



International Conference on Computational Science, ICCS 2010

# On twisted factorizations of block tridiagonal matrices

Wilfried N. Gansterer<sup>a</sup>, Gerhard König<sup>b</sup>

<sup>a</sup>University of Vienna, Research Lab Computational Technologies and Applications

<sup>b</sup>University of Vienna, Department of Computational Biological Chemistry

## Abstract

Non-symmetric and symmetric twisted block factorizations of block tridiagonal matrices are discussed. In contrast to non-blocked factorizations of this type, localized pivoting strategies can be integrated which improves numerical stability without causing any extra fill-in. Moreover, the application of such factorizations for approximating an eigenvector of a block tridiagonal matrix, given an approximation of the corresponding eigenvalue, is outlined. A heuristic strategy for determining a suitable starting vector for the underlying inverse iteration process is proposed.

© 2012 Published by Elsevier Ltd. Open access under [CC BY-NC-ND license](https://creativecommons.org/licenses/by-nc-nd/4.0/).

**Keywords:** twisted factorizations, twisted block factorizations, block tridiagonal eigenvalue problem, eigenvector computation

## 1. Introduction

In this paper, we discuss strategies for efficiently computing twisted block factorizations of a block tridiagonal matrix  $W(p)$  with  $p$  square diagonal blocks. Such factorizations form the basis for approximating eigenvectors of  $W(p)$  if approximations of the corresponding eigenvalues are available.

In the most general setting considered here,  $W(p)$  does not have to be symmetric:

$$W(p) := \begin{pmatrix} B_1 & C_1 & & & & \\ A_2 & B_2 & C_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & A_{p-1} & B_{p-1} & C_{p-1} & \\ & & & A_p & B_p & \end{pmatrix}. \tag{1}$$

The dimensions  $b_i$  of the  $p$  quadratic diagonal blocks  $B_i$  ( $i = 1, \dots, p$ ) are called *block sizes* in the following and determine shape and size of the  $p - 1$  subdiagonal blocks  $A_i$  ( $i = 2, \dots, p$ ), and of the  $p - 1$  superdiagonal blocks  $C_i$  ( $i = 1, \dots, p - 1$ ). In many situations, symmetric  $W(p)$  is of particular interest. In this case  $B_i = B_i^T$  for  $i = 1, \dots, p$ , and  $C_i = A_{i+1}^T$  for  $i = 1, \dots, p - 1$ .

In this paper, we discuss non-symmetric as well as symmetry-preserving factorizations of  $W(p)$  and compare them to related work in the literature. The general structure of these factorizations is  $W(p) = PLDU$  (in the non-symmetric case) or  $W(p) = PLDL^T P^T$  (in the symmetric case), where  $L$  and  $U$  are block tridiagonal with identity matrices along the diagonal,  $D$  is block diagonal, and  $P$  is also block diagonal with permutation matrices along the diagonal.

Email addresses: [wilfried.gansterer@univie.ac.at](mailto:wilfried.gansterer@univie.ac.at) (Wilfried N. Gansterer), [gerhard@mdy.univie.ac.at](mailto:gerhard@mdy.univie.ac.at) (Gerhard König)

### 1.1. Motivation

Since block tridiagonal structure can be considered a generalization of band structure, matrices as defined in (1) arise in many situations. In particular, they can be an intermediate result of a reduction step of a general dense matrix, for example, in a block tridiagonalization process [1] or in a bandwidth reduction process [2, 3]. In particular, the context of block tridiagonalization as a preprocessing step for computing spectral information of a dense symmetric matrix provides motivation for investigating suitable factorizations of  $W(p)$ , as outlined in the following.

The block tridiagonal divide-and-conquer (BD&C) method [4, 5] allows for computing eigenvalues and eigenvectors of symmetric  $W(p)$  without reducing it to tridiagonal form. However, it turns out that in some constellations the accumulation of the eigenvector information in this divide-and-conquer process can become the main performance limiting factor—in particular, in cases where reduced accuracy approximations (with respect to the “full” accuracy determined by the problem instance and its condition as well as by the given floating-point arithmetic) are not sufficient. Thus, a central question arising in this context is the following: Given approximate eigenvalues, is it possible to find efficient alternatives to the eigenvector accumulation process in the BD&C method for efficiently computing the corresponding eigenvectors of a symmetric block tridiagonal matrix?

