comp\_vector_k(B_i)\|_2 = O(tol/relgap_k)$. Note that $true\_vector_k(B_i) = V^T \cdot true\_vector_k(true(B_j))$ and $comp\_vector_k(B_i) = \hat{V}^T \cdot comp\_vector_k(B_j)$, where the difference between V and $\hat{V}$ is bounded in Lemma 5.14. Consider the following triangle inequality:

$$\begin{aligned}
&\|true\_vector_k(B_i) \Leftrightarrow comp\_vector_k(B_i)\|_2 \\
&\quad = \left\| V^T \cdot true\_vector_k(true(B_j)) \Leftrightarrow \hat{V}^T \cdot comp\_vector_k(B_j) \right\|_2 \\
&\quad \le \left\| V^T \cdot true\_vector_k(true(B_j)) \Leftrightarrow \hat{V}^T \cdot true\_vector_k(true(B_j)) \right\|_2 + \\
&\qquad \left\| \hat{V}^T \cdot true\_vector_k(true(B_j)) \Leftrightarrow \hat{V}^T \cdot true\_vector_k(B_j) \right\|_2 + \qquad (6.2)
\end{aligned}$$

$$\begin{aligned}&\left\|\hat{V}^T \cdot true\_vector_k(B_j) \Leftrightarrow \hat{V}^T \cdot comp\_vector_k(B_j)\right\|_2 \\ \equiv\ & I_1 + I_2 + I_3\end{aligned}$$

By Lemma 5.14, $I_1 = O(\varepsilon)$. By Theorem 2.12 and Lemma 5.11, $I_2 = O(\varepsilon/relgap_k)$. By the induction hypothesis, $I_3 = O(tol/relgap_k)$. Thus, the sum $I_1 + I_2 + I_3 = O(tol/relgap_k)$ as desired. This completes the induction.

Now we consider more rigorous bounds. Explicit formulas for such bounds would be quite complicated and pessimistic and shed little new light on the problem. However, it is illuminating to use the tree to explain how errors accumulate. We use the fact that if V_1 and V_2 are orthogonal, and δV_1 and δV_2 are small perturbations, then to first order $\|V_1V_2 \Leftrightarrow (V_1 + \delta V_1)(V_2 + \delta V_2)\|_2 \leq \|\delta V_1\|_2 + \|\delta V_2\|_2$. This means that to first order perturbation bounds simply add as we proceed up the tree. Thus, every time we move along an edge, we add the error contributed by that edge. For leaf nodes which correspond to 2 by 2 submatrices, we use error bounds for the special subroutine discussed in section 3. For stopping edges we used the bounds of Theorem 4.1. For nodes whose exiting edge is the first shifted QR edge, we can use the error bounds for the conventional SVD algorithm [GK65]. For zero-shift QR edges we use the analysis of the last paragraph. Only the edges connecting a leaf to the root contribute to the error for the singular vector corresponding to the leaf. □

7 Flows and the SVD Algorithm

For a matrix A, let A_- denote its strictly lower triangular part and set $\pi_0(A) = A_- \Leftrightarrow A_-^T$. We will consider flows on invertible real matrices A of the form

$$\frac{dA}{dt} = A(\pi_0(F(A^T A))) \Leftrightarrow (\pi_0(F(A\,A^T)))A\ , \qquad A(t=0) = A_0\ , \tag{7.1}$$

where F is a smooth, real-valued function on $(0,\infty)$.

Such flows were first considered in the singular value context by M. Chu [Chu], who analyzed the Toda case, $F(x) = x$. For SVD we will set $F(x) = \log x$, but initially, for reasons of general interest and to suggest additional possibilities, we will consider general F.

Convention. By the QR factorization of a real, invertible matrix X, we mean $X = QR$, where Q is orthogonal and R is upper triangular with positive diagonal entries (see [GVL83]).

Theorem 7.2 *Equation (7.1) has a unique, global solution $A(t)$ which preserves the singular values of $A(t)$. The flow can be solved explicitly, as follows. Let*

$$e^{t\,F(A_0^T A_0)} = Q_1(t)\ R_1(t) \tag{7.3}$$

$$e^{t\,F(A_0 A_0^T)} = Q_2(t)\ R_2(t) \tag{7.4}$$

be the QR-factorizations of $e^{tF(A_0^T A_0)}$ and $e^{tF(A_0 A_0^T)}$ respectively. Then

$$A(t) = Q_2^T(t)\ A_0\ Q_1(t)\ . \tag{7.5}$$

Finally, (7.1) preserves bidiagonality i.e. if A_0 is bidiagonal, then $A(t)$ is bidiagonal for all $t > 0$. Moreover the signs of its nonzero entries are preserved.

Proof. Differentiation of (7.3) and (7.4), give

$$\frac{d}{dt}Q_1 = Q_1\,\pi_0(Q_1^T F(A_0^T A_0)Q_1) = Q_1\,\pi_0(F((Q_2^T A_0 Q_1)^T (Q_2^T A_0 Q_1))) = Q_1 \pi_0(F(A(t)^T A(t)))$$

and

$$\frac{d}{dt}Q_2 = Q_2 \pi_0(Q_2^T\,F(A_0 A_0^T)Q_2) = Q_2 \pi_0(F((Q_2^T A_0 Q_1)(Q_2^T A_0 Q_1)^T)) = Q_2 \pi_0(F(A(t)\,A(t)^T))\,.$$

Thus

$$\begin{aligned}\frac{d}{dt}A(t) &= [\Leftrightarrow \pi_0(F(A(t)\ A(t)^T))\ Q_2^T]\ A_0 Q_1 + Q_2^T A_0 [Q_1 \pi_0(F(A(t)^T A(t)))] \\ &= A(t)\pi_0(F(A(t)^T A(t))) \Leftrightarrow \pi_0(F(A(t)A(t)^T))\ A(t)\ ,\end{aligned}$$

which is equation (7.1). From (7.3), (7.4), (7.5),

$$\begin{aligned} A(t) &= (R_2(t)\ e^{-t\,F(A_0 A_0^T)})\ A_0\ (e^{t\,F(A_0^T A_0)}(R_1(t))^{-1}) \\ &= R_2(t)\ A_0 (R_1(t))^{-1}\ , \end{aligned} \tag{7.6}$$

from which we learn that (7.1) preserves upper triangularity. On the other hand, we also have

$$\begin{aligned} A^T(t)\ A(t) &= Q_1(t)^T\ A_0^T A_0 Q_1(t) \\ &= R_1(t)\ e^{-tF(A_0^T A_0)} A_0^T A_0\ e^{t\,F(A_0^T A_0)}(R_1(t))^{-1} \\ &= R_1(t)\ A_0^T A_0 (R_1(t))^{-1}\ , \end{aligned} \tag{7.7}$$

so that $A^T(t)\,A(t)$ is upper Hessenberg, and hence tridiagonal, by symmetry. It follows that $A(t)$ is bidiagonal. Furthermore, it follows from (7.6) and (7.7) that

$$\text{sgn}\ A_{ii}(t) = \text{sgn}(A_0)_{ii}\ , \quad \text{sgn}\ A_{i\,i+1}(t) = \text{sgn}(A_0)_{i\,i+1}\ . \tag{7.8}$$

In particular, if a_p and b_q are positive initially, they are positive for all time.

Finally, the preservation of singular values is immediate from (7.5), and this proves the theorem. □

The above result is due to Chu [Chu86], and is modeled on related results ([Sym80], [Sym82]) and Deift-Nanda-Tomei [DNT83], for the symmetric eigenvalue problem. The relationship between the singular value flow (7.1) and Toda-type eigenvalue flows ([Sym80], [Sym82], [DNT83], [DLNT86], [DLT89], [Wat84]), is described by the following theorem, whose proof is immediate.

Theorem 7.9 *Under the map*

$$A \mapsto T(A) = A^T A \tag{7.10}$$

equation (7.1) is transformed into

$$\frac{dT}{dt} = [T,\ \pi_0(F(T))]\ . \tag{7.11}$$

□

Remark 7.12 The perfect shuffle $A \mapsto \begin{pmatrix} 0 & A^T \\ A & 0 \end{pmatrix} \mapsto S(A)$ of Section 2, transforms the singular value problem for A into an eigenvalue problem for S. One might ask what happens to (7.1) under the map $A \mapsto S$. One finds that (7.1) is transformed into the Toda-type isospectral deformation

$$\frac{dS}{dt} = [S,\ \pi_0(F(S^2))]\ . \tag{7.13}$$

Of particular interest is the case $F(x) = x$. Here (7.11) becomes the Toda flow, $\frac{dT}{dt} = [T, \pi_0(T)]$, but (7.13) reduces to $\frac{dS}{dt} = [S, \pi_0(S^2)]$; in the case where A is bidiagonal and S takes the form (2.2), this is the so-called Kac-van Moerbeke lattice [KvM75].

The flow that is directly related to the SVD algorithm corresponds to the choice $F(x) = \log x$. To see this note that (7.11) becomes

$$\frac{dT}{dt} = [T, \pi_0(\log T)]\ , \qquad T(0) \equiv A_0^T A_0\ ,$$

whose solution is given by (7.7),

$$T(t) = Q_1^T(t)\ T(0)\ Q_1(t)\ .$$

But from (7.3),

$$T(0) = e^{\log T(0)} = Q_1(1)\ R_1(1)\ ,$$

and so

$$T(1) = R_1(1)\ Q_1(1)\ .$$

Thus

$$A^T(0)\ A(0) = T(0) \mapsto T(1) = A(1)^T A(1)$$

is just one step of QR, and hence, in the bidiagonal case,

$$A(0) \mapsto A(1)$$

is one step of SVD.

In summary, we have proven the following basic result.

Theorem 7.14 *Let A_0 be bidiagonal and let $F(x) = \log x$. Then the integer time evaluation of the solution $A(t)$ of (7.1) gives precisely the iterates of the SVD algorithm, $A_0, A_1, \ldots, A_k, \ldots$. Thus*

$$A(k) = A_k\ , \qquad k = 0, 1, 2, \ldots\ . \tag{7.15}$$

We will call the flow induced by (7.1) in the case $F(x) = \log x$, the *SVD flow*.

In the classical case of the Toda flow, where T is tridiagonal and $F(x) = x$, Moser [Mos75] proved the remarkable result that the solution $T(t)$ of (7.11) converges to a diagonal matrix as $t \to \infty$. The same is true if T is a full symmetric matrix, provided $F(x)$ is strictly monotonic on spec T (see, e.g., [DLT85]). In the bidiagonal case, the convergence of $T(t) = A(t)^T A(t)$ in turn implies that $A(t)$ also converges to a diagonal matrix (see [Chu86]). For the reader's convenience we will present a (new) proof of the convergence of $A(t)$, calculating en route the leading asymptotics as $t \to \infty$. By Theorem 7.14, this of course gives an independent proof of the convergence of SVD and, by Theorem 7.9, also QR.

To fix notation, let

$$A = B = \begin{pmatrix} a_1 & b_1 & & \bigcirc \\ & \ddots & \ddots & \\ & & & b_{n-1} \\ \bigcirc & & & a_n \end{pmatrix}\ , \quad a_i, b_i > 0\ ,$$

with singular values $\sigma_1 > \sigma_2 > \ldots > \sigma_n > 0$, and let

$$T = B^T B = \begin{pmatrix} c_1 & d_1 & & \bigcirc \\ d_1 & & \ddots & \\ & \ddots & \ddots & d_{n-1} \\ \bigcirc & & d_{n-1} & c_n \end{pmatrix}\ ,$$

so that

$$c_i = a_i^2 + b_{i-1}^2 \ , \qquad 1 \le i \le n \ , \tag{7.16}$$

$$d_i = a_i b_i \ , \qquad 1 \le i \le n \Leftrightarrow 1 \ . \tag{7.17}$$

(Here $b_0 \equiv 0$.) Let $v_i = (v_i(1), \ldots, v_i(n))^T$, $1 \le i \le n$, denote the unit right singular vectors for B, $Tv_i = \sigma_i^2 v_i$, normalized so that $v_i(1) > 0$. Set $b^2 = \sum_{i=1}^{n-1} b_i^2$.

Theorem 7.18 *(Asymptotics for SVD). Let $B(t)$ be the solution of (7.1) with $B(0) = B_0$ bidiagonal and $F(x)$ increasing on spec $B^T B$. Then as $t \to \infty$*

$$a_j(t) = \sigma_j \left(1 + \frac{b_j^2}{2(\sigma_{j+1}^2 \Leftrightarrow \sigma_j^2)}(1 + O(b^2)) + \frac{b_{j-1}^2}{2(\sigma_{j-1}^2 \Leftrightarrow \sigma_j^2)}(1 + O(b^2)) \right) \tag{7.19}$$

and

$$b_j(t) \sim \frac{1}{\sigma_j} [\frac{\prod_{k \le j}(\sigma_k^2 \Leftrightarrow \sigma_{j+1}^2)}{\prod_{k \le j-1}(\sigma_k^2 \Leftrightarrow \sigma_j^2)}] \ [\frac{v_{j+1}(1)}{v_j(1)}] \, e^{(F(\sigma_{j+1}^2) - F(\sigma_j^2))t} \tag{7.20}$$

(Here $\prod_{k \le j-1}(\sigma_k^2 \Leftrightarrow \sigma_j^2) \equiv 1$ for $j = 1$).

Proof. From (7.7) and (7.3),

$$\begin{aligned} v_j(1,t) &= (e_1, Q_1^T(t)\, v_j) \\ &= ((R_1(t))^{-1} e_1, \ e^{tF(T_0)} v_j) \\ &= \frac{e^{tF(\sigma_j^2)}}{(R_1(t))_{11}} v_j(1) \ , \end{aligned}$$

and hence

$$v_j(1,t) = \frac{e^{tF(\sigma_j^2)} v_j(1)}{(\sum_{m=1}^n e^{2tF(\sigma_m^2)} \ v_m^2(1))^{1/2}}. \tag{7.21}$$

In particular, as $F(x)$ is increasing,

$$v_j(1,t) \to \delta_{j1} \tag{7.22}$$

as $t \to \infty$.

