Computing the Bidiagonal SVD through an Associated Tridiagonal Eigenproblem

Osni Marques[1] and Paulo B. Vasconcelos[2]

[1] Lawrence Berkeley National Laboratory, oamarques@lbl.gov
[2] Faculdade de Economia and CMUP, Universidade do Porto, pjv@fep.up.pt

Abstract. In this paper, we present an algorithm for the singular value decomposition (SVD) of a bidiagonal matrix by means of the eigenpairs of an associated symmetric tridiagonal matrix. The algorithm is particularly suited for the computation of a subset of singular values and corresponding vectors. We focus on a sequential implementation, discuss special cases and other issues. We use a large set of bidiagonal matrices to assess the accuracy of the implementation and to identify potential shortcomings. We show that the algorithm can be up to three orders of magnitude faster than existing algorithms, which are limited to the computation of a full SVD.

1 Introduction

It is well known that the singular value decomposition (SVD) of a matrix $A \in \mathbb{R}^{m\times n}$, namely $A = USV^T$, with left singular vectors $U = [u_1, u_2, \ldots u_n]$, right singular vectors $V = [v_1, v_2, \ldots v_n]$, and singular values $S = diag(s_1, s_2, \ldots s_n)$, $s_1 \geq s_2 \geq \ldots s_n \geq 0$, can be obtained through the eigenpairs (λ, x) of the matrices $C_{n\times n} = A^T A$ and $C_{m\times m} = AA^T$. However, if A is square and orthogonal $C_{n\times n}$ and $C_{m\times m}$ are both the identity and provide little information about the singular vectors of A, which are not unique: $A = (AV)IV^T$ is the SVD of A for any orthogonal matrix V. A potential difficulty for some algorithms (e.g. the one presented in this paper) is large clusters of close singular values, as this may have an impact on the orthogonality of the computed singular vectors.

Alternatively, the SVD can be obtained through the augmented matrix [1]

$$C = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix} = J \begin{bmatrix} -S & 0 \\ 0 & S \end{bmatrix} J^T,\ J = \begin{bmatrix} U & U \\ -V & V \end{bmatrix} / \sqrt{2}, \tag{1}$$

such that the eigenvalues of C are $\pm s$ and its eigenvectors are mapped into the singular vectors of A (scaled by $\sqrt{2}$) in a very structured manner.

In practical calculations, the SVD of a full matrix A involves the reduction of A to bidiagonal form B through orthogonal transformations, i.e. $A = \hat{U}B\hat{V}^T$. The singular values are thus preserved; the singular vectors of B need to be back transformed into those of A.

If B is an upper bidiagonal matrix with $(a_1, a_2, \ldots a_n)$ on the main diagonal and $(b_1, b_2, \ldots b_{n-1})$ on the off diagonal, we can replace A with B in (1) to obtain

Table 1. LAPACK's (bidiagonal, BD) SVD and (tridiagonal, ST) eigensolvers.

routine	usage	algorithm
`BDSQR`	all s and (opt.) u and/or v	implicit QL or QR
`BDSDC`	all s and (opt.) u and v	divide-and-conquer
`STEQR`	all λ's and (opt.) x	implicit QL or QR
`STEVX`	selected λ's and (opt.) x	bisection & inverse iteration
`STEDC`	all λ's and (opt.) x	divide-and-conquer
`STEMR`	selected λ's and (opt.) x	MRRR

$C = P\,T_{GK}\,P^T$, where T_{GK} is the Golub-Kahan symmetric tridiagonal matrix,

$$T_{GK} = tridiag\begin{pmatrix} & a_1 & & b_1 & & a_2 & & b_2 \ldots & b_{n-1} & & a_n & \\ 0 & & 0 & & 0 & & 0 & \ldots & & 0 & & 0 \\ & a_1 & & b_1 & & a_2 & & b_2 \ldots & b_{n-1} & & a_n & \end{pmatrix}, \tag{2}$$

and the perfect shuffle $P = [e_{n+1}, e_1, e_{n+2}, e_2, e_{n+3}, \ldots e_{2n}]$, were the e's are the columns of the identity matrix of dimension $2n$. Then, if the eigenpairs of T_{GK} are $(\pm s, z)$, with $\|z\| = 1$, and from (1), we obtain $z = P(u^T, \pm v^T)/\sqrt{2}$ [6]. Thus, we can extract the SVD of B from the eigendecomposition of T_{GK}.