The idea pursued in this paper is based on representing  $W(p)$  as a product of two block tridiagonal matrices with special structure (for every diagonal block, either the superdiagonal block or the subdiagonal block is nonzero). Equivalently,  $W(p)$  can be represented as a product of three matrices (a block diagonal and two block tridiagonals with the structure mentioned before, but identity matrices along the diagonal). Based on such a representation, the idea is to design a fast and efficient inverse iteration process for computing a desired eigenvector.

The candidate representations of  $W(p)$  are *twisted block factorizations*, a blocked generalization of twisted factorizations of tridiagonal matrices, which have been considered earlier (see, for example, [6]). Among all possible twisted block factorizations of shifted  $W(p)$  one is selected as the basis for inverse iteration with a properly chosen starting vector. This idea is motivated by central components of the MRRR algorithm for computing eigenvectors of a symmetric tridiagonal matrix summarized in [7].

In this paper, we discuss algorithms for computing twisted block factorizations of  $W(p)$ . We distinguish non-symmetric and symmetry-preserving factorizations and compare them to approaches appearing in the literature. When the ultimate objective is to compute an eigenvector of  $W(p)$  based on these factorizations and on a given approximation of the corresponding eigenvalue, central algorithmic questions are (i) how to pick one of the twisted block factorizations and—related to that—(ii) how to choose the starting vector for the inverse iteration process. We also discuss two strategies based on the factorization methods discussed, one of them based on [6], the other based on a heuristic which seems not to have been investigated before.

### 1.2. Related Work

Most existing studies concerning twisted factorizations focussed on *tridiagonal* matrices and were motivated by the objective to efficiently calculate their eigenvectors [7, 8, 9, 10, 6, 11, 12].

In 1990, Demmel and Kahan showed that the Cholesky factorization of a tridiagonal matrix into two bidiagonals can be used to compute all eigenvalues of a symmetric definite tridiagonal matrix to high accuracy [13]. Later, it was also shown that bidiagonal  $LDL^T$  representations of tridiagonal matrices often determine eigenvalues with high relative accuracy [8].

Such results formed the basis for the development of very efficient methods for the calculation of eigenvectors of tridiagonal matrices. Based on Fernando’s solution to Wilkinson’s problem [9] Parlett and Dhillon [6] suggested to use twisted factorizations of tridiagonal matrices to determine a good starting vector for inverse iteration. This is justified by the fact that the position of the largest component in the eigenvector to be computed is associated with the minimal diagonal element of the twisted factorizations. The proper choice of a starting vector based on twisted factorizations leads to stability and rapid convergence of the inverse iteration process.

Less work has been done so far on banded or block tridiagonal matrices. Meurant [14] reviewed the connections between the inverse of a block tridiagonal matrix and its twisted factorization, thereby deducing formulas for the blocks of the inverse of a block tridiagonal matrix. Parlett and Dhillon [6] discussed a blocked extension of the tridiagonal case and they also suggested a strategy for determining a good initial approximation to an eigenvector of  $W(p)$ .

Very recently, Vömel and Slemons published a theoretical treatment of twisted factorizations of banded matrices [15]. They gave a proof of the existence of two twisted factorizations of a given banded matrix by using a double factorization of the twisted block. They also discuss the connections to the inverse of the matrix and consider the use of their twisted factorizations for inverse iteration on band matrices.

However, Vömel and Slemons focus on *non-blocked* twisted factorizations of a banded matrix. When pivoting is introduced for enhancing numerical stability, their approach does in general not preserve the structure of block tridiagonal or banded matrices due to fill-in. In order to address both aspects—numerical stability and preservation of block tridiagonal structure—our focus is on twisted *block* factorizations. By integrating localized pivoting within blocks in the factorization process we can improve numerical stability *without* causing fill-in. The approach discussed in this paper is more directly related to the twisted block factorizations indicated in [6]. However, this paper does not explicitly discuss the pivoting issue. Moreover, we propose a computationally less expensive alternative for determining a starting vector for inverse iteration.

