

Studies on Accurate Singular Value Decomposition for Bidiagonal Matrices

Munehiro NAGATA

Studies on Accurate Singular Value Decomposition for Bidiagonal Matrices

Munehiro NAGATA

March 2016

Contents

Preface	iii
1 Introduction	1
1.1 Preliminaries	2
1.2 Algorithms compute accurate singular values for bidiagonal matrices	4
1.3 Preconditioning for computing singular vectors	6
1.4 Organization of the thesis	10
2 Error analysis of the m2dLVs algorithms for computing bidi- agonal singular values	11
2.1 The mdLVs transformation	11
2.2 Mixed forward-backward error analysis	15
2.3 Forward error analysis	19
3 Accurate similarity transformation derived from the discrete Lotka-Volterra system for bidiagonal singular values	25
3.1 The dLV_∞ similarity transformation	26
3.2 Mixed forward-backward stability of the dLV_∞ similarity transformation	32
3.3 Forward stability of the dLV_∞ similarity transformation	35
3.4 Numerical examples	38
4 Toward Accurate Singular Vectors of Bidiagonal Matrices	43
4.1 Arbitrary-stride reduction	44
4.2 Block diagonalization with tridiagonal blocks	48
4.3 An equivalent transformation	50

4.4	Extended condition numbers of coefficient matrices	51
5	Concluding remarks	55

Preface

The author, for SVD of bidiagonal matrices, has analyzed numerical stability of singular value computation algorithms and preconditioning toward accurately solving linear systems which correspond to compute singular vectors. More precisely, at first, the author showed a new algorithm which is algebraically equivalent to the original mdLVs algorithm, and confirmed numerical stability of the new algorithm. Additionally, he analyzed behavior of the discrete Lotka-Volterra (dLV) system as the free parameter $\delta \rightarrow \infty$, where the dLV system is the basis of the mdLVs algorithm. Moreover, the author presented a numerical method which transforms a given linear system whose coefficient matrix is positive definite and tridiagonal into another linear system whose coefficient matrix has a smaller condition number than that of coefficient matrix of the given linear system.

Backgrounds of author's research are as follows. The I-SVD is contrived from the mdLVs algorithm and the algorithm applying the twisted factorization based on the dLV type transformation. These algorithms are presented by Iwasaki *et al.* and compute the singular values and singular vectors of a given bidiagonal matrix, respectively. The mdLVs algorithm is a shifted version of the dLV algorithm and it is known that numerical experiments of the mdLVs algorithm show its competitiveness with existing algorithms for computing singular values from the view point of accuracy. However, its numerical stability had not shown in theory(Issue 1). In addition, it had not known that the behavior of the dLV system as the free parameter $\delta \rightarrow \infty$ (Issue 2) though it is known that the convergence speed of the dLV system grows faster as δ becomes larger.

On the other hand, singular vectors computed with the I-SVD algorithm are as almost accurate as those computed with existing algorithms. Computing singular vectors corresponds to solving a linear system whose coefficient matrix has a large condition number, namely, an ill-conditioned linear system. In general, since it is known that accuracy of the solution is not good enough in the case when we solve a given ill-conditioned linear system, it is desirable

that we apply reducing the condition number of coefficient matrix which is called preconditioning before we solve the system. But, this process is not practiced(Issue 3) in the I-SVD algorithm.

The main purpose of author's research is contributing some useful results gained by working on the above three issues improving accuracy of the I-SVD algorithm.

The main results which the author has obtained are as follows. In [A1], the author *et al.* have shown a new algorithm which is algebraically equivalent to the original mdLVs algorithm and its numerical stability. In addition, they quantitatively have estimated relative changes of singular values which rounding error causes. In [A2], the author *et al.* have studied the behavior of the dLV system as $\delta \rightarrow \infty$ and derived a new recursion formula corresponding to a similarity transformation of matrix. Then, they have shown that the recursion formula is applicable for computing singular values similarly to the original dLV algorithm and it is numerically stable and forward stable. In addition, they quantitatively have estimated relative changes of singular values which rounding error cause and shown that upper bounds to the relative changes are smaller than those of the dLV algorithm. In [A3], the author *et al.* have shown that, with an appropriate nonsingular matrix S , if we transform an M -tridiagonal matrix such that all entries are zero except for the diagonal, $(i, i + M)$ and $(i + M, i)$ entries into a $2M$ -tridiagonal matrix $\tilde{A}(= SA)$, the condition number of $\tilde{A}$ becomes smaller than that of A . We have proven it by applying the result shown by Wang *et al.* in the case when A is tridiagonal(1-tridiagonal). Additionally, we have shown that block diagonalizing $2M$ -tridiagonal matrices with appropriate similarity transformation enables us to implement preconditioning for computing singular vectors with the I-SVD algorithm.

Chapter 1

Introduction

Singular value decomposition (SVD, for simplicity) is one of the matrix decomposition methods in linear algebra. It is known that SVD is applied to processing[29] and data search[22] in information processing. For a rectangular matrix $A \in \mathbf{R}^{m \times n} (m \geq n)$, SVD is expressed as follows.

$$A = U \begin{pmatrix} \Sigma \\ O \end{pmatrix} V^T,$$

where $U \in \mathbf{R}^{m \times m}$ and $V \in \mathbf{R}^{n \times n}$ are orthogonal matrices, $\Sigma \in \mathbf{R}^{n \times n}$ is a diagonal matrix whose i -th diagonal element $\sigma_i(A)$ is the i -th singular value of A , and $O \in \mathbf{R}^{(m-n) \times n}$ is a zero matrix. It is noted that $\sigma_1(A) \geq \sigma_2(A) \geq \dots \geq \sigma_n(A) \geq 0$. The i -th column of U is called the i -th left singular vector and the i -th column of V are called the i -th right singular vector, respectively.

One of commonly used algorithms for SVD consists of the following 4 phases:

1. Apply a preprocessing to transform a given dense matrix A into a bidiagonal matrix B without changing the singular values.
2. Compute the singular values of B .
3. Compute the singular vectors of B .
4. Apply a postprocessing to transform the computed singular vectors of B into those of A .

In this thesis, we discuss accuracy in the 2nd and 3rd phases.

When SVD is executed with numerical algorithms in finite precision arithmetic, a greater or less extent, almost certainly, numerical errors (ex. rounding error) arise. Then, it concerns all computational scientists and engineers who

demand the accuracy of numerical solutions. Thus, in order to remove their concern, we discuss numerical stability or condition of singular value computation algorithm and singular vector computation algorithm which are or will be incorporated to the integrable singular value decomposition (I-SVD) algorithm[26, 35, 36], where the I-SVD algorithm is an algorithm for the SVD of B , namely, it corresponds to the 2nd and 3rd phases. It implies that the I-SVD algorithm becomes applicable to the SVD for arbitrary A by adding some appropriate preprocessing(corresponds to the 1st phase) and postprocessing(corresponds to the 4th phase). The reason why we choose the I-SVD algorithm is that it has potential for progress in accuracy. It is mentioned in Section 1.2 and 1.3.

This chapter is organized as follows. In Section 1.1, we confirm the senses of *numerical stability* and *condition* for discussing accuracy of numerical algorithms. In Section 1.2, we mention some algorithms which computes accurate singular values for bidiagonal matrices. In Section 1.3, we mention some algorithms which computes accurate singular vectors for bidiagonal matrices, and show an approach toward computing more accurate singular vectors with preconditioning, which is used for solving linear systems. In Section 1.4, we give organization on and after Chapter 2.

1.1 Preliminaries

Understanding backward errors and forward errors is vital to showing numerical stability of numerical algorithms. Let $\hat{y}$ be an approximation to $y = f(x)$, which is computed in an arithmetic of precision u , where f is a real scalar function of a real scalar variable x . Then, since it does not necessarily mean that it exactly holds that $\hat{y} = y$, we can interpret $\hat{y}$ as the exact solution to the input data $x + \Delta x$, namely, $\hat{y} = f(x + \Delta x)$. The values of $|\Delta x|$ and $|\Delta x/x|$ and the process of estimating them are called backward errors and backward error analysis, respectively. The term “backward” means the errors are *reflected back* into the original problem [9]. Backward error analysis on numerical algorithms was provided by Wilkinson [39, 40] about 50 years ago. According to [14], “although a number of more recent books have included analysis of rounding errors, none has treated the subject in the same depth as Wilkinson.”

Through backward error analysis, we can grasp rounding errors as perturbations of the input data and resolve the question of bounding the forward error into perturbation theory, where the forward error means the absolute or relative error of $\hat{y}$. If a numerical algorithm which computes $y = f(x)$ works

out $\hat{y}$ with a small backward error for any x , then the algorithm is called backward stable. It is noted that the definition of the term “small” depends on context. On the other hand, in the case that there exists the weaker relational expression $\hat{y} + \Delta y = f(x + \Delta x)$ than $\hat{y} = f(x + \Delta x)$, instead of backward stability, numerical stability is argued with the following mixed forward-backward error result

$$\hat{y} + \Delta y = f(x + \Delta x), \quad |\Delta y| \leq \epsilon|y|, \quad |\Delta x| \leq \eta|x|.$$

If ϵ and η are small, an algorithm computing $y = f(x)$ is numerically stable. This definition is restricted to numerical algorithms whose rounding errors occupy the most of ones since *stability* holds various intention in other disciplines and fields of numerical analysis.

Other than stability, *condition* is often used as an index measuring the quality of a problem. The condition of a problem indicates how much perturbations in the data affect the solution of the problem. Especially, a problem is called ill-conditioned if small perturbations in the data bring relatively large perturbations in the solution. Otherwise, a problem is called well-conditioned. It is noted here that condition is different from the degree of difficulty in solving the problem.