Table 1 lists the current LAPACK subroutines intended for the computation of the SVD of bidiagonal matrices, and eigenvalues and eigenvectors of tridiagonal matrices. The tradeoffs (performance, accuracy) of these eigensolvers have been thoroughly examined in [3]. We are interested in how the symmetric tridiagonal (ST) subroutines could be applied to (2), specially for the computation of subsets of eigenpairs, which in turn could reduce the computational costs when a full SVD is not needed (or for the computations of subsets in parallel). While `STEDC` could be potentially redesigned to compute a subset of eigenvectors, saving some work but only at the top level of recursion of the divide-and-conquer algorithm, `STEVX` and `STEMR` offer more straighforward alternatives. `STEVX` performs bisection to find selected eigenvalues followed by inverse iteration to find their eigenvectors, for an $O(n)$ cost per eigenpair. `STEVX` can occasionally fail to provide orthogonal eigenvectors when the eigenvalues are too closely clustered. In contrast, `STEMR` uses a much more sophisticated algorithm called MRRR [4, 5] to guarantee orthogonality. An improved version of the MRRR algorithm targeting T_{GK} in order to compute the SVD has been proposed in [6]; however, our experiments with an implementation given in [6] produced vectors with inadequate level of orthogonality, for relatively simple matrices. Therefore, we have decided to adopt `STEVX` for computing eigenvalues and eigenvectors of (2), even though it has known failure modes that we discuss later.

The main contribution of this paper is to discuss an implementation of an algorithm for the SVD of a bidiagonal matrix obtained from eigenpairs of a tridiagonal matrix T_{GK}. This implementation is called `BDSVDX`, introduced in LAPACK 3.6.0. While the associated formulation is not necessarily new, as mentioned above, its actual implementation requires care in order to deal correctly with multiple or tightly clustered singular values, or some cases of splitting. To the best of our knowledge, no such implementation has been done and exhaustively tested. In concert with `BDSVDX` we have also developed `GESVDX`, which takes a general matrix A, reduces it to bidiagonal form B, invokes `BDSVDX`, and then maps the output of `BDSVDX` into the SVD of A. In LAPACK, the current counterparts of `GESVDX` are `GESVD` and `GESDD`, which are based on the BD subroutines listed in Table 1 and can only compute all singular values (and optionally singular vectors). This can be much more expensive if only a few singular values and vectors are desired.

The rest of the paper is organized as follows. First, we discuss how singular values are mapped into the eigenvalue spectrum. Then, we discuss special cases, the criterion for splitting a bidiagonal matrix, and other implementation details. Next, we show the results of our tests with `BDSVDX` using a large set of bidiagonal matrices, to assess both accuracy and computational performance. Finally, we discuss limitations and opportunities for future work.

2 Mapping singular values into eigenvalues

Similarly to `BDSQR` and `BDSDC`, `BDSVDX` allows the computation of singular values only or singular values and the corresponding singular vectors. Borrowing features from `STEVX`, `BDSVDX` can be used in three modes, through a character variable `RANGE`. If `RANGE`="A", all singular values will be found: `BDSVDX` will compute the smallest (negative or zero) n eigenvalues of the corresponding T_{GK}. If `RANGE`="V", all singular values in the half-open interval (VL,VU] will be found: `BDSVDX` will compute the eigenvalues of the corresponding T_{GK} in the interval (-VU,-VL]. If `RANGE`="I", the IL-th through IU-th singular values will be found: the indices IL and IU are mapped into values (similar to VL and VU) by applying bisection to T_{GK}. VL, VU, IL and IU are arguments of `BDSVDX` (which are mapped into similar arguments for `STEVX`).

For a bidiagonal matrix B of dimension n, if singular vectors are requested, `BDSVDX` returns an array Z of dimension $2n \times p$, where $p \leq n$ is a function of `RANGE`. Each column of Z will contain $(u_i^T, v_i^T)^T$ corresponding to singular value s_i, i.e. (using Matlab notation) $Z = [\ U;\ V\ ]$. `STEVX` returns eigenvalues (and corresponding vectors) in ascending order, so we target the negative part of the eigenvalue spectrum (i.e. $-S$) in (1). Therefore, the absolute values of the returned eigenvalues give us the singular values in the desired order, $s_1 \geq s_2 \geq \ldots s_n \geq 0$. We only need to change the signs of the entries in the eigenvectors that are reloaded to V. We note that `BDSVDX` inherits some shortcomings from `STEVX`: in extreme situations bisection may fail to converge, or not all eigenvalues with indices IL:IU can be found, or inverse iteration fails to converge after the allowed number of iterations is reached.