### 1.3. Synopsis

In Section 2, forward and backward block LU factorizations of a block tridiagonal matrix with local pivoting are reviewed. In Section 3 non-symmetric as well as symmetry-preserving twisted block factorizations of  $W(p)$  are discussed. The computation of an eigenvector to a given eigenvalue approximation using inverse iteration based on these factorizations is the topic of Section 4, where we also distinguish several strategies for choosing a starting vector. Conclusions are given in Section 5.

## 2. Block LU Factorizations of a Block Tridiagonal Matrix

As outlined before, the basic idea discussed in this paper is the factorization of block tridiagonal  $W(p)$  into two block tridiagonals or, alternatively, into two block tridiagonals and a block diagonal. In this section, we summarize block LU factorizations of  $W(p)$  with integrated *local* pivoting. This forms the basis for the twisted block factorizations of  $W(p)$  discussed in Section 3. We assume that all factorizations outlined in these two sections exist.

### 2.1. Scalar LU Factorization

Scalar LU factorization decomposes any given matrix  $M$  into a unit lower triangular matrix  $L$  and an upper triangular matrix  $U$  [16]. In the standard *forward process*, the subdiagonal elements in column  $k$  of the matrix  $M$  are eliminated by multiplication with an elimination matrix whose subdiagonal elements of the respective column  $k$  are given as  $M_{ik}/M_{kk}$  ( $i = k + 1, \dots, n$ ). In exact arithmetic, this process breaks down if at some point  $M_{kk} = 0$ . In floating-point arithmetic, stability problems may arise if  $M_{kk}$  is very small. In order to cope with this problem, pivoting strategies have been developed. Partial pivoting, for example, identifies the largest element in absolute value of the current column  $k$  and interchanges the corresponding rows of the submatrix yet to be processed in order to make  $M_{kk}$  as large as possible. Formally, these row interchanges can be represented as the application of a permutation matrix  $P$  from the left:

$$M = PLU.$$

In the following, we generalize this scalar forward LU factorization process to a block-based LU factorization process with local pivoting for block tridiagonal  $W(p)$ .

### 2.2. Forward Block LU Factorization

When generalized to a block tridiagonal matrix  $W(p)$  defined as in (1), the resulting factors  $L$  and  $U$  will be lower and upper block bidiagonal, respectively. We illustrate the process for  $p = 4$ . Based on the ansatz

$$W(4) = \begin{pmatrix} P_1 & & & \\ & P_2 & & \\ & & P_3 & \\ & & & P_4 \end{pmatrix} \begin{pmatrix} L_1 & & & \\ M_2 & L_2 & & \\ & M_3 & L_3 & \\ & & M_4 & L_4 \end{pmatrix} \begin{pmatrix} U_1 & N_1 & & \\ & U_2 & N_2 & \\ & & U_3 & N_3 \\ & & & U_4 \end{pmatrix} \quad (2)$$

$$= \begin{pmatrix} P_1 L_1 U_1 & P_1 L_1 N_1 & & & \\ P_2 M_2 U_1 & P_2 L_2 U_2 + P_2 M_2 N_1 & P_2 L_2 N_2 & & \\ & P_3 M_3 U_2 & P_3 L_3 U_3 + P_3 M_3 N_2 & P_3 L_3 N_3 & \\ & & P_4 M_4 U_3 & P_4 L_4 U_4 + P_4 M_4 N_3 & \end{pmatrix},$$

the defining equations for the block LU factorization process with local pivoting can be derived block by block. Starting from the top left corner (in “forward” direction), the first step is to factorize  $B_1 = P_1 L_1 U_1$  using partial pivoting. Then, from the equations

$$\begin{aligned} P_1 L_1 N_1 &= C_1 \\ P_2 M_2 U_1 &= A_2 \end{aligned} \quad (3)$$

the matrices  $N_1$  and  $M'_2 := P_2 M_2$  can be computed as solutions of two triangular systems.  $B_1$  has to be non singular, so that for arbitrary  $C_1$  and  $A_2$  these linear systems have a unique solution. Note that the permutation matrix from the *next* partial pivoting step appears already implicitly in  $M'_2$  without being known explicitly.