We assess the condition of a problem with a condition number and determine whether or not a problem is ill-conditioned. For example, we derive the condition number for f as follow [14]. Let f be twice differentiable, then, we have

$$\begin{aligned} \hat{y} - y &= f(x + \Delta x) - f(x) \\ &= f'(x)\Delta x + \frac{f''(x + \theta\Delta x)}{2!}\Delta x^2, \quad \theta \in (0, 1). \end{aligned}$$

By considering the relative error of y , we obtain

$$\frac{\hat{y} - y}{y} = \left(\frac{xf'(x)}{f(x)} \right) \frac{\Delta x}{x} + O(\Delta x^2).$$

Since it implies that

$$\left| \frac{\hat{y} - y}{y} \right| \approx \left| \frac{xf'(x)}{f(x)} \right| \left| \frac{\Delta x}{x} \right|,$$

the quality

$$\kappa(x) = \left| \frac{xf'(x)}{f(x)} \right|$$

is called the *condition number* of f . Hence, when $\kappa(x)$ is large, it may be that small perturbations in input data x give large perturbations in the solution y , namely, the problem is ill-conditioned. If $\mathbf{x}$ or $\mathbf{f}$ is a vector, then, the condition number is defined in a similar way using norms, and it measures the *maximum* relative change for *some* vectors Δx .

In addition, in general, the condition of linear system $A\mathbf{x} = \mathbf{b}$ is evaluated with the condition number of the coefficient matrix $A \in \mathbf{R}^{m \times m}$. Let $\hat{\mathbf{x}}$ be the solution of the perturbed linear system $(A + E)\hat{\mathbf{x}} = \mathbf{b}$, then, by taking into account that $\hat{\mathbf{x}} - \mathbf{x} = -A^{-1}E\hat{\mathbf{x}}$, we derive

$$\frac{\|\hat{\mathbf{x}} - \mathbf{x}\|}{\|\mathbf{x}\|} \approx \frac{\|\hat{\mathbf{x}} - \mathbf{x}\|}{\|\hat{\mathbf{x}}\|} \leq \kappa(A) \cdot \frac{\|E\|}{\|A\|},$$

where $\kappa(A) = \|A\| \cdot \|A^{-1}\|$ just then denotes the matrix condition number [9]. If $\|A\|$ is the matrix 2-norm, the matrix condition number is written as

$$\kappa(A) = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}. \quad (1.1)$$

1.2 Algorithms compute accurate singular values for bidiagonal matrices

By treading the beaten track, namely, using rounding error analysis and perturbation theory, depending on the algorithm, we can evaluate its backward and forward errors, and show its numerical stability. In singular value computation for bidiagonal matrices, the QR algorithm with shifts (QRs algorithm) [7] and differential quotient difference with shifts (dqds) algorithm [11] are known as typical numerical ones which compute accurate singular values.

The QRs algorithm for bidiagonal matrices is developed by Demmel and Kahan, and it is applied with the QR decomposition. The QR decomposition is a matrix operation which factors a given matrix A into a product of an orthogonal matrix Q and an upper triangular matrix R with nonnegative diagonal elements, that is, $A = QR$ in matrix expression. The QR algorithm with no shift computes singular values of given m -by- m real bidiagonal matrix B by the iteration $A_k = Q_k R_k$ and $A_{k+1} := R_k Q_k$ for $k = 0, 1, \dots$, which is called the QR iteration. Note that $A_0 := B^T B$ or $B B^T$. Demmel and Kahan have shown upper bounds to accumulated forward errors by the QR iteration with the fact that relative perturbations of singular values is expressed the product of the relative perturbation of each matrix element. It implies that the QR algorithm is forward stable since those upper bounds are small. Then, it is

known that singular values computed with the QR algorithm are absolutely accurate.

The dqds algorithm for bidiagonal matrices is derived from the progressive quotient difference with shifts (pqds) algorithm of Rutishauser by introducing temporary variables. Those variables reduce concern for cancellation by subtraction in finite precision arithmetic. Additionally, it is known that the dqds algorithm is numerically stable. Fernando and Parlett have shown it with mixed forward-backward error analysis, and estimated relative perturbations of singular values on the dqds algorithm as well as Demmel and Kahan have shown. As those perturbations are significantly small, singular values computed with the dqds algorithm have high relative accuracy.

Besides numerical stability, Fernando and Parlett presented a shift strategy. Since the dqds algorithm is algebraically equivalent to the pqds algorithm, we can be perceived that the dqds algorithm is the Cholesky decomposition with shift of tridiagonal matrices obtained from a product of bidiagonal matrices, that is,

$$(B^{(n+1)})^\top B^{(n+1)} = B^{(n)}(B^{(n)})^\top - \tau^{(n)}I,$$

where $\tau^{(n)}$ such that $0 \leq \tau^{(n)} < \sigma_{\min}^2(B^{(n)})$ is called a shift, which accelerates convergence speed of the algorithm, and $B^{(0)}$ is a given bidiagonal matrix A . By adopting good values of shift, the number of iteration reduces. It produces a good result that not just computation time becomes smaller but also accuracy rises from the point of view that numerical errors accumulate. Thus, we may argue that a good shift strategy leads to accuracy.

Compared to the QRs algorithm and dqds algorithm, the modified discrete Lotka-Volterra with shifts (mdLVs) algorithm is a newer one for singular value computation of bidiagonal matrices [19] and incorporated into the I-SVD algorithm for computing singular value decomposition. It has been proposed as a shifted version of the dLV algorithm which is based on the finite discrete Lotka-Volterra (dLV) system,

$$\begin{cases} u_k^{(n+1)}(1 + \delta^{(n+1)}u_{k+1}^{(n+1)}) = u_k^{(n)}(1 + \delta^{(n)}u_{k-1}^{(n)}), & k = 1, 2, \dots, 2m-1, \\ u_0^{(n)} \equiv 0, \quad u_{2m}^{(n)} \equiv 0, \quad \delta^{(n)} > 0, & n = 0, 1, \dots, \end{cases} \quad (1.2)$$

where $u_k^{(n)}$ and $\delta^{(n)}$ denote the values of u_k and δ at the discrete time $t = \sum_{i=0}^{n-1} \delta^{(i)}$, respectively. The finite dLV system is yielded with a time discretization of the Lotka-Volterra system [15, 42], which is known in biological mathematics as a prey-predator model. In more detail, it describes the individual number of each species at the continuous time, where every species is

numbered such that the k th species preys on the $(k+1)$ th and is simultaneously food for the $(k-1)$ th.

It is shown in [16] that the dLV system (1.2) with any positive constant $\delta^{(n)}$ for $n = 0, 1, \dots$ is applicable for computing the singular values of the bidiagonal matrix B . Under the initial settings $u_{2k-1}^{(0)} = (b_{k,k})^2 / (1 + \delta^{(0)} u_{2k-2}^{(0)})$ and $u_{2k}^{(0)} = (b_{k,k+1})^2 / (1 + \delta^{(0)} u_{2k-1}^{(0)})$, it holds that $\lim_{n \rightarrow \infty} u_{2k-1}^{(n)} = (\sigma_k(B))^2$ and $\lim_{n \rightarrow \infty} u_{2k}^{(n)} = 0$ where $\sigma_k(B)$ denotes the singular value of B with the decreasing order $\sigma_1(B) > \sigma_2(B) > \dots > \sigma_m(B)$. The discussion in [17] also grasps the case where positive and bounded $\delta^{(n)}$ is variable for $n = 0, 1, \dots$. Namely, the singular values of B are given through the sequence of the similarity transformations based on the dLV system (1.2). As a method for the SVD computation, the dLV system has been given a favorable comment that “numerical experiments show its strong competitiveness with existing SVD software packages” by Chu [6]. In addition, according to numerical examples in [20], there are cases when singular values computed with the mdLVs algorithm are more accurate than those computed the QR algorithm and the dqds algorithm. However, it has not been shown that the mdLVs algorithm is numerically stable and forward stable. Hence, it is worth modifying the mdLVs algorithm into numerical stable one. It is remarkable that Fernando and Parlett has succeeded to show numerical stability of the dqds algorithm which is algebraically equivalent to the pqds algorithm in [11] though numerical stability of the pqds algorithm has not been shown.

As previously stated, we can freely choose the parameter $\delta^{(n)}$ in the mdLVs algorithm provided that it is positive and bounded. From the point of view that numerical errors accumulate, it is desirable that the free parameter $\delta^{(n)}$ is determined such that the number of iteration reduces. Since the convergence speed of $u_{2k-1}^{(n)}$ and $u_{2k}^{(n)}$ in the dLV system grow faster as $\delta^{(n)}$ becomes larger [16], we should choose the large value as much as possible. Thus, it is worth confirming how the dLV system behaves as the limit $\delta^{(n)} \rightarrow \infty$.

1.3 Preconditioning for computing singular vectors

After computing singular values, the I-SVD algorithm solve following linear system whose coefficient matrix is tridiagonal,

$$(B^\top B - \sigma_i^2 I) \mathbf{v}_i = \mathbf{0}. \quad (1.3)$$

Since the coefficient matrix $A = B^\top B - \sigma_i^2(B)I$ is normal, from (1.1) it holds that

$$\kappa(A) = \frac{|\lambda(A)|_{\max}}{|\lambda(A)|_{\min}}, \quad (1.4)$$

where $|\lambda(A)|_{\max} = \max_{1 \leq i \leq m} |\lambda_i(A)|$ and $|\lambda(A)|_{\min} = \min_{1 \leq i \leq m} |\lambda_i(A)|$. Hence, the more the computed singular value $\hat{\sigma}_i(B)$ is accurate, the more the linear system (1.3) becomes ill-conditioned. In this case, in order to compute accurate singular vectors, we must devise an algorithm for solving the linear system (1.3).

Parlett and Dhillon presented the multiple relatively robust representations (MR³) algorithm [28], which computes the eigenvectors of given real symmetric tridiagonal matrix $B^\top B$ by applying the twisted factorization to the coefficient matrix of (1.3). It is known that the MR³ algorithm accurately computes eigenvector $\mathbf{v}_i$ corresponding to eigenvalue $\lambda_i(B^\top B)$ in the case when the relative gap between $\lambda_i(B^\top B)$ and its nearest eigenvalue $\lambda_{i+1}(B^\top B)$ is large. Thus, by applying the twisted factorization, Parlett and Dhillon found a method for computing accurately orthogonal eigenvectors from the ill-conditioned problem whose coefficient matrix has the large relative gap.