3 Splitting: special cases

The criterion for splitting in `BDSVDX` is the same that is used in `STEQR` and is discussed in [7]. We first form the matrix T_{GK} and check for splitting in two phases, starting with the off diagonal entries of T_{GK} with even indices (i.e. the b's). If, for a given i, $b_i = 0$ (or it is tiny enough to be set to zero) the matrix B splits and the SVD for each resulting (square) submatrix of B can be obtained independently. This effect is propagated into the associated T_{GK}, i.e. the eigenvalues and eigenvectors of each submatrix of T_{GK} can be obtained independently. We then check the off diagonal entries of T_{GK} with odd indices (i.e. the a's). If, for a given j, $a_j = 0$ (or tiny enough), we end up with rectangular bidiagonal matrices, which do not have equal numbers of left and right singular vectors. This complicates our simple approach for extracting singular vectors of B from eigenvectors of T_{GK}. The problem can be reduced to one of the three special cases illustrated below with small matrices.

Zero in the interior. If $n = 5$ and $a_3 = 0$, we have the following SVD:

$$B = bidiag\begin{pmatrix} & b_1 & & b_2 & & b_3 & & b_4 & \\ a_1 & & a_2 & & 0 & & a_4 & & a_5 \end{pmatrix} = \begin{bmatrix} U_1 & \\ & U_2 \end{bmatrix}\begin{bmatrix} S_1 & \\ & S_2 \end{bmatrix}\begin{bmatrix} V_1^T & \\ & V_2^T \end{bmatrix},$$

where U_1 and V_2 are 2-by-2, U_2 and V_1 are 3-by-3, S_1 is 2-by-3 (its third column contains only zeros), and S_2 is 3-by-2 (its third row contains only zeros). If we construct $T_{GK}^{(1)}$ and $T_{GK}^{(2)}$ matrices as

$$T_{GK}^{(1)} = tridiag\begin{pmatrix} & a_1 & & b_1 & & a_2 & & b_2 & \\ 0 & & 0 & & 0 & & 0 & & 0 \\ & a_1 & & b_1 & & a_2 & & b_2 & \end{pmatrix}, \quad T_{GK}^{(2)} = tridiag\begin{pmatrix} & b_3 & & a_4 & & b_4 & & a_5 & \\ 0 & & 0 & & 0 & & 0 & & 0 \\ & b_3 & & a_4 & & b_4 & & a_5 & \end{pmatrix},$$

then the first three columns of their respective eigenvector matrices are

$$Z_{5\times3}^{(1)} = \begin{bmatrix} v_{1,1}^{(1)} & v_{1,2}^{(1)} & v_{1,3}^{(1)} \\ u_{1,1}^{(1)} & u_{1,2}^{(1)} & 0 \\ v_{2,1}^{(1)} & v_{2,2}^{(1)} & v_{2,3}^{(1)} \\ u_{2,1}^{(1)} & u_{2,2}^{(1)} & 0 \\ v_{3,1}^{(1)} & v_{3,2}^{(1)} & v_{3,3}^{(1)} \end{bmatrix} D^{-1}, \quad Z_{5\times3}^{(2)} = \begin{bmatrix} u_{1,1}^{(2)} & u_{1,2}^{(2)} & u_{1,3}^{(2)} \\ v_{1,1}^{(2)} & v_{1,2}^{(2)} & 0 \\ u_{2,1}^{(2)} & u_{2,2}^{(2)} & u_{2,3}^{(2)} \\ v_{2,1}^{(2)} & v_{2,2}^{(2)} & 0 \\ u_{3,1}^{(2)} & u_{3,2}^{(2)} & u_{3,3}^{(2)} \end{bmatrix} D^{-1}$$