Rewriting the next equation  $B_2 = P_2 L_2 U_2 + M'_2 N_1$  into

$$B_2 - M'_2 N_1 = P_2 L_2 U_2 \quad (4)$$

reveals that the next step is to factorize the Schur complement  $B_2 - M'_2 N_1$  with partial pivoting in order to compute  $P_2$ ,  $L_2$ , and  $U_2$ . Note that only at this point  $P_2$  is computed explicitly (as mentioned before, so far it was only contained implicitly in the solution of Eqn. (3)).

Now we can proceed with solving linear systems for  $N_2$  and  $M'_3 := P_3 M_3$ , factorizing  $B_3 - M'_3 N_2$ , solving for  $N_3$  and  $M'_4 := P_4 M_4$ , and finally factorizing  $B_4 - M'_4 N_3$  (again assuming that all linear systems have a unique solution). As a result, the entire block LU factorization (2) in forward direction of  $W(4)$  has been constructed.

### 2.3. Backward Block LU Factorization

The block tridiagonal LU factorization can also be performed backwards, starting from the factorization of the lower right block  $B_p$ . In this case, the resulting  $L$  and  $U$  will be upper and lower block bidiagonal, respectively. Again, we illustrate the process for  $p = 4$ . Based on the ansatz

$$\begin{aligned} W(4) &= \begin{pmatrix} P_1 & & & \\ & P_2 & & \\ & & P_3 & \\ & & & P_4 \end{pmatrix} \begin{pmatrix} L_1 & M_1 & & \\ & L_2 & M_2 & \\ & & L_3 & M_3 \\ & & & L_4 \end{pmatrix} \begin{pmatrix} U_1 & & & \\ N_2 & U_2 & & \\ & N_3 & U_3 & \\ & & N_4 & U_4 \end{pmatrix} \quad (5) \\ &= \begin{pmatrix} P_1 L_1 U_1 + P_1 M_1 N_2 & P_1 M_1 U_2 & & & \\ P_2 L_2 N_2 & P_2 L_2 U_2 + P_2 M_2 N_3 & P_2 M_2 U_3 & & \\ & P_3 L_3 N_3 & P_3 L_3 U_3 + P_3 M_3 N_4 & P_3 M_3 U_4 & \\ & & P_4 L_4 N_4 & P_4 L_4 U_4 & \end{pmatrix} \end{aligned}$$

we start with factorizing  $B_4 = P_4 L_4 U_4$  using local partial pivoting. From the equations

$$\begin{aligned} P_3 M_3 U_4 &= C_3 \\ P_4 L_4 N_4 &= A_4 \end{aligned}$$

the matrices  $N_4$  and  $M'_3 := P_3 M_3$  can be computed as solutions of two linear systems, assuming (as before) that  $B_4$  is non singular. Then,  $B_3$  can be rewritten as  $B_3 = P_3 L_3 U_3 + M'_3 N_4$ , which leads to

$$B_3 - M'_3 N_4 = P_3 L_3 U_3.$$

Thus, factorizing the Schur complement  $B_3 - M'_3 N_4$  with local partial pivoting yields  $P_3$ ,  $L_3$ , and  $U_3$ . Proceeding analogously to the forward case (but in backward direction) determines the remaining unknown submatrices in the factorization (5).

### 3. Twisted Block Factorizations

A twisted block factorization of  $W(p)$  is the result of combining some forward factorization steps as reviewed in Section 2.2 with some backward factorization steps as reviewed in Section 2.3. We use the abbreviation  $TF(k)$  ( $k = 1, 2, \dots, p$ ) for a twisted block factorization which combines  $k - 1$  forward steps and  $p - k$  backward steps. Note that for  $k = p$  we get the forward factorization from Section 2.2, and for  $k = 1$  we get the backward factorization from Section 2.3. We denote the diagonal block at position  $k$ , where forward and backward elimination steps meet, as “twisted block”.

In Section 3.1, we discuss non-symmetric twisted block factorizations with local pivoting, which yield non-symmetric representations of  $W(p)$ . Symmetry-preserving variants, which may sometimes be relevant for symmetric  $W(p)$ , are the topic of Section 3.2.