Now recall (see e.g. [GvL]) that the rows

$$\begin{matrix} v_1(1) & v_2(1) & \cdots & v_n(1) \\ v_1(2) & v_2(2) & \cdots & v_n(2) \\ & \vdots & & \end{matrix}$$

of the matrix of eigenvectors V for the tridiagonal matrix T can be computed by applying the Gram-Schmidt procedure to the row vectors

$$\begin{matrix} v_1(1) & v_2(1) & \cdots & v_n(1) \\ \sigma_1^2 v_1(1) & \sigma_2^2 v_2(1) & \ldots & \sigma_n^2 v_n(1) \\ & \vdots & & \\ \sigma_1^{2(k-1)} v_1(1) & \sigma_2^{2(k-1)} v_2(1) & \ldots & \sigma_n^{2(k-1)} v_n(1) \ . \end{matrix}$$

In particular if $v(1,t) = (v_1(1,t),\ldots,v_n(1,t))$, $v(2,t) = (v_1(2,t),\ldots,v_n(2,t))$ denote the first two rows of the matrix of eigenvectors $V(t)$ for $T(t)$; then

$$1 = (v(1,t)\wedge v(2,t),\ v(1,t)\wedge v(2,t)) = \sum_{i<k}\left(\begin{vmatrix} v_i(1,t) & v_k(1,t) \\ v_i(2,t) & v_k(2,t)\end{vmatrix}\right)^2 \ .$$

But by Gram-Schmidt $v_m(2,t) = \lambda_1(t)\sigma_m^2 v_m(1,t) + \lambda_2(t)v_m(1,t)$ where $\lambda_1(t)$ and $\lambda_2(t)$ are independent of m, $\lambda_1(t) \neq 0$. Using (7.21) we find

$$\frac{\begin{vmatrix} v_i(1,t) & v_k(1,t) \\ v_i(2,t) & v_k(2,t)\end{vmatrix}^2}{\begin{vmatrix} v_1(1,t) & v_2(1,t) \\ v_1(2,t) & v_2(2,t)\end{vmatrix}^2} = \frac{\begin{vmatrix} 1 & 1 \\ \sigma_i^2 & \sigma_k^2\end{vmatrix}^2}{\begin{vmatrix} 1 & 1 \\ \sigma_1^2 & \sigma_2^2\end{vmatrix}^2}\left(\frac{v_i(1)v_k(1)}{v_1(1)v_2(1)}\right)^2 e^{2[F(\sigma_i^2)+F(\sigma_k^2)-F(\sigma_1^2)-F(\sigma_2^2)]t} \ ,$$

which, by the monotonicity of F, converges to zero as $t \to \infty$, unless $i = 1$ and $k = 2$. Thus

$$\begin{vmatrix} v_1(1,t) & v_2(1,t) \\ v_1(2,t) & v_2(2,t)\end{vmatrix} \Leftrightarrow\!\!\to \pm 1$$

and as $|v_1(2,t)| \leq 1$ and $v_2(1,t) \to 0$ by (7.22), we conclude that

$$v_j(2,t) \to \pm\delta_{j2} \tag{7.23}$$

Continuing by induction we learn that

$$v_j(k,t) \to \pm\delta_{jk} \tag{7.24}$$

as $t \to \infty$, and hence

$$T(t) = V(t)\ \Sigma^2 V(t) \to \Sigma^2 = \mathrm{diag}(\sigma_1^2,\ldots,\sigma_n^2)\ . \tag{7.25}$$

In particular $d_i(t) = a_i(t)\ b_i(t) \to 0$. But det $T(t) = (\det\ A(t))^2 = \prod_{i=1}^n a_i^2(t) =$ constant $\neq 0$, as (7.11) is isospectral, and $\max_i a_i(t)$ is bounded by (7.5), which implies $\|A(t)\| = \|A_0\|$. It follows that $b_i(t) \to 0$, and from (7.17) (and (7.8)),

$$a_i(t) = \sqrt{c_i(t) \Leftrightarrow b_{i-1}^2(t)} \to \sigma_i \ .$$

The eigenvalue equation for $T(t)$ implies

$$v_i(k+1,t) = \frac{v_i(1,t)}{(\prod_{m=1}^k d_m)}\det(\sigma_i^2 \Leftrightarrow T(t))_k \tag{7.26}$$

where $(\sigma_i^2 \Leftrightarrow T(t))_k$ is $k \times k$ matrix formed from the first k rows and columns of $\sigma_i^2 \Leftrightarrow T(t)$. In particular, setting $i = k+1$ and using (7.21), (7.24), (7.25) in (7.26), we find

$$\prod_{m=1}^k d_m(t) \sim \pm\frac{v_{k+1}(1)}{v_1(1)}\left(\prod_{m=1}^k (\sigma_{k+1}^2 \Leftrightarrow \sigma_m^2)\right) e^{(F(\sigma_{k+1}^2)-F(\sigma_1^2))t} \ ,$$

which leads directly to (7.20).

To obtain the asymptotics for $a_i(t)$, conjugate the eigenvalue equation for T using a diagonal matrix to the form

$$\begin{pmatrix} c_1 \Leftrightarrow \sigma_i^2 & d_1^2 & & & \\ 1 & (c_2 \Leftrightarrow \sigma_i^2) & d_2^2 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & (c_{n-1} \Leftrightarrow \sigma_i^2) & d_{n-1}^2 \\ & & & 1 & (c_n \Leftrightarrow \sigma_i^2) \end{pmatrix} \begin{pmatrix} \tilde{v}_i(1) \\ \\ \vdots \\ \\ \tilde{v}_i(1) \end{pmatrix} = 0 \ .$$

Applying Cramer's rule to the leading $(i \Leftrightarrow 1)\times(i \Leftrightarrow 1)$ matrix and to the trailing $(n \Leftrightarrow i)\times(n \Leftrightarrow i)$ matrix, one sees easily that

$$\tilde{v}_i(i \Leftrightarrow 1)/\tilde{v}_i(i) = \frac{d_{i-1}^2}{\sigma_i^2 \Leftrightarrow \sigma_{i-1}^2}(1 + O(\sum_j d_j^2))$$

and

$$\tilde{v}_i(i+1) \ / \ \tilde{v}_i(i) = \frac{1}{\sigma_i^2 \Leftrightarrow \sigma_{i+1}^2}(1 + O(\sum_j d_j^2)) \ ,$$

respectively. Inserting these relations in the i^{th} equation

$$c_i \Leftrightarrow \sigma_i^2 = \Leftrightarrow \frac{\tilde{v}_i(i \Leftrightarrow 1)}{\tilde{v}_i(i)} \Leftrightarrow d_i^2 \frac{\tilde{v}_i(i+1)}{\tilde{v}_i(i)} \ ,$$

and using (7.16), the result follows. □

Remark 7.27 From (7.24), $v_j(k,t) \to \pm\delta_{jk}$ as $t \to \infty$. The choice of signs can be determined from (7.26). Indeed as $t \to \infty$, $v_{k+1}(k+1,t) \sim (\text{pos.}) \times \prod_{i=1}^{k}(\sigma_{k+1}^2 \Leftrightarrow \sigma_i^2)$, and so

$$v_j(k,t) \to (\Leftrightarrow 1)^{k+1}\delta_{jk} \ .$$

Remark 7.28 The form of (7.19) suggests a proof using more standard techniques in numerical analysis. Indeed (7.19) can easily be proved by applying two consecutive Jacobi rotations in the planes $(j \Leftrightarrow 1, j)$ and $(j, j+1)$ respectively, and then using second order perturbation theory.

Remark 7.29 Note that a linearization of (7.1) around the equilibrium point $A(\infty) = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$, would also give the asymptotic rates (7.19) and (7.20), but without the precise constants.

Remark 7.30 The asymptotics in (7.20) can be used to calculate the scattering matrix for the classical Toda lattice (see [Mos75]) directly.

8 The Hamiltonian Structure for the Flows

In this section we show that equation (7.1) is Hamiltonian. More precisely, we show that there is a Poisson bracket $\{\cdot,\cdot\}_S$, the *Sklyanin bracket*, defined on the space of matrices, and a Hamiltonian H_F, such that the differential equations generated by $(H_F, \{\cdot,\cdot\}_S)$,

$$\frac{d}{dt}\phi(A(t)) = \{\phi, H_F\}_S(A(t)) \ , \qquad A(0) = A_0 \ , \tag{8.1}$$

for all observables ϕ, are equivalent to (7.1).

A general reference for the Hamiltonian mechanics used in this paper is [Arn78]; a description of the Sklyanin bracket can be found in [Sem84]. The Sklyanin bracket can be defined in great generality on groups and on associative algebras (see also [LPar]), but we will restrict ourselves to the case where the underlying space is $M_n(\mathrm{R})$, the algebra of real $n \times n$ matrices.

We begin with some notation and definitions. The space $M_n(\mathrm{R})$ carries a natural ($gl(n,\mathrm{R})$-ad-) invariant pairing

$$(A, B) \equiv \mathrm{tr}\ AB \ , \tag{8.2}$$

$$(A, [B, C]) = \Leftrightarrow([B, A], C) \ , \tag{8.3}$$

where A, B, C belong to $M_n(\mathrm{R})$ and $[\cdot,\cdot]$ denotes the standard commutator. Denote by X_a (resp. $\widetilde{X}_a$) the ($Gl(n,\mathrm{R})$-) left-invariant (($Gl(n,\mathrm{R})$-) right-invariant, resp.) vector field generated by $a \in M_n(\mathrm{R})$. Thus for smooth functions $\phi : M_n(\mathrm{R}) \to \mathrm{R}$,

$$X_a\phi(g) = \frac{d}{dt}|_{t=0}\phi(g\, e^{ta}) \equiv (D'\phi(g), a) \tag{8.4}$$

$$\widetilde{X}_a\phi(g) = \frac{d}{dt}|_{t=0}\phi(e^{ta}g) \equiv (D\,\phi(g), a) \tag{8.5}$$

for all $g \in M_n(\mathrm{R})$. Clearly

$$D'\phi(g) = (\nabla\phi(g))^T g \tag{8.6}$$

$$D\phi(g) = g(\nabla\phi(g))^T \ , \tag{8.7}$$

where $\nabla\phi(g)$ is the matrix with entries $\frac{\partial\phi}{\partial g_{ij}}(g)$. Standard computations for Lie brackets show that

$$[X_a, X_b] = X_{[a,b]} \tag{8.8}$$

$$[\widetilde{X}_a, \widetilde{X}_b] = \Leftrightarrow\widetilde{X}_{[a,b]} \ , \tag{8.9}$$

and

$$[X_a, \widetilde{X}_b] = 0 \ . \tag{8.10}$$

Finally, a linear map $R : M_n(\mathrm{R}) \to M_n(\mathrm{R})$ is said to solve the *modified Yang-Baxter* equation (mYB) if

$$[R(A), R(B)] \Leftrightarrow R([A, R(B)] + [R(A), B]) = \Leftrightarrow[A, B] \tag{8.11}$$

for all $A, B \in M_n(\mathrm{R})$. (Such an R is an example of a *classical r-matrix* — see [Sem84]).

Theorem 8.12 *Suppose R and R' are skew symmetric*

$$(A, R(B)) = \Leftrightarrow(R(A), B) \ , \qquad (A, R'(B)) = \Leftrightarrow(R'(A), B) \tag{8.13}$$

solutions of (mYB). Then

$$\{\phi, \psi\}_{R,R'}(g) \equiv (R(D\phi(g)),\, D\psi(g)) + (R'(D'\phi(g)),\, D'\psi(g)) \tag{8.14}$$

defines a Poisson bracket on $M_n(\mathbf{R})$.

Proof. The only point to check is the Jacobi identity, $\{\phi_1, \{\phi_2, \phi_3\}_{R,R'}\}_{R,R'}+$ cyclic permutations (c.p.) $= 0$. Set

$$\begin{aligned}\{\phi, \psi\}_r(g) &\equiv (R(D\phi(g)),\, D\psi(g))\\ \{\phi, \psi\}_\ell(g) &\equiv (R'(D'\phi(g),\, D'\psi(g)) \ .\end{aligned}$$

We have

$$\begin{aligned}(D'\{\phi_2, \phi_3\}_\ell(g), a) &= \frac{d}{dt}|_{t=0}\{\phi_2, \phi_3\}_\ell(g\, e^{ta})\\ &= (R'(D'\phi_2(g)),\, \frac{d}{dt}|_{t=0}D'\phi_3(g\, e^{ta})) \Leftrightarrow(\frac{d}{dt}|_{t=0}D'\phi_2(ge^{ta}),\, R'(D'\phi_3(g)))\\ &= X_a X_{R'(D'\phi_2(g))}\phi_3(g) \Leftrightarrow X_a X_{R'(D'\phi_3(g))}\phi_2(g) \ ,\end{aligned}$$

which implies

$$\begin{aligned}\{\phi_1, \{\phi_2, \phi_3\}_\ell\}_\ell + \text{c.p.} &= \frac{1}{2}(R'(D'\phi_1),\, D'\{\phi_2, \phi_3\}_2) + \text{c.p.}\\ &= X_{R'(D'\phi_1)}\, X_{R'(D'\phi_2)}\phi_3 \Leftrightarrow X_{R'(D'\phi_1)}\, X_{R'(D'\phi_3)}\phi_2 + \text{c.p.}\\ &= [X_{R'(D'\phi_2)}, X_{R'(D'\phi_3)}]\phi_1 + \text{c.p.}\\ &= X_{[R'(D'\phi_2),R'(D'\phi_3)]}\phi_1 + \text{c.p.} \ , \qquad \text{by (8.8)},\\ &= (D'\phi_1,\, [R'(D'\phi_2),\, R'(D'\phi_3)]) + \text{c.p.}\end{aligned}$$

Similarly

$$\{\phi_1, \{\phi_2, \phi_3\}_r\}_r = \Leftrightarrow(D\phi_1, [R(D\phi_2), R(D\phi_3)]) + \text{c.p.},$$

and

$$\begin{aligned}\{\phi_1, \{\phi_2, \phi_3\}_r\}_\ell + \{\phi_1, \{\phi_2, \phi_3\}_\ell\}_r &= [X_{R'(D'\phi_2)}, \widetilde{X}_{R(D\phi_3)}]\phi_1 + \text{c.p.}\\ &= 0 \ , \qquad \text{by (8.10)},\end{aligned}$$

Thus

$$\begin{aligned}\{\phi_1, \{\phi_2, \phi_3\}_{R,R'}\}_{R,R'} + \text{c.p.} &= \{\phi_1, \{\phi_2, \phi_3\}_\ell\}_\ell + \{\phi_1, \{\phi_2, \phi_3\}_r\}_r + \text{c.p.}\\ &= (D'\phi_1, [R'(D'\phi_2), R'(D'\phi_3)]) \Leftrightarrow(D\phi_1, [R(D\phi_2), R(D\phi_3)]) + \text{c.p.}\\ &= \Leftrightarrow(D'\phi_1, [D'\phi_2, D'\phi_3]) + (D\phi_1, [D\phi_2, D\phi_3]) \ ,\end{aligned}$$

where in the last step we have used (mYB), (8.13) and (8.3). Direct substitution of (8.6) and (8.7) show that the last two terms cancel, and this proves the theorem. □

Set

$$R(A) \equiv A_+ \Leftrightarrow A_- \ , \qquad R'(A) \equiv \Leftrightarrow R(A) \tag{8.15}$$

where A_+ is the strict upper part of A and A_- is strict lower part of A as before. A straightforward calculation shows that R and R' are skew and solve (mYB).