where $Z_{5\times3}^{(1)}$ and $Z_{5\times3}^{(2)}$ show how the entries of the eigenvectors corresponding to the three smallest (negative) eigenvalues of $T_{GK}^{(1)}$, $\lambda_1^{(1)} < \lambda_2^{(1)} < \lambda_3^{(1)}$, and $T_{GK}^{(2)}$, $\lambda_1^{(2)} < \lambda_2^{(2)} < \lambda_3^{(2)}$ relate to the entries of U_1, U_2, V_1 and V_2, where $v_{ij}^{(1)}$ are the entries of V_1 and so on. Note that the left and right singular vectors corresponding to s_3 are in different matrices, with $D = diag(\sqrt{2}, \sqrt{2}, 1)$. (The array Z returned by `BDSVDX` would be, in Matlab notation, $Z = [\ Z_{5\times2}^{(1a)}\ \ Z_{5\times3}^{(1b)};\ 0\ \ Z_{5\times3}^{(2)}\ ]$, where $Z_{5\times2}^{(1a)}$ contains the first two columns of $Z_{5\times3}^{(1)}$, while $Z_{5\times3}^{(1b)}$ has zeros in its two first columns and the last column of $Z_{5\times3}^{(1)}$ in its last column.)

Zero at the top. If $n = 4$ and $a_1 = 0$, we have the following SVD:

$$B = bidiag\begin{pmatrix} & b_1 & & b_2 & & b_3 & \\ 0 & & a_2 & & a_3 & & a_4 \end{pmatrix} = \begin{bmatrix} U \end{bmatrix}\begin{bmatrix} 0 & \\ & S \end{bmatrix}\begin{bmatrix} 1 & \\ & V^T \end{bmatrix},$$

Table 2. Relation between the eigenvectors of T_{GK} and the entries of U and V for a zero at the top or bottom of B.

zero at the top	zero at the bottom
$Z^{(1)}_{8\times 5} = \begin{bmatrix} 1 & \\ & Z^{(1)}_{7\times 4} \end{bmatrix} (D^{(1)})^{-1}$	$Z^{(2)}_{8\times 5} = \begin{bmatrix} Z^{(2)}_{7\times 4} & \\ & 1 \end{bmatrix} (D^{(2)})^{-1}$
$D^{(1)} = diag(1, \sqrt{2}, \sqrt{2}, \sqrt{2}, 1)$	$D^{(2)} = diag(\sqrt{2}, \sqrt{2}, \sqrt{2}, 1, 1)$
columns of $Z^{(1)}_{7\times 4}$:	columns of $Z^{(2)}_{7\times 4}$:
$(u_{1,1}\ \ v_{1,1}\ \ u_{2,1}\ \ v_{2,1}\ \ u_{3,1}\ \ v_{3,1}\ \ u_{4,1})^T$	$(v_{1,1}\ \ u_{1,1}\ \ v_{2,1}\ \ u_{2,1}\ \ v_{3,1}\ \ u_{3,1}\ \ v_{4,1})^T$
$(u_{1,2}\ \ v_{1,2}\ \ u_{2,2}\ \ v_{2,2}\ \ u_{3,2}\ \ v_{3,2}\ \ u_{4,2})^T$	$(v_{1,2}\ \ u_{1,2}\ \ v_{2,2}\ \ u_{2,2}\ \ v_{3,2}\ \ u_{3,2}\ \ v_{4,2})^T$
$(u_{1,3}\ \ v_{1,3}\ \ u_{2,3}\ \ v_{2,3}\ \ u_{3,3}\ \ v_{3,3}\ \ u_{4,3})^T$	$(v_{1,3}\ \ u_{1,3}\ \ v_{2,3}\ \ u_{2,3}\ \ v_{3,3}\ \ u_{3,3}\ \ v_{4,3})^T$
$(u_{1,4}\ \ 0\ \ u_{2,4}\ \ 0\ \ u_{3,4}\ \ 0\ \ u_{4,4})^T$	$(v_{1,4}\ \ 0\ \ v_{2,4}\ \ 0\ \ v_{3,4}\ \ 0\ \ v_{4,4})^T$

where U is 4-by-4, S is 3-by-3, and V is 3-by-3. If we construct a T_{GK} from B, its first row and column will be zero, and the entries of the eigenvectors corresponding to the five smallest eigenvalues of T_{GK} (again, related explicitly to singular values of B) relate to the entries of U and V as shown in Table 2. (The array Z returned by `BDSVDX` would be formed by taking the last four columns of $Z^{(1)}_{8\times 5}$; its last column is concatenated with the first column of $Z^{(1)}_{8\times 5}$.)