#### 3.1. Unsymmetric Twisted Block Factorizations

In the following, we illustrate  $TF(3)$  of  $W(4)$ , where two elimination steps are done in forward direction, and one in backward direction. In order to distinguish the steps done in forward and backward direction, the blocks constructed in the forward direction are marked by the superscript “+”, while the blocks constructed in the backward direction are marked by the superscript “-”. Note that forward and backward elimination processes are completely independent of each other until the computation of the blocks in the row where forward and backward factorization meet.

Based on the ansatz

$$\begin{aligned}
 W(4) &= \begin{pmatrix} P_1^+ & & & \\ & P_2^+ & & \\ & & P_3 & \\ & & & P_4^- \end{pmatrix} \begin{pmatrix} L_1^+ & & & \\ M_2^+ & L_2^+ & & \\ & M_3^+ & L_3 & M_3^- \\ & & & L_4^- \end{pmatrix} \begin{pmatrix} U_1^+ & N_1^+ & & \\ & U_2^+ & N_2^+ & \\ & & U_3 & \\ & & & N_4^- & U_4^- \end{pmatrix} \tag{6} \\
 &= \begin{pmatrix} P_1^+ L_1^+ U_1^+ & & & & & & & \\ P_2^+ M_2^+ U_1^+ & P_1^+ L_1^+ N_1^+ & & & & & & \\ & P_2^+ L_2^+ U_2^+ + P_2^+ M_2^+ N_1^+ & & & & & & \\ & & P_3 M_3^+ U_2^+ & & & & & \\ & & & P_3 L_3 U_3 + P_3 M_3^+ N_2^+ + P_3 M_3^- N_4^- & & & & \\ & & & & P_4^- L_4^- N_4^- & & & \\ & & & & & P_3 M_3^- U_4^- & & \\ & & & & & & P_4^- L_4^- U_4^- & \end{pmatrix},
 \end{aligned}$$

we again derive the defining equations block by block.

In the forward direction, the first step is to factorize  $B_1 = P_1^+ L_1^+ U_1^+$ . Then,  $N_1^+$  and  $M_2^+ := P_2^+ M_2^+$  can be computed as solutions of two linear systems. As in (4), the Schur complement of  $B_2$  is factorized for computing  $P_2^+$ ,  $L_2^+$ , and  $U_2^+$ . Using this information,  $N_2^+$  and  $M_3^+ := P_3 M_3^+$  are computed as solutions of two linear systems.

At this point, the forward part of  $TF(3)$  is completed, and the next steps are conducted in backward direction. After factorizing  $B_4 = P_4^- L_4^- U_4^-$ ,  $N_4^-$  and  $M_3^- := P_3 M_3^-$  are computed as solutions of two linear systems.

Finally, we can work on the third block row where both factorizations meet. The diagonal block  $B_3$  in this row can be expressed as  $B_3 = P_3 L_3 U_3 + M_3^+ N_2^+ + M_3^- N_4^-$ . Thus, from factorizing

$$B_3 - M_3^+ N_2^+ - M_3^- N_4^- = P_3 L_3 U_3$$

we obtain  $P_3$ ,  $L_3$ , and  $U_3$  and thus have determined all blocks in factorization (6).

In some situations (for example, when computing eigenvectors of  $W(p)$  as discussed in Section 4) it is convenient to reformulate the factorization (6) as

$$W(4) = PLDU$$

with block diagonal  $D$  and block tridiagonals  $L$  and  $U$ , which have identity matrices along the diagonal. In our example, for  $TF(3)$  of  $W(4)$

$$L = \begin{pmatrix} I & & & \\ M_2^+ (L_1^+)^{-1} & I & & \\ & M_3^+ (L_2^+)^{-1} & I & M_4^- (L_4^-)^{-1} \\ & & & I \end{pmatrix},$$

$$D = \begin{pmatrix} L_1^+ U_1^+ & & & \\ & L_2^+ U_2^+ & & \\ & & L_3 U_3 & \\ & & & L_4^- U_4^- \end{pmatrix},$$

and

$$U = \begin{pmatrix} I & (U_1^+)^{-1} N_1^+ & & \\ & I & (U_2^+)^{-1} N_2^+ & \\ & & I & \\ & & (U_4^-)^{-1} N_3^- & I \end{pmatrix}.$$

### 3.2. Symmetric Twisted Block Factorizations

So far, the discussion was dominated by non-symmetric LU-based factorizations. Nevertheless, for symmetric  $W(p)$  we can also construct a symmetric factorization  $W(p) = PLDL^T P^T$  with block diagonal  $D$ . This construction proceeds analogously to the method summarized in Section 3.1. The main difference is that a symmetric indefinite factorization, for example, a Bunch-Kaufman factorization [17], has to be used for factorizing diagonal blocks.