For SVD, the Sklyanin bracket is defined by

$$\begin{aligned} \{\phi, \psi\}_S(A) &\equiv \{\phi, \psi\}_{R,-R}(A) \\ &= (R(D\phi(A)), D\psi(A)) \Leftrightarrow (R(D'\phi(A)), (D'\psi(A))) \end{aligned} \tag{8.16}$$

Remark 8.17 The R matrix in (8.15) also arises in the study of the Cholesky eigenvalue algorithm (see [DLT89]).

By Theorem 8.12, $\{\cdot,\cdot\}_S$ gives a Poisson bracket on $M_n(\mathbf{R})$. But more is true: $\{\cdot,\cdot\}_S$ restricts as a Poisson bracket to the submanifolds

$$\{A \in M_n(\mathbf{R}) : \det\ A = c\}$$

for any constant $c \neq 0$. Indeed, from the formula $(\nabla \log \det\ A)^T = A^{-1}$, we see that $D \log \det(A) = D' \log \det(A) = I$, which implies $R(D \log \det(A)) = R(D' \log \det\ A) = 0$. Hence $\{\det, \phi\}_S = 0$ for all functions ϕ.

We now show that (7.1) is Hamiltonian.

Theorem 8.18 *Let $F(x)$ be a smooth real-valued function on $(0, \infty)$ and let $G_F(x) = \Leftrightarrow \int^x \frac{F(s)}{2s} ds$ be a primitive of $\Leftrightarrow F(x)/2x$. Then the equation*

$$\frac{d}{dt}\phi(A(t)) = \{\phi, H_F\}_S(A(t)) \ , \qquad A(0) = A_0 \ ,$$

generated by $H_F(A) = tr\ G_F(A^T A)$ on $\{A \in M_n(\mathbf{R}) : det\ A = \ det\ A_0 \neq 0\}$, is equivalent to (7.1).

Proof. From (8.17),

$$\dot{\phi} = \{\phi, H_F\}_S = \Leftrightarrow(\nabla\phi^T(A), (R(A\nabla H_F^T(A)))A) + (\nabla\phi^T(A), AR(\nabla H_F^T(A)A)) \ ,$$

so that

$$\dot{A} = A\, R(\nabla H_F^T(A)\, A) \Leftrightarrow (R(A\, \nabla H_F^T(A)))\, A \ . \tag{8.19}$$

But by differentiation,

$$\begin{aligned} \nabla H_F(A) &= 2AG_F'(A^T A) \\ &= \Leftrightarrow A(A^T A)^{-1}\, F(A^T A) \\ &= \Leftrightarrow (A^T)^{-1}\, F(A^T A) \ , \end{aligned}$$

which implies

$$\dot{A} = \Leftrightarrow A\, R(F(A^T A)) + R(F(AA^T))\, A \ .$$

As $\pi_0(S) = \Leftrightarrow R(S)$ for any symmetric matrix S, this proves the theorem. □

In particular

$$H_{SVD}(A) \equiv \Leftrightarrow\frac{1}{4}\mathrm{tr}(\log(A^TA))^2 \qquad (8.20)$$

generates the SVD flow. Also, $H = \Leftrightarrow\frac{1}{2}\,\mathrm{tr}(A^TA)$ generates the Toda-SVD flow in [Chu86].

The bracket $\{\cdot,\cdot\}_S$ is highly degenerate and the determination of the associated symplectic leaves (see [Wei85], [Sem85]) is in general extremely difficult. We have, however, the following happy fact.

Theorem 8.21 *The set $\mathcal{B}_\Delta$ of bidiagonal matrices B with positive entries a_p, b_q and fixed determinant,*

$$det\ B = \prod_{p=1}^{n} a_p = \Delta \qquad (8.22)$$

is a $(2n\Leftrightarrow 2)$-dimensional symplectic leaf for the Sklyanin bracket $\{\cdot,\cdot\}_S$. Moreover, $\mathcal{B}_\Delta$ has a global Darboux coordinate system given by

$$x_i \equiv \log b_i\ , \qquad 1 \le i \le n\Leftrightarrow 1\ , \qquad (8.23)$$

$$y_i \equiv \log\prod_{j=1}^{i} a_j\ , \quad 1 \le i \le n\Leftrightarrow 1\ , \qquad (8.24)$$

$$\{x_i, x_j\}_S = 0\ , \quad \{y_i, y_j\}_S = 0\ , \quad \{x_i, y_j\}_S = \delta_{ij}\ . \qquad (8.25)$$

Proof. We compute $\{\phi,\psi\}_S(B)$ at a bidiagonal matrix B. Set $\eta(m) = \Leftrightarrow 1, 0, 1$ if m is negative, zero, or positive respectively.

Insert the formulae

$$\begin{array}{rcl}
(B\nabla\phi^T(B))_{ij} & = & a_i\phi_{ji} + b_i\phi_{j,i+1} \\
(R(B\nabla\phi^T(B)))_{ij} & = & \eta(j\Leftrightarrow i)\,(a_i\phi_{ji} + b_i\phi_{j,i+1}) \\
(\nabla\phi^T(B)B)_{ij} & = & a_j\phi_{ji} + b_{j-1}\phi_{j-1,i} \\
(R(\nabla\phi^T(B)B))_{ij} & = & \eta(j\Leftrightarrow i)\,(a_j\phi_{ji} + b_{j-1}\phi_{j-1,i})
\end{array}$$

and their analogs for ψ, into (8.17), to obtain after some algebra

$$\begin{array}{rcl}
\{\phi,\psi\}_S(B) & = & \sum_{i,j}(\eta(j\Leftrightarrow i)\Leftrightarrow\eta(j+1\Leftrightarrow i))a_ib_j\phi_{ji}\psi_{i,j+1} \\
 & & +\sum_{i,j}(\eta(j\Leftrightarrow i)\Leftrightarrow\eta(j+1\Leftrightarrow i))\,a_jb_i\,\phi_{j,i+1}\psi_{ij} \\
 & = & \sum_{i}(\ \Leftrightarrow a_ib_{i-1}\phi_{i-1,i}\psi_{ii}\Leftrightarrow a_ib_i\phi_{ii}\psi_{i,i+1} + a_ib_i\psi_{i,i+1}\psi_{ii} + a_{i+1}b_i\phi_{i+1,i+1}\psi_{i,i+1})\ .
\end{array}$$

Changing variables, $a, b \to x, y$, on $\mathcal{B}_\Delta$ now leads to

$$\{\phi,\psi\}_S(B) = \{\phi,\psi\}_S(B(x,y)) = \sum_{i=1}^{n-1}(\frac{\partial\phi}{\partial x_i}\frac{\partial\psi}{\partial y_i}\Leftrightarrow\frac{\partial\phi}{\partial y_i}\frac{\partial\psi}{\partial x_i})\ , \qquad (8.26)$$

which is the canonical, non-degenerate Poisson bracket on R^{2n-2}, and (8.25) follows. □

For later convenience, we introduce the notation

$$y_n \equiv \log \prod_{j=1}^{n} a_j = \log \Delta \tag{8.27}$$

Remark 8.28 Formula (8.26) makes explicit the fact that $\mathcal{B}_\Delta$ is a symplectic leaf of $\{\cdot,\cdot\}_S$. In particular $\{\phi,\psi\}_S(B(x,y))$ depends *only* on the values of ϕ and ψ on $\mathcal{B}_\Delta$.

Remark 8.29 The fact that $\mathcal{B}_\Delta$ is a natural phase space for the Hamiltonian version of SVD is in striking parallel to the fact that T_τ, the tridiagonal matrices with prescribed trace τ (and with nonzero off diagonal elements), provides a natural phase space for the Hamiltonian version of QR. The relevant Poisson structure for QR is given by the Lie-Poisson structure on the dual of the Lie algebra of the lower triangular group (see [Kos79], [Adl79]; see also [DLNT86]). However, the map $B \mapsto B^T B$ from $\mathcal{B}_\Delta$ to T_τ is not symplectic and the relationship between the two Poisson structures is not clear. On the other hand the perfect shuffle $B \mapsto S$ of Section 2, induces a Poisson structure on the space of tridiagonal matrices S, of type (2.2), with fixed determinant. In particular, this shows that the Kac-van Moerbeke lattice (see Remark 7.12) is Hamiltonian on the space of such matrices S.

We conclude this section by proving **Fact 1** of the Introduction. Thus, if $M(j,i)$ is the Jacobian of the iterated SVD map from B_i to B_j expressed in the variables $\log b_1, \ldots, \log b_{n-1}$, $\log a_1, \ldots, \log a_n$, then

$$\lambda \in \text{spec } M(j,i) \Leftrightarrow \lambda^{-1} \in \text{ spec } M(j,i) \ .$$

For a bidiagonal matrix B set

$$\beta_i = \log b_i \ , \qquad 1 \le i \le n \Leftrightarrow 1 \ , \tag{8.30}$$

$$\alpha_i = \log a_i \ , \qquad 1 \le i \le n \ . \tag{8.31}$$

so that

$$\begin{pmatrix} \beta \\ \alpha \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & N \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \ , \tag{8.32}$$

where N is the $n \times n$ matrix $\begin{pmatrix} 1 & & & \bigcirc \\ \Leftrightarrow 1 & \ddots & & \\ & \ddots & \ddots & \\ \bigcirc & & \Leftrightarrow 1 & 1 \end{pmatrix}$. Note $N^{-1} = \begin{pmatrix} 1 & & & \bigcirc \\ 1 & 1 & & \\ \vdots & \ddots & & \\ & & 1 & \\ 1 & 1 & 1 & 1 \end{pmatrix}$.

Let $\begin{pmatrix} \beta^i \\ \alpha^i \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & N \end{pmatrix} \begin{pmatrix} x^i \\ y^i \end{pmatrix}$, $\begin{pmatrix} \beta^j \\ \alpha^j \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & N \end{pmatrix} \begin{pmatrix} x^j \\ y^j \end{pmatrix}$ be the coordinates of the i^{th} and j^{th} SVD iterates B_i and B_j, with $i < j$. Then

$$M(j,i) \equiv \frac{\partial(\beta^j,\alpha^j)}{\partial(\beta^i,\alpha^i)} = \begin{pmatrix} 1 & 0 \\ 0 & N \end{pmatrix} \frac{\partial(x^j,y^j)}{\partial(x^i,y^i)} \begin{pmatrix} 1 & 0 \\ 0 & N^{-1} \end{pmatrix} \ . \tag{8.33}$$

Under the iteration, $y_n^j = \log\det\, B^j = \log\det\, B^i = y_n^i$. Hence

$$\frac{\partial(x^j, y^j)}{\partial(x^i, y^i)} = \begin{pmatrix} & \frac{\partial x_1^j}{\partial y_n^i} \\ \frac{\partial(x_1^j,\ldots,x_{n-1}^j,y_1^j,\ldots,y_{n-1}^j)}{\partial(x_1^i,\ldots,x_{n-1}^i,y_1^i,\ldots,y_{n-1}^i)} & \vdots \\ & \frac{\partial y_{n-1}^j}{\partial y_n^i} \\ 0\cdots 0 & 1 \end{pmatrix} . \tag{8.34}$$

Now recall the following standard fact from Hamiltonian mechanics ([Ar]): let J denote the standard matrix $\begin{pmatrix} 0 & I \\ \Leftrightarrow I & 0 \end{pmatrix}$ and suppose $(x(t;x,y),\, y(t;x,y))$ is the solution of a Hamiltonian system of equations in $\mathbb{R}^{2n-2}$ in canonical form

$$\frac{d}{dt}\begin{pmatrix} x \\ y \end{pmatrix} = J\,\nabla H = \begin{pmatrix} 0 & I \\ \Leftrightarrow I & 0 \end{pmatrix} \begin{pmatrix} H_x \\ H_y \end{pmatrix} , \quad (x(0;x,y),\, y(0;x,y)) = (x,y) \ , \tag{8.35}$$

for some Hamiltonian $H : \mathbb{R}^{2n-2} \to \mathbb{R}$. Then for any t, the Jacobian $D = \frac{\partial(x(t;x,y),\, y(t;x,y))}{\partial(x,y)}$ is symplectic, i.e. $D^T J\, D = J$. But $\det(D^T \Leftrightarrow \lambda) = \det(JD^{-1}J^{-1} \Leftrightarrow \lambda) = \det(D^{-1} \Leftrightarrow \lambda)$. Thus if D is symplectic,

$$\lambda \in \text{spec } D \Leftrightarrow \lambda^{-1} \in \text{spec } D \ . \tag{8.36}$$

Finally from Theorems 7.14, 8.18 and 8.21, $x_1^j, \ldots, x_{n-1}^j$, $y_1^j, \ldots, y_{n-1}^j$ is the time $t = j \Leftrightarrow i$ evaluation in canonical variables of a Hamiltonian flow with initial data $x_1^i, \ldots, x_{n-1}^i$, $y_1^i, \ldots, y_{n-1}^i$. Hence the matrix $\frac{\partial(x_1^j,\ldots,x_{n-1}^j,y_1^j,\ldots,y_{n-1}^j)}{\partial(x_1^i,\ldots,x_{n-1}^i,y_1^i,\ldots,y_{n-1}^i)}$ is symplectic and (8.36) holds. **Fact 1** now follows from (8.33) and (8.34).