Zero at the bottom. If $n = 4$ and $a_4 = 0$, we have the following SVD:

$$B = bidiag \begin{pmatrix} & b_1 & & b_2 & & b_3 & \\ a_1 & & a_2 & & a_3 & & 0 \end{pmatrix} = \begin{bmatrix} U & \\ & 1 \end{bmatrix} \begin{bmatrix} S & \\ & 0 \end{bmatrix} \begin{bmatrix} V^T \end{bmatrix},$$

where U is 3-by-3, S is 3-by-3, and V is 4-by-4. If we construct a T_{GK} from B, its last row and column will be zero, the entries of the eigenvectors corresponding to the five smallest eigenvalues of T_{GK} (again, related explicitly to singular values of B) relate to the entries of U and V as shown in Table 2. (The array Z returned by `BDSVDX` would be formed by taking the first four columns of $Z^{(2)}_{8\times 5}$; its last column is concatenated with the last column of $Z^{(2)}_{8\times 5}$.)

4 Reorthogonalization of vectors

As discussed earlier, $z_i = P(u_i^T, -v_i^T)^T/\sqrt{2}$ $(i \leq 1 \leq n)$. We could simply create $\hat{u}_i$ with the even entries of z_i and $\hat{v}_i$ with the odd entries of z_i and multiply those vectors by $\sqrt{2}$ in order to obtain u_i and v_i. However, in our implementation we explicitly normalize $\hat{u}_i$ and $\hat{v}_i$. This allows us to check how far the norms of $\hat{u}_i$ and $\hat{v}_i$ are from $\frac{1}{\sqrt{2}}$, which may be the case if z_i is associated with a small λ. Then, if needed, we apply a Gram-Schmidt reorthogonalization to $\hat{u}_i$ and $\hat{v}_i$. Our test for triggering a reorthogonalization is based on $|\|\hat{u}\| - \frac{1}{\sqrt{2}}| \geq tol$ (similarly for $\hat{v}$), $tol = \sqrt{\varepsilon}$, where ε is the machine precision. However, we have identified matrices for which this test is not sufficient, which suggests the need for a strategy that takes into account the separation of λ's. This is the case, for example, of the bidiagonal defined as $a_i = 10^{-(2i-1)}, i = 1, 2, \ldots 8$, $b_i = 10^{-(2i-2)}, i = 1, 2, \ldots 7$, for which $s_1 \approx 1.005$, $s_7 \approx 10^{-12}$ and $s_8 \approx 10^{-22}$. The eigenvectors of the corresponding T_{GK} associated with eigenvalues $-s_7$ and $-s_8$ are not well defined, resulting in singular vectors that are not orthogonal to work precision.

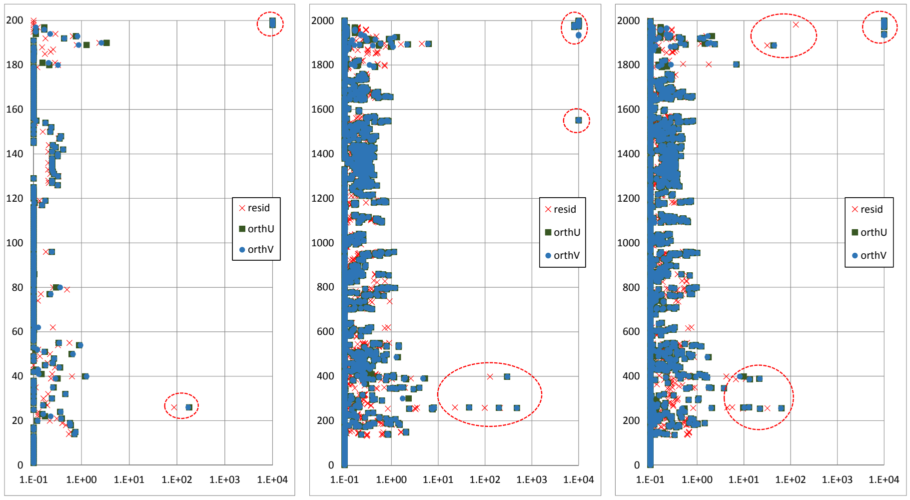
5 Numerical experiments

We have used a large set of bidiagonal matrices to test BDSVDX, on a typical Intel-based computer, in double and single precisions, using different compilers. Here we report results in double precision only, with the gnu compiler. Most of the test matrices in our testbed are derived from symmetric tridiagonal matrices described in [2] (also used in [3]). In this case, we factor $T - \nu I = LL^T$ (Cholesky) for a proper value of ν (obtained from the Gerschgorin bounds of T), then set $B = L^T$. The testbed also includes bidiagonals generated with random entries.