It suffices to illustrate the first two steps (in order to simplify the notation, we omit the superscripts for the direction of the factorization process): In the forward direction, we first factorize  $B_1 = P_1 L_1 D_1 L_1^T P_1^T$ , where  $P_1$  is a permutation and  $D_1$  is a direct sum of  $1 \times 1$  and  $2 \times 2$  pivot blocks. Then,  $M_1' := P_2 M_1$  can be computed as the solution of the linear system  $M_1' D_1 L_1^T P_1^T = A_2$ , followed by the symmetric factorization of the updated diagonal block  $B_2 - P_2 M_1 D_1 M_1^T P_2^T$ . The process is continued until the twisted block, and performed analogously in the backward direction starting with the symmetric factorization of  $B_p$ .

## 4. Computing an Eigenvector of $W(p)$

In this section we outline an approach for approximating an eigenvector  $v$  of  $W(p)$  based on the twisted block factorizations summarized in Section 3, assuming that an approximation  $\hat{\lambda}$  of the corresponding eigenvalue  $\lambda$  is available. Analogously to the ideas leading to [7] the approach is based on one step (or a few steps) of inverse iteration on the shifted matrix  $W(p) - \hat{\lambda}I$ . It uses a properly chosen twisted block factorization of the block tridiagonal matrix  $W(p) - \hat{\lambda}I$  for determining a suitable starting vector.

### 4.1. Review Inverse Iteration

Given the eigenvalue approximation  $\hat{\lambda} \approx \lambda$  ( $\hat{\lambda}$  is called “shift” in the following), inverse iteration computes an approximation  $\hat{v}$  for the eigenvector  $v$  as follows:

1. **initialize**  $\hat{v}_{(0)}, i := 0$
2. **repeat**
3.   **solve**  $(W(p) - \hat{\lambda}I)y_{(i+1)} = \hat{v}_{(i)}$
4.    $\hat{v}_{(i+1)} := y_{(i+1)} / \|y_{(i+1)}\|_2$
5.    $i := i + 1$
6. **until convergence**

In general, a random starting vector  $\hat{v}_{(0)}$  is considered appropriate [18]. However, for the special structure of  $W(p)$  considered in this paper it is possible to determine a better starting vector from twisted factorizations of  $W(p)$ , as already indicated in [6].

#### 4.2. Inverse Iteration Based on Twisted Block Factorizations

Given a twisted block factorization  $W(p) - \hat{\lambda}I = PLDL^T P^T$  as discussed in Section 3.2, the solver step 3 of the inverse iteration process involves solving three linear systems, one block diagonal system and two combined forward and back (block) substitutions.

Note that except for the extreme cases TF( $p$ ) and TF(1), the factor  $L$  has nonzero blocks above and below the main block diagonal. Thus, in the general case, solving a linear system with one of these factors resulting from a twisted block factorization TF( $k$ ) involves forward substitution down to the block number  $k - 1$ , and backward substitution up to the block number  $p - k$  until the two directions meet at block number  $k$ .

So far, we have not specified which one of the  $p$  possible twisted block factorizations to use in the inverse iteration process. The choice of one of these factorizations determines the starting vector  $\hat{v}_{(0)}$ . The idea is that with a properly chosen starting vector very few iterations (ideally only one) of the inverse iteration process should suffice for computing a good approximation of the eigenvector  $v$ .