On the other hand, in contradistinction to the twisted factorization based on the qd type transformation presented by Parlett and Dhillon, Iwasaki *et al.* presented a new method for computing the twisted factorization based on the dLV type transformation [18]. This method is incorporated in the I-SVD algorithm, and it corresponds to computing the left singular vectors of B in SVD. It is known that singular vectors computed with the I-SVD algorithm are as accurate in the sense of SVD and orthogonality with respect to singular vectors as those computed with the QR algorithm.

In general, solving an ill-conditioned problem such as (1.3) has a bad influence on accuracy of the solution. Hence, it is desirable that we apply preconditioning to the ill-conditioned problem. In the case when we rewrite an original linear system $A\mathbf{x} = \mathbf{b}$ into another linear system

$$M_P^{-1}A\mathbf{x} = M_P^{-1}\mathbf{b} \quad (1.5)$$

with a nonsingular $M_P \in \mathbf{R}^{m \times m}$, the new linear system is called a *preconditioned system* and M_P is the *preconditioning matrix* or *preconditioner*. Also, we can also rewrite an original linear system $A\mathbf{x} = \mathbf{b}$ as follows.

$$M_L^{-1}AM_R^{-1}(M_R\mathbf{x}) = M_L^{-1}\mathbf{b}, \quad (1.6)$$

where $M_L, M_R \in \mathbf{R}^{m \times m}$ are nonsingular and called the *left preconditioner* and the *right preconditioner*, respectively. Note that the preconditioned systems (1.5) and (1.6) have the same solution as the original system in exact arithmetic. In order to make computed singular vectors more accurate, we should put off solving the linear system and transform the original system (1.3) into an appropriate preconditioned system.

Preconditioning is applied to improve convergence since it is known that the condition number of coefficient matrix involves convergence of iterative method, for example, conjugate gradient method [24, 32]. As one of preconditioned conjugate gradient method, the incomplete Cholesky-conjugate gradient (ICCG) method is well-known [23, 25]. In the ICCG method, a positive-definite symmetric coefficient matrix A is decomposed as follows.

$$A = LL^\top + E \quad (1.7)$$

or

$$A = \check{L}\check{D}\check{L}^\top + E,$$

where $\check{D}$ is a diagonal matrix, L and $\check{L}$ are tridiagonal matrices such that $L = \check{L}\check{D}^{1/2}$ and E is an error matrix. Then, the original linear system is preconditioned as follows.

$$L^{-1}A(L^\top)^{-1}(L^\top \mathbf{x}) = L^{-1}\mathbf{b}, \quad (1.8)$$

where we use the (1.7) decomposition. Thus, we can expect to improve convergence since the coefficient matrix $L^{-1}A(L^\top)^{-1}$ of the preconditioned system (1.8) will be an approximate identity matrix, namely, the condition number is approximately equal to 1.

For the linear system (1.3), we also consider an appropriate preconditioning which reduces the condition number of the coefficient matrix $A = B^\top B - \sigma_i^2(B)I$. By recalling that the MR³ algorithm and the twisted factorization based on the dLV type transformation have accurate results, it is natural that we reduce the condition number of A with a nonsingular matrix S at first and then, rewrite it into a matrix to which the MR³ algorithm and the twisted factorization based on the dLV type transformation is applicable by similarity transformation with some appropriate nonsingular matrix P . In matrix expression, it is written as follows.

$$P^{-1}SAP(P^{-1}\mathbf{x}) = \mathbf{0}. \quad (1.9)$$

In this precondition system, the left preconditioner $M_L = (P^{-1}S)^{-1}$ and the right preconditioner $M_R = P^{-1}$.

In 2010, Wang *et al.* showed in [38] that the ratio of maximal eigenvalue to minimal eigenvalue of the pentadiagonal coefficient matrix is smaller than that of given tridiagonal matrix if it is positive definite and has the same sign in each pair of symmetric subdiagonal entries, where the pentadiagonal coefficient matrix is obtained from given tridiagonal matrix with the 1st step of the cyclic reduction method. In addition, it has been shown that the 1st step of the cyclic reduction method for the original coefficient matrix A is given as a form of SA . These facts make us expect to improve the ill-conditioned problem (1.3) though $B^\top B - \hat{\sigma}_i^2(B)I$ is not positive definite in the case when computed singular value $\hat{\sigma}_i(B)$ is larger than exact singular value $\sigma_i(B)$.

The cyclic reduction method is a classical technique for solving linear systems with tridiagonal coefficient matrices [10, 12]. This method involves a sequence of transformations of coefficient matrices such that two bands (except for the diagonal band and zero-values bands) move stepwise away from the diagonal band. In the final step of the cyclic reduction method, the coefficient matrix is transformed into a diagonal matrix. The cyclic reduction method has been extended to the block cyclic reduction method for application to block tridiagonal linear systems that appear when solving partial differential equations, such as Poisson's equation [4, 34].

Another extension of the cyclic reduction method is the stride reduction method. The $2M$ -stride reduction method is applicable to linear systems with coefficient matrices of the form:

$$A := \begin{pmatrix} b_1 & & a_1 & & & \\ & \ddots & & \ddots & & \\ c_{M+1} & & b_{M+1} & & \ddots & \\ & \ddots & & \ddots & & a_{m-M} \\ & & \ddots & & \ddots & \\ & & & c_m & & b_m \end{pmatrix}. \quad (1.10)$$

In this thesis, we refer to the band matrix A as an M -tridiagonal matrix. A $2M$ -tridiagonal matrix has the same form as that in (1.10), but replacing the suffix M with $2M$. These appear in coefficient matrices of linear systems following $2M$ -stride reduction of linear systems with M -tridiagonal coefficient matrices. It is clear that 2-stride reduction is equal to the 1st step of the cyclic reduction method. Therefore, the 1st step of the cyclic reduction method may be regarded as a special case of stride reduction. This case is referred to

hereinafter as 2-stride reduction.

The ratios of maximal eigenvalue to minimal eigenvalue of coefficient matrices have not yet been fully studied at all steps of the cyclic reduction method for obtaining a linear system with a diagonal coefficient matrix. In this thesis, for simplicity, we call the ratio (1.4) of maximal eigenvalue to minimal eigenvalue of matrix the extended condition number. If A is normal, then, the extended condition number is equal to the matrix condition number of A . Thus, by investigating how condition numbers and extended condition numbers of coefficient matrices behave in the stride reduction method, we may find some properties which is useful for solving the linear system to compute singular vectors.

1.4 Organization of the thesis

On and after Chapter 2, this thesis is organized as follows.

In Chapter 2, at first, the m2dLVs transformation is introduced in order to improve accuracy of the mdLVs algorithm, where the m2dLVs transformation is algebraically equivalent to the original mdLVs transformation which is the recursion formula in the mdLVs algorithm. Then, numerical stability of the m2dLVs transformation, the relative perturbations with respect to singular values and the forward errors in the m2dLVs transformation are discussed.

In Chapter 3, by taking the limit $\delta^{(n)} \rightarrow \infty$ in the dLV system, a new recursion formula of the similarity transformation is derived. We call the new recursion formula the dLV_∞ similarity transformation in this thesis. Then, it is shown that the dLV_∞ similarity transformation is applicable to singular value computation and numerically and forward stable. Moreover, some numerical example are given in order to confirm numerical and forward stability of the dLV_∞ similarity transformation.

In Chapter 4, the arbitrary-stride reduction method is mentioned and it is clarified that the arbitrary-tridiagonal matrix is transformed by the permutation matrix into a tridiagonal matrix without changing the eigenvalues of the matrix. Then, it is shown that $2M$ -stride reduction is expressed with a combination of 2-stride reduction and row and column permutations. Moreover, using the findings of Wang *et al.* [38], a remarkable property with respect to the condition numbers or extended condition numbers in the arbitrary-stride reduction method. At the end of the chapter, two numerical examples are presented in order to confirm the property.

Chapter 5 describe concluding remarks of the thesis.

Chapter 2

Error analysis of the m2dLVs algorithms for computing bidiagonal singular values

In Chapter 1, we describe some features of the mdLVs algorithm. From the point of view of computational scientists and engineers who demand the accuracy of numerical solutions, it is a problem that numerical stability of the mdLVs algorithm which is incorporated into the I-SVD algorithm has not shown. Actually, due to difficulty of showing numerical stability, we rewrite the main recursive formula in the mdLVs algorithm (the mdLVs transformation) into another algebraically equivalent one (the m2dLVs transformation). The main purpose of this chapter is to show numerical stability of the m2dLVs transformation.

This chapter is organized as follows. In Section 2.1, we briefly explain the features of the mdLVs transformation. With respect to the mdLVs transformation, we present a perturbation theorem on singular values in Section 2.2, and then, by taking account of a part of the results in Section 2.2, we also estimate the forward errors in Section 2.3.

2.1 The mdLVs transformation

In this section, we give a short explanation for the mdLVs transformation based on the dLV system (1.2).