Remark 8.37 The Hamiltonian H_{SVD} on the (2n-2)-dimensional leaf $\mathcal{B}_\Delta$ is completely integrable in the sense of Liouville. The commuting integrals are the singular values $\sigma_1, \ldots, \sigma_{n-1}$ (recall $\prod_{i=1}^n \sigma_i = \Delta$ is a Casimir), and the associated angles are suitable combinations of the logarithms of the first components of the unit singular vectors. We leave the details to the interested reader (cf. [Mos75], [DLNT86], for example).

9 Asymptotics of $\|M(j,i)\|$

By the results of Sections 7 and 8, the SVD flow on bidiagonal matrices $B = \begin{pmatrix} a_1 & b_1 & & \bigcirc \\ & \ddots & \ddots & \\ & & \ddots & b_{n-1} \\ \bigcirc & & & a_n \end{pmatrix}$ takes the form

$$\begin{aligned} \frac{dx_i}{dt} &= \frac{\partial H_{SVD}}{\partial y_i}(x_1,\ldots,y_{n-1};y_n)\ , \quad 1 \le i \le n \Leftrightarrow 1\ , \\ \frac{dy_i}{dt} &= \Leftrightarrow \frac{\partial H_{SVD}}{\partial x_i}(x_1,\ldots,y_{n-1};y_n)\ , \quad 1 \le i \le n \Leftrightarrow 1\ , \\ \frac{dy_n}{dt} &= 0 \\ x_i(0) &= x_i\ , \quad y_i(0) = y_i\ , \quad 1 \le i \le n \Leftrightarrow 1\ , \quad y_n(0) = y_n(t) = y_n\ , \end{aligned} \tag{9.1}$$

in the canonical coordinates $x_i = \log b_i$, $y_i = \log \prod_{j=1}^{i} a_j$, where $H_{SVD} = \Leftrightarrow \frac{1}{4}\mathrm{tr}(\log(B^T(x,y)B(x,y)))^2$. This leads to the equation

$$\frac{d}{dt}K_n = \begin{pmatrix} J & \nabla_n^2 H_{SVD} \\ 0 & \ldots \quad 0 \end{pmatrix} K_n\ ,\ K_n(0) = I\ , \tag{9.2}$$

for the full $(2n \Leftrightarrow 1) \times (2n \Leftrightarrow 1)$ Jacobian matrix $K_n(t) = \frac{\partial(x_1(t),\ldots,x_{n-1}(t),y_1(t),\ldots,y_n(t))}{\partial(x_1,\ldots,x_{n-1},y_1,\ldots,y_n)}$, where J is again the standard $(2n \Leftrightarrow 2) \times (2n \Leftrightarrow 2)$ matrix $\begin{pmatrix} 0 & I \\ \Leftrightarrow I & 0 \end{pmatrix}$ and $\nabla_n^2 H_{SVD}$ is the $(2n \Leftrightarrow 2) \times (2n \Leftrightarrow 1)$ Hessian matrix

$$\nabla_n^2 H_{SVD} = \left(\frac{\partial^2 H_{SVD}}{\partial z_i \partial z_j}\right)_{1 \le i \le 2n-2,\, 1 \le j \le 2n-1} \tag{9.3}$$

$$(z_1,\ldots,z_{2n-1}) \equiv (x_1,\ldots,x_{n-1},y_1,\ldots,y_n)\ . \tag{9.4}$$

Our goal in this section is to evaluate $K_n(t)$ as $t \to \infty$. By (8.33), the asymptotics for $M(j,i)$ will then follow. Inserting $F(x) = \log x$ in Theorem 7.18, we obtain

$$a_i(t) = \sigma_i + O(b^2)\ , \quad 1 \le i \le n \tag{9.5}$$

and

$$b_i(t) \sim b_i^\infty\left(\frac{\sigma_{i+1}}{\sigma_i}\right)^{2t}\ , \quad 1 \le i \le n \Leftrightarrow 1 \tag{9.6}$$

as $t \to \infty$, where $b_i^\infty > 0$ and $b^2 = \sum_{j=1}^{n-1} b_j^2$ as before.

A convenient formula for H_{SVD} is given by the spectral representation

$$\mathrm{tr}(\log B^T B)^2 = \mathrm{tr}\int_C \frac{(\log s)^2}{s \Leftrightarrow B^T B}\frac{ds}{2\pi i}\ , \tag{9.7}$$

where $\mathcal{C}$ is the counterclockwise contour

$$\begin{array}{ccccc} \times & \times & \times & \dots & \times \\ 0 & \sigma_n^2 & \sigma_{n-1}^2 & \dots & \sigma_1^2 \end{array} \quad ,$$

$$\mathcal{C}$$

from which we obtain, after one integration by parts,

$$\frac{\partial^2}{\partial z_j \partial z_k} \operatorname{tr}(\log B^T B)^2 = \operatorname{tr} \int_{\mathcal{C}} \frac{\log s}{s} \left[\frac{\partial^2 B^T B}{\partial z_j \partial z_k} \frac{1}{s \Leftrightarrow B^T B} + \frac{\partial B^T B}{\partial z_j} \frac{1}{s \Leftrightarrow B^T B} \frac{\partial B^T B}{\partial z_k} \frac{1}{s \Leftrightarrow B^T B} \right] \frac{ds}{\pi i} \, . \tag{9.8}$$

Now $\partial B / \partial x_m = e_{m,m+1} b_m$, $1 \leq m \leq n \Leftrightarrow 1$, where e_{ij} is the standard $n \times n$ matrix with 1 in the (i,j) position, and zero elsewhere. Thus if z_j or z_k lies in the set $\{x_1, \dots, x_{n-1}\}$, then $\frac{\partial^2}{\partial z_j \partial z_k} \operatorname{tr}(\log B^T B)^2$ is of order b, and hence is exponentially decreasing. (Here we use (9.5) and (9.6) to bound $\max_{s \in \mathcal{C}} \|(S \Leftrightarrow B^T B)^{-1}\|$, etc.)

The leading order contribution comes from the derivatives $\partial^2 / \partial y_j \partial y_k$. We find

$$\frac{\partial B^T B}{\partial y_j} = 2 a_j^2 e_{jj} \Leftrightarrow 2 a_{j+1}^2 e_{j+1,j+1} + O(b) \tag{9.9}$$

and

$$\begin{aligned} \frac{\partial^2 B^T B}{\partial y_j \partial y_k} = & \quad \delta_{jk}(4 a_j^2 e_{jj} + 4 a_{j+1}^2 e_{j+1,j+1}) \\ & + \quad \delta_{j-1,k}(\Leftrightarrow 4 \, a_j^2) e_{jj} + \delta_{j+1,k}(\Leftrightarrow 4 \, a_{j+1}^2) e_{j+1,j+1} \\ & + \quad O(b) \, , \end{aligned} \tag{9.10}$$

where $a_{n+1} \equiv 0$, etc. Substituting (9.9) and (9.10) in (9.8), we find

$$\begin{aligned} \frac{\partial^2}{\partial y_j \partial y_k} \operatorname{tr}(\log B^T B) = & \quad \operatorname{tr} \int_{\mathcal{C}} \frac{\log s}{s} \Bigg[\delta_{jk}(4 \sigma_j^2 e_{jj} + 4 \sigma_{j+1}^2 e_{j+1,j+1}) \begin{pmatrix} (s \Leftrightarrow \sigma_1^2)^{-1} & & \bigcirc \\ & \ddots & \\ \bigcirc & & (s \Leftrightarrow \sigma_n^2)^{-1} \end{pmatrix} \\ & + \delta_{j-1,k}(\Leftrightarrow 4 \sigma_j^2) e_{ij} \begin{pmatrix} (s \Leftrightarrow \sigma_1^2)^{-1} & & \bigcirc \\ & \ddots & \\ \bigcirc & & (s \Leftrightarrow \sigma_n^2)^{-1} \end{pmatrix} \\ & + \delta_{j+1,k}(\Leftrightarrow 4 \sigma_{j+1}^2) e_{j+1,j+1} \begin{pmatrix} (s \Leftrightarrow \sigma_1^2)^{-1} & & \bigcirc \\ & \ddots & \\ \bigcirc & & (s \Leftrightarrow \sigma_n^2)^{-1} \end{pmatrix} \\ & + (2 \sigma_j^2 e_{jj} \Leftrightarrow 2 \sigma_{j+1}^2 e_{j+1,j+1}) \begin{pmatrix} (s \Leftrightarrow \sigma_1^2)^{-1} & & \bigcirc \\ & \ddots & \\ \bigcirc & & (s \Leftrightarrow \sigma_n^2)^{-1} \end{pmatrix} \end{aligned}$$

$$\times(2\sigma_k^2 e_{kk} \Leftrightarrow 2\sigma_{k+1}^2 e_{k+1,k+1}) \begin{pmatrix} (s \Leftrightarrow \sigma_1^2)^{-1} & & \bigcirc \\ & \ddots & \\ \bigcirc & & (s \Leftrightarrow \sigma_n^2)^{-1} \end{pmatrix} \Bigg] \frac{ds}{\pi i}$$

$$+O(b) \ .$$

After performing the integrals, this leads to

$$\frac{\partial^2}{\partial y_j \partial y_k} \operatorname{tr}(\log B^T B)^2 = 16\,\delta_{jk} \Leftrightarrow 8\,\delta_{j-1,k} \Leftrightarrow 8\,\delta_{j+1,k} + O(b) \ , \quad 1 \le j \le n \Leftrightarrow 1 \ , \quad 1 \le k \le n \ ,$$

and hence

$$\begin{pmatrix} J & \nabla_n^2 H_{SVD} \\ 0 \quad \ldots & 0 \end{pmatrix} = \begin{pmatrix} O_{n-1,n-1} & L \\ O_{n,n-1} & O_{n,n} \end{pmatrix} + O(b) \tag{9.11}$$

where L is the $(n \Leftrightarrow 1) \times n$ matrix

$$L = \begin{pmatrix} \Leftrightarrow 4 & 2 & & & & & \bigcirc \\ 2 & \Leftrightarrow 4 & 2 & & & & \\ & 2 & \Leftrightarrow 4 & & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & & & \\ & & & 2 & \Leftrightarrow 4 & 2 & \\ \bigcirc & & & & 2 & \Leftrightarrow 4 & 2 \end{pmatrix} \tag{9.12}$$

Equation (9.2) now takes the form

$$\frac{dK_n}{dt} = \begin{pmatrix} 0 & L \\ 0 & 0 \end{pmatrix} K_n + C(t) K_n \ , \quad K_n(0) = I \ , \tag{9.13}$$

where by (9.6),

$$\|C(t)\| \le c_0\, e^{-\delta t} \ , \quad \delta = \min_{1 \le i \le n-1} \log(\sigma_i^2/\sigma_{i+1}^2) > 0 \ , \tag{9.14}$$

for some positive constant c_0. Rewriting (9.13) in the standard way (see e.g. [CL55]) in integral form, we obtain after iteration

$$\begin{aligned} K_n(t) &= e^{\left(\begin{smallmatrix} 0 & L \\ 0 & 0 \end{smallmatrix}\right)t} \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} (1 + o(1)) \\ &= \begin{pmatrix} A_{11} + t\,L\,A_{21} & A_{12} + t\,L\,A_{22} \\ A_{21} & A_{22} \end{pmatrix} (1 + o(1)) \end{aligned} \tag{9.15}$$

for suitable constant matrices A_{ij}. Here the terms $o(1)$ are exponentially decreasing. Finally, as in (8.33),

$$M(t,0) = \frac{\partial(\beta(t), \alpha(t))}{\partial(\beta, \alpha)} = \begin{pmatrix} B_{11} + t\,,_n B_{21} & B_{12} + t\,,_n B_{22} \\ B_{21} & B_{22} \end{pmatrix} (1 + o(1)) \tag{9.16}$$

for suitable B_{ij}, where $,_n$ is the $(n \Leftrightarrow 1) \times n$ matrix,

$$,_n = LN^{-1} = \begin{pmatrix} \Leftrightarrow 2 & 2 & & & \bigcirc \\ & \Leftrightarrow 2 & 2 & & \\ & & \ddots & \ddots & \\ & & & & \\ \bigcirc & & & \Leftrightarrow 2 & 2 \end{pmatrix} , \tag{9.17}$$

which appears in (1.9).