#### 4.3. Choice of Starting Vector

In the following, we mention two strategies for determining the starting vector  $\hat{v}_{(0)}$  for the inverse iteration process on  $W(p) - \hat{\lambda}I$ . We restrict ourselves to starting vectors  $\hat{v}_{(0)}$  with entry one in position  $j$  and zeros in all other positions. When solving a block bidiagonal system with such a vector  $\hat{v}_{(0)}$  as the right hand side, all entries of the solution vector below position  $j$  will be zero. Thus, this position  $j$  is in fact the “starting position” of the back- or forward substitution process and we can identify this starting position with the starting vector  $\hat{v}_{(0)}$  (since  $j$  completely determines  $\hat{v}_{(0)}$ ).

The first (SVD-based) strategy has been mentioned in [6]. It determines a good initial approximation to the eigenvector sought based on the block with the smallest singular value over all diagonal blocks in the matrices  $D$  over all possible twisted block factorizations  $PLDU$ .

Due to the potentially higher multiplicity of eigenvalues in block tridiagonal matrices (compared to irreducible tridiagonal matrices), such block-oriented strategies seem important for another reason: Identifying a starting block allows for determining  $b_i$  different scalar starting positions and thus potentially for approximating  $b_i$  different eigenvectors for an eigenvalue with multiplicity higher than one. Nevertheless, [15] already hints at a possible drawback of this strategy for determining a starting vector: in some situations, the requirement of computing singular value decompositions may be a limiting factor, for example, for large block sizes  $b_i$  (relative to  $n$ ).

Thus, we propose an alternative heuristic scalar strategy: Determine the position  $m$  of the diagonal element with minimum absolute value over all diagonal elements from the blocks  $U_i$  in the matrices  $D$  over all possible twisted block  $PLDU$  factorizations. Then, choose the starting vector  $\hat{v}_0$  for the inverse iteration process with entry one at position  $m$ , and use the factorization which contains this minimum diagonal element  $|D_{mm}|$  for solving the linear system. This heuristic has advantages in computational cost over the one derived from [6], but it remains to be investigated whether any analytical results can be derived and whether it achieves competitive numerical results in practical computations.

#### 4.4. Numerical Example

The applicability of the approach based on twisted block factorizations is demonstrated with a numerical example. Using a preliminary draft implementation of the twisted block factorizations discussed in Section 3, we applied the SVD-based strategy from Section 4.3 for determining starting vectors and computed all eigenvectors of a random symmetric matrix  $W(p)$  with  $n = 1000$  and  $b_i = 5$  ( $i = 1, \dots, p$ ). The eigenvalues used as shifts were computed using the routine LAPACK/dsyevd.

The numerical results are very encouraging. The average residual over all computed eigenvectors was  $1.3277 \cdot 10^{-14}$ , and only one residual was larger than  $10^{-12}$  (see Fig. 1). The average scalar product between corresponding eigenvectors computed by our method and computed by LAPACK/dsyevd was  $1.2234 \cdot 10^{-16}$ .

Moreover, preliminary experiences indicate that the method based on twisted block factorizations can achieve improvements in runtime performance compared to competing tridiagonalization-based approaches using LAPACK routines.

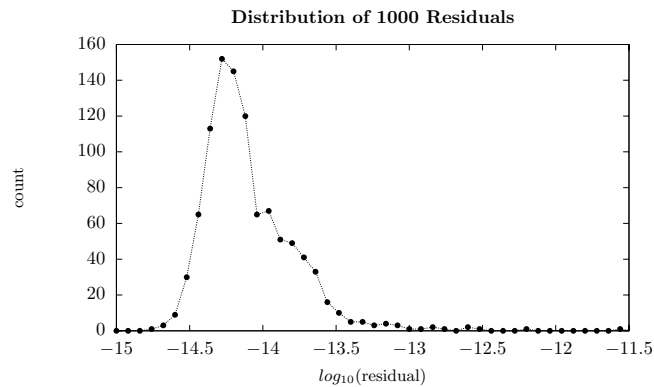


Figure 1: Residuals for all eigenvectors of a random symmetric matrix  $W(p)$  with  $n = 1000$  and  $b_i = 5$  ( $i = 1, \dots, p$ ) computed via twisted block factorizations and the SVD-based strategy for determining the starting vector (bin size 0.08 at the log-scale).