Some integrable systems have their matrix representation called the Lax

representation. The Lax representation for the dLV system (1.2) is written as

$$L^{(n+1)}R^{(n+1)} - \frac{1}{\delta^{(n+1)}}I = R^{(n)}L^{(n)} - \frac{1}{\delta^{(n)}}I, \quad (2.1)$$

$$L^{(n)} := \begin{pmatrix} q_1^{(n)} & & & \\ & 1 & q_2^{(n)} & \\ & & \ddots & \ddots \\ & & & 1 & q_m^{(n)} \end{pmatrix}, \quad R^{(n)} := \begin{pmatrix} 1 & e_1^{(n)} & & \\ & 1 & \ddots & \\ & & \ddots & e_{m-1}^{(n)} \\ & & & 1 \end{pmatrix},$$

$$q_k^{(n)} := \frac{1}{\delta^{(n)}}(1 + \delta^{(n)}u_{2k-2}^{(n)})(1 + \delta^{(n)}u_{2k-1}^{(n)}), \quad e_k^{(n)} := \delta^{(n)}u_{2k-1}^{(n)}u_{2k}^{(n)},$$

where I is the m -by- m identity matrix. The equality in the each entry leads to the dLV system (1.2). Let us introduce the tridiagonal matrix,

$$Y^{(n)} := L^{(n)}R^{(n)} - \frac{1}{\delta^{(n)}}I = \begin{pmatrix} w_1^{(n)} & w_1^{(n)}w_2^{(n)} & & \\ 1 & w_2^{(n)} + w_3^{(n)} & \ddots & \\ & \ddots & \ddots & w_{2m-3}^{(n)}w_{2m-2}^{(n)} \\ & & 1 & w_{2m-2}^{(n)} + w_{2m-1}^{(n)} \end{pmatrix}, \quad (2.2)$$

where $w_k^{(n)} := u_k^{(n)}(1 + \delta^{(n)}u_{k-1}^{(n)})$. Then it follows from (2.1) and (2.2) that

$$Y^{(n+1)} = R^{(n)}Y^{(n)}(R^{(n)})^{-1}. \quad (2.3)$$

Hereinafter, we call the similarity transformation from $Y^{(n)}$ to $Y^{(n+1)}$ with the dLV transformation. Equation (2.3) implies that the eigenvalues of $Y^{(n)}$ are kept constant under the evolution from n to $n+1$.

The dLV transformation differs from the similarity transformations associated with the qd algorithm in [11] and appearing in [3, 1] essentially. The dLV transformation is not a simple LR transformations. Though, in fact, $Y^{(n)}$ can be decomposed into the LR form,

$$Y^{(n)} := \bar{L}^{(n)}\bar{R}^{(n)},$$

$$\bar{L}^{(n)} := \begin{pmatrix} w_1^{(n)} & & & \\ 1 & w_3^{(n)} & & \\ & \ddots & \ddots & \\ & & 1 & w_{2m-1}^{(n)} \end{pmatrix}, \quad \bar{R}^{(n)} := \begin{pmatrix} 1 & w_2^{(n)} & & \\ & 1 & \ddots & \\ & & \ddots & w_{2m-2}^{(n)} \\ & & & 1 \end{pmatrix},$$

the product $\bar{R}^{(n)}\bar{L}^{(n)}$ is not equal to $Y^{(n+1)}$ given by the dLV transformation.

Moreover, let us introduce the diagonal matrix,

$$G^{(n)} := \text{diag} \left(\prod_{j=1}^{m-1} \sqrt{w_{2j-1}^{(n)} w_{2j}^{(n)}}, \prod_{j=2}^{m-1} \sqrt{w_{2j-1}^{(n)} w_{2j}^{(n)}}, \dots, \sqrt{w_{2m-3}^{(n)} w_{2m-2}^{(n)}}, 1 \right).$$

Then, by taking account of it in (2.3) with (2.2), we get

$$Y_s^{(n+1)} = [(G^{(n+1)})^{-1} R^{(n)} G^{(n)}] Y_s^{(n)} [(G^{(n+1)})^{-1} R^{(n)} G^{(n)}]^{-1}, \quad (2.4)$$

where $Y_s^{(n)} := (G^{(n)})^{-1} Y^{(n)} G^{(n)}$ is the symmetric tridiagonal matrix,

$$Y_s^{(n)} = \begin{pmatrix} w_1^{(n)} & \sqrt{w_1^{(n)} w_2^{(n)}} & & & \\ \sqrt{w_1^{(n)} w_2^{(n)}} & w_2^{(n)} + w_3^{(n)} & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \sqrt{w_{2m-3}^{(n)} w_{2m-2}^{(n)}} & \sqrt{w_{2m-2}^{(n)} + w_{2m-1}^{(n)}} & \\ & & & & \end{pmatrix}.$$

The symmetric tridiagonal matrix $Y_s^{(n)}$ can be decomposed as

$$Y_s^{(n)} = (B^{(n)})^\top B^{(n)}, \quad B^{(n)} := \begin{pmatrix} \sqrt{w_1^{(n)}} & \sqrt{w_2^{(n)}} & & & \\ & \sqrt{w_3^{(n)}} & \ddots & & \\ & & \ddots & \sqrt{w_{2m-2}^{(n)}} & \\ & & & \sqrt{w_{2m-1}^{(n)}} & \end{pmatrix}. \quad (2.5)$$

Equations (2.4) and (2.5) claim that the dLV transformation sequentially generates $B^{(1)}, B^{(2)}, \dots$ whose singular values are equal to those of $B^{(0)} = B$. The variables $w_1^{(n+1)}, w_2^{(n+1)}, \dots, w_{2m-1}^{(n+1)}$ in $B^{(n+1)}$ are computed from $w_1^{(n)}, w_2^{(n)}, \dots, w_{2m-1}^{(n)}$ in $B^{(n)}$ as follows.

$$\begin{cases} u_k^{(n)} = w_k^{(n)} / (1 + \delta^{(n)} * u_{k-1}^{(n)}), & k = 1, 2, \dots, 2m-1, \\ w_{k-1}^{(n+1)} = u_{k-1}^{(n)} * (1 + \delta^{(n)} * u_k^{(n)}), & k = 2, 3, \dots, 2m-1, \end{cases}$$

where $u_0^{(n)} = 0$ and $u_{2m}^{(n)} = 0$. As is described in Chapter 1, $u_{2k-1}^{(n)} \rightarrow (\sigma(B))^2$ and $u_{2k}^{(n)} \rightarrow 0$ as $n \rightarrow \infty$. So, $w_{2k-1}^{(n)} \rightarrow (\sigma_k(B))^2$ and $w_{2k}^{(n)} \rightarrow 0$ as $n \rightarrow \infty$.

In order to accelerate the convergence, a shift of origin is introduced into the dLV transformation [19]. The mdLVs transformation is presented as a shifted version of the dLV transformation and the recursion formula is expressed as

follows.

$$\begin{cases} u_k^{(n)} = w_k^{(n)} / (1 + \delta^{(n)} * u_{k-1}^{(n)}), & k = 1, 2, \dots, 2m-1, \\ v_k^{(n)} = u_k^{(n)} * (1 + \delta^{(n)} * u_{k+1}^{(n)}), & k = 1, 2, \dots, 2m-1, \\ w_{2k-1}^{(n+1)} = v_{2k-2}^{(n)} + v_{2k-1}^{(n)} - w_{2k-2}^{(n+1)} - \tau^{(n)}, & k = 1, 2, \dots, m, \\ w_{2k}^{(n+1)} = v_{2k-1}^{(n)} * (v_{2k}^{(n)} / w_{2k-1}^{(n+1)}), & k = 1, 2, \dots, m-1, \end{cases} \quad (2.6)$$

where $u_0^{(n)} = 0$, $u_{2m}^{(n)} = 0$, $w_0^{(n+1)} = 0$, $v_k^{(n)}$ denote new auxiliary variables, and $\tau^{(n)}$ denotes the shift such that $0 \leq \tau^{(n)} < (\sigma_m(B^{(n)}))^2$. However, it has not shown that the 3rd and 4th recursion formulas in (2.6) is numerically stable. Hence, we replace them with the numerical stable recursion formulas based on differential form of the stationary qd transform (dstqds) shown in [8]. Herewith, since we have another new transformation, in order to distinguish it from the original mdLVs transformation, we call it the m2dLVs transformation. Specifically, the recursion formula of the m2dLVs transformation is given as

$$\begin{cases} u_k^{(n)} = w_k^{(n)} / (1 + \delta^{(n)} * u_{k-1}^{(n)}), & k = 1, 2, \dots, 2m-1, \\ v_k^{(n)} = u_k^{(n)} * (1 + \delta^{(n)} * u_{k+1}^{(n)}), & k = 1, 2, \dots, 2m-1, \\ d_1^{(n)} = -\tau^{(n)}, \\ w_{2k-1}^{(n+1)} = d_k^{(n)} + v_{2k-1}^{(n)}, & k = 1, 2, \dots, m, \\ w_{2k}^{(n+1)} = v_{2k-1}^{(n)} * (v_{2k}^{(n)} / w_{2k-1}^{(n+1)}), & k = 1, 2, \dots, m-1, \\ d_{k+1}^{(n)} = d_k^{(n)} * (v_{2k}^{(n)} / w_{2k-1}^{(n+1)}) - \tau^{(n)}, & k = 1, 2, \dots, m-1, \end{cases}$$

where $d_k^{(n)}$ denote new auxiliary variables. It is noted that $0 < w_k^{(n+1)} < M_0$ and $0 < u_k \leq v_k < M_1$, for any n , if $0 < w_k^{(0)} < M_0$ and $0 \leq \sum_{n=0}^{\infty} \tau^{(n)} < (\sigma_m(B))$, and it is shown similarly to Lemma 4.1 in [19] because the m2dLVs transformation is algebraically equivalent to the original mdLVs transformation. The m2dLVs transformation generates the matrix transformation

$$\begin{aligned} Y_v^{(n)} &= R^{(n)} Y^{(n)} (R^{(n)})^{-1}, \quad Y^{(n+1)} = Y_v^{(n)} - \tau^{(n)} I, \\ Y_v^{(n)} &:= G_v^{(n)} (B_v^{(n)})^\top B_v^{(n)} (G_v^{(n)})^{-1}, \\ B_v^{(n)} &:= \begin{pmatrix} \sqrt{v_1^{(n)}} & \sqrt{v_2^{(n)}} & & \\ & \sqrt{v_3^{(n)}} & \ddots & \\ & & \ddots & \sqrt{v_{2m-2}^{(n)}} \\ & & & \sqrt{v_{2m-1}^{(n)}} \end{pmatrix}, \\ G_v^{(n)} &= \text{diag} \left(\prod_{j=1}^{m-1} \sqrt{v_{2j-1}^{(n)} v_{2j}^{(n)}}, \prod_{j=2}^{m-1} \sqrt{v_{2j-1}^{(n)} v_{2j}^{(n)}}, \dots, \sqrt{v_{2m-3}^{(n)} v_{2m-2}^{(n)}}, 1 \right). \end{aligned}$$

If $\tau^{(n)} < (\sigma_m(B^{(n)}))^2$, then $w_{2k-1}^{(n)} \rightarrow (\sigma_k(B))^2 - \sum_{N=0}^n \tau^{(N)}$ and $w_{2k}^{(n)} \rightarrow 0$ as $n \rightarrow \infty$. Though it is proved in [19] for the original mdLVs transformation, it also holds in the m2dLVs transformation since it is algebraically equivalent to the original mdLVs transformation.