To test accuracy, we compute $resid = \|U^T BV - S\|/(\|B\| \times n \times \varepsilon)$, $orthU = \|I - U^T U\|/(n \times \varepsilon)$, and $orthV = \|I - V^T V\|/(n \times \varepsilon)$. To test RANGE="I" or RANGE="V" for a given B, we build the corresponding T_{GK} prior to invoking BDSVDX and compute its eigenvalues using bisection. Then, for RANGE="V" we generate n_V pairs of random indices IL and IU, map those indices into the eigenvalues of T_{GK}, perturb the eigenvalues slightly to obtain corresponding pairs VL and VU, and then invoke BDSVDX n_V times. For RANGE="I" we simply generate n_I pairs of random indices IL and IU, and then invoke BDSVDX n_I times.

Fig. 1a shows the accuracy of BDSVDX, all singular values and vectors, for 200 bidiagonal matrices with dimensions ranging from 9 to 4006. Figs. 1b-1c show the accuracy of BDSVDX for the same matrices of Fig. 1a, with $n_I = 10$ (random) pairs of IL, IU, and $n_V = 10$ (random) pairs of VL, VU for each matrix. In the figures, the matrices (y-axis) are ordered according to their condition numbers, which range from 1.0 to $> 10^{200}$. For convenience, we use floor and ceiling functions to bound the results in the x-axis, setting its limits to 10^{-2} and 10^{+4}.

As can be seen in Fig. 1a, the great majority of the results are adequately below 1.0. We consider the outliers to be the ones above 100 and mark them with an ellipsis. Matrix 26 is a bidiagonal matrix obtained from a tridiagonal matrix with highly clustered eigenvalues. Its dimension is 1260, its condition number is 2.2668, and its 136 largest eigenvalues have 12 digits in common (its spectrum contains other large clusters). Matrices 198-200 are more difficult: their entries are taken randomly from the interval $[2 * log(\varepsilon), -2 * log(\varepsilon)]$, therefore ranging from ε^{-2} to ε^2 (this is a notoriously hard case, borrowed from the LAPACK testers), and their dimensions are 125, 250 and 500, respectively. For $n = 500$, $s_1 = 1.47 \times 10^{+31}$ and $s_n = 1.34 \times 10^{-284}$ (as computed by BDSQR). For these matrices, resid is $O(10^{-8})$ but $orthU$ and $orthV$ are $O(10^{+13})$. As expected, the effect of large clusters of singular values of matrix 26 and the oddities of matrices 198-200, are propagated to Figs. 1b and 1c. Figure 1b contains additional outliers: case 398 corresponds to a bidiagonal similar to matrix 26 in Fig. 1a; cases 1551 and 1552 are related to a bidiagonal of dimension 1000, obtained from a tridiagonal with one eigenvalue equal to 1.0 and all others equal to $1/\sqrt{\varepsilon}$.



(a) `RANGE`="A" (b) `RANGE`="I" (c) `RANGE`="V"

Fig. 1. *resid*, *orthU*, *orthV* (x-axis, log scale) of `BDSVDX` for `RANGE`="A","I" and"V", double precision. (1a) 200 matrices (y-axis), increasing condition numbers; (1b) $n_I = 10$ for each matrix of `RANGE`="A"; (1c) $n_V = 10$ for each matrix of `RANGE`="A".