## 5. Conclusions

We have summarized and discussed strategies for computing non-symmetric as well as symmetric twisted factorizations of block tridiagonal matrices. In contrast to some earlier approaches, we focussed on twisted *block* factorizations which integrate localized pivoting while preserving the original structure. Moreover, we discussed the application of such factorizations in the context of approximating eigenvectors of block tridiagonal matrices based on inverse iteration. For this context, we proposed a heuristic for deriving a suitable starting vector for the inverse iteration process which may be an interesting alternative to an earlier approach in the literature.

In the future, we will on the one hand work on the analysis of the proposed heuristic in order to analyze the quality of the resulting starting vector. On the other hand, we will work on questions related to efficient implementations of the concepts developed in this paper, on their experimental evaluation and on the comparison with competing concepts in terms of the resulting quality of the eigenvector approximation as well as in terms of computational efficiency.

## References

- [1] Y. Bai, W. N. Gansterer, R. C. Ward, Block tridiagonalization of “effectively” sparse symmetric matrices, *ACM Trans. Math. Softw.* 30 (2004) 326–352.
- [2] C. H. Bischof, B. Lang, X. Sun, Parallel tridiagonalization through two-step band reduction, in: *Proceedings of the 1994 Scalable High-Performance Computing Conference*, Washington D.C., 1994, pp. 23–27.
- [3] C. H. Bischof, B. Lang, X. Sun, A framework for symmetric band reduction, *ACM Trans. Math. Software* 26 (2000) 581–601.
- [4] W. N. Gansterer, R. C. Ward, R. P. Muller, An extension of the divide-and-conquer method for a class of symmetric block-tridiagonal eigenproblems, *ACM Trans. Math. Softw.* 28 (2002) 45–58.
- [5] W. N. Gansterer, R. C. Ward, R. P. Muller, W. A. Goddard, III, Computing approximate eigenpairs of symmetric block tridiagonal matrices, *SIAM J. Sci. Comput.* 25 (2003) 65–85.
- [6] B. N. Parlett, I. S. Dhillon, Fernando’s solution to Wilkinson’s problem: An application of double factorization, *Linear Algebra Appl.* 267 (1997) 247–279.
- [7] I. S. Dhillon, B. N. Parlett, C. Vömel, The design and implementation of the MRRR algorithm, *ACM Trans. Math. Softw.* 32 (4) (2006) 533–560.
- [8] I. S. Dhillon, B. N. Parlett, Multiple representations to compute orthogonal eigenvectors of symmetric tridiagonal matrices, *Linear Algebra Appl.* 387 (2004) 1–28.
- [9] K. V. Fernando, On computing an eigenvector of a tridiagonal matrix .I. Basic results, *SIAM J. Matrix Anal. Appl.* 18 (4) (1997) 1013–1034.
- [10] B. N. Parlett, For tridiagonals T replace T with LDLt, *J. Comput. Appl. Math.* 123 (1-2) (2000) 117–130.
- [11] B. N. Parlett, I. S. Dhillon, Relatively robust representations of symmetric tridiagonals, *Linear Algebra Appl.* 309 (2000) 121–151.
- [12] B. Parlett, O. Marques, An implementation of the dqds algorithm (positive case), *Linear Algebra Appl.* 309 (1-3) (2000) 217–259.
- [13] J. W. Demmel, W. Kahan, Accurate singular values of bidiagonal matrices, *SIAM J. Sci. Stat. Comput.* 11 (1990) 873–912.
- [14] G. Meurant, A review on the inverse of symmetric tridiagonal and block tridiagonal matrices, *SIAM J. Matrix Anal. Appl.* 13 (3) (1992) 707–728.
- [15] C. Vömel, J. Slemons, Twisted factorization of a banded matrix, *BIT* 49 (2) (2009) 433–447.
- [16] G. H. Golub, C. F. Van Loan, *Matrix Computations*, 3rd Edition, Johns Hopkins University Press, Baltimore, MD, 1996.



- [17] J. R. Bunch, L. C. Kaufman, Some stable methods for calculating inertia and solving symmetric linear equations, *Math. Comp.* 31 (1977) 163–179.
- [18] I. C. F. Ipsen, Computing an eigenvector with inverse iteration, *SIAM Rev.* 39 (1997) 254–291.