2.2 Mixed forward-backward error analysis

In this section, from the viewpoint of the mixed forward-backward error analysis, we evaluate the perturbations on singular values after the m2dLVs transformation in floating point arithmetic.

It is hardly expected that the operations $\text{op} = +, -, *, /$ are carried out without numerical errors in floating point arithmetic. In order to evaluate numerical errors in floating point arithmetic, we use the standard model of floating point arithmetic [14],

$$\text{fl}(x \text{ op } y) := (x \text{ op } y)(1 + \epsilon) = (x \text{ op } y)(1 + \epsilon')^{-1}, \quad |\epsilon|, |\epsilon'| \leq \varepsilon, \quad (2.7)$$

where x and y are floating point numbers, $\text{fl}(x \text{ op } y)$ is the output of $x \text{ op } y$ in floating point arithmetic, and ε is the machine precision.

Here, let $U^{(n)} := \{u_1^{(n)}, u_2^{(n)}, \dots, u_{2m-1}^{(n)}\}$, $\hat{V}^{(n)} := \{\hat{v}_1^{(n)}, \hat{v}_2^{(n)}, \dots, \hat{v}_{2m-1}^{(n)}\}$ and $W^{(n)} := \{w_1^{(n)}, w_2^{(n)}, \dots, w_{2m-1}^{(n)}\}$ be the sets of variables appearing in the m2dLVs transformation in floating point arithmetic. Let $\hat{U}^{(n)} := \{\hat{u}_1^{(n)}, \hat{u}_2^{(n)}, \dots, \hat{u}_{2m-1}^{(n)}\}$ with $\hat{u}_k^{(n)} := u_k^{(n)} + \Delta u_k^{(n)}$ be the set that the perturbations are added to $U^{(n)}$. Let us assume that $\tilde{W}^{(n)} := \{\tilde{w}_1^{(n)}, \tilde{w}_2^{(n)}, \dots, \tilde{w}_{2m-1}^{(n)}\}$ is the set where $\hat{u}_k^{(n)} = \tilde{w}_k^{(n)} / (1 + \delta^{(n)} \hat{u}_{k-1}^{(n)})$ for $k = 1, 2, \dots, 2m-1$. Then it is obvious that $\tilde{W}^{(n)}$ is a perturbed set from $W^{(n)}$. Moreover, let us introduce a new set $\check{V}^{(n)} := \{\check{v}_1^{(n)}, \check{v}_2^{(n)}, \dots, \check{v}_{2m-1}^{(n)}\}$ where $\check{v}_k^{(n)} := \hat{u}_k^{(n)}(1 + \delta^{(n)} \hat{u}_{k+1}^{(n)})$ for $k = 1, 2, \dots, 2m-1$. Then $\check{V}^{(n)}$ can be regarded as be computed from $\tilde{W}^{(n)}$ in exact arithmetic, and be rounded into $\hat{V}^{(n)} := \{\hat{v}_1^{(n)}, \hat{v}_2^{(n)}, \dots, \hat{v}_{2m-1}^{(n)}\}$ in floating point arithmetic. So, it is realized that the transformation from $W^{(n)}$ to $\hat{V}^{(n)}$ is the composition of three mappings, a mapping from $W^{(n)}$ to $\tilde{W}^{(n)}$, a mapping from $\tilde{W}^{(n)}$ to $\check{V}^{(n)}$ and a mapping from $\check{V}^{(n)}$ to $\hat{V}^{(n)}$, as shown in Fig. 2.1. Similarly, the transformation from $\hat{V}^{(n)}$ to $\hat{W}^{(n+1)}$ is representable as the composition of three mappings associated with the perturbed sets $\check{V}^{(n)} := \{\check{v}_1^{(n)}, \check{v}_2^{(n)}, \dots, \check{v}_{2m-1}^{(n)}\}$ and $\check{W}^{(n+1)} := \{\check{w}_1^{(n+1)}, \check{w}_2^{(n+1)}, \dots, \check{w}_{2m-1}^{(n+1)}\}$.

Without loss of generality, let us assume that the entries of $W^{(n)}$ and $\delta^{(n)}$ are the floating point numbers. Then $\tilde{w}_k^{(n)}$, $\hat{v}_k^{(n)}$ and $\check{v}_k^{(n)}$, $\hat{w}_k^{(n+1)}$ are expressed in terms of $w_k^{(n)}$, $\check{v}_k^{(n)}$ and $\hat{v}_k^{(n)}$, $\check{w}_k^{(n+1)}$, respectively, in the following theorem,

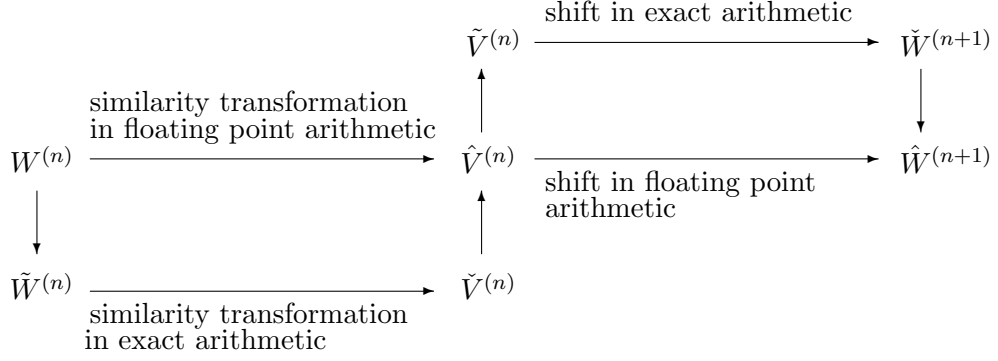


Figure 2.1: The m2dLVs transformation in floating point arithmetic.

which is the main result for showing numerical stability of the m2dLVs transformation.

Theorem 2.2.1. *The entries of $W^{(n)}$, $\tilde{W}$, $\check{V}^{(n)}$ and $\hat{V}^{(n)}$ satisfy*

$$\tilde{w}_k^{(n)} = w_k^{(n)} \prod_{j=1}^3 (1 + \xi_{j,k-1}), \quad k = 1, 2, \dots, 2m-1, \quad (2.8)$$

$$\hat{v}_k^{(n)} = \check{v}_k^{(n)} \prod_{j=3}^5 (1 + \xi_{j,k+1})^{-1}, \quad k = 1, 2, \dots, 2m-1, \quad (2.9)$$

where all the ξ 's are equal to or smaller than the machine epsilon ε . Moreover, the entries of $\hat{V}^{(n)}$, $\tilde{V}^{(n)}$, $\check{W}^{(n+1)}$ and $\hat{W}^{(n+1)}$ satisfy

$$\begin{cases} \tilde{v}_{2k-1}^{(n)} = \hat{v}_{2k-1}^{(n)} (1 + \eta_{1,k}), & k = 1, 2, \dots, m, \\ \tilde{v}_{2k}^{(n)} = \hat{v}_{2k}^{(n)} (1 + \eta_{2,k})(1 + \eta_{3,k})(1 + \eta_{4,k}), & k = 1, 2, \dots, m-1, \end{cases} \quad (2.10)$$

$$\begin{cases} \hat{w}_{2k-1}^{(n+1)} = \check{w}_{2k-1}^{(n+1)} [(1 + \eta_{1,k})(1 + \eta_{2,k})]^{-1}, & k = 1, 2, \dots, m, \\ \hat{w}_{2k}^{(n+1)} = \check{w}_{2k}^{(n+1)} [(1 + \eta_{4,k})(1 + \eta_{5,k})]^{-1}, & k = 1, 2, \dots, m-1, \end{cases} \quad (2.11)$$

where all the η 's are equal to or smaller than ε .

Proof. The equality $\hat{u}_k^{(n)} = \text{fl}(w_k^{(n)} / \text{fl}(1 + \text{fl}(\delta^{(n)} * \hat{u}_{k-1}^{(n)})))$ leads to

$$\hat{u}_k^{(n)} = \frac{w_k^{(n)} (1 + \xi_{1,k-1})(1 + \xi_{2,k-1})}{1 + \delta^{(n)} \hat{u}_{k-1}^{(n)} (1 + \xi'_{3,k-1})^{-1}}, \quad (2.12)$$

where $|\xi_{1,k-1}|, |\xi_{2,k-1}|, |\xi'_{3,k-1}| \leq \varepsilon$. Since there exists $\xi_{3,k-1}$ such that $|\xi_{3,k-1}| \leq |\xi'_{3,k-1}|$ and $1 + \delta^{(n)} \hat{u}_{k-1}^{(n)} (1 + \xi'_{3,k-1})^{-1} = (1 + \delta^{(n)} \hat{u}_{k-1}^{(n)}) (1 + \xi_{3,k-1})^{-1}$, we may

rewrite (2.12) as

$$\hat{u}_k^{(n)} = \frac{w_k^{(n)}}{1 + \delta^{(n)} \hat{u}_{k-1}^{(n)}} \prod_{j=1}^3 (1 + \xi_{j,k-1}). \quad (2.13)$$

Let us recall that $\hat{u}_k^{(n)} = \tilde{w}_k^{(n)} / (1 + \delta^{(n)} \hat{u}_{k-1}^{(n)})$. Then, by combining it with (2.13), we get (2.8). Similarly, from $\hat{v}_k^{(n)} = \text{fl}(\hat{u}_k^{(n)} * \text{fl}(1 + \text{fl}(\delta^{(n)} * \hat{u}_{k+1}^{(n)})))$ and $\tilde{v}_k^{(n)} = \hat{u}_k^{(n)} (1 + \delta^{(n)} \hat{u}_{k+1}^{(n)})$, we derive (2.9).