To compute B_{21}, B_{22}, note that for any t,

$$\sigma_\ell(B') = \sigma_\ell(B) \ , \quad 1 \leq \ell < n \ ,$$

where $B' = B(t)$. Thus for $1 \leq m \leq n \Leftrightarrow 1$, $1 \leq \ell \leq n$,

$$\sum_{i=1}^{n} \frac{\partial \sigma_\ell(B')}{\partial a_i'} a_i' \frac{\partial \alpha_i'}{\partial \beta_m} + \sum_{i=1}^{n-1} \frac{\partial \sigma_\ell(B')}{\partial b_i'} b_i' \frac{\partial \beta_i'}{\partial \beta_m} = \frac{\partial \sigma_\ell}{\partial \beta_m} \ . \tag{9.18}$$

By regular perturbation theory applied to the perfect shuffle S of B (see (2.2)), we see that (in the notation of Section 2)

$$\begin{aligned} \frac{\partial \sigma_\ell(B')}{\partial a_i'} &= 2h_\ell^+(2i \Leftrightarrow 1)\, h_\ell^+(2i) \\ &= u_\ell'(i)\ v_\ell'(i) \ , \end{aligned}$$

where u_ℓ', v_ℓ' are the unit eigenvectors of $B'(B')^T$ and $(B')^T B'$ respectively, chosen such that $u_\ell'(1), v_\ell'(1) > 0$. But by Remark 7.27, $v_\ell'(i) \to (\Leftrightarrow 1)^{i+1}\delta_{\ell i}$ as $t \to \infty$. A similar analysis shows that the same is true for $u_\ell'(i)$; hence $\frac{\partial \sigma_\ell(B')}{\partial a_i'} \to \delta_{\ell i}$. On the other hand, by (9.16), $\partial\beta_i'/\partial\beta_m$ grows at worst linearly as $t \to \infty$, but b_i' decreases exponentially and $\partial\sigma_\ell(B')/\partial b_i'$ is bounded, again by regular perturbation theory. Inserting this information into (9.18), we learn that

$$\lim_{t\to\infty} \frac{\partial \alpha_\ell'}{\partial \beta_m} = \frac{1}{\sigma_\ell} \frac{\partial \sigma_\ell}{\partial \beta_m} \ .$$

Thus

$$B_{21} = \left(\frac{\partial \log \sigma_i}{\partial \beta_m} \right)_{1 \leq i \leq n,\, 1 \leq m \leq n-1} \tag{9.19}$$

and similarly

$$B_{22} = \left(\frac{\partial \log \sigma_i}{\partial \alpha_m} \right)_{1 \leq i \leq n,\, 1 \leq m \leq n} \ . \tag{9.20}$$

Also

$$,_n B_{21} = \begin{pmatrix} \frac{\partial \log \sigma_2^2/\sigma_1^2}{\partial \beta_1} & \cdots & \frac{\partial \log \sigma_2^2/\sigma_1^2}{\partial \beta_{n-1}} \\ \vdots & & \vdots \\ \frac{\partial \log \sigma_n^2/\sigma_{n-1}^2}{\partial \beta_1} & \cdots & \frac{\partial \log \sigma_n^2/\sigma_{n-1}^2}{\partial \beta_{n-1}} \end{pmatrix} \tag{9.21}$$

and

$$,\, {}_nB_{22} = \begin{pmatrix} \frac{\partial \log \sigma_2^2/\sigma_1^2}{\partial \alpha_1} & \cdots & \frac{\partial \log \sigma_2^2/\sigma_1^2}{\partial \alpha_n} \\ \vdots & & \vdots \\ \frac{\partial \log \sigma_n^2/\sigma_{n-1}^2}{\partial \alpha_1} & \cdots & \frac{\partial \log \sigma_n^2/\sigma_{n-1}^2}{\partial \alpha_n} \end{pmatrix} . \tag{9.22}$$

Recalling that for any matrix A, $\|A\|_\infty = \max_i \sum_j |A_{ij}|$, we have obtained the following theorem.

Theorem 9.23 *Under the bidiagonal SVD flow (9.1), the Jacobian matrix* $M(t,0) = \frac{\partial(\beta(t),\alpha(t))}{\partial(\beta,\alpha)}$, *satisfies*

$$M(t,0) = \begin{pmatrix} B_{11} + t\, ,\, {}_nB_{21} & B_{12} + t\, ,\, {}_nB_{22} \\ B_{21} & B_{22} \end{pmatrix} (1 + o(1))$$

where the term $o(1)$ *is exponentially decreasing as* $t \to \infty$, *and the constant matrices* B_{21}, B_{22}, $,\, {}_nB_{21}$, $,\, {}_nB_{22}$ *satisfy (9.19)-(9.22) respectively.*

Also

$$\|M(t,0)\|_\infty \leq (8n \Leftrightarrow 4)t + O(1) \tag{9.24}$$

as $t \to \infty$.

Proof. The point to note is that, by (the proof of the) inequality (2.4), the entries of B_{21} and B_{22} are bounded by 1 and the entries of $,\, {}_nB_{21}$, $,\, {}_nB_{22}$ are bounded by 4. □

Remark 9.25 From (9.5) and (9.6),

$$\alpha_i(t) = \log a_i(t) = \log \sigma_i + o(1)\ , \quad 1 \leq i \leq n\ , \tag{9.26}$$

and

$$\beta_i(t) = \log b_i(t) = t\, \log(\sigma_{i+1}^2/\sigma_i^2) + O(1)\ , \quad 1 \leq i \leq n \Leftrightarrow 1\ , \tag{9.27}$$

The content of Theorem 9.23 is that the leading asymptotics can be differentiated with respect to the initial data. This in turn suggests an alternative proof of Theorem 9.23: if (9.26) and (9.27) can be shown to hold uniformaly for all initial data in a *complex* neighborhood of B, then Theorem 9.23 follows immediately from Cauchy's formula. This approach can indeed be carried out, but we present no details. (In this connection we refer the reader to [Mos75], where an analysis of the asymptotics of the classical Toda lattice (tridiagonal, $F(x) = x$) with complex initial data, is presented.) In Section 10, however, we will present (the outline of) a third proof of Theorem 9.23 using iterates of the gradient of one step of the SVD algorithm, and which does not utilize the underlying flows.

Finally we note that the proof of (9.24) shows that, more generally,

$$\|M(t_2,t_1)\|_\infty \leq (8n \Leftrightarrow 4)(t_2 \Leftrightarrow t_1) + O(1)\ , \tag*{(9.24)$'$}$$

as $t_2 \Leftrightarrow t_1 \to \infty$. Moreover the estimate is uniform in $t_2 > t_1 \geq 0$. Recalling yet again the relationship between the SVD flow and the SVD algorithm, **Fact 4** in the introduction is finally proven by setting $t_1 = i$ and $t_2 = j$ in (9.24)$'$.

10 The Spectrum of the One-step Jacobian of SVD

We consider one step of SVD taking the bidiagonal matrix B to the bidiagonal matrix B'. Our goal in this section is to analyze the $(2n \Leftrightarrow 2) \times (2n \Leftrightarrow 2)$ Jacobian K of the map $B \to B'$, expressed in terms of x, y variables,

$$\begin{aligned} K &= \frac{\partial(x_1', \ldots, x_{n-1}', y_1', \ldots, y_{n-1}')}{\partial(x_1, \ldots, x_{n-1}, y_1, \ldots, y_{n-1})} \\ &\equiv \begin{pmatrix} \frac{\partial x'}{\partial x} & \frac{\partial x'}{\partial y} \\ \frac{\partial y'}{\partial x} & \frac{\partial y'}{\partial y} \end{pmatrix} , \end{aligned}$$

and, in particular, to prove **Fact 2** and **Fact 3** of the Introduction. Observe that K is the leading $(2n \Leftrightarrow 2) \times (2n \Leftrightarrow 2)$ submatrix of the full Jacobian $K_n(t = 1)$ of Section 9; K is symplectic by the results of Section 8.

We will use the notation

$$A_i \equiv \prod_{j=1}^{i} a_j = e^{y_i} , \quad 1 \le i \le n . \tag{10.1}$$

As before $b_i = e^{x_i}$, $1 \le i \le n \Leftrightarrow 1$.

Our first result is formula (10.19) below, which computes K to relevant orders in b_i.

The SVD algorithm can be implemented by applying a sequence of 2×2 rotations to the matrix B (see [GVL83]). A straightforward induction using these rotations leads to the following formulae for the entries of B'.

$$A_j' = s_j / r_j , \quad 1 \le j \le n , \tag{10.2}$$

$$b_j' = \frac{r_{j+1}}{r_j s_j} s_{j-1} b_j a_{j+1} , \quad 1 \le j \le n \Leftrightarrow 1 , \tag{10.3}$$

where r_j, s_j satisfy the recurrences

$$r_{i+1}^2 = A_{i+1}^2 + r_i^2 b_{i+1}^2 , \quad i \ge 0 , \tag{10.4}$$

and

$$s_{i+1}^2 = r_{i+1}^4 + s_i^2 b_{i+1}^2 a_{i+2}^2 , \quad i \ge 0 , \tag{10.5}$$

where $r_0 = s_0 \equiv 1$.

These recurrence relations imply

$$r_i^2 = \sum_{j=0}^{i} A_j^2 \Big(\prod_{k=j+1}^{i} b_k^2 \Big) \tag{10.6}$$

and

$$s_i^2 = \sum_{j=0}^{i} r_j^4 \Big(\frac{\prod_{k=j+1}^{i} b_k^2}{A_{j+1}^2} \Big) A_{i+1}^2 , \tag{10.7}$$

where $\prod_{k=j+1}^{i} b_k^2 \equiv 1$ if $j = i$. Observe that

$$r_i^2 = r_i^2(A_1, \ldots, A_i, b_1, \ldots, b_i) \tag{10.8}$$

$$s_i^2 = s_i^2(A_1, \ldots, A_{i+1}, b_1, \ldots, b_i) \ , \tag{10.9}$$

and hence from (10.2) and (10.3),

$$x_i' = x_i'(x_1, \ldots, x_{i+1}, y_1, \ldots, y_{i+1}) \tag{10.10}$$

$$y_i' = y_i'(x_1, \ldots, x_i, y_1, \ldots, y_{i+1}) \ . \tag{10.11}$$

Expansion to third order in $b^2 = \sum_{i=1}^{n-1} b_i^2$, yields

$$r_i^2 = A_i^2 + A_{i-1}^2 b_i^2 + A_{i-2}^2 b_{i-1}^2 b_i^2 + O(b^6) \ , \tag{10.12}$$

$$\begin{aligned} s_i^2 = \ & A_i^4 + (2A_i^2 A_{i-1}^2 + A_{i-1}^4 A_{i+1}^2 A_i^{-2}) b_i^2 + (2A_i^2 A_{i-2}^2 \\ & +2A_{i-1}^2 A_{i-2}^2 A_{i+1}^2 A_i^{-2} + A_{i-2}^4 A_{i+1}^2 A_{i-1}^{-2}) b_{i-1}^2 b_i^2 + A_{i-1}^4 b_i^4 + O(b^6) \ , \end{aligned} \tag{10.13}$$

$$\begin{aligned} r_i^2 s_i^2 = \ & A_i^6 + (3A_i^4 A_{i-1}^2 + A_{i-1}^4 A_{i+1}^2) b_i^2 + (3A_i^4 A_{i-2}^2 + 2A_{i-1}^2 A_{i-2}^2 A_{i+1}^2 \\ & +A_i^2 A_{i-2}^4 A_{i+1}^2 A_{i-1}^{-2}) b_{i-1}^2 b_i^2 + (3A_i^2 A_{i-1}^4 + A_{i-1}^6 A_{i+1}^2 A_{i-1}^{-2}) b_i^4 + O(b^6) \ , \end{aligned} \tag{10.14}$$

$$\begin{aligned} r_{i+1}^2 s_{i-1}^2 = \ & A_{i+1}^2 A_{i-1}^4 + (2A_{i-1}^2 A_{i-2}^2 A_{i+1}^2 + A_{i-2}^4 A_i^2 A_{i+1}^2 A_{i-1}^{-2}) b_{i-1}^2 \\ & +(A_i^2 A_{i-1}^4) b_{i+1}^2 + (A_{i+2}^2 A_{i-2}^4) b_{i-1}^4 + (2A_{i-1}^2 A_{i-3}^2 A_{i+1}^2 + 2A_{i-2}^2 A_{i-3}^2 A_i^2 A_{i+1}^2 A_{i-1}^{-2} \\ & +A_{i-3}^4 A_i^2 A_{i+1}^2 A_{i-2}^{-2}) b_{i-1}^2 b_{i-2}^2 + (2A_{i-1}^2 A_{i-2}^2 A_i^2 \\ & +A_{i-2}^4 A_i^4 A_{i-1}^{-2}) b_{i-1}^2 b_{i+1}^2 + A_{i-1}^6 b_i^2 b_{i+1}^2 + O(b^6) \ . \end{aligned} \tag{10.15}$$

We first compute $\partial x'/\partial x$, which is lower Hessenberg by (10.10). From (10.3)

$$\Leftrightarrow \frac{\partial x_i'}{\partial x_j} = \delta_{ij} + \frac{1}{2} \frac{\frac{\partial}{\partial x_j} r_{i+1}^2 s_{i-1}^2}{r_{i+1}^2 s_{i-1}^2} \Leftrightarrow \frac{1}{2} \frac{\frac{\partial}{\partial x_j}(r_i^2 s_i^2)}{r_i^2 s_i^2} \ . \tag{10.16}$$

