Abstract

We present a stable and efficient divide-and-conquer algorithm for computing the singular value decomposition of an $N \times N$ lower bidiagonal matrix. Previous divide-and-conquer algorithms all suffer from a potential loss of orthonormality among the computed singular vectors unless extended precision arithmetic is used. We then show how to use the fast multipole method of Carrier, Greengard and Rokhlin to speed up this algorithm from $O(N^2)$ time to $O(N \log_2 N)$ time for computing all the singular values, and from $O(N^3)$ time to $O(N^2)$ time for computing all the singular values and singular vectors. We also present a generalization that computes the singular value decomposition of a lower banded matrix.

A Divide-And-Conquer Algorithm for the Bidiagonal *SVD*[†]

Ming Gu and Stanley C. Eisenstat

Research Report YALEU/DCS/RR-933
16 December 1992

[†] This research was supported in part by U. S. Army Research Office contract DAAL03-91-G-0032.

1. Introduction

Given an $(N+1) \times N$ lower bidiagonal matrix[1]

$$B = \begin{pmatrix} \alpha_1 & & & \\ \beta_1 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_{N-1} & \alpha_N \\ & & & \beta_N \end{pmatrix} \quad , \qquad (1.1)$$

its *singular value decomposition* (SVD) is

$$B = X \begin{pmatrix} \Omega \\ 0 \end{pmatrix} Y^T \quad ,$$

where X and Y are $(N+1) \times (N+1)$ and $N \times N$ orthonormal matrices, respectively; Ω is an $N \times N$ non-negative diagonal matrix; and 0 is a row of zero elements. The columns of X and Y are the *left singular vectors* and the *right singular vectors* of B, respectively; and the diagonal entries of Ω are the *singular values* of B. This problem arises when one computes the SVD of a general matrix by first reducing it to bidiagonal form [9, 11, 12, 26]. In this paper, we propose a bidiagonal divide-and-conquer algorithm (BDC) for solving this problem.

BDC first divides B as follows:

$$B = \begin{pmatrix} B_1 & \alpha_k e_k & 0 \\ 0 & \beta_k e_1 & B_2 \end{pmatrix} \quad ,$$

where B_1 and B_2 are lower bidiagonal matrices, each of which is a submatrix of B. BDC next recursively computes the SVDs of B_1 and B_2, and then computes orthonormal matrices $(Q \;\; q)$ and W such that

$$B = (Q \;\; q) \begin{pmatrix} M \\ 0 \end{pmatrix} W^T \quad ,$$

where

$$M = \begin{pmatrix} z_1 & & & \\ z_2 & d_2 & & \\ \vdots & & \ddots & \\ z_N & & & d_N \end{pmatrix}$$

is an $N \times N$ matrix with non-zero elements only on its diagonal and in its first column. BDC computes the singular values of B by computing the SVD of M

$$M = U \Omega V^T \quad ,$$

[1] An $N \times N$ lower bidiagonal matrix can be put into the form (1.1) by appending a zero row.

where U and V are orthonormal matrices[2]. *BDC* then computes the singular vector matrices of B as $(QU \quad q)$ and WV, respectively (see Section 2).

Since error is associated with computation, a *numerical SVD* of B or M is usually defined as a decomposition of the form

$$B = \hat{X} \begin{pmatrix} \hat{\Omega} \\ 0 \end{pmatrix} \hat{Y}^T + O(\epsilon \; \|B\|_2) \quad \text{or} \quad M = \hat{U}\hat{\Omega}\hat{V}^T + O(\epsilon \; \|M\|_2) \quad , \qquad (1.2)$$

where ϵ is the machine precision; $\hat{\Omega}$ is diagonal; and $\hat{X}$ and $\hat{Y}$ or $\hat{U}$ and $\hat{V}$ are *numerically orthonormal.* An algorithm that produces such a decomposition is said to be *backward stable* [26].

While the singular values of B and M are always well-conditioned with respect to perturbations, the singular vectors can be extremely sensitive to perturbations [12, 24, 26]. That is, $\hat{\Omega}$ can be guaranteed to be close to Ω, but $\hat{X}$, $\hat{Y}$, $\hat{U}$ and $\hat{V}$ can be very different from X, Y, U and V, respectively. Thus one is usually content with backward stable algorithms for computing the *SVD* of B or M.

The divide-and-conquer algorithm developed by Jessup and Sorensen [21] uses basically the same dividing strategy. The scheme it uses for computing the *SVD* of M is based on work in [4, 5, 8]. While it can compute the singular values of M to high absolute accuracy, in the presence of close singular values it can have difficulties in computing numerically orthonormal singular vectors, unless extended precision arithmetic is used [21, 22, 25].

In this paper we develop a new scheme for computing the *SVD* of M. It uses an approach similar to that of Jessup and Sorensen [21] for computing the singular values. But it uses a completely different approach for computing the singular vectors, one that is backward stable. The amount of work is roughly the same, yet it does not require the use or simulation of extended precision arithmetic (see Section 3). Since it uses this scheme, *BDC* is backward stable as well. Moreover, *BDC* uses a different procedure for handling deflation, which allows it to be asymptotically twice as fast as that of Jessup and Sorensen [21] (see Section 4).

There are three other divide-and-conquer algorithms for the bidiagonal *SVD*. Arbenz and Golub [3] use a similar dividing strategy that reduces the problem of computing the *SVD* of B to that of M^T. Arbenz [1] and Gragg, Thornton and Warner [13] compute the *SVD* of B by computing the eigendecomposition of the matrix $\begin{pmatrix} 0 & B^T \\ B & 0 \end{pmatrix}$. All three algorithms can be unstable as noted above. The techniques presented in this paper can be used to stablize and speed up these algorithms as well.

[2] Since $\begin{pmatrix} M \\ 0 \end{pmatrix}$ is obtained from B by orthonormal transformations, M and B have the same singular values.

BDC computes all the singular values in $O(N^2)$ time and all the singular values and singular vectors in $O(N^3)$ time. We show that by using the fast multipole method of Carrier, Greengard and Rokhlin [6, 14], *BDC* can be accelerated to compute all the singular values in $O(N \log_2 N)$ time and all the singular values and singular vectors in $O(N^2)$ time. These asymptotic times are better than the corresponding worst-case time requirements ($O(N^2)$ and $O(N^3)$) for the Golub-Kahan algorithm [9, 11, 12] and bisection with inverse iteration [19, 20]. This is an important advantage of *BDC* for large matrices.

We take the usual model of arithmetic[3]

$$fl(\alpha \circ \beta) = (\alpha \circ \beta)\,(1+\xi) \quad ,$$

where α and β are floating point numbers; $\circ$ is one of $+, -, \times$, and $\div$; $fl(\alpha \circ \beta)$ is the floating point result of the operation $\circ$; and $|\xi| \le \epsilon$. We also require that

$$fl(\sqrt{\alpha}) = \sqrt{\alpha}\,(1+\xi)$$

for any positive floating point number α. For simplicity, we ignore the possibility of overflow and underflow.

Section 2 presents the dividing strategy; Section 3 presents a scheme for computing the *SVD* of M; Section 4 presents the deflation procedure; Section 5 discusses the application of the fast multipole method to speed up *BDC*; and Section 6 generalizes *BDC* to compute the *SVD* of a lower banded matrix.

2. "Dividing" the Matrix

Given an $(N+1) \times N$ lower bidiagonal matrix B, *BDC* recursively divides B into two subproblems as follows (cf. [21]):

$$B = \begin{pmatrix} B_1 & \alpha_k e_k & 0 \\ 0 & \beta_k e_1 & B_2 \end{pmatrix} \quad , \tag{2.1}$$

where B_1 and B_2 are $k \times (k-1)$ and $(N-k+1) \times (N-k)$ lower bidiagonal matrices respectively; and e_j is the j-th unit vector of appropriate dimension. Usually $k < N$ is taken to be $\lfloor N/2 \rfloor$.

Assume that we are given the *SVD*s

$$B_i = (Q_i \;\; q_i) \begin{pmatrix} D_i \\ 0 \end{pmatrix} W_i^T \quad ,$$

where $(Q_i \;\; q_i)$ and W_i are orthonormal matrices; and D_i is a non-negative diagonal matrix. Let l_1^T and λ_1 be the last row and last component of Q_1 and q_1, respectively; and let f_2^T and

[3] This model excludes machines like CRAYs and CDC Cybers that do not have a guard digit. *BDC* can easily be modified for such machines.