The variables $\hat{w}_{2k-1}^{(n+1)}$, $\hat{w}_{2k}^{(n+1)}$ and $\hat{d}_{k+1}^{(n)}$ are representable as

$$\begin{cases} \hat{w}_{2k-1}^{(n+1)} = (\hat{d}_k^{(n)} + \hat{v}_{2k-1}^{(n)}) (1 + \eta_{2,k})^{-1}, \\ \hat{w}_{2k}^{(n+1)} = \frac{\hat{v}_{2k-1}^{(n)} \hat{v}_{2k}^{(n)}}{\hat{w}_{2k-1}^{(n+1)}} (1 + \eta_{3,k}) (1 + \eta_{5,k})^{-1}, \end{cases} \quad (2.14)$$

$$\hat{d}_{k+1}^{(n)} = \left[\frac{\hat{d}_k^{(n)} \hat{v}_{2k}^{(n)}}{\hat{w}_{2k-1}^{(n+1)}} (1 + \eta_{3,k}) (1 + \eta_{4,k}) - \tau^{(n)} \right] (1 + \eta_{1,k+1})^{-1}, \quad (2.15)$$

respectively, where all the η 's are equal to or smaller than ε . It is noted here that the entries of $\tilde{W}^{(n)}$ and $\tilde{V}^{(n)}$ satisfy

$$\begin{cases} \tilde{w}_{2k-1}^{(n+1)} = \tilde{d}_k^{(n)} + \tilde{v}_{2k-1}^{(n)}, \\ \tilde{w}_{2k}^{(n+1)} = \tilde{v}_{2k-1}^{(n)} \frac{\tilde{v}_{2k}^{(n)}}{\tilde{w}_{2k-1}^{(n+1)}}. \end{cases} \quad (2.16)$$

Let us assume that (2.10) and $\tilde{d}_k^{(n)} = \hat{d}_k^{(n)} (1 + \eta_{1,k})$. Then, by combining it with (2.14)–(2.16), we have (2.11). $\square$

From Theorem 2.2.1, we see that $\tilde{W}^{(n)}$, $\hat{V}^{(n)}$, $\tilde{V}^{(n)}$ and $\hat{W}^{(n+1)}$ have small relative differences from $W^{(n)}$, $\check{V}^{(n)}$, $\hat{V}^{(n)}$ and $\check{W}^{(n+1)}$, respectively. By taking account that it is a mixed forward-backward error result, Theorem 2.2.1 shows that the m2dLVs transformation is numerically stable.

Let $\tilde{B}^{(n)}$, $\check{B}_v^{(n)}$, $\hat{B}_v^{(n)}$, $\tilde{B}_v^{(n)}$, $\check{B}^{(n+1)}$ and $\hat{B}^{(n+1)}$ be the upper bidiagonal matrices associated with $\tilde{W}^{(n)}$, $\check{V}^{(n)}$, $\hat{V}^{(n)}$, $\tilde{V}^{(n)}$, $\check{W}^{(n+1)}$ and $\hat{W}^{(n+1)}$, respectively in the same way as $B^{(n)}$ associated with $W^{(n)}$. Corollary 2 in Demmel-Kahan [7] helps us to estimate the relative differences of singular values of $B^{(n)}$, $\check{B}_v^{(n)}$, $\hat{B}_v^{(n)}$, $\tilde{B}_v^{(n)}$, $\check{B}^{(n+1)}$ and $\hat{B}^{(n+1)}$.

Theorem 2.2.2 (Demmel-Kahan [7]). *Let $\sigma_1(B^{(n)}) \geq \dots \geq \sigma_m(B^{(n)})$ be the singular values of $B^{(n)}$. If $\tilde{B}_{ii}^{(n)} = \alpha_{2i-1} B_{ii}^{(n)}$, $\tilde{B}_{i,i+1}^{(n)} = \alpha_{2i} B_{i,i+1}^{(n)}$, $\alpha_j \neq 0$, then*

$$\frac{1}{\bar{\alpha}} \leq \frac{\sigma_k(\tilde{B}^{(n)})}{\sigma_k(B^{(n)})} \leq \bar{\alpha},$$

where $\bar{\alpha} = \prod_{j=1}^{2m-1} \max(|\alpha_j|, |\alpha_j^{-1}|)$.

By combining Theorem 2.2.1 with Theorem 2.2.2, we get a theorem for the singular values of $B^{(n)}$, $\check{B}_v^{(n)}$, $\hat{B}_v^{(n)}$, $\tilde{B}_v^{(n)}$, $\check{B}^{(n+1)}$ and $\hat{B}^{(n+1)}$.

Theorem 2.2.3. *The singular values of $B^{(n)}$, $\tilde{B}^{(n)}$, $\check{B}_v^{(n)}$, $\hat{B}_v^{(n)}$, $\tilde{B}_v^{(n)}$, $\check{B}^{(n+1)}$ and $\hat{B}^{(n+1)}$ satisfy*

$$\sigma_i(\tilde{B}^{(n)}) = \sigma_i(B^{(n)}) \exp[3(m-1)\varepsilon_i], \quad i = 1, 2, \dots, m, \quad (2.17)$$

$$\sigma_i(\check{B}_v^{(n)}) = \sigma_i(\tilde{B}^{(n)}), \quad i = 1, 2, \dots, m, \quad (2.18)$$

$$\sigma_i(\hat{B}_v^{(n)}) = \sigma_i(\check{B}_v^{(n)}) \exp[3(m-1)\check{\varepsilon}_i], \quad i = 1, 2, \dots, m, \quad (2.19)$$

$$\sigma_i(\tilde{B}_v^{(n)}) = \sigma_i(\hat{B}_v^{(n)}) \exp[2(m-1)\check{\varepsilon}'_i], \quad i = 1, 2, \dots, m, \quad (2.20)$$

$$\sigma_i(\check{B}^{(n+1)}) = \sigma_i(\tilde{B}_v^{(n)}) - \tau^{(n)}, \quad i = 1, 2, \dots, m, \quad (2.21)$$

$$\sigma_i(\hat{B}^{(n+1)}) = \sigma_i(\check{B}^{(n+1)}) \exp[(2m-1)\check{\varepsilon}'_i], \quad i = 1, 2, \dots, m, \quad (2.22)$$

where $|\varepsilon_i|, |\check{\varepsilon}_i|, |\check{\varepsilon}'_i|, |\check{\varepsilon}_i| \leq \varepsilon$.

Proof. From the relationships of $\tilde{W}^{(n)}$, $\check{V}^{(n)}$, $\tilde{V}^{(n)}$ and $\check{W}^{(n)}$, we can easily get (2.18) and (2.21).

It is obvious from Theorem 2.2.1 that

$$\sqrt{\tilde{w}_k^{(n)}} = \sqrt{w_k^{(n)}} \prod_{j=1}^3 \sqrt{1 + \xi_{j,k-1}}.$$

It is noted here that $\max(1 + \xi_{j,k-1}, (1 + \xi_{j,k-1})^{-1}) \leq 1 + \varepsilon$. With the help of Theorem 2.2.2, we may estimate the relative difference in any singular value between $B^{(n)}$ and $\tilde{B}^{(n)}$ by using the product of all the relative differences of $\sqrt{\tilde{w}_k^{(n)}}$ from $\sqrt{w_k^{(n)}}$ for $k = 1, 2, \dots, 2m-1$, namely,

$$\begin{aligned} \frac{\sigma_i(\tilde{B}^{(n)})}{\sigma_i(B^{(n)})} &\leq \prod_{k=1}^{2m-1} \max \left(\prod_{j=1}^3 \sqrt{1 + \xi_{j,k-1}}, \prod_{j=1}^3 \sqrt{(1 + \xi_{j,k-1})^{-1}} \right) \\ &\leq \prod_{k=1}^{2m-1} \prod_{j=1}^3 \max \left(\sqrt{1 + \xi_{j,k-1}}, \sqrt{(1 + \xi_{j,k-1})^{-1}} \right) \\ &\leq (1 + \varepsilon)^{3(m-1)} \\ &\leq \exp[3(m-1)\varepsilon], \end{aligned} \quad (2.23)$$

and

$$\frac{\sigma_i(\tilde{B}^{(n)})}{\sigma_i(B^{(n)})} \geq \exp[-3(m-1)\varepsilon]. \quad (2.24)$$

By introducing ε_i such that $|\varepsilon_i| \leq \varepsilon$, we can rewrite (2.23) and (2.24) as (2.17).

Similarly, we derive (2.19), (2.20) and (2.22). $\square$

Theorem 2.2.3 suggests that the m2dLVs transformation in floating point arithmetic gives rise to the sufficiently small perturbations on singular values.

2.3 Forward error analysis

In this section, we investigate the forward stability of the m2dLVs transformation in floating point arithmetic. We also give an upper bound of the forward errors which occur in the m2dLVs transformation.

Let us begin our analysis with introducing a lemma for the forward errors which occurs in the mapping from $W^{(n)}$ to $\hat{V}^{(n)}$.