Using (10.15) and (10.16), and the fact that

$$\frac{\partial}{\partial x_i}(\text{anything}) = (\text{something}) b_i^2 \ ,$$

we obtain

$$(\frac{\partial x'}{\partial x} \Leftrightarrow I) =$$

$$\left(\begin{array}{cccc} \Leftrightarrow b_i^2(\frac{3}{\sigma_1^2} & b_2^2(\frac{1}{\sigma_2^2} & 0 & \cdots \quad 0 \\ +\frac{\sigma_2^2}{\sigma_1^4} + O(b^2)) & +O(b^2)) & & \\ b_1^2(\frac{2}{\sigma_1^2} + \frac{a_2^2}{a_1^4} + O(b^2)) & \Leftrightarrow b_2^2(\frac{3}{\sigma_2^2} + \frac{\sigma_3^2}{\sigma_2^4} + O(b^2)) & b_3^2(\frac{1}{\sigma_3^2} + O(b^2)) & \\ +\frac{a_2^2}{a_1^4} + O(b^2)) & +\frac{\sigma_3^2}{\sigma_2^4} + O(b^2)) & & \\ O(b_1^2 b^2) & b_2^2(\frac{2}{\sigma_2^2} + \frac{\sigma_3^2}{\sigma_2^4} + O(b^2)) & & \\ & +\frac{\sigma_3^2}{\sigma_2^4} + O(b^2)) & & \\ & \vdots & \ddots & \ddots \\ & & & 0 \\ & & & b_{n-1}^2(\frac{1}{\sigma_{n-1}^2} + O(b^2)) \\ O(b_1^2 b^2) & O(b_2^2 b^2) \ \cdots & b_{n-2}^2(\frac{2}{\sigma_{n-2}^2} + \frac{\sigma_{n-1}^2}{\sigma_{n-2}^4} + O(b^2)) & \Leftrightarrow b_{n-1}^2(\frac{3}{\sigma_{n-1}^2} + \frac{\sigma_n^2}{\sigma_{n-1}^4} + O(b^2)) \end{array} \right) . \tag{10.17}$$

Similar formulae can be obtained for $\frac{\partial x'}{\partial y}$, $\frac{\partial y'}{\partial x}$ and $\frac{\partial y'}{\partial y}$. In particular $\frac{\partial y'}{\partial x}$ is lower triangular and

$$\frac{\partial y'}{\partial x} = \begin{pmatrix} p_1^2 b_1^2 & 0 & \dots & 0 \\ O(b_1^2 b_2^2) & p_2^2 b_2^2 & & \\ \vdots & \vdots & \ddots & \\ & & & 0 \\ O(b_1^2 b_{n-1}^2) & O(b_2^2 b_{n-1}^2) & \dots & p_{n-1}^2 b_{n-1}^2 \end{pmatrix} , \qquad (10.18)$$

where

$$p_i = \sqrt{\frac{1}{\sigma_i^2} + \frac{\sigma_{i+1}^2}{\sigma_i^4} + O(b^2)} , \quad 1 \le i \le n \Leftrightarrow 1 . \qquad (10.19)$$

The final formulae can be written in the form

$$K = I + \begin{pmatrix} E_1 B^2 & E_2 + O_0(b^2) \\ B(1 + BE_3 B)B & B^2 E_4 \end{pmatrix} \qquad (10.20)$$

where

$$B = \operatorname{diag}(p_1 b_1, \dots, p_{n-1} b_{n-1}) , \qquad (10.21)$$

$$E_2 = \begin{pmatrix} \Leftrightarrow 4 & 2 & & & \bigcirc \\ 2 & \Leftrightarrow 4 & \ddots & & \\ & \ddots & \ddots & & \\ & & & & 2 \\ \bigcirc & & & 2 & \Leftrightarrow 4 \end{pmatrix} , \quad O_0(b^2) \text{ is lower Hessenberg,} \qquad (10.22)$$

$$\begin{array}{ll} E_1, E_4 & \text{are lower Hessenberg with entries} \\ & (E_1)_{i,i+1}, (E_4)_{i,i+1} \text{ of the form} \\ & (\text{positive constant } + O(b^2)) , \end{array} \qquad (10.23)$$

and

$$E_3 \text{ is strictly lower triangular.} \qquad (10.24)$$

Note first that (10.20) immediately proves **Fact 3** of the Introduction,

$$K \to K_\infty = \begin{pmatrix} 1 & E_2 \\ 0 & 1 \end{pmatrix}$$

as $t = k \to \infty$. Formula (10.20) can also be used to give an alternative proof of the asymptotics of $K_n(t)$, as mentioned in Remark 9.25. Let $K(j)$ be the leading $(2n \Leftrightarrow 2) \times (2n \Leftrightarrow 2)$ submatrix of $K_n(t)$ evaluated at time $t = j$. We will show that

$$K(j) = \begin{pmatrix} A'_{11} + jE_2 A'_{12} & A'_{12} + j\, E_2 A'_{22} \\ A'_{21} & A'_{22} \end{pmatrix} (1 + o(1)) , \qquad (10.25)$$

and leave the (rather lengthy) remaining details to the reader. From (10.20) and (9.6) we have for k large

$$K(k+1, k) = K_\infty + \epsilon_k ,$$

where

$$\|\epsilon_j\| \leq c\,\rho^j\;, \quad 0 < \rho < 1\;, \quad c \text{ constant.}$$

Writing

$$\begin{aligned} K(j) &= \textstyle\prod_{k=0}^{j-1} K(k+1,k) \\ &= K_\infty^j \textstyle\prod_{k=0}^{j-1} \left(1 + K_\infty^{-(k+1)} \epsilon_k K_\infty^k\right)\;. \end{aligned}$$

But

$$\|K_\infty^{-(k+1)} \epsilon_k\, K_\infty^k\| \leq c' k\, \rho^k$$

for large k. Thus

$$\begin{pmatrix} A'_{11} & A'_{12} \\ A'_{21} & A'_{22} \end{pmatrix} \equiv \lim_{j\to\infty} \prod_{k=0}^{j-1} (1 + K_\infty^{-(k+1)} \epsilon_k K_\infty^k)$$

exists and the convergence is exponential. Formula (10.25) now follows.

Our next result towards the proof of **Fact 2** shows that the eigenvalues λ_j, $1 \leq j \leq 2n \Leftrightarrow 2$, of $K = K(t+1,t)$ eventually lie in Gershgorin-type disks contained in a fixed wedge with vertex $\lambda = 1$, symmetric about $1 + i\mathbf{R}$, and with aperture 2θ less than π.

Figure 1:

In particular (spec $K(t+1,t)) \cap \mathbf{R}$ is eventually empty.

Observe first that $K \Leftrightarrow I = K(t+1,t) \Leftrightarrow I$ can be rewritten as

$$K \Leftrightarrow I = \begin{pmatrix} 1 & 0 \\ 0 & B \end{pmatrix} \begin{pmatrix} E_1 B & E_2 + O_0(b^2) \\ 1 + BE_3B & BE_4 \end{pmatrix} \begin{pmatrix} B & 0 \\ 0 & 1 \end{pmatrix} \qquad (10.26)$$

so that the eigenvalue problem

$$(K \Leftrightarrow I)\begin{pmatrix} f' \\ g' \end{pmatrix} = \nu \begin{pmatrix} f' \\ g' \end{pmatrix} , \quad \lambda = 1 + \nu , \tag{10.27}$$

reduces to the system

$$(E_2 + O(b^2))g = (\nu B^{-1} \Leftrightarrow E_1 B) f \tag{10.28}$$

$$(1 + BE_2 B) f = (\nu B^{-1} \Leftrightarrow BE_4) g \tag{10.29}$$

where

$$\nu = \lambda \Leftrightarrow 1 \quad \text{and} \quad \begin{pmatrix} f \\ g \end{pmatrix} = \begin{pmatrix} Bf' \\ g' \end{pmatrix} \neq 0 . \tag{10.30}$$

For $t \to \infty$, $K(t+1,t) \to \begin{pmatrix} 1 & E_2 \\ 0 & 1 \end{pmatrix}$, and so

$$\nu \to 0 . \tag{10.31}$$

As E_3 is strictly lower triangular, $(1 + BE_3 B)^{-1}$ exists (even if the b_j are not small) and so $g \neq 0$. We normalize

$$\|g\|^2 = \sum_{j=1}^{n-1} |g_j|^2 = 1 . \tag{10.32}$$

We have

$$\begin{aligned} (E_2 + O_0(b^2))g \quad &= (\nu B^{-1})(1 + BE_3 B)^{-1}(\nu B)^{-1} g + E_1 B(1 + BE_3 B)^{-1}(BE_4) g \\ &\quad \Leftrightarrow E_1 B(1 + BE_3 B)^{-1}(\nu B^{-1}) g \Leftrightarrow (\nu B^{-1})(1 + BE_3 B)^{-1}(BE_4) g \\ &= I + II + III + IV \ . \end{aligned}$$

Now

$$\begin{aligned} I \quad &= (\nu B^{-1})^2 + \nu^2 \textstyle\sum_{k=1}^{n-1} (\Leftrightarrow 1)^{k-1} (B^{-1}(BE_3 B)^k B^{-1}) g \\ &= (\nu B^{-1})^2 + O_1(\nu^2) \end{aligned} \tag*{$(10.33)_I$}$$

where $O_1(\nu^2)$ is strictly lower triangular,

$$II = O_2(b^4) + E_1 B^2 E_4 \ , \tag*{$(10.33)_{II}$}$$

where we note that

$$E_1 B^2 E_4 = O_2'(b^2) + \begin{pmatrix} 0 & 0 & O(b_2^2) & 0 & \dots \\ 0 & 0 & 0 & O(b_3^2) & 0 \\ & & & \ddots & \\ & & & & \dots \end{pmatrix}, O_2'(b) \text{ is lower Hessenberg,}$$

$$III = O_3(\nu) \ , \tag*{$(10.33)_{III}$}$$

and

$$IV = O_4(\nu) \ , \tag*{$(10.33)_{IV}$}$$

where $O_3(\nu)$, $O_4(\nu)$ are again lower Hessenberg.

Thus the eigenvalue problem becomes

$$(E_2 \Leftrightarrow (\nu B^{-1})^2)g = (O_1(\mu^2) + O_2(b^4) \Leftrightarrow O_0(b^2) + E_1 B^2 E_4 + O_3(\nu) + O_4(\nu))g \ . \qquad (10.34)$$

For our present purposes all we need is that

$$(E_2 \Leftrightarrow (\nu B^{-1})^2)g = o_5(1) \ , \quad \|g\| = 1 \ , \qquad (10.35)$$

which implies

$$\begin{aligned} \min_j |4 + (\nu/p_j b_j)^2| \ &\leq \left(\textstyle\sum_{j=1}^{n-1} | \Leftrightarrow 4 \Leftrightarrow (\nu/p_j b_j)^2|^2 |g_j|^2 \right)^{1/2} \\ &\leq \|Y\|_2 + o_5(1) \ , \end{aligned}$$

where Y is the $(n \Leftrightarrow 1) \times (n \Leftrightarrow 1)$ matrix

$$Y = \begin{pmatrix} 0 & 2 & & & 0 \\ 2 & & & & \\ & \ddots & \ddots & & \\ & & & & 2 \\ 0 & & & 2 & 0 \end{pmatrix} \ , \quad \|Y\|_2 = 4 \cos \frac{\pi}{n} < 4 \ . \qquad (10.36)$$

We conclude, finally, that if ν is an eigenvalue of $K(t+1,t) \Leftrightarrow I$, then eventually ν must lie in one of the $2(n \Leftrightarrow 2)$ Gershgorin-type disks $D_j^{\pm}$, $1 \leq j \leq n \Leftrightarrow 1$,

$$\{z : |4 + (z/p_j b_j)^2| < \rho_n\} \ , \qquad (10.37)_j$$

where

$$\|Y\|_2 < \rho_n \equiv \frac{1}{2}(\|Y\|_2 + 4) < 4 \qquad (10.38)$$

In particular, this establishes Fig. 1 with $\sin 2\theta = \rho_n/4$.

Inserting a free parameter s, $0 \leq s \leq 1$, in (10.26), as follows,

$$K_s \Leftrightarrow I \equiv \begin{pmatrix} 1 & 0 \\ 0 & B \end{pmatrix} \begin{pmatrix} sE_1 B & \Leftrightarrow 4I + s(Y + O_0(b^2)) \\ 1 + sBE_2 B & sBE_4 \end{pmatrix} \begin{pmatrix} B & 0 \\ 0 & 1 \end{pmatrix} \ ,$$

simple bookkeeping shows that we are led to an eigenvalue problem with spectrum lying in the same disks $(10.37)_j$, uniformly for $0 \leq s \leq 1$ as $t \to \infty$. But the centers of the disks are the eigenvalues of $K_0 \Leftrightarrow I = K_s|_{s=0} \Leftrightarrow I$, and a standard continuity argument in s now shows that each disk $D_j^{\pm}$ contains one eigenvalue of $K \Leftrightarrow I$. (If two disks overlap this means that their union contains two eigenvalues of $K \Leftrightarrow I$, etc.)

Recall from (9.6) that

$$b_i(t) \sim b_i^{\infty} \left(\frac{\sigma_{i+1}}{\sigma_i} \right)^{2t} \ , \quad 1 \leq i \leq n \Leftrightarrow 1 \ , \quad b_i^{\infty} > 0 \ .$$

In the case that the numbers σ_{i+1}/σ_i, $1 \leq i \leq n \Leftrightarrow 1$, are distinct we obtain immediately a proof of **Fact 2**. For in this case the disks $D_j^{\pm}$ are eventually disjoint, and each of the translated disks $1 + D_j^{\pm}$ contains precisely one of the $2(n \Leftrightarrow 2)$ eigenvalues $\{\lambda\}$ of K. Thus the eigenvalues are simple. Suppose $\lambda \in 1 + D_j^{\pm}$ for some j. Then $\overline{\lambda}^{-1}$ is also an eigenvalue by **Fact 1**, and must lie in the same disk, by a simple computation using $(10.37)_j$. By simplicity, we must have $\lambda = \overline{\lambda}^{-1}$, i.e. the roots lie on the unit circle.

Remark 10.39. The idea for the above proof of **Fact 2** was suggested to the authors by Gene Wayne (cf. [dlLW89]).