φ_2 be the first row and first component of Q_2 and q_2, respectively. Plugging these into (2.1), we get

$$
\begin{aligned}
B &= \begin{pmatrix} (Q_1 \;\; q_1)\begin{pmatrix} D_1 \\ 0 \end{pmatrix} W_1^T & \alpha_k e_k & 0 \\ 0 & \beta_k e_1 & (Q_2 \;\; q_2)\begin{pmatrix} D_2 \\ 0 \end{pmatrix} W_2^T \end{pmatrix} \\
&= \begin{pmatrix} Q_1 & q_1 & 0 & 0 \\ 0 & 0 & Q_2 & q_2 \end{pmatrix} \begin{pmatrix} D_1 W_1^T & \alpha_k Q_1^T e_k & 0 \\ 0 & \alpha_k q_1^T e_k & 0 \\ 0 & \beta_k Q_2^T e_1 & D_2 W_2^T \\ 0 & \beta_k q_2^T e_1 & 0 \end{pmatrix} \\
&= \begin{pmatrix} q_1 & Q_1 & 0 & 0 \\ 0 & 0 & Q_2 & q_2 \end{pmatrix} \begin{pmatrix} \alpha_k \lambda_1 & 0 & 0 \\ \alpha_k l_1 & D_1 & 0 \\ \beta_k f_2 & 0 & D_2 \\ \beta_k \varphi_2 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & W_1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & W_2 \end{pmatrix}^T \quad . \qquad (2.2)
\end{aligned}
$$

Note that there is only one non-zero element in the first and last rows of the middle matrix ($\alpha_k \lambda_1$ and $\beta_k \varphi_2$, respectively). We apply a Givens rotation to zero out $\beta_k \varphi_2$. Let

$$
r_0 = \sqrt{(\alpha_k \lambda_1)^2 + (\beta_k \varphi_2)^2} \quad , \quad c_0 = \frac{\alpha_k \lambda_1}{r_0} \quad \text{and} \quad s_0 = \frac{\beta_k \varphi_2}{r_0} \quad . \qquad (2.3)
$$

Then we have

$$
\begin{aligned}
B &= \begin{pmatrix} c_0 q_1 & Q_1 & 0 & -s_0 q_1 \\ s_0 q_2 & 0 & Q_2 & c_0 q_2 \end{pmatrix} \begin{pmatrix} r_0 & 0 & 0 \\ \alpha_k l_1 & D_1 & 0 \\ \beta_k f_2 & 0 & D_2 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & W_1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & W_2 \end{pmatrix}^T \\
&= (Q \;\; q) \begin{pmatrix} M \\ 0 \end{pmatrix} W^T \qquad (2.4)
\end{aligned}
$$

where

$$
Q = \begin{pmatrix} c_0 q_1 & Q_1 & 0 \\ s_0 q_2 & 0 & Q_2 \end{pmatrix} \quad , \quad q = \begin{pmatrix} -s_0 q_1 \\ c_0 q_2 \end{pmatrix} \quad \text{and} \quad W = \begin{pmatrix} 0 & W_1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & W_2 \end{pmatrix} \quad ,
$$

and

$$
M = \begin{pmatrix} r_0 & 0 & 0 \\ \alpha_k l_1 & D_1 & 0 \\ \beta_k f_2 & 0 & D_2 \end{pmatrix} \quad .
$$

In equation (2.4), B is reduced to $\begin{pmatrix} M \\ 0 \end{pmatrix}$ by orthonormal transformations $(Q \;\; q)$ and

W. M can be non-zero only on its diagonal and in its first column. Let $U\Omega V^T$ be the SVD of M computed using the scheme described in Section 3. Plugging it into (2.4), we obtain

$$B = (Q \ \ q)\begin{pmatrix} U\Omega V^T \\ 0 \end{pmatrix} W^T = (QU \ \ q)\begin{pmatrix} \Omega \\ 0 \end{pmatrix}(WV)^T = X\begin{pmatrix} \Omega \\ 0 \end{pmatrix}Y^T \ ,$$

where $X = (QU \ \ q)$ and $Y = WV$.

The singular values of B are the diagonal elements of Ω. The singular vectors of B are the columns of X and Y. To compute the SVDs of B_1 and B_2, this process (equations (2.1) and (2.4)) can be recursively applied until the subproblems are sufficiently small. These small subproblems are then solved using the Golub-Kahan algorithm [9, 11]. There can be at most $O(\log_2 N)$ levels of recursion.

Equations (2.1) and (2.4) also suggest a recursion for computing only the singular values. Let f_1^T and φ_1 be the first row of Q_1 and the first component of q_1, respectively; let l_2^T and λ_2 be the last row of Q_2 and last component of q_2, respectively; let f^T and φ be the first row of Q and q, respectively; and let l^T and λ be the last row of Q and q, respectively. Suppose that D_i, f_i, λ_i, l_i and φ_i are given for $i = 1, 2$. Then we can compute Ω, f, λ, l, and φ by computing r_0, s_0 and c_0 using equation (2.3); computing the SVD of $M = U\Omega V^T$; and computing

$$\lambda = -s_0\lambda_1 \ , \ \ \varphi = c_0\varphi_2 \ , \ \ f^T = (f_1^T \ 0)\,U \quad \text{and} \ l^T = (0 \ l_2^T)\,U \ .$$

The divide-and-conquer algorithms in [7, 16] for the symmetric tridiagonal eigenproblem have a similar recursion for the eigenvalues.

3. Computing the SVD of M

In this section we present a scheme for finding the SVD of the matrix

$$M = \begin{pmatrix} z_1 & & & \\ z_2 & d_2 & & \\ \vdots & & \ddots & \\ z_n & & & d_n \end{pmatrix} \ , \tag{3.1}$$

where $D = diag(d_1, d_2, \ldots, d_n)$ with[4] $0 \equiv d_1 \le d_2 \le \ldots \le d_n$; and $z = (z_1, z_2, \ldots, z_n)^T$. We further assume that

$$d_{j+1} - d_j \ge \tau\|M\|_2 \quad \text{and} \quad |z_j| \ge \tau\|M\|_2 \ , \tag{3.2}$$

where τ is a small multiple of ϵ to be specified in Section 3.3. Any matrix of the form (3.1) can be reduced to one that satisfies these conditions by the deflation procedure described in Section 4.1 and a simple permutation.

[4] d_1 is introduced to simplify the presentation.

The following lemma characterizes the singular values and singular vectors of M.

LEMMA 1 (JESSUP AND SORENSEN [21]). *Let* $U\Omega V^T$ *be the SVD of* M *with*

$$U = (u_1, \ldots, u_n) \quad , \quad \Omega = \mathrm{diag}(\omega_1, \ldots, \omega_n) \quad \textit{and} \quad V = (v_1, \ldots, v_n) \quad ,$$

where $0 < \omega_1 < \ldots < \omega_n$. *Then*

$$MM^T = D^2 + zz^T = U\Omega^2 U^T \quad .$$

is the eigendecomposition of MM^T. *The singular values* $\{\omega_i\}_{i=1}^n$ *satisfy the* interlacing property

$$0 = d_1 < \omega_1 < d_2 < \ldots < d_n < \omega_n < d_n + ||z||_2 \quad ,$$

and the secular equation

$$f(\omega) = 1 + \sum_{k=1}^{n} \frac{z_k^2}{d_k^2 - \omega^2} = 0 \quad .$$

The singular vectors satisfy

$$u_i = \left(\frac{z_1}{d_1^2 - \omega_i^2}, \ldots, \frac{z_n}{d_n^2 - \omega_i^2} \right)^T \Bigg/ \sqrt{\sum_{k=1}^{n} \frac{z_k^2}{(d_k^2 - \omega_i^2)^2}} \quad , \tag{3.3}$$

$$v_i = \left(-1, \frac{d_2 z_2}{d_2^2 - \omega_i^2}, \ldots, \frac{d_n z_n}{d_n^2 - \omega_i^2} \right)^T \Bigg/ \sqrt{1 + \sum_{k=2}^{n} \frac{(d_k z_k)^2}{(d_k^2 - \omega_i^2)^2}} \quad . \tag{3.4}$$

On the other hand, given D and all the singular values, we can construct a matrix with the same structure as (3.1).

LEMMA 2. *Given a diagonal matrix* $D = \mathrm{diag}(d_1, d_2, \ldots, d_n)$ *and a set of numbers* $\{\hat{\omega}_i\}_{i=1}^n$ *satisfying the interlacing property*

$$0 \equiv d_1 < \hat{\omega}_1 < d_2 < \ldots < d_n < \hat{\omega}_n \quad , \tag{3.5}$$

there exists a matrix

$$\hat{M} = \begin{pmatrix} \hat{z}_1 & & & \\ \hat{z}_2 & d_2 & & \\ \vdots & & \ddots & \\ \hat{z}_n & & & d_n \end{pmatrix}$$

whose singular values are $\{\hat{\omega}_i\}_{i=1}^n$. *The vector* $\hat{z} = (\hat{z}_1, \hat{z}_2, \ldots, \hat{z}_n)^T$ *is determined by*

$$|\hat{z}_i| = \sqrt{(\hat{\omega}_n^2 - d_i^2) \prod_{k=1}^{i-1} \frac{(\hat{\omega}_k^2 - d_i^2)}{(d_k^2 - d_i^2)} \prod_{k=i}^{n-1} \frac{(\hat{\omega}_k^2 - d_i^2)}{(d_{k+1}^2 - d_i^2)}} \quad , \tag{3.6}$$

where the sign of $\hat{z}_i$ *can be chosen arbitrarily.*

Proof: Assume that $\hat{z}$ exists. By Lemma 1 we have

$$\det\left(D^2 + \hat{z}\hat{z}^T - \omega^2 I\right) = \prod_{k=1}^{n} \left(\hat{\omega}_k^2 - \omega^2\right) \quad .$$