Lemma 2.3.1. *The auxiliary variables $\hat{u}_k^{(n)}$ and $\hat{v}_k^{(n)}$ in floating point arithmetic are representable in terms of $u_k^{(n)}$ and $v_k^{(n)}$ given in exact arithmetic as*

$$\hat{u}_k^{(n)} = u_k^{(n)}(1 + \mu_k), \quad k = 1, 2, \dots, 2m-1, \quad (2.25)$$

$$\hat{v}_k^{(n)} = v_k^{(n)}(1 + \nu_k), \quad k = 1, 2, \dots, 2m-1, \quad (2.26)$$

respectively, where

$$|\mu_k| \leq 3(k-1)\varepsilon, \quad k = 1, 2, \dots, 2m-1, \quad (2.27)$$

$$\begin{cases} |\nu_k| \leq 6k\varepsilon + O(\varepsilon^2), & k = 1, 2, \dots, 2m-2, \\ |\nu_{2m-1}| \leq 6(m-1)\varepsilon + O(\varepsilon^2). \end{cases} \quad (2.28)$$

Proof. For fixed k , it holds that

$$\hat{u}_k^{(n)} = \frac{w_k^{(n)}}{1 + \delta^{(n)}\hat{u}_{k-1}^{(n)}} \prod_{j=1}^3 (1 + \epsilon_{j,k}), \quad (2.29)$$

where $|\epsilon_{j,k}| \leq \varepsilon$. From $\hat{u}_0^{(n)} = 0$ and $w_1^{(n)} = u_1^{(n)}$, it is obvious that $\hat{u}_1^{(0)} = u_1^{(0)}$. Let us assume that $\hat{u}_{k-1}^{(n)} = u_{k-1}^{(n)} \prod_{j=2}^{k-1} \prod_{i=1}^3 (1 + \epsilon'_{i,j})$ where $|\epsilon'_{i,j}| \leq \varepsilon$. Then there exists $\epsilon_{i,j}$ such that $|\epsilon_{i,j}| \leq \varepsilon$ and

$$\frac{1}{1 + \delta^{(n)}\hat{u}_{k-1}^{(n)}} = \frac{1}{1 + \delta^{(n)}u_{k-1}^{(n)}} \prod_{j=2}^{k-1} \prod_{i=1}^3 (1 + \epsilon_{i,j}).$$

So, by taking account of it in (2.29), we get

$$\hat{u}_k^{(n)} = u_k^{(n)} \prod_{j=2}^k \prod_{i=1}^3 (1 + \epsilon_{i,j}). \quad (2.30)$$

By induction for k , from (2.30) we derive (2.27).

Moreover, by combining (2.30) with

$$\hat{v}_k^{(n)} = \hat{u}_k^{(n)}(1 + \delta^{(n)}\hat{u}_{k+1}^{(n)}) \prod_{j=4}^6 (1 + \epsilon_{j,k}), \quad (2.31)$$

where $|\epsilon_{j,k}| \leq \varepsilon$, we have (2.28). $\square$

Then, we estimate the forward errors which occurs in the mapping from $W^{(n)}$ to $\tilde{V}^{(n)}$. From Theorem 2.2.1 and Lemma 2.3.1, it follows that

$$\begin{cases} \tilde{v}_{2k-1}^{(n)} = v_{2k-1}^{(n)}(1 + \nu_{2k-1})(1 + \eta_{1,k}), & k = 1, 2, \dots, m, \\ \tilde{v}_{2k}^{(n)} = v_{2k}^{(n)}(1 + \nu_{2k}) \prod_{j=1}^3 (1 + \eta_{j+1,k}), & k = 1, 2, \dots, m-1, \end{cases}$$

where $|\nu_j| \leq 12(m-1)\varepsilon + O(\varepsilon^2)$ for $j = 1, 2, \dots, 2m-1$. This leads to an upper bound of the forward errors which occurs in the mapping from $W^{(n)}$ to $\tilde{V}^{(n)}$ as follows.

$$\max_k \left\{ \left| \frac{\tilde{v}_k^{(n)} - v_k^{(n)}}{v_k^{(n)}} \right| \right\} \leq (12m-9)\varepsilon + O(\varepsilon^2). \quad (2.32)$$

In general, the forward errors in a transformation can be bounded by using the product of the backward errors and the condition number, which is shown in [14]. Hence, by interpreting the forward errors with respect to $\tilde{V}^{(n)}$ as the backward errors with respect to $\check{W}^{(n+1)}$ and defining the relative componentwise condition number as the amount which means how much the backward errors impact the forward errors with respect to $\check{W}^{(n+1)}$, we estimate the forward errors with respect to $\hat{W}^{(n+1)}$, quantitatively. Bueno and Dopico has already succeeded in estimating the forward errors in the Darboux transformation and the Christoffel transformation with a relative componentwise condition number. Thus, we also introduce the relative componentwise condition number as follows.

Definition 2.3.1 (cf. Bueno-Marcellan-Dopico [1, 3]). *The relative componentwise condition number in the m2dLVs transformation with respect to perturbations associated with the backward errors is defined by*

$$\kappa(V^{(n)}) := \limsup_{\varepsilon \rightarrow 0} \left\{ \max_k \left\{ \left| \frac{\Delta w_k^{(n+1)}}{\varepsilon w_k^{(n+1)}} \right| \right\} : |\Delta v_k^{(n)}| \leq \varepsilon |v_k^{(n)}| \right\}, \quad (2.33)$$

where

$$\max_k \left\{ \left| \frac{\Delta w_k^{(n+1)}}{\varepsilon w_k^{(n+1)}} \right| \right\} := \max \left\{ \left| \frac{\Delta w_1^{(n+1)}}{\varepsilon w_1^{(n+1)}} \right|, \left| \frac{\Delta w_2^{(n+1)}}{\varepsilon w_2^{(n+1)}} \right|, \dots, \left| \frac{\Delta w_{2m-1}^{(n+1)}}{\varepsilon w_{2m-1}^{(n+1)}} \right| \right\}.$$

The entries of $W^{(n+1)}$ are the rational functions with respect to the entries of $V^{(n)}$, and are differentiable with respect to those of $V^{(n)}$. Then, according

to [5], we may rewrite $\kappa(V^{(n)})$ in terms of the partial derivatives by the entries of $V^{(n)}$, namely,

$$\kappa(V^{(n)}) := \max\{\kappa_1^{(n)}, \kappa_2^{(n)}, \dots, \kappa_{2m-1}^{(n)}\}, \quad (2.34)$$

where

$$\kappa_k^{(n)} := \sum_{j=1}^k \left| \frac{v_j^{(n)}}{w_k^{(n+1)}} \frac{\partial w_k^{(n+1)}}{\partial v_j^{(n)}} \right|, \quad k = 1, 2, \dots, 2m-1. \quad (2.35)$$

Here we prepare a lemma for $(v_j^{(n)}/w_k^{(n+1)})(\partial w_k^{(n+1)}/\partial v_j^{(n)})$ in (2.35).

Lemma 2.3.2. *The partial derivative $\partial w_k^{(n+1)}/\partial v_j^{(n)}$ satisfies*

$$\frac{v_{2j-1}^{(n)}}{w_{2k-1}^{(n+1)}} \frac{\partial w_{2k-1}^{(n+1)}}{\partial v_{2j-1}^{(n)}} = \begin{cases} -\frac{d_j^{(n)}}{w_{2k-1}^{(n+1)}} \prod_{i=j}^{k-1} \frac{w_{2i}^{(n+1)}}{w_{2i-1}^{(n+1)}}, & j = 1, 2, \dots, k-2, \\ -\frac{d_{k-1}^{(n)}}{w_{2k-1}^{(n+1)}} \frac{w_{2k-2}^{(n+1)}}{w_{2k-3}^{(n+1)}}, & j = k-1, \\ \frac{v_{2k-1}^{(n)}}{w_{2k-1}^{(n+1)}}, & j = k, \end{cases} \quad (2.36)$$

$$\frac{v_{2j}^{(n)}}{w_{2k-1}^{(n+1)}} \frac{\partial w_{2k-1}^{(n+1)}}{\partial v_{2j}^{(n)}} = \begin{cases} \frac{d_j^{(n)}}{w_{2k-1}^{(n+1)}} \frac{v_{2j}^{(n+1)}}{w_{2j-1}^{(n+1)}} \prod_{i=j+1}^{k-1} \frac{w_{2i}^{(n+1)}}{w_{2i-1}^{(n+1)}}, & j = 1, 2, \dots, k-2, \\ \frac{d_{k-1}^{(n)}}{w_{2k-1}^{(n+1)}} \frac{v_{2k-2}^{(n+1)}}{w_{2k-3}^{(n+1)}}, & j = k-1, \\ 0, & j = k, \end{cases} \quad (2.37)$$

$$\frac{v_{2j-1}^{(n)}}{w_{2k}^{(n+1)}} \frac{\partial w_{2k}^{(n+1)}}{\partial v_{2j-1}^{(n)}} = \begin{cases} \frac{d_j^{(n)}}{w_{2k-1}^{(n+1)}} \prod_{i=j}^{k-1} \frac{w_{2i}^{(n+1)}}{w_{2i-1}^{(n+1)}}, & j = 1, 2, \dots, k-2, \\ \frac{d_{k-1}^{(n)}}{w_{2k-1}^{(n+1)}} \frac{w_{2k-2}^{(n+1)}}{w_{2k-3}^{(n+1)}}, & j = k-1, \\ \frac{d_k^{(n)}}{w_{2k-1}^{(n+1)}}, & j = k, \end{cases} \quad (2.38)$$

$$\frac{v_{2j}^{(n)}}{w_{2k}^{(n+1)}} \frac{\partial w_{2k}^{(n+1)}}{\partial v_{2j}^{(n)}} = \begin{cases} -\frac{d_j^{(n)}}{w_{2k-1}^{(n+1)}} \frac{v_{2j}^{(n)}}{w_{2j-1}^{(n+1)}} \prod_{i=j+1}^{k-1} \frac{w_{2i}^{(n+1)}}{w_{2i-1}^{(n+1)}}, & j = 1, 2, \dots, k-2, \\ -\frac{d_{k-1}^{(n)}}{w_{2k-1}^{(n+1)}} \frac{v_{2k-2}^{(n)}}{w_{2k-3}^{(n+1)}}, & j = k-1, \\ 1, & j = k. \end{cases} \quad (2.39)$$