The remainder of this section is devoted to proving **Fact 1** in the (nongeneric) case where the numbers σ_{i+1}/σ_i are not distinct.

We show first that the eigenvalues $1 + \nu$ are eventually geometrically simple. From (10.34), $(10.33)_I$-$(10.33)_{IV}$, we see that the eigenvalue equation has the form

$$\begin{pmatrix} \Leftrightarrow 4 \Leftrightarrow (\frac{\nu}{p_1 b_1})^2 + o(1) & 2 + o(1) & O(b_2^2) & 0 & & \dots \\ * & \Leftrightarrow 4 \Leftrightarrow (\frac{\nu}{p_2 b_2})^2 + o(1) & 2 + o(1) & O(b_3^2) & 0 & \dots \\ * & * & \Leftrightarrow 4 \Leftrightarrow (\frac{\nu}{p_3 b_3})^2 + o(1) & 2 + o(1) & O(b_4^2) & 0 \dots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ & & & & & \end{pmatrix} \begin{pmatrix} g_1 \\ \cdot \\ \cdot \\ \cdot \\ g_{n-1} \end{pmatrix} = 0 \tag{10.40}$$

where the terms $*$ are bounded as $t \to \infty$. Now if ν were geometrically double, we could take $g_1 = 0$. But by induction

$$\det \begin{pmatrix} 2 + o(1) & O(b_2^2) & 0 & \dots & \\ \Leftrightarrow 4 \Leftrightarrow (\nu / b_2 p_2)^2 + o(1) & 2 + o(1) & O(b_3^2) & 0 & \cdots \\ & \ddots & \ddots & & \\ & & & & \end{pmatrix} = 2^{n-2} + o(1) \neq 0 \ ,$$

which implies $(g_1, \dots, g_{n-1})^T = 0$, contradicting $g \neq 0$. Thus the geometric multiplicity is one.

Remark 10.41. One can show that the eigenvalue equation always takes the form (10.40), even when the b_i's are not small, provided we replace $O(b_2^2)$, $O(b_3^2)$, ... by certain *positive* quantities. This implies, in particular, that the geometric multiplicity of any eigenvalue of $K(t, t+1)$ is at most 2, for all t.

Next we show that for the eigenvalue problem (10.27), as $t \to \infty$,

$$\left(\begin{pmatrix} \overline{f'} \\ \overline{g'} \end{pmatrix} , \ iJ \begin{pmatrix} f' \\ g' \end{pmatrix} \right) > 0 \qquad \text{if} \quad \operatorname{Im} \lambda > 0 \ , \tag{10.42}$$

and

$$\left(\begin{pmatrix} \overline{f'} \\ \overline{g'} \end{pmatrix} , \ iJ \begin{pmatrix} f' \\ g' \end{pmatrix} \right) < 0 \qquad \text{if} \quad \operatorname{Im} \lambda < 0 \ , \tag{10.43}$$

where $(\cdot, \cdot) = (\cdot, \cdot)_m$ denotes the real Euclidean inner product in $\mathbf{R}^m$. Indeed, from (10.27),

$$\begin{aligned} (E_2 + O_0(b^2))g' \ &= (\nu \Leftrightarrow E_1 B^2) f' \\ (1 + B^2 E_3) B^2 f' \ &= (\nu \Leftrightarrow B^2 E_4) g' \end{aligned}$$

which implies

$$\begin{aligned} f' &= B^{-2}(1+B^2E_3)^{-1}(\nu \Leftrightarrow B^2E_4)g' \\ &= (\nu\, B^{-2})g' + O(1)\ , \end{aligned}$$

where we have normalized $\|g'\| = 1$. Thus

$$(\overline{g}', f') = \nu\|B^{-1}g'\|^2 + O(1)\ . \tag{10.44}$$

Now suppose $|\nu\|B^{-1}g'\|^2|$ remains bounded as $t \to \infty$. Then $(\overline{g}', f')$ remains bounded by (10.44) and as $t \to \infty$

$$(\overline{g}', (E_2g')) = O(b^2) + \nu(\overline{g}', f') \Leftrightarrow (\overline{g}', E_1B^2f') \to 0$$

as $b \to 0$, $\nu \to 0$ and $B^2f' = (1+B^2E_3)^{-1}(\nu \Leftrightarrow B^2E_4)g' \to 0$. But this contradicts $|(\overline{g}', E_2g')| \geq$ inf spec$(\Leftrightarrow E_2) > 0$. Hence $\nu\|B^{-1}g'\|^2 \to \infty$ as $t \to \infty$. Formulae (10.42) and (10.43) now follow from (10.44) and Figure 1.

Suppose $\lambda \neq \overline{\lambda}^{-1}$ is an eigenvalue of K, $K\binom{f'}{g'} = \lambda\binom{f'}{g'}$. Then as K is symplectic,

$$\left(\begin{pmatrix}\overline{f}'\\ \overline{g}'\end{pmatrix},\ iJ\begin{pmatrix}f'\\ g'\end{pmatrix}\right) = \left(\overline{K\begin{pmatrix}f'\\ g'\end{pmatrix}},\ iJ\,K\begin{pmatrix}f'\\ g'\end{pmatrix}\right) = \overline{\lambda}\,\lambda\left(\begin{pmatrix}\overline{f}'\\ g'\end{pmatrix},\ iJ\begin{pmatrix}f'\\ g'\end{pmatrix}\right)\ ,$$

which implies $\left(\binom{\overline{f}'}{\overline{g}'}\ ,\ iJ\binom{f'}{g'}\right) = 0$ as $\overline{\lambda}\lambda \neq 1$, contradicting (10.42), (10.43). (Note Im $\lambda \neq 0$, by Figure 1). Thus $|\lambda|^2 = \lambda\overline{\lambda} = 1$. Furthermore, by our previous calculations, λ has geometric multiplicity one. Suppose λ corresponds to a nontrivial Jordan block,

$$\begin{aligned} K &= U\begin{pmatrix} \lambda & 1 & & \\ 0 & \lambda & & \\ & 0 & & \\ \vdots & & \ddots & \ddots \\ & \vdots & & \end{pmatrix}U^{-1}\ , \\ K^T &= U^{-T}\begin{pmatrix} \lambda & 0 & \dots \\ 1 & \lambda & 0 \\ & \ddots & \ddots \\ & & \end{pmatrix}U^T\ . \end{aligned}$$

for some invertible matrix U. Set $\binom{f'}{g'} = Ue_1$; then $K\binom{f'}{g'} = \lambda\binom{f'}{g'}$. But $K^TJ\binom{\overline{f}'}{\overline{g}'} = JK^{-1}\binom{\overline{f}'}{\overline{g}'} = \overline{\lambda}^{-1}J\binom{\overline{f}'}{\overline{g}'}$, so that $J\overline{f} = \sum_{j\neq 1} c_jU^{-T}e_j$ for suitable constants c_j, where $j = 1$ is excluded as $U^{-T}e_1$ is *not* an eigenvector of K^T. Thus

$$\left(\begin{pmatrix}\overline{f}'\\ \overline{g}'\end{pmatrix}, iJ\begin{pmatrix}f'\\ g'\end{pmatrix}\right) = \Leftrightarrow\left(\begin{pmatrix}f'\\ g'\end{pmatrix},\ iJ\begin{pmatrix}\overline{f}'\\ \overline{g}'\end{pmatrix}\right) = \Leftrightarrow\sum_{j\neq 1} c_j(Ue_1, U^{-T}e_j) = 0\ ,$$

again contradicting (10.42), (10.43). We conclude that λ is algebraically simple, and this concludes the proof of **Fact 2**.

Figure 2:

In the initial numerical experiments mentioned in the Introduction, in addition to **Facts 1, 2** and **3**, it was observed that whenever an eigenvalue λ came off the unit circle for finite $t = j$, the break occurred through $\lambda = \Leftrightarrow 1$,
The preceding computations give some insight into, but, as yet, not a complete proof of, this phenomenon. Indeed, if $\lambda_1 \neq \lambda_2$ are two distinct eigenvalues of K, $Kh_i = \lambda_i h_i$, $i = 1, 2$, with positive imaginary parts, then arguing as above, $(\overline{h}_1, iJh_2) = 0$. But then for t large, when the spectrum of $K(t+1,t)$ is simple, we must have

$$(\overline{h}, iJh) > 0 \tag{10.45}$$

for all nonzero h in $V_+(t) \equiv \text{span}\{w : \; K(t+1,t)w = \lambda w, \text{ Im } \lambda > 0\}$. But $V_+(t)$ is clearly continuous in t as long as the spectrum of $K(t+1,t)$ does not cross the real axis. Let $t_0 < \infty$ be the last time for which spec $K(t+1,t) \cap \mathrm{R} \neq \emptyset$. (Such a time may, of course, not exist.) It follows by continuity that (10.45) holds for all $t > t_0$, and using arguments similar to those above, one obtains the following *result*: viewed backwards in time from $t = \infty$, the matrix $K(t+1,t)$ is diagonalizable with spec $K(t+1,t) \subset \{\lambda : |\lambda| = 1\}$, until such a time t_0 that an eigenvalue touches the real axis at $\lambda = +1$ or $\lambda = \Leftrightarrow 1$.

At this point, however, it is not clear how to rule out the case $\lambda = 1$.

Remark 10.46 The reader familiar with dynamical stability theory will recognize that the above computations are modeled on the strong stability theory of Krein ([Kre50, Kre55]; see also [GL55], [Mos58]) for symplectic matrices.

11 Numerical Experiments

In this section we summarize numerical experiments which support the error analysis of sections 5 and 6. The first set of experiments determines the average number of QR steps necessary for convergence, and were performed in [DK88]; we just cite the needed data here. The second set describes the growth of $\|M(t)\|$ for the same set of test problems. The first set of experiments were performed using Fortran on a SUN 4/260 in IEEE standard double precision floating point arithmetic [IEE85]; the machine precision $\varepsilon = 2^{-53} \approx 10^{-16}$ and the range of representable numbers is approximately $10^{\pm 308}$. The second set of experiments were performed using Matlab on the same SUN 4/260 using the same arithmetic.

The test matrices were the same 105 bidiagonal matrices in 12 classes used in [DK88]:

Class 1: These eight matrices are graded in the usual way from large at the upper left to small at the lower right. Four of the matrices are 10 by 10 and four are 20 by 20. The singular values range from 1 to 10^{-90} in some examples.

Class 2: This class is identical to class 1 except the order of the entries on the diagonal and superdiagonal are reversed. Thus these matrices are graded from small at the upper left corner to large at the lower right.

Class 3: These eight 20 by 20 and 40 by 40 matrices are obtained by abutting those in class 1 with their reversals in class 2. Thus each matrix is small at the upper left, large in the middle, and small again at the lower right.

Class 4: These eight 20 by 20 and 40 by 40 matrices are obtained by abutting those in class 2 with their reversals in class 1. Thus each matrix is large at the upper left, small in the middle, and large again at the lower right.

Class 5: These eight matrices are obtained from class 1 by reversing the order of the superdiagonals. Thus the diagonal is graded from large at the upper left to small at the lower right, and the superdiagonal is graded in the opposite direction.

Class 6: These eight matrices are obtained from class 5 by reversing the order of both the diagonals and superdiagonals. Thus the diagonal is graded from small at the upper left to large at the lower right, and the superdiagonal is graded in the opposite direction.

Class 7: These sixteen matrices are all small on the diagonal and mostly large on the offdiagonal. The diagonals range from 10^{-2} down to 10^{-16} and the diagonals are mostly 1 with occasional small values.

Classes 8–11: The ten 20 by 20 matrices in each class are generated by letting each bidiagonal entry be a random number of the form $r \cdot 10^i$, where r is a random number uniformly distributed between ⇔.5 and .5, and i is a random integer. In class 8, i is uniformly distributed from 0 to ⇔15. In class 9, i is uniformly distributed from 0 to ⇔10. In class 10, i is uniformly distributed from 0 to ⇔5. In class 11, i is identically 0. Thus, in class 11 each matrix entry is simply uniformly distributed on [⇔.5, .5].

Class 12: This one 41 by 41 matrix is graded in as in class 1, with the ratio of adjacent entries being $10^{-.1} \approx .79$. Each offdiagonal entry is identical to the diagonal entry below it. This very dense grading leads to different convergence properties than for the matrices in class 1, which is why we put this example in a separate class.

Now we describe the results of the first experiment, which computes the number of QR steps needed for convergence with relative error tolerance $tol = 100\varepsilon \approx 10^{-14}$. This is the number m in the statement of Theorem 6.1. Actually, we compute a related quantity which is more closely related to the actual work done: the number of "QR inner loops" divided by $n(n+1)/2$, where n is the matrix dimension and one "QR inner loop" is one pass through the inner loop of the QR algorithm (shifted or unshifted). The reason for choosing this statistic is as follows. The usual rule of thumb for the number of QR steps it takes to compute the SVD is two steps per singular value [Par80]. If convergence always takes place at the end of the matrix, this means there will be 2 steps on a matrix of length i, for $i = n, n \Leftrightarrow 1, \ldots, 3$ (two by two matrices are handled specially). Thus, since one QR step on a matrix of length i consists of i "QR inner loops", we expect an average of about $n(n+1)$ "QR inner loops" for the entire SVD. Thus, the quantity "QR inner loops" divided by $n(n+1)/2$ should be a measure of the difficulty of computing the SVD of a matrix which is independent of dimension, and we expect it to equal 2 on the average. For each of the twelve problem classes, and for both the algorithm of section 3 and the standard SVD algorithm [BDMS79], the minimum, average and maximum of this statistic is presented in Table 1.