On the other hand,

$$\begin{aligned} \det\left(D^2 + \hat{z}\hat{z}^T - \omega^2 I\right) &= \det\left(I + \left(D^2 - \omega^2 I\right)^{-1} \hat{z}\hat{z}^T\right) \det\left(D^2 - \omega^2 I\right) \\ &= \left(1 + \sum_{k=1}^{n} \frac{\hat{z}_k^2}{d_k^2 - \omega^2}\right) \prod_{k=1}^{n} \left(d_k^2 - \omega^2\right) \quad . \end{aligned}$$

Combining these two equations,

$$\prod_{k=1}^{n} \left(\hat{\omega}_k^2 - \omega^2\right) = \left(1 + \sum_{k=1}^{n} \frac{\hat{z}_k^2}{d_k^2 - \omega^2}\right) \prod_{k=1}^{n} \left(d_k^2 - \omega^2\right) \quad .$$

Setting $\omega = d_i$, we get

$$\hat{z}_i^2 = \frac{\prod_k \left(\hat{\omega}_k^2 - d_i^2\right)}{\prod_{k \neq i} \left(d_k^2 - d_i^2\right)} \quad .$$

Because of the interlacing property (3.5), the expression on the right hand side is positive. Taking square roots we get (3.6). Working backward, if $\hat{z}$ is given by (3.6), then the singular values of $\hat{M}$ are $\{\hat{\omega}_i\}_{i=1}^n$. ■

3.1. Computing the Singular Vectors of M

For each *exact* singular value ω_i, equations (3.3) and (3.4) give the corresponding *exact* singular vectors. Observe that if ω_i was given, then we could compute each difference $d_k^2 - \omega_i^2$ to high relative accuracy as $(d_k - \omega_i)(d_k + \omega_i)$. We could also compute each product and each ratio to high relative accuracy, and thus compute u_i and v_i to component-wise high relative accuracy.

In practice we can only hope to compute an approximation $\hat{\omega}_i$ to ω_i. But problems can arise if we approximate u_i and v_i by

$$\hat{u}_i = \left(\frac{z_1}{d_1^2 - \hat{\omega}_i^2}, \ldots, \frac{z_n}{d_n^2 - \hat{\omega}_i^2}\right)^T \Bigg/ \sqrt{\sum_{k=1}^{n} \frac{z_k^2}{\left(d_k^2 - \hat{\omega}_i^2\right)^2}}$$

and

$$\hat{v}_i = \left(-1, \frac{d_2 z_2}{d_2^2 - \hat{\omega}_i^2}, \ldots, \frac{d_n z_n}{d_n^2 - \hat{\omega}_i^2}\right)^T \Bigg/ \sqrt{1 + \sum_{k=2}^{n} \frac{(d_k z_k)^2}{\left(d_k^2 - \hat{\omega}_i^2\right)^2}}$$

(i.e., replace ω_i by $\hat{\omega}_i$ in equations (3.3) and (3.4) as in [21]). For even if $\hat{\omega}_i$ is close to ω_i, the approximate ratios $z_k/(d_k^2 - \hat{\omega}_i^2)$ and $d_k z_k/(d_k^2 - \hat{\omega}_i^2)$ can still be very different from the exact ratios $z_k/(d_k^2 - \omega_i^2)$ and $d_k z_k/(d_k^2 - \omega_i^2)$, resulting in singular vectors very different from u_i and v_i. And when all the approximate singular values $\{\hat{\omega}_i\}_{i=1}^n$ are computed and all the corresponding singular vectors are approximated in this manner, the resulting singular vector matrices may not be orthonormal.

Lemma 2 allows us to overcome this problem. After we have computed all the approximate singular values $\{\hat{\omega}_i\}_{i=1}^n$ of M, we find a *new* matrix $\hat{M}$ whose *exact* singular values are $\{\hat{\omega}_i\}_{i=1}^n$, and then compute the singular vectors of $\hat{M}$ using Lemma 1. Note that each difference

$$\hat{\omega}_k^2 - d_i^2 = (\hat{\omega}_k - d_i)(\hat{\omega}_k + d_i) \quad \text{and} \quad d_k^2 - d_i^2 = (d_k - d_i)(d_k + d_i)$$

in (3.6) can be computed to high relative accuracy. Each ratio and each product can also be computed to high relative accuracy. Thus $|\hat{z}_i|$ can be computed to high relative accuracy. We choose the sign of $\hat{z}_i$ to be the sign of z_i. Substituting the *exact* singular values $\{\hat{\omega}_i\}_{i=1}^n$ and the computed $\hat{z}$ into equations (3.3) and (3.4), each singular vector of $\hat{M}$ can again be computed to component-wise high relative accuracy. Consequently, after all the singular vectors are computed, the singular vector matrices of $\hat{M}$ will be numerically orthonormal.

To ensure the existence of $\hat{M}$, we need $\{\hat{\omega}_i\}_{i=1}^n$ to satisfy the interlacing property (3.5). But since the exact singular values of M satisfy the same interlacing property (see Lemma 1), this is only an accuracy requirement on the computed singular values, and is not an additional restriction on M. We can use the *SVD* of $\hat{M}$ as an approximation to the *SVD* of M. Since

$$M = \hat{M} + \begin{pmatrix} z_1 - \hat{z}_1 & & & \\ z_2 - \hat{z}_2 & 0 & & \\ \vdots & & \ddots & \\ z_n - \hat{z}_n & & & 0 \end{pmatrix} ,$$

we have

$$|\hat{\omega}_i - \omega_i| \le \|M - \hat{M}\|_2 \le \|z - \hat{z}\|_2 \quad .$$

Such a substitution is backward stable (see (1.2)) as long as $\hat{z}$ is close to z (cf. [17]).

3.2. Computing the Singular Values of M

In order to guarantee that $\hat{z}$ is close to z, we must ensure that the approximations $\{\hat{\omega}_i\}_{i=1}^n$ to the singular values are sufficiently accurate. The key is the stopping criterion for the root-finder, which requires a slight reformulation of the secular equation (cf. [4, 17]).

Consider the singular value $\omega_i \in (d_i, d_{i+1})$, where $1 \le i \le n-1$; the case $i = n$ is considered later. ω_i is a root of the secular equation

$$f(\omega) = 1 + \sum_{j=1}^{n} \frac{z_j^2}{d_j^2 - \omega^2} = 0 \quad .$$

We first assume that[5] $\omega_i \in (d_i, \frac{d_i+d_{i+1}}{2})$. Let $\delta_j = d_j - d_i$ and let

$$\psi(\mu) \equiv \sum_{j=1}^{i} \frac{z_j^2}{(\delta_j - \mu)(d_j + d_i + \mu)} \quad \text{and} \quad \phi(\mu) \equiv \sum_{j=i+1}^{n} \frac{z_j^2}{(\delta_j - \mu)(d_j + d_i + \mu)} \quad .$$

Setting $\omega = d_i + \mu$, we have

$$f(\mu + d_i) = 1 + \psi(\mu) + \phi(\mu) \equiv g(\mu) \quad .$$

We seek the root $\mu_i = \omega_i - d_i \in (0, \delta_{i+1}/2)$ of $g(\mu) = 0$.

An important property of $g(\mu)$ is that we can compute each difference $\delta_j - \mu$ to high relative accuracy for any $\mu \in (0, \delta_{i+1}/2)$. Indeed, since $\delta_i = 0$, we have $fl(\delta_i - \mu) = -fl(\mu)$; since $fl(\delta_{i+1}) = fl(d_{i+1} - d_i)$ and $0 < \mu < (d_{i+1} - d_i)/2$, we can compute $fl(\delta_{i+1} - \mu)$ as $fl(fl(d_{i+1} - d_i) - fl(\mu))$; and in a similar fashion, we can compute $\delta_j - \mu$ to high relative accuracy for any $j \neq i, i+1$.

Because we can also compute $d_j + d_i + \mu$ (a sum of positive terms) to high relative accuracy, we can compute each ratio $z_j^2/((\delta_j - \mu)(d_j + d_i + \mu))$ in $g(\mu)$ to high relative accuracy for any $\mu \in (0, \delta_{i+1}/2)$. And, since both $\psi(\mu)$ and $\phi(\mu)$ are sums of terms of the same sign, we can bound the error in computing $g(\mu)$ by

$$\eta n(1 + |\psi(\mu)| + |\phi(\mu)|) \quad ,$$

where η is a small multiple of ϵ that is independent of n and μ.