Finally, Fig. 2 compares the times taken by `BDSQR`, `BDSDC` and `BDSVDX` on 12 bidiagonals with dimensions ranging from 494 to 2003 (a sample of matrices from Fig. 1a). For `BDSVDX`, we compute all singular values/vectors, the largest 20% and 10% singular values/vectors, and the largest 5 singular values/vectors. For each matrix, the timings are normalized with respect to the time taken by `BDSQR` (y-axis, log scale). As expected, `BDSVDX` is not competitive for all or a relatively large set of singular values/vectors, the gains become apparent at about 10%. In particular, `BDSVDX` is 3 orders of magnitude faster than `BDSQR` and 2 orders of magnitude faster than `BDSDC` for the computation of the largest 5 singular values and vectors of the largest matrix.

6 Conclusions

This paper presented an algorithm for the computation of the SVD of a bidiagonal matrix by means of the eigenpairs of an associated tridiagonal matrix. The implementation, `BDSVDX` (included in the LAPACK 3.6.0 release), provides for the computation of a subset of singular values/vectors, which is important for many large dimensional problems that do not require the full set. Our experiments revealed that this feature can lead to impressive gains in computing times, when compared with existing implementations that are limited to the computation of the full SVD. The implementation discussed here offers opportunities for parallelism, for example by assigning different subsets of values and vectors to different processes.

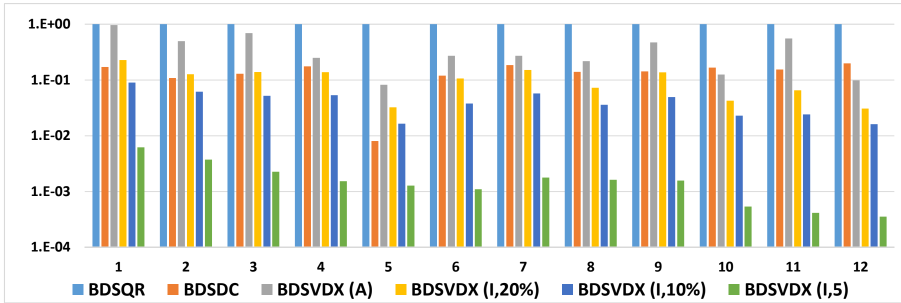


Fig. 2. Normalized times (y-axis, log scale) for `BDSQR`, `BDSDC` and `BDSVDX` on 12 bidiagonals whose dimensions range from 494 to 2003 (x-axis, increasing size), double precision. `BDSVDX`: all, the largest 20% and 10%, and the largest 5 singular values/vectors. For each matrix, the timings are normalized with respect to the time taken by `BDSQR`, which is typically the slowest.

Numerical results on a large set of test matrices substantiated the accuracy of the implementation; the exceptions are matrices with very large condition numbers or highly clustered singular values. Interestingly, we have verified (results not shown) that the accuracy is not so much dependent on the condition number of the singular vectors, $\kappa_{u,v} = \min(\frac{1}{\min_i gap_i} \frac{1}{s_1}, \frac{1}{\varepsilon})$, $gap_i = \min_{j \neq i} |\sigma_i - \sigma_j|$, as we had originally thought. On the other hand, we have identified pathological cases (typically very small singular values) for which the computed singular vectors may not be orthogonal to work precision. A more robust strategy to cope with such cases needs to be investigated; it will be a priority in our future work.

References

[1] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe and H. van der Vorst, editors, Templates for the solution of Algebraic Eigenvalue Problems: A Practical Guide. SIAM, Philadelphia, 2000.

[2] O. Marques, J. Demmel, C. Voemel and B. N. Parlett, A testing infrastructure for symmetric tridiagonal eigensolvers, ACM TOMS, 35, 2008.

[3] J. Demmel, O. Marques, C. Voemel and B. N. Parlett, Performance and accuracy of LAPACK's symmetric tridiagonal eigensolvers, SIAM J. Sci. Comput., 30:1508-1526, 2008.

[4] I. S. Dhillon and B. N. Parlett, Multiple representations to compute orthogonal eigenvectors of symmetric tridiagonal matrices, Linear Algebra Appl., 387:1-28, 2004.

[5] I. S. Dhillon, B. N. Parlett and C. Voemel, The design and implementation of the MRRR algorithm, ACM TOMS, 32:533-560, 2006.

[6] P. Willems and B. Lang, A framework for the MR^3 algorithm: theory and implementation, SIAM J. Sci. Comput., 35:740-766, 2013.

[7] J. Demmel and W. Kahan, Accurate singular values of bidiagonal matrices, SIAM J. Sci. and Stat. Comput., 11:873912, 1990.