Table 1: QR inner loops / $(n(n+1)/2)$

Class	Standard SVD			New SVD		
	Min	Avg	Max	Min	Avg	Max
1	.60	.90	1.33	.09	.49	1.11
2	.60	1.94	3.07	.09	.49	1.11
3	.61	.85	1.19	.56	.82	1.19
4	.32	1.04	1.80	.35	.60	1.04
5	.07	.45	1.11	.09	.57	1.42
6	.07	.40	.93	.09	.57	1.42
7	.10	1.32	2.31	.10	1.04	1.85
8	.41	.64	.95	.26	.49	.77
9	.79	.94	1.29	.57	.75	.93
10	1.07	1.29	1.57	1.04	1.22	1.48
11	1.97	2.26	2.52	2.06	2.20	2.41
12	1.53	1.53	1.53	2.96	2.96	2.96

Since Class 11 corresponds to matrices with uniform random entries, we see that the rule of thumb of 2 QR steps per singular value is justified. In the worst case, Class 12, the new SVD algorithm takes 3 QR steps per eigenvalue. In the other classes it takes many fewer step to reach convergence. Thus, in practice we can bound the number of QR steps by $m = 3n$ in order to obtain an upper bound depending only on the matrix dimension n

in Theorem 6.1. Of course, after the algorithm has been run m is easily available, so that Theorem 6.1 could be used to get less pessimistic bounds.

The second set of experiments measured $\|M(j,0)\|_\infty$ for the same test cases as above. We computed $M(j,0)$ as follows. The first order perturbation theory in Lemmas 5.2 and 5.4 can be seen as computing the linear operator $M(i+1,i)$ which maps the relative errors $\epsilon_{a_1},\ldots,\epsilon_{a_n},\epsilon_{b_1},\ldots,\epsilon_{b_{n-1}}$ in the entries of the bidiagonal matrix B to the relative errors $\epsilon_{a'_1},\ldots,\epsilon_{a'_n},\epsilon_{b'_1},\ldots,\epsilon_{b'_{n-1}}$ in the entries of the bidiagonal matrix B' after one zero-shift QR step. The entries of this matrix are computed as products of the 2 by 2 matrices appearing in the proofs of Lemmas 5.2 and 5.4, and so the entries of $M(i+1,i)$ are complicated polynomials in the sines of cosines of rotation angles occuring during the running of the algorithm. We obtain $M(j,0) = M(j,j\Leftrightarrow 1)\cdot M(1,0)$ via matrix multiplication.

The experiments were performed by taking each one of the 105 test matrices and running zero-shift QR until $M(i+1,i)$ had converged to its asymptotic value, and the graph of $\|M(j,0)\|_\infty$ versus j had converged to a straight line; this convergence was determined by examining the graph. (This is not the same as computing all the $M(j,i)$ arising during the running of the overall hybrid algorithm on the test cases, but is nonetheless a thorough test of our predicted upper bound $(8n\Leftrightarrow 4)(j\Leftrightarrow i)+O(1)$ of Theorem 9.23 on $\|M(j,i)\|_\infty$.) For each test matrix the computed values of $\|M(j,0)\|_\infty$ were analyzed as follows:

1. Let $j_{\max}$ be the number of QR steps taken and n the matrix dimension.

2. Let $s \equiv (\|M(j_{\max},0)\|_\infty \Leftrightarrow \|M(j_{\max-1},0)\|_\infty)/n$ be the asymptotic rate of growth of $\|M(j,0)\|_\infty$ (divided by n).

3. Let $r \equiv \max_{1\le j\le j_{\max}}(\|M(j,0)\|_\infty \Leftrightarrow n\cdot s\cdot j)$. Then for all $1\le j\le j_{\max}$ we have $\|M(j,0)\|_\infty \le nsj+r$. In other words, the line $nsj+r$ is the tightest affine upper bound to $\|M(j,0)\|_\infty$.

4. Let $t = \max_{1\le j\le j_{\max}}(nsj+r\Leftrightarrow\|M(j,0)\|_\infty)$ be the maximum amount the straight line $nsj+r$ overestimates $\|M(j,0)\|_\infty$.

Table 2: Growth Statistics for $\|M(j,0)\|_\infty$

Class	$j_{\max}$		$\max s$	$\max r$	$\max t$
	min	max			
1	5	20	2.00	-10.00	.26
2	10	80	2.00	-.02	1.23
3	10	80	2.18	-8.67	4.63
4	20	90	1.86	-7.16	3.34
5	20	90	2.10	-1.00	16.02
6	20	90	2.10	1.00	18.00
7	20	90	4.01	-19.00	18.00
8	30	30	5.06	24.73	151.28
9	40	40	4.00	18.16	59.93
10	40	40	3.41	26.29	49.11
11	40	40	4.15	8.71	27.58

Table 2 summarizes the values of s, r and t computed. Columns 2 and 3 give the minimum and maximum values of $j_{\max}$ for the given class. Columns 4, 5 and 6 give the maximum values of r, s and t, respectively, in each class.

The largest overshoot 151.28 corresponds to a 16% change in $nsj_{\max} + r$, and small deviation from linearity. Excluding this case, the maximum deviation is less than 5%. So even though the analysis leading to the linear growth bound on $\|M(j,0)\|_\infty$ was asymptotic, we find linear growth sets in quite early, much earlier than we can currently explain.

The smallest values $\underline{s}$ and $\underline{r}$ which satisfy $n\underline{s}j + \underline{r} \geq nsj + r$ for all 105 (s,r) pairs and all j are $\underline{s} = 5.06$ and $\underline{r} = 0$. This justifies the claims made earlier and used in the error analyses of sections 5 and 6.

12 Conclusions

We have proven that the singular values of a bidiagonal matrix are much less sensitive to relative perturbations in the matrix entries than previously thought: the uncertainty of singular vector v_i is proportional to the reciprocal of the *relative gap* $\min_{j\neq i} |\sigma_i \Leftrightarrow \sigma_j|/(\sigma_i+\sigma_j)$ rather than the reciprocal of the *absolute gap* $\min_{j\neq i} |\sigma_i \Leftrightarrow \sigma_j|/\sigma_{\max}$. When the matrix has two or more tiny singular values, the relative gap can be much larger than the absolute gap, so the bound is much tighter.

We have also shown that the algorithm in [DK88] is capable of computing the singular values to this higher accuracy. The proof involves a new analysis of the stopping criterion, as well as showing that rounding errors during the zero-shift QR algorithm accumulate slowly. This latter analysis is facilitated by associating to zero-shift QR a Hamiltonian differential equation which interpolates the iterates of the algorithm. In contrast to many eigenvalue algorithms where the underlying Hamiltonian structures are Lie-Poisson structures (see, e.g., [DLNT86, DLT89]), here the underlying structure is a so-called Sklyanin structure. The canonical variables on the appropriate symplectic leaves for the bidiagonal case turn out to be linear combinations of logarithms of the matrix entries. The differential equation shows that these canonical variables are relatively insensitive to changes in the initial conditions. Since the canonical variables are essentially logarithms of matrix entries, this means that the logarithms of the matrix entries are insensitive. This in turn means that small relative errors (such as rounding errors) in the matrix entries grow slowly.

If the initial matrix is not bidiagonal, then reduction to bidiagonal form may introduce errors so large as to swamp the finer bounds of the bidiagonal SVD. One possibility is to use Jacobi's method if the original matrix is dense [DV90]. A situation where bidiagonal reduction is sufficiently accurate is the symmetric positive definite tridiagonal eigenproblem, where the reduction is performed via Cholesky [BD88].

The algorithm described here will be part of the LAPACK linear algebra library for supercomputers [DDDC*87].

13 Acknowledgements

Percy Deift acknowledges the support of NSF grant DMS-8802305. James Demmel acknowledges the support of NSF grants DCR-8552474 and ASC-8715728 and DARPA grant F49620-87-C-0065. Luen-Chau Li acknowledges the support of NSF grant DMS-8704097. Carlos Tomei thanks CNPq, Brazil. The authors also thank W. Kahan, who while looking over the shoulder of the second author at a column of numbers noticed that they appeared in reciprocal pairs. They also thank Gene Wayne, who suggested the idea of the proof of Fact 2. Finally, they acknowledge the help of Socrates Rivera who made many of the initial calculations in section 10.

References

[Adl79] M. Adler. On a trace functional for formal pseudodifferential operators and the symplectic structure of the Korteweg-de Vries type equations. *Inv. Math.*, 50:219–248, 1979.

[Arn78] V. I. Arnold. *Mathematical Methods of Classical Mechanics.* Springer-Verlag, New York, 1978.

[BD88] J. Barlow and J. Demmel. *Computing Accurate Eigensystems of Scaled Diagonally Dominant Matrices.* Computer Science Dept. Technical Report 421, Courant Institute, New York, NY, December 1988. submitted to SIAM J. Num. Anal.

[BDMS79] J. Bunch, J. Dongarra, C. Moler, and G. W. Stewart. *LINPACK User's Guide.* SIAM, Philadelphia, PA, 1979.

[Chu86] M. Chu. A differential equation approach to the singular value decomposition of bidiagonal matrices. *Lin. Alg. Appl*, 80:71–80, 1986.

[CL55] E. A. Coddington and N. Levinson. *Theory of ordinary differential equations.* McGraw-Hill, New York, 1955.

[DDDC*87] J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, and D. Sorensen. *Prospectus for the Development of a Linear Algebra Library for High-Performance Computers.* Mathematics and Computer Science Division Report ANL/MCS-TM-97, Argonne National Laboratory, Argonne, IL, September 1987.

[Dem84] J. Demmel. Underflow and the reliability of numerical software. *SIAM Journal on Scientific and Statistical Computing*, 5(4):887–919, Dec 1984.

[DK70] C. Davis and W. Kahan. The rotation of eigenvectors by a perturbation III. *SIAM Journal of Numerical Analysis*, 7:248–263, 1970.

[DK88] J. Demmel and W. Kahan. Accurate Singular Values of Bidiagonal Matrices. *SIAM J. Sci. Stat. Comp.*, v. 11, n. 5, pp. 873–912, Sept 1990.

[dlLW89] R. de la Llave and C. E. Wayne. Whiskered and low dimensional tori in nearly integrable Hamiltonian systems. Preprint, 1989.

[DLNT86] P. Deift, L. C. Li, T. Nanda, and C. Tomei. The Toda flow on a generic orbit is integrable. *Comm. Pure Appl. Math.*, 39:183–232, 1986.

[DLT85] P. Deift, L. C. Li, and C. Tomei. Toda flows with infinitely many variables. *Jour. Func. Anal.*, 64:358–402, 1985.

[DLT89] P. Deift, L. C. Li, and C. Tomei. Matrix factorizations and integrable systems. *Comm. Pure Appl. Math.*, 42:443–521, 1989.

[DNT83] P. Deift, T. Nanda, and C. Tomei. Differential equations for the symmetric eigenvalue problem. *SIAM J. Num. Anal.*, 20:1–22, 1983.

[Dri87] K. R. Driessel. On finding the singular values and singular vectors of a matrix by means of an isosingular gradient flow. Idaho State University, preprint, 1987.

[DV90] J. Demmel and K. Veselić. Jacobi's Method is More Accurate than QR. to appear in SIAM J. Mat. Anal. Appl.

[GK65] G. Golub and W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. *SIAM Journal of Numerical Analysis (Series B)*, 2(2):205–224, 1965.

[GL55] I. M. Gelfand and V. B. Lidsky. On the structure of regions of stability of linear canonical systems of differential equations with periodic coefficients (transl. AMS 2(8)(1958)143-181). *Usp. Math. Nauk.*, 10(1):3–40, 1955.

[GVL83] G. Golub and C. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, MD, 1983.

[IEE85] IEEE Standard for Binary Floating Point Arithmetic, ANSI/IEEE Std 754-1985. IEEE, 1985.

[Kah68] W. Kahan. *Accurate Eigenvalues of a Symmetric Tridiagonal Matrix.* Computer Science Dept. Technical Report CS41, Stanford University, Stanford, CA, July 1966 (revised June 1968).

[Kos79] B. Kostant. The solution to a generalized Toda lattice and representation theory. *Adv. in Math.*, 34:195–338, 1979.

[Kre50] M. G. Krein. A generalization of several investigations of A. M. Lyapounov. *Dokl. Akad. Nauk*, 73:445–448, 1950.

[Kre55] M. G. Krein. The basic propositions of the theory of λ-zones of stability of a canonical system of linear differential equations with periodic coefficients, Pamyati A. A. Androvna. *Izvestia Akad. Nauk.*, 413–498, 1955.

[KvM75] M. Kac and P. van Moerbeke. On some periodic Toda lattice. *Proc. Nat. Acad. Sci., USA*, 72:1627–1629, 1975.

[LPar] L. C. Li and S. Parmentier. Nonlinear Poisson structures and r-matrices. *Comm. Math Phys.*, to appear.

[Mos58] J. Moser. New aspects in the theory of stability of Hamiltonian systems. *Comm. Pure Appl. Math.*, 11:81–114, 1958.

[Mos75] J. Moser. *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, New York, Berlin, Heidelberg, 1975.

[Par80] B. Parlett. *The Symmetric Eigenvalue Problem.* Prentice Hall, Englewood Cliffs, NJ, 1980.

[Sem84] M. Semenov-Tyan-Shanskii. What is a classical r-matrix? *Funct. Anal. Appl.*, 259–272, 1984.

[Sem85] M. Semenov-Tyan-Shanskii. Dressing transformations and Poisson group actions. *Publ. RIMS*, 21:1237–1260, 1985.

[Sym80] W. W. Symes. Hamiltonian group actions and integrable systems. *Physica*, 1D:339–374, 1980.

[Sym82] W. W. Symes. The QR algorithm for the finite nonperiodic Toda lattice. *Physica*, 4D:275–280, 1982.

[Wat84] D. S. Watkins. Isospectral flows. *SIAM Review*, 26:379–391, 1984.

[Wei85] A. Weinstein. The local structure of Poisson manifolds. *J. Diff. Geom.*, 18:523–557, 1985.

[Wil65] J. H. Wilkinson. *The Algebraic Eigenvalue Problem.* Oxford University Press, Oxford, 1965.