We now assume that $\omega_i \in [\frac{d_i+d_{i+1}}{2}, d_{i+1})$. Let $\delta_j = d_j - d_{i+1}$ and let

$$\psi(\mu) \equiv \sum_{j=1}^{i} \frac{z_j^2}{(\delta_j - \mu)(d_j + d_{i+1} + \mu)} \quad \text{and} \quad \phi(\mu) \equiv \sum_{j=i+1}^{n} \frac{z_j^2}{(\delta_j - \mu)(d_j + d_{i+1} + \mu)} \quad .$$

Setting $\omega = d_{i+1} + \mu$, we seek the root $\mu_i = \omega_i - d_{i+1} \in [\delta_i/2, 0)$ of the equation

$$g(\mu) \equiv f(\mu + d_{i+1}) = 1 + \psi(\mu) + \phi(\mu) = 0 \quad .$$

For any $\mu \in [\delta_i/2, 0)$, we can compute each difference $\delta_j - \mu$ to high relative accuracy. Since $|\mu| \leq |\delta_i|/2 \leq d_{i+1}/2$, we can compute each sum $d_j + d_{i+1} + \mu$ to high relative accuracy as $d_j + (d_{i+1} + \mu)$. Thus we can again compute each ratio $z_j^2/((\delta_j - \mu)(d_j + d_{i+1} + \mu))$ to high relative accuracy and bound the error in computing $g(\mu)$ as before.

Finally we consider the case $i = n$. Let $\delta_j = d_j - d_n$ and let

$$\psi(\mu) \equiv \sum_{j=1}^{n} \frac{z_j^2}{(\delta_j - \mu)(d_j + d_n + \mu)} \quad \text{and} \quad \phi(\mu) \equiv 0 \quad .$$

[5] This can easily be checked by computing $f(\frac{d_i+d_{i+1}}{2})$. If $f(\frac{d_i+d_{i+1}}{2}) > 0$, then $\omega_i \in (d_i, \frac{d_i+d_{i+1}}{2})$, otherwise $\omega_i \in [\frac{d_i+d_{i+1}}{2}, d_{i+1})$.

Setting $\omega = d_n + \mu$, we seek the root $\mu_n = \omega_n - d_n \in (0, \|z\|_2)$ of the equation

$$g(\mu) \equiv f(\mu + d_n) = 1 + \psi(\mu) + \phi(\mu) = 0 \quad .$$

Again, for any $\mu \in (0, \|z\|_2)$, we can compute each ratio $z_j^2/((\delta_j - \mu)(d_j + d_n + \mu))$ to high relative accuracy, and we can bound the error in computing $g(\mu)$ as before.

In practice the root-finder cannot make any progress at a point μ where it is impossible to determine the sign of $g(\mu)$ numerically. Thus we propose the stopping criterion

$$|g(\mu)| \leq \eta n \,(1 + |\psi(\mu)| + |\phi(\mu)|) \quad , \tag{3.7}$$

where, as before, $\eta n(1+|\psi(\mu)|+|\phi(\mu)|)$ is an upper bound on the round-off error in computing $g(\mu)$. Note that for each i, there is at least one floating point number that satisfies this stopping criterion numerically, namely $\mathit{fl}(\mu_i)$.

We have not specified the scheme for finding the root of $g(\mu)$. We can use the bisection method or the rational interpolation strategies in [4, 13, 23]. What is most important is the stopping criterion and the fact that, with the reformulation of the secular equation given above, we can find a μ that satisfies it.

3.3. Numerical Stability

In this subsection we show that $\hat{z}$ is indeed close to z, as long as the root-finder guarantees that each computed μ_i satisfies the stopping criterion.

Since $f(\omega_i) = 0$, we have

$$1 = -\sum_{j=1}^{n} \frac{z_j^2}{d_j^2 - \omega_i^2} \leq \sum_{j=1}^{n} \frac{z_j^2}{|d_j^2 - \omega_i^2|} \quad ,$$

and the stopping criterion (3.7) implies that the computed singular value $\hat{\omega}_i$ satisfies

$$|f(\hat{\omega}_i)| \leq \eta n \left(\sum_{j=1}^{n} \frac{z_j^2}{|d_j^2 - \omega_i^2|} + \sum_{j=1}^{n} \frac{z_j^2}{|d_j^2 - \hat{\omega}_i^2|} \right) \quad .$$

Since

$$f(\hat{\omega}_i) = f(\hat{\omega}_i) - f(\omega_i) = (\hat{\omega}_i^2 - \omega_i^2) \sum_{j=1}^{n} \frac{z_j^2}{(d_j^2 - \hat{\omega}_i^2)(d_j^2 - \omega_i^2)} \quad ,$$

it follows that

$$|\hat{\omega}_i^2 - \omega_i^2| \sum_{j=1}^{n} \frac{z_j^2}{|(d_j^2 - \hat{\omega}_i^2)(d_j^2 - \omega_i^2)|} \leq \eta n \left(\sum_{j=1}^{n} \frac{z_j^2}{|d_j^2 - \hat{\omega}_i^2|} + \sum_{j=1}^{n} \frac{z_j^2}{|d_j^2 - \omega_i^2|} \right) \quad . \tag{3.8}$$

Note that for any j,

$$\frac{1}{|d_j^2 - \hat{\omega}_i^2|} + \frac{1}{|d_j^2 - \omega_i^2|} \leq \frac{2}{|(d_j^2 - \hat{\omega}_i^2)(d_j^2 - \omega_i^2)|^{\frac{1}{2}}} + \frac{|\hat{\omega}_i^2 - \omega_i^2|}{|(d_j^2 - \hat{\omega}_i^2)(d_j^2 - \omega_i^2)|} \quad .$$

Plugging this into (3.8) and using the Cauchy-Schwartz inequality, we get

$$\begin{aligned} |\hat{\omega}_i^2 - \omega_i^2| \sum_{j=1}^{n} \frac{z_j^2}{|(d_j^2 - \hat{\omega}_i^2)(d_j^2 - \omega_i^2)|} & \\ & \leq \frac{2\eta n}{1 - \eta n} \sum_{j=1}^{n} \frac{z_j^2}{|(d_j^2 - \hat{\omega}_i^2)(d_j^2 - \omega_i^2)|^{\frac{1}{2}}} \\ & \leq \frac{2\eta n}{1 - \eta n} \|z\|_2 \sqrt{\sum_{j=1}^{n} \frac{z_j^2}{|(d_j^2 - \hat{\omega}_i^2)(d_j^2 - \omega_i^2)|}} \quad , \end{aligned}$$

or

$$\begin{aligned} |\hat{\omega}_i^2 - \omega_i^2| & \leq \frac{2\eta n}{1 - \eta n} \|z\|_2 \Big/ \sqrt{\sum_{j=1}^{n} \frac{z_j^2}{|(d_j^2 - \hat{\omega}_i^2)(d_j^2 - \omega_i^2)|}} \\ & \leq \frac{2\eta n \|z\|_2}{(1 - \eta n)|z_j|} \sqrt{|(d_j^2 - \hat{\omega}_i^2)(d_j^2 - \omega_i^2)|} \\ & \leq \frac{2\eta n \|z\|_2}{(1 - \eta n)|z_j|} \left(|d_j^2 - \omega_i^2| + \frac{1}{2} |\hat{\omega}_i^2 - \omega_i^2| \right) \quad . \end{aligned}$$

Letting $\xi_j = 2\eta n \|z\|_2 / ((1 - \eta n)|z_j|)$, this implies that

$$|\hat{\omega}_i^2 - \omega_i^2| \leq \frac{\xi_j}{1 - \frac{1}{2}\xi_j} |d_j^2 - \omega_i^2| \tag{3.9}$$

for every $1 \leq j \leq n$, provided that $\xi_j < 2$.

Let $\hat{\omega}_i^2 - \omega_i^2 = \alpha_{ij}(d_j^2 - \omega_i^2)/z_j$ for all i and j. Suppose that we pick $\tau = 2\eta n^2$ in (3.2). Then $|z_j| \geq 2\eta n^2 \|z\|_2$. Assume further that $\eta n < 1/100$. Then $\xi_j \leq 2/3$, and (3.9) implies that $|\alpha_{ij}| \leq \alpha \equiv 4\eta n \|z\|_2$ for all i and j. Thus

$$|\hat{z}_i| = \sqrt{\frac{\prod_j (\hat{\omega}_j^2 - d_i^2)}{\prod_{j \neq i} (d_j^2 - d_i^2)}} = \sqrt{\frac{\prod_j (\omega_j^2 - d_i^2)(1 + \alpha_{ji}/z_i)}{\prod_{j \neq i} (d_j^2 - d_i^2)}} = |z_i| \sqrt{\prod_{j=1}^{n} \left(1 + \frac{\alpha_{ji}}{z_i}\right)}$$

and, since $\hat{z}_i$ and z_i have the same sign,

$$\begin{aligned} |\hat{z}_i - z_i| & = |z_i| \left| \sqrt{\prod_{j=1}^{n} \left(1 + \frac{\alpha_{ji}}{z_i}\right)} - 1 \right| \leq |z_i| \left(\left(1 + \frac{\alpha}{|z_i|}\right)^{\frac{n}{2}} - 1 \right) \\ & \leq |z_i| \left(\exp\left(\frac{\alpha n}{2|z_i|}\right) - 1 \right) \leq (e - 1)\, \alpha n / 2 \\ & \leq 4\eta n^2 \|z\|_2 \quad , \end{aligned} \tag{3.10}$$

where we have used the fact that $\alpha n/(2|z_i|) \leq 1$ and that $(\exp(x) - 1)/x \leq e - 1$ for $0 < x \leq 1$.

One factor of n in τ and (3.10) comes from the stopping criterion (3.7). This is quite conservative and could be reduced to $\log_2 n$ by using a binary tree structure for summing up the terms in $\psi(\mu)$ and $\phi(\mu)$. The other factor of n comes from the upper bound for $\prod_j(1+\alpha_{ji}/z_i)$. This also seems quite conservative. Thus we might expect the factor of n^2 in τ and (3.10) to be more like $O(n)$ in practice.

4. Deflation

4.1. Deflation for M

Let

$$M = \begin{pmatrix} z_1 & & & \\ z_2 & d_2 & & \\ \vdots & & \ddots & \\ z_n & & & d_n \end{pmatrix} ,$$

where $D = diag(d_1, d_2, \dots, d_n)$ with $d_1 \equiv 0$ and $d_i \geq 0$; and $z = (z_1, z_2, \dots, z_n)^T$. We now show that we can backward stably reduce M to a similar matrix which further satisfies (cf. (3.2))

$$|d_i - d_j| \geq \tau \|M\|_2 \quad \text{for} \quad i \neq j \ , \qquad \text{and} \qquad |z_i| \geq \tau\|M\|_2 \ ,$$

where τ is specified in Section 3.3. We illustrate the reductions for $n = 3$. Similar reductions appear in [21].

First assume that $|z_1| < \tau\|M\|_2$, then we set z_1 to $\tau\|M\|_2$:

$$M = \begin{pmatrix} z_1 & & \\ z_2 & d_2 & \\ z_3 & & d_3 \end{pmatrix} = \begin{pmatrix} \tau\|M\|_2 & & \\ z_2 & d_2 & \\ z_3 & & d_3 \end{pmatrix} + O(\tau\|M\|_2) \ . \tag{4.1}$$

The matrix M is perturbed by $O(\tau\|M\|_2)$. The perturbed matrix has the same structure as M and satisfies $|z_1| \geq \tau\|M\|_2$.

Next assume that $|z_i| < \tau\|M\|_2$ for $i \geq 2$. We illustrate the reduction for $i = 3$. We set z_i to zero:

$$M = \begin{pmatrix} z_1 & & \\ z_2 & d_2 & \\ z_3 & & d_3 \end{pmatrix} = \begin{pmatrix} z_1 & & \\ z_2 & d_2 & \\ 0 & & d_3 \end{pmatrix} + O(\tau\|M\|_2) \ . \tag{4.2}$$

The matrix M is perturbed by $O(\tau\|M\|_2)$. d_i is an approximate singular value of M and is deflated. The $(n-1) \times (n-1)$ leading principle submatrix of the perturbed matrix has the same structure as M but is of smaller dimensions.

Now assume that $|d_i - d_1| = |d_i| < \tau\|M\|_2$. We illustrate the reduction for $i = 3$. Let $r = \sqrt{z_i^2 + z_1^2}$, $s = z_i/r$ and $c = z_1/r$. We set d_i to zero and apply a Givens rotation to zero out z_i:

$$\begin{aligned}
&\begin{pmatrix} c & & s \\ & 1 & \\ -s & & c \end{pmatrix}\begin{pmatrix} z_1 & & \\ z_2 & d_2 & \\ z_3 & & d_3 \end{pmatrix} \\
&= \begin{pmatrix} c & & s \\ & 1 & \\ -s & & c \end{pmatrix}\begin{pmatrix} z_1 & & \\ z_2 & d_2 & \\ z_3 & & 0 \end{pmatrix} + O(\tau\|M\|_2) \\
&= \begin{pmatrix} r & & \\ z_2 & d_2 & \\ 0 & & 0 \end{pmatrix} + O(\tau\|M\|_2) \quad .
\end{aligned} \tag{4.3}$$

The matrix M is perturbed by $O(\tau\|M\|_2)$. 0 is an approximate singular value of M and is deflated. The $(n-1)\times(n-1)$ leading principle submatrix of the matrix in (4.3) has the same structure as M but is of smaller dimensions.

Finally assume that $|d_i - d_j| < \tau\|M\|_2$ for $i, j \geq 2$. We illustrate the reduction for $i = 3$ and $j = 2$. Let $r = \sqrt{z_i^2 + z_j^2}$, $s = z_i/r$ and $c = z_j/r$. We set d_j to d_i and apply a Givens rotation to zero out z_i:

$$\begin{aligned}
&\begin{pmatrix} 1 & & \\ & c & s \\ & -s & c \end{pmatrix}\begin{pmatrix} z_1 & & \\ z_2 & d_2 & \\ z_3 & & d_3 \end{pmatrix}\begin{pmatrix} 1 & & \\ & c & -s \\ & s & c \end{pmatrix} \\
&= \begin{pmatrix} 1 & & \\ & c & s \\ & -s & c \end{pmatrix}\begin{pmatrix} z_1 & & \\ z_2 & d_3 & \\ z_3 & & d_3 \end{pmatrix}\begin{pmatrix} 1 & & \\ & c & -s \\ & s & c \end{pmatrix} + O(\tau\|M\|_2) \\
&= \begin{pmatrix} z_1 & & \\ r & d_3 & \\ 0 & & d_3 \end{pmatrix} + O(\tau\|M\|_2) \quad .
\end{aligned} \tag{4.4}$$

The matrix M is perturbed by $O(\tau\|M\|_2)$. d_i is an approximate singular value of M and is deflated. The $(n-1)\times(n-1)$ leading principle submatrix of the matrix in (4.4) has the same structure as M but is of smaller dimensions.

4.2. Local Deflation for *BDC*

In the dividing strategy (see (2.4)), we write

$$B = (Q \;\; q)\begin{pmatrix} M \\ 0 \end{pmatrix} W^T = (QU \;\; q)\begin{pmatrix} \Omega \\ 0 \end{pmatrix}(WV)^T \quad , \tag{4.5}$$

where

$$Q = \begin{pmatrix} c_0 q_1 & Q_1 & 0 \\ s_0 q_2 & 0 & Q_2 \end{pmatrix} \; , \quad W = \begin{pmatrix} 0 & W_1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & W_2 \end{pmatrix} \quad \text{and} \quad M = \begin{pmatrix} r_0 & 0 & 0 \\ \alpha_k l_1 & D_1 & 0 \\ \beta_k f_2 & 0 & D_2 \end{pmatrix} \quad ;$$

l_1^T is the last row of Q_1; f_2^T is first row of Q_2; and $U\Omega V^T$ is the SVD of M.

Note that both Q and W are block matrices with some zero blocks. Since the main cost of BDC is in computing the matrix-matrix products QU and WV, we would like to take advantage of this structure. In this subsection we design a deflation procedure for BDC that gets a speedup of roughly a factor of 2 by doing so. This approach is not used in [21].

If $|r_0| < \tau \|M\|_2$, then we can apply reduction (4.1). If the vector $(\alpha_k l_1^T, \beta_k f_2^T)$ has some components with small absolute value, then we can apply reduction (4.2). In both cases the block structure of Q and W is preserved. If D_1 has a small diagonal element, then we can apply reduction (4.3). And if D_1 has two close diagonal elements, then we can apply reduction (4.4). Again in both cases the block structure is preserved. We can do the same when D_2 has a small diagonal element or has two close diagonal elements.

However, when D_1 has a diagonal element that is close to a diagonal element in D_2 and we apply reduction (4.4), the block structure of Q and W is changed. To illustrate, assume that after applying a permutation the first diagonal element of D_1 is close to the last diagonal element of D_2. Let

$$Q_1 = (\tilde{q}_1 \;\; \tilde{Q}_1) \, , \quad Q_2 = (\tilde{Q}_2 \;\; \tilde{q}_2) \;\; , \quad W_1 = (\tilde{w}_1 \;\; \tilde{W}_1) \;\; \text{and} \quad W_2 = (\tilde{W}_2 \;\; \tilde{w}_2) \;\; ;$$

let

$$\alpha_k l_1 = \begin{pmatrix} z_2 \\ \tilde{z}_1 \end{pmatrix} \quad \text{and} \quad \beta_k f_2 = \begin{pmatrix} \tilde{z}_2 \\ z_N \end{pmatrix} \quad ;$$

and let $D_1 = diag(d_2, \tilde{D}_1)$ and $D_2 = diag(\tilde{D}_2, d_N)$. Define

$$M_1 = \begin{pmatrix} r_0 & & & & \\ z_2 & d_N & & & \\ \tilde{z}_1 & & \tilde{D}_1 & & \\ \tilde{z}_2 & & & \tilde{D}_2 & \\ z_N & & & & d_N \end{pmatrix} \; .$$

Then $\|M_1 - M\|_2 < \tau \|M\|_2$. We apply a Givens rotation G to zero out z_N:

$$GMG^T \;\; = \;\; GM_1G^T + O(\tau \|M\|_2)$$

$$
\begin{aligned}
&= \begin{pmatrix} r_0 & & & & \\ r & d_N & & & \\ \tilde{z}_1 & & \tilde{D}_1 & & \\ \tilde{z}_2 & & & \tilde{D}_2 & \\ 0 & & & & d_N \end{pmatrix} + O(\tau\|M\|_2) \\
&\equiv \begin{pmatrix} \tilde{M}_1 & 0 \\ 0 & d_N \end{pmatrix} + O(\tau\|M\|_2) \quad ,
\end{aligned}
$$

where G is defined by $r = \sqrt{z_2^2 + z_N^2}$, $c = z_2/r$ and $s = z_N/r$. Plugging this into (4.5), we have

$$
\begin{aligned}
B &= (Q \;\; q)\begin{pmatrix} M \\ 0 \end{pmatrix} W^T \\
&= (QG^T \;\; q)\begin{pmatrix} GMG^T \\ 0 \end{pmatrix} (WG^T)^T \\
&= \begin{pmatrix} \tilde{X}_1 & \tilde{x} & q \end{pmatrix} \begin{pmatrix} \tilde{M}_1 & 0 \\ 0 & d_N \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \tilde{Y}_1 & \tilde{y} \end{pmatrix} + O(\tau\|M\|_2) \quad , \qquad (4.6)
\end{aligned}
$$

where

$$
\tilde{X}_1 = \begin{pmatrix} c_0 q_1 & c\tilde{q}_1 & \tilde{Q}_1 & 0 \\ s_0 q_2 & s\tilde{q}_2 & 0 & \tilde{Q}_2 \end{pmatrix} \quad \text{and} \quad \tilde{x} = \begin{pmatrix} -s\tilde{q}_1 \\ c\tilde{q}_2 \end{pmatrix} \quad ,
$$

and

$$
\tilde{Y}_1 = \begin{pmatrix} 0 & c\tilde{w}_1 & \tilde{W}_1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & s\tilde{w}_2 & 0 & \tilde{W}_2 \end{pmatrix} \quad \text{and} \quad \tilde{y} = \begin{pmatrix} -s\tilde{w}_1 \\ 0 \\ c\tilde{w}_2 \end{pmatrix} \quad .
$$

d_N is an approximate singular value of B and can be deflated. The corresponding approximate left and right singular vectors are $\tilde{x}$ and $\tilde{y}$, respectively. The matrix

$$
\tilde{M}_1 = \begin{pmatrix} r_0 & & & \\ r & d_N & & \\ \tilde{z}_1 & & \tilde{D}_1 & \\ \tilde{z}_2 & & & \tilde{D}_2 \end{pmatrix}
$$

has the same structure as M but is of smaller dimensions.

We deflate $\tilde{M}_1$ in a similar fashion until $\tilde{D}_1$ does not have any diagonal element that is close to a diagonal element of $\tilde{D}_2$. Thus, after this procedure, B can be written as

$$
B = \begin{pmatrix} \tilde{X}_1 & \tilde{X}_2 & q \end{pmatrix} \begin{pmatrix} \tilde{M}_1 & 0 \\ 0 & \tilde{\Omega}_2 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \tilde{Y}_1 & \tilde{Y}_2 \end{pmatrix}^T + O(\tau\|B\|_2) \quad . \qquad (4.7)
$$

$\tilde{\Omega}_2$ is a diagonal matrix whose diagonal elements are the deflated singular values; and the columns of $\tilde{X}_2$ and $\tilde{Y}_2$ are the corresponding approximate left and right singular vectors. $\tilde{M}_1$ is of the form

$$\tilde{M}_1 = \begin{pmatrix} r_0 & & & \\ \tilde{z}_0 & \tilde{D}_0 & & \\ \tilde{z}_1 & & \tilde{D}_1 & \\ \tilde{z}_2 & & & \tilde{D}_2 \end{pmatrix} ,$$

where the dimension of $\tilde{D}_0$ is the number of these deflations; $\tilde{D}_i$ contains the diagonal elements of D_i not affected by deflation; and $\tilde{z}_0$, $\tilde{z}_1$ and $\tilde{z}_2$ are defined accordingly. $\tilde{X}_1$ and $\tilde{Y}_1$ are of the form

$$\tilde{X}_1 = \begin{pmatrix} c_0 q_1 & \tilde{Q}_{0,1} & \tilde{Q}_1 & 0 \\ s_0 q_2 & \tilde{Q}_{0,2} & 0 & \tilde{Q}_2 \end{pmatrix} \quad \text{and} \quad \tilde{Y}_1 = \begin{pmatrix} 0 & \tilde{W}_{0,1} & \tilde{W}_1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & \tilde{W}_{0,2} & 0 & \tilde{W}_2 \end{pmatrix} , \qquad (4.8)$$

where the column dimension of $\tilde{Q}_{0,1}$, $\tilde{Q}_{0,2}$, $\tilde{W}_{0,1}$ and $\tilde{W}_{0,2}$ is the number of these deflations; the columns of $\tilde{Q}_1$ and $\tilde{Q}_2$ are those of Q_1 and Q_2 not affected by deflation; and the columns of $\tilde{W}_1$ and $\tilde{W}_2$ are those of W_1 and W_2 not affected by deflation.

Let $\tilde{U}_1\tilde{\Omega}_1\tilde{V}_1^T$ be the *SVD* of $\tilde{M}_1$. Then

$$\begin{aligned} B &= \begin{pmatrix} \tilde{X}_1 & \tilde{X}_2 & q \end{pmatrix} \begin{pmatrix} \tilde{U}_1\tilde{\Omega}_1\tilde{V}_1^T & 0 \\ 0 & \tilde{\Omega}_2 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \tilde{Y}_1 & \tilde{Y}_2 \end{pmatrix}^T + O(\tau\|B\|_2) \\ &= \begin{pmatrix} \tilde{X}_1\tilde{U}_1 & \tilde{X}_2 & q \end{pmatrix} \begin{pmatrix} \tilde{\Omega}_1 & 0 \\ 0 & \tilde{\Omega}_2 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \tilde{Y}_1\tilde{V}_1 & \tilde{Y}_2 \end{pmatrix}^T + O(\tau\|B\|_2) \quad . \end{aligned}$$

Thus $\begin{pmatrix} \tilde{X}_1\tilde{U}_1 & \tilde{X}_2 & q \end{pmatrix}$ and $\begin{pmatrix} \tilde{Y}_1\tilde{V}_1 & \tilde{Y}_2 \end{pmatrix}$ are approximate left and right singular vector matrices of B, respectively. The matrices $\tilde{X}_1\tilde{U}_1$ and $\tilde{Y}_1\tilde{V}_1$ can be computed while taking advantage of the block structure of $\tilde{X}_1$ and $\tilde{Y}_1$ in (4.8).

4.3. Global Deflation for *BDC*

To illustrate *global* deflation, we look at 2 levels of the dividing strategy (cf. (2.4)):

$$B = \begin{pmatrix} B_1 & \alpha_{i+j}e_{i+j} & \\ & \beta_{i+j}e_1 & B_2 \end{pmatrix} = \begin{pmatrix} B_{1,1} & \alpha_i e_i & & & \\ & \beta_i e_1 & B_{1,2} & \alpha_{i+j}e_j & \\ & & & \beta_{i+j}e_1 & B_2 \end{pmatrix} , \qquad (4.9)$$

where B_1, B_2, $B_{1,1}$ and $B_{1,2}$ are principle submatrices of B of dimensions $(i+j)\times(i+j-1)$, $(N-i-j+1)\times(N-i-j)$, $i\times(i-1)$ and $j\times(j-1)$, respectively.

Let $X_{1,2}\begin{pmatrix} D_{1,2} \\ 0 \end{pmatrix} W_{1,2}^T$ be the *SVD* of $B_{1,2}$, and let $(f_{1,2}^T \ \ \varphi_{1,2})$ and $(l_{1,2}^T \ \ \lambda_{1,2})$ be the first and last rows of $X_{1,2}$, respectively. Then

$$\begin{aligned} B &= \begin{pmatrix} B_{1,1} & \alpha_i e_i & & \\ & \beta_i e_1 & X_{1,2}\begin{pmatrix} D_{1,2} \\ 0 \end{pmatrix} W_{1,2}^T & \alpha_{i+j} e_j & \\ & & & \beta_{i+j} e_1 & B_2 \end{pmatrix} \\ &= \bar{X} \begin{pmatrix} B_{1,1} & \alpha_i e_i & & & \\ & \beta_i f_{1,2} & D_{1,2} & \alpha_{i+j} l_{1,2} & \\ & \beta_i \varphi_{1,2} & 0 & \alpha_{i+j} \lambda_{1,2} & \\ & & & \beta_{i+j} e_1 & B_2 \end{pmatrix} \bar{Y}^T \ \ , \end{aligned} \tag{4.10}$$

where[6] $\bar{X} = diag(I_i, X_{1,2}, I_{N-i-j+1})$ and $\bar{Y} = diag(I_{i-1}, 1, W_{1,2}, 1, I_{N-i-j})$.

Let $\bar{d}_s$ be the s-th diagonal element of $D_{1,2}$. Then $\bar{d}_s$ is also the $(i+s)$-th diagonal element of the middle matrix in (4.10). Let $\bar{f}_s$ and $\bar{l}_s$ be the s-th components of $f_{1,2}$ and $l_{1,2}$, respectively. If we ignore all zero components, then the $(i+s)$-th column and row of the middle matrix in (4.10) are $\bar{d}_s$ and $(\beta_i \bar{f}_s, \bar{d}_s, \alpha_{i+j}\bar{l}_s)$, respectively. Thus if both $|\beta_i \bar{f}_s|$ and $|\alpha_{i+j}\bar{l}_s|$ are small, then we can perturb them to zero. B has $\bar{d}_s$ as an approximate singular value with the $(i+s)$-th columns of P and R as the corresponding approximate left and right singular vectors, respectively. This singular value and its singular vectors can be deflated from all subsequent subproblems. We call this *global* deflation.

Consider the deflation procedure for computing the *SVD* in Section 4.2. If $|\beta_i \bar{f}_s|$ is small, then it can be perturbed to zero. This is a local deflation if only $|\beta_i \bar{f}_s|$ is small, and a global deflation if $|\alpha_{i+j}\bar{l}_s|$ is also small.

5. Acceleration by the Fast Multipole Method

Suppose that we want to evaluate the complex function

$$\Phi(x) = \sum_{j=1}^{n} c_j \varphi(x - x_j) \tag{5.1}$$

at m points in the complex plane, where $\{c_j\}_{j=1}^n$ are constants and $\varphi(x)$ is one of $\log(x)$, $1/x$ and $1/x^2$. The direct computation takes $O(nm)$ time. But the *fast multipole method* (*FMM*) proposed by Carrier, Greengard and Rokhlin [6, 14] takes only $O(n+m)$ time to approximate $\Phi(x)$ at these points to a precision specified by the user[7]. In this section we briefly describe how *FMM* can be used to accelerate *BDC*. A more detailed description appears in [18] in the context of updating the singular value decomposition.

[6] I_i is an $i \times i$ identity matrix.

[7] The constant hidden in the O notation depends on the logarithm of the precision.

Let

$$M = \begin{pmatrix} z_1 & & & \\ z_2 & d_2 & & \\ \vdots & & \ddots & \\ z_n & & & d_n \end{pmatrix} ,$$

where $D = diag(d_1, d_2, \ldots, d_n)$ with $0 \equiv d_1 < d_2 < \ldots < d_n$; and $z = (z_1, z_2, \ldots, z_n)^T$ with $z_i \neq 0$. Let $U\Omega V^T$ denote the SVD of M with

$$U = (u_1, \ldots, u_n) \ , \quad \Omega = diag(\omega_1, \ldots, \omega_n) \quad \text{and} \quad V = (v_1, \ldots, v_n) \ .$$

Consider the cost of computing $U^T x$ for a vector $x = (x_1, \ldots, x_n)^T$. By equation (3.3) in Lemma 1, the i-th component $u_i^T x$ of $U^T x$ can be written as $\Phi_1(\omega_i)/\sqrt{\Phi_2(\omega_i)}$, where

$$\Phi_1(\omega) = \sum_{k=1}^{n} \frac{x_k z_k}{d_k^2 - \omega^2} \quad \text{and} \quad \Phi_2(\omega) = \sum_{k=1}^{n} \frac{z_k^2}{(d_k^2 - \omega^2)^2} \ .$$

Thus we can compute $U^T x$ by evaluating $\Phi_1(\omega)$ and $\Phi_2(\omega)$ at n points. Since these two functions are of the form (5.1), we can do this in $O(n)$ time using *FMM*. To achieve better efficiency, we modify *FMM* to take advantage of the fact that all the computations are real (see [15, 18]).

Let B be a lower bidiagonal matrix of dimensions $(N+1) \times N$. When BDC is used to compute all the singular values and singular vectors, the main cost for each subproblem is the computation of $\tilde{X}_1 U$ and $\tilde{Y}_1 V$ (see (4.7)), where $\tilde{X}_1$ and $\tilde{Y}_1$ are column orthonormal matrices[8]. Each row of $\tilde{X}_1 U$ is of the form $x^T U = (U^T x)^T$, and there are $O(n)$ rows. Thus the cost of computing $\tilde{X}_1 U$ is $O(n^2)$ using *FMM*. A similar result holds for computing $\tilde{Y}_1 V$. There are $\log_2 N$ levels of recursion and 2^{k-1} subproblems at the k-th level, each of size $O(N/2^{k-1})$. Thus the cost at the k-th level is $O(N^2/2^k)$, and the total time is $O(N^2)$.

We may also have to apply orthonormal matrices P and R to the left and right singular vector matrices of B, respectively, e.g., when B is obtained by reducing a dense matrix to bidiagonal form [12, 24, 26]. For each subproblem, we can apply the left and right singular vector matrices of the corresponding M directly to P and R. The cost for each subproblem is $O(Nn)$ using *FMM*, and there are $O(N/n)$ subproblems at each level. Thus the cost at each level is $O(N^2)$, and the total time is $O(N^2 \log_2 N)$.

When BDC is used to compute only the singular values, the main cost for each subproblem is computing two vectors of the form $x^T U$, finding all the roots of the reformulated secular equation, and computing the vector $\hat{z}$. We now show how to find all the roots of M and all the components of $\hat{z}$ in $O(n)$ time.

[8] We view $\tilde{X}_1$ and $\tilde{Y}_1$ as dense matrices to simplify the presentation, even though *FMM* is more efficient when it exploits their block structure (see (4.8)).

A root-finder computes successive iterates for each singular value ω_i. We assume that the number of iterations for each root is bounded. The main cost for each iteration is in evaluating the function

$$g(\mu) = 1 + \psi(\mu) + \phi(\mu) = 1 + \sum_{j=1}^{n} \frac{z_j^2}{(\delta_j - \mu)(d_j + d_i + \mu)} \quad .$$

To compute all the singular values simultaneously, we must evaluate $g(\mu)$ at $O(n)$ points. The function $g(\mu)$ is similar to the form (5.1), and thus we can evaluate $g(\mu)$ at these points in $O(n)$ time using *FMM*[9]. In other words, all the singular values of M can be computed in $O(n)$ time.

To compute $\hat{z}$, we note that equation (3.6) can be rewritten as

$$|\hat{z}_i| = \sqrt{(\hat{\omega}_n^2 - d_i^2)} \exp\left(\frac{1}{2}\sum_{k=1}^{i-1} \log\left(\frac{\hat{\omega}_k^2 - d_i^2}{d_k^2 - d_i^2}\right) + \frac{1}{2}\sum_{k=i}^{n-1} \log\left(\frac{\hat{\omega}_k^2 - d_i^2}{d_{k+1}^2 - d_i^2}\right)\right) \quad 1 \le i \le n \quad .$$

Thus we can compute all of the components of $\hat{z}$ in $O(n)$ time using *FMM*.

We have shown that when computing all the singular values of B, we can solve each subproblem in $O(n)$ time. Since there are $O(N/n)$ subproblems at each level, the cost at each level is $O(N)$, and thus the total time is $O(N \log_2 N)$.

6. Computing the *SVD* of a Banded Matrix

We now generalize *BDC* to compute the *SVD* of a banded matrix. This problem arises when one uses the block Lanczos algorithm to compute the *SVD* of a sparse matrix [10, 12]. Arbenz [2] has similarly generalized divide-and-conquer algorithms for the symmetric tridiagonal eigenproblem to solve the symmetric banded eigenproblem.

Suppose for simplicity that B is an $(N+K) \times N$ lower $(K+1)$-diagonal matrix with $K \ll N$. We recursively divide B into two subproblems:

$$B = \begin{pmatrix} B_{1,1} & B_{1,2} & 0 \\ 0 & B_{2,2} & B_{2,3} \end{pmatrix} \tag{6.1}$$

where $B_{1,1}$ and $B_{2,3}$ are $(k+K) \times k$ and $(N-k) \times (N-K-k)$ lower $(K+1)$-diagonal matrices, respectively; $B_{1,2}$ is a $(k+K) \times K$ matrix with non-zero elements only on the lowest K diagonals; and $B_{2,2}$ is an $(N-k) \times K$ matrix with non-zero elements only on the highest K diagonals. Usually $k < N$ is taken to be $\lfloor (N-K)/2 \rfloor$.

Let

$$B_{1,1} = (Q_1 \;\; S_1) \begin{pmatrix} D_1 \\ 0 \end{pmatrix} W_1^T = Q_1 D_1 W_1^T$$

[9] See [15, 18] for a version of *FMM* that computes $\psi(\mu)$ and $\phi(\mu)$ and their derivatives at $O(n)$ points in $O(n)$ time. This is needed for the root-finders in [4, 5, 13, 23] and to check the stopping criterion.

be the SVD of $B_{1,1}$, and let

$$B_{2,3} = (Q_2 \;\; S_2)\begin{pmatrix} D_2 \\ 0 \end{pmatrix} W_2^T = Q_2 D_2 W_2^T$$

be the SVD of $B_{2,3}$. Plugging these into (6.1), we have

$$\begin{aligned} B &= \begin{pmatrix} (Q_1 \;\; S_1)\begin{pmatrix} D_1 \\ 0 \end{pmatrix} W_1^T & B_{1,2} & 0 \\ 0 & B_{2,2} & (Q_2 \;\; S_2)\begin{pmatrix} D_2 \\ 0 \end{pmatrix} W_2^T \end{pmatrix} \\ &= \begin{pmatrix} S_1 & Q_1 & & \\ & & Q_2 & S_2 \end{pmatrix} \begin{pmatrix} Z_{0,1} & 0 & 0 \\ Z_1 & D_1 & 0 \\ Z_2 & 0 & D_2 \\ Z_{0,2} & 0 & 0 \end{pmatrix} \begin{pmatrix} & W_1 & \\ I_K & & \\ & & W_2 \end{pmatrix}^T \quad , \qquad (6.2) \end{aligned}$$

where

$$Z_{0,1} = S_1^T B_{1,2} \;\; , \;\; Z_1 = Q_1^T B_{1,2} \;\; , \;\; Z_{0,2} = S_2^T B_{2,2} \;\; \text{and} \;\; Z_2 = Q_2^T B_{2,2} \; .$$

The matrix $\begin{pmatrix} Z_{0,1} \\ Z_{0,2} \end{pmatrix}$ is $2K \times K$. There exists an orthonormal matrix $\begin{pmatrix} G_{1,1} & G_{1,2} \\ G_{2,1} & G_{2,2} \end{pmatrix}$ such that

$$\begin{pmatrix} Z_{0,1} \\ Z_{0,2} \end{pmatrix} = \begin{pmatrix} G_{1,1} & G_{1,2} \\ G_{2,1} & G_{2,2} \end{pmatrix} \begin{pmatrix} Z_0 \\ 0 \end{pmatrix} \quad ,$$

where Z_0 is a $K \times K$ lower triangular matrix. Plugging this into (6.2), we have

$$B = \begin{pmatrix} S_1 G_{1,1} & Q_1 & & S_1 G_{2,1} \\ S_2 G_{1,2} & & Q_2 & S_2 G_{2,2} \end{pmatrix} \begin{pmatrix} Z_0 & 0 & 0 \\ Z_1 & D_1 & 0 \\ Z_2 & 0 & D_2 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} & W_1 & \\ I_K & & \\ & & W_2 \end{pmatrix}^T \quad . \qquad (6.3)$$

The middle matrix in (6.3) is lower triangular and can only have non-zero elements on its diagonal and in its first K columns. Partition

$$\begin{pmatrix} Z_0 \\ Z_1 \\ Z_2 \end{pmatrix} = \begin{pmatrix} \tilde{Z}_0 & 0 \\ \tilde{Z} & z \end{pmatrix} \quad ,$$

where $\tilde{Z}_0$ is $(K-1) \times (K-1)$ lower triangular; and $z = (r_0, z_1^T, z_2^T)^T$ with z_i being the last column of Z_i and r_0 being the last diagonal element of Z_0. Let

$$M = \begin{pmatrix} r_0 & 0 & 0 \\ z_1 & D_1 & 0 \\ z_2 & 0 & D_2 \end{pmatrix} \quad ,$$

and let $U\Omega V^T$ be the SVD of M computed using the scheme described in Section 3. Then the middle matrix in (6.3) can be rewritten as

$$\begin{pmatrix} \tilde{Z}_0 & 0 \\ \tilde{Z} & U\Omega V^T \end{pmatrix} = \begin{pmatrix} I_{K-1} & \\ & U \end{pmatrix} \begin{pmatrix} \tilde{Z}_0 & 0 \\ U^T\tilde{Z} & \Omega \end{pmatrix} \begin{pmatrix} I_{K-1} & \\ & V \end{pmatrix}^T ,$$

where the middle matrix is lower triangular and can only have non-zero elements on its diagonal and in its first $K-1$ columns. Thus the SVD of the middle matrix in (6.3) can be computed by applying this procedure K times.

To compute the SVDs of $B_{1,1}$ and $B_{2,3}$, we can recursively apply equations (6.1) and (6.3) to $B_{1,1}$ and $B_{2,3}$ until the subproblems are sufficiently small. These small subproblems are then solved using the Golub-Kahan algorithm [9, 11]. There can be at most $O(\log_2 N)$ levels of recursion. This algorithm takes $O(KN^3)$ time to compute both the singular values and the singular vectors. Similar to the bidiagonal case, there is an $O(K^2N^2)$ time divide-and-conquer algorithm for computing only the singular values. These times can be reduced to $O(KN^2)$ and $O(K^2N\log_2 N)$, respectively, by using FMM. These reduced times are better than the corresponding worst-case time requirements ($O(N^3)$ and $O(KN^2)$) for the banded QR algorithm [24].

References

[1] P. Arbenz, *Divide-and-conquer algorithms for the computation of the SVD of bidiagonal matrices*, Tech. Report 89, Department Informatik, ETH, Zürich, Switzerland, 1988.

[2] ——, *Divide-and-conquer algorithms for the bandsymmetric eigenvalue problem*, Tech. Report 170, Department Informatik, ETH, Zürich, Switzerland, 1991.

[3] P. Arbenz and G. H. Golub, *On the spectral decomposition of Hermitian matrices modified by low rank perturbations with applications*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 40–58.

[4] J. R. Bunch and C. P. Nielsen, *Updating the singular value decomposition*, Numer. Math., 31 (1978), pp. 111–129.

[5] J. R. Bunch, C. P. Nielsen, and D. C. Sorensen, *Rank-one modification of the symmetric eigenproblem*, Numer. Math., 31 (1978), pp. 31–48.

[6] J. Carrier, L. Greengard, and V. Rokhlin, *A fast adaptive multipole algorithm for particle simulations*, SIAM J. Sci. Stat. Comput., 9 (1988), pp. 669–686.

[7] J. J. M. Cuppen, *A divide and conquer method for the symmetric tridiagonal eigenproblem*, Numer. Math., 36 (1981), pp. 177–195.

[8] G. H. Golub, *Some modified matrix eigenvalue problems*, SIAM Review, 15 (1973), pp. 318–334.

[9] G. H. Golub and W. Kahan, *Calculating the singular values and pseudo-inverse of a matrix*, SIAM J. Numer. Anal., 2 (1965), pp. 205–224.

[10] G. H. Golub, F. T. Luk, and M. Overton, *A block Lanczos method for computing the singular values and corresponding singular vectors of a matrix*, ACM Trans. Math. Soft., 7 (1981), pp. 149–169.

[11] G. H. Golub and C. Reinsch, *Singular value decomposition and least squares solutions*, Numer. Math., 14 (1970), pp. 403–420.

[12] G. H. Golub and C. F. van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, second ed., 1989.

[13] W. B. Gragg, J. R. Thornton, and D. D. Warner, *Parallel divide and conquer algorithms for the symmetric tridiagonal eigenproblem and bidiagonal singular value problem*, in Proceedings of 23rd Annual Pittsburgh Conference, University of Pittsburgh School of Engineering, vol. 23 of Modelling and Simulation, 1992.

[14] L. Greengard and V. Rokhlin, *A fast algorithm for particle simulations*, J. Comp. Phys., 73 (1987), pp. 325–348.

[15] M. Gu and S. C. Eisenstat, *A fast divide-and-conquer method for the symmetric tridiagonal eigenproblem.* Presented at the Fourth SIAM Conference on Applied Linear Algebra, Minneapolis, Minnesota, September 1991.

[16] ——, *A divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem*, Tech. Report YALEU/DCS/RR-932, Department of Computer Science, Yale University, 1992.

[17] ——, *A stable and efficient algorithm for the rank-one modification of the symmetric eigenproblem*, Tech. Report YALEU/DCS/RR-916, Department of Computer Science, Yale University, 1992.

[18] ——, *A fast algorithm for updating the singular value decomposition.* To appear, 1993.

[19] E. R. Jessup, *Parallel Solution of the Symmetric Tridiagonal Eigenproblem*, Ph.D. thesis, Department of Computer Science, Yale University, 1989.

[20] E. R. Jessup and I. C. F. Ipsen, *Improving the accuracy of inverse iteration*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 550–572.

[21] E. R. Jessup and D. C. Sorensen, *A parallel algorithm for computing the singular value decomposition of a matrix.* Revision of Tech. Report ANL-MCS-TM-102, Argonne National Laboratory, July 1991.

[22] W. Kahan, *Rank-1 perturbed diagonal's eigensystem.* Unpublished manuscript, July 1989.

[23] R.-C. Li, *Solving secular equations stably and efficiently.* Unpublished manuscript, October 1992.

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

[25] D. C. Sorensen and P. T. P. Tang, *On the orthogonality of eigenvectors computed by divide-and-conquer techniques*, SIAM J. Numer. Anal., 28 (1991), pp. 1752–1775.

[26] G. W. Stewart, *Introduction to Matrix Computations*, Academic Press, New York, 1973.