Proof. From $d_{k+1}^{(n)} = d_k^{(n)} * (v_{2k}^{(n)} / w_{2k-1}^{(n+1)}) - \tau^{(n)}$, we derive

$$\frac{\partial d_k^{(n)}}{\partial v_{2j-1}^{(n)}} = \begin{cases} -\frac{d_j^{(n)}}{v_{2j-1}^{(n+1)}} \prod_{i=j}^{k-1} \frac{w_{2i}^{(n+1)}}{w_{2i-1}^{(n+1)}}, & j = 1, 2, \dots, k-1, \\ -\frac{d_{k-1}^{(n)}}{w_{2k-3}^{(n+1)}} \frac{v_{2k-2}^{(n)}}{w_{2k-3}^{(n+1)}}, & j = k-1, \end{cases} \quad (2.40)$$

$$\frac{\partial d_k^{(n)}}{\partial v_{2j}^{(n)}} = \begin{cases} \frac{d_j^{(n)}}{w_{2j-1}^{(n+1)}} \prod_{i=j+1}^{k-1} \frac{w_{2i}^{(n+1)}}{w_{2i-1}^{(n+1)}}, & j = 1, 2, \dots, k-1, \\ \frac{d_{k-1}^{(n)}}{w_{2k-3}^{(n+1)}}, & j = k-1. \end{cases} \quad (2.41)$$

By combining (2.40) and (2.41) with

$$\begin{aligned} \frac{\partial w_{2k-1}^{(n+1)}}{\partial v_{2j-1}^{(n)}} &= \begin{cases} \frac{\partial d_k^{(n)}}{\partial v_{2j-1}^{(n)}}, & j = 1, 2, \dots, k-1, \\ 1, & j = k, \end{cases} \\ \frac{\partial w_{2k-1}^{(n+1)}}{\partial v_{2j}^{(n)}} &= \begin{cases} \frac{\partial d_k^{(n)}}{\partial v_{2j}^{(n)}}, & j = 1, 2, \dots, k-1, \\ 0, & j = k, \end{cases} \\ \frac{\partial w_{2k}^{(n+1)}}{\partial v_{2j-1}^{(n)}} &= \begin{cases} -\frac{w_{2k}^{(n+1)}}{w_{2k-1}^{(n+1)}} \frac{\partial w_{2k-1}^{(n+1)}}{\partial v_{2j-1}^{(n)}}, & j = 1, 2, \dots, k-1, \\ \frac{d_k^{(n)}}{w_{2k-1}^{(n+1)}} \frac{v_{2k}^{(n)}}{w_{2k-1}^{(n+1)}}, & j = k, \end{cases} \\ \frac{\partial w_{2k}^{(n+1)}}{\partial v_{2j}^{(n)}} &= \begin{cases} -\frac{w_{2k}^{(n+1)}}{w_{2k-1}^{(n+1)}} \frac{\partial w_{2k-1}^{(n+1)}}{\partial v_{2j}^{(n)}}, & j = 1, 2, \dots, k-1, \\ \frac{v_{2k-1}^{(n)}}{w_{2k-1}^{(n+1)}}, & j = k, \end{cases} \end{aligned}$$

we have (2.36)–(2.39). $\square$

Consequently, from (2.32)–(2.35) with Theorem 2.2.1 and Lemma 2.3.2, we have a theorem for an upper bound of the forward errors in the m2dLVs transformation.

Theorem 2.3.1. *For $k = 1, 2, \dots, 2m - 1$, the relative forward errors with respect to $w_k^{(n+1)}$ are bounded as*

$$\max_k \left\{ \left| \frac{w_k^{(n+1)} - \hat{w}_k^{(n+1)}}{w_k^{(n+1)}} \right| \right\} \leq [(12m - 9)\kappa(V^{(n)}) + 2]\varepsilon + O(\varepsilon^2),$$

where $\kappa(V^{(n)}) = \max\{\kappa_1^{(n)}, \kappa_2^{(n)}, \dots, \kappa_{2m-1}^{(n)}\}$ with

$$\begin{aligned} \kappa_{2k-1}^{(n)} &= \sum_{j=1}^{k-2} \left| \frac{d_j^{(n)}}{w_{2k-1}^{(n+1)}} \frac{v_{2j}^{(n)} + w_{2j}^{(n+1)}}{w_{2j-1}^{(n+1)}} \prod_{i=j+1}^{k-1} \frac{w_{2i}^{(n+1)}}{w_{2i-1}^{(n+1)}} \right| \\ &\quad + \left| \frac{d_{k-1}^{(n)}}{w_{2k-1}^{(n+1)}} \frac{v_{2k-2}^{(n)} + w_{2k-2}^{(n+1)}}{w_{2k-3}^{(n+1)}} \right| + \frac{v_{2k-1}^{(n)}}{w_{2k-1}^{(n+1)}}, \quad k = 1, 2, \dots, m, \\ \kappa_{2k}^{(n)} &= \sum_{j=1}^{k-2} \left| \frac{d_j^{(n)}}{w_{2k-1}^{(n+1)}} \frac{v_{2j}^{(n)} + w_{2j}^{(n+1)}}{w_{2j-1}^{(n+1)}} \prod_{i=j+1}^{k-1} \frac{w_{2i}^{(n+1)}}{w_{2i-1}^{(n+1)}} \right| \\ &\quad + \left| \frac{d_{k-1}^{(n)}}{w_{2k-1}^{(n+1)}} \frac{v_{2k-2}^{(n)} + w_{2k-2}^{(n+1)}}{w_{2k-3}^{(n+1)}} \right| + \left| \frac{d_k^{(n)}}{w_{2k-1}^{(n+1)}} \right| + 1, \quad k = 1, 2, \dots, m - 1. \end{aligned}$$

This theorem shows an upper bound of the *total* forward errors accumulated in the process which computes $\hat{W}^{(n+1)}$ from $W^{(n)}$. Note that Theorem 2.3.1 differs from (2.32) since (2.32) shows an upper bound of the forward errors with respect to $\check{V}^{(n)}$ which is the set of variables appearing in the middle of the m2dLVs transformation in floating point arithmetic. (Recall Figure 2.1.)

Chapter 3

Accurate similarity transformation derived from the discrete Lotka-Volterra system for bidiagonal singular values

As described in Chapter 1, the convergence speed of $u_{2k-1}^{(n)}$ and $u_{2k}^{(n)}$ in the dLV system grow faster as $\delta^{(n)}$ becomes larger [16]. From the point of view that numerical errors accumulate, in this chapter, we confirm how the dLV system behaves as the limit $\delta^{(n)} \rightarrow \infty$.

This chapter is organized as follows. In Section 3.1, we first derive the recursion formula which gives the dLV_∞ similarity transformation from the original dLV transformation [16, 17]. And then, in order to show it is applicable to singular value computation, we prove that the dLV_∞ can generate the singular values of B . Next, in Section 3.2 and Section 3.3, we confirm that the dLV_∞ similarity transformation has numerical stability similarly to Section 2.2 and evaluate the forward errors appearing in the dLV_∞ variables, respectively. The results in these sections guarantee accuracy of the dLV_∞ similarity transformation. Finally, in Section 3.4, we give some numerical examples with the original dLV algorithm, the algorithm based on the dLV_∞ similarity transformation and the dqd algorithm, and numerically demonstrate the relative forward errors in the variables appearing in the dLV_∞ similarity transformation. The reason why we use the dqd algorithm is unraveled in Section 3.1.

3.1 The dLV_∞ similarity transformation

In this section, we first derive a recursion formula of the similarity transformation by taking the limit $\delta^{(n)} \rightarrow \infty$ in the dLV similarity transformation, and next show that it is applicable for computing the singular values of bidiagonal B .

According to [16, 17], the recursion formula of the dLV similarity transformation is written as

$$\begin{cases} u_k^{(n)} = \frac{w_k^{(n)}}{1 + \delta^{(n)} u_{k-1}^{(n)}}, & k = 1, 2, \dots, 2m-1, \\ w_k^{(n+1)} = u_k^{(n)} (1 + \delta^{(n)} u_{k+1}^{(n)}), & k = 1, 2, \dots, 2m-1, \\ u_0^{(n)} := 0, \quad u_{2m}^{(n)} := 0, & n = 0, 1, \dots \end{cases} \quad (3.1)$$

Equation (3.1) is derived from the dLV system (1.2) by introducing the variable $w_k^{(n)} := u_k^{(n)} (1 + \delta^{(n)} u_{k-1}^{(n)})$. Hereinafter, let $\delta^{(n)} = \delta$ for $n = 0, 1, \dots$ since the limit $\delta^{(n)} \rightarrow \infty$ for $n = 0, 1, \dots$ is equivalent to the limit $\delta \rightarrow \infty$ in (3.1). We give a lemma in the process for getting $u_1^{(n)}, u_2^{(n)}, \dots, u_{2m-1}^{(n)}$ from given $w_1^{(n)}, w_2^{(n)}, \dots, w_{2m-1}^{(n)}$ in the 1st equation of (3.1).

Lemma 3.1.1. *Let $w_1^{(n)} > 0, w_2^{(n)} > 0, \dots, w_{2m-1}^{(n)} > 0$. Then it holds that, for $n = 0, 1, \dots$,*

$$d_k^{(n)} := \lim_{\delta \rightarrow \infty} u_{2k-1}^{(n)} = \frac{w_{2k-1}^{(n)}}{1 + s_{k-1}^{(n)}} > 0, \quad k = 1, 2, \dots, m, \quad (3.2)$$

$$\lim_{\delta \rightarrow \infty} u_{2k}^{(n)} = 0, \quad k = 1, 2, \dots, m-1, \quad (3.3)$$

$$s_k^{(n)} := \lim_{\delta \rightarrow \infty} \delta u_{2k}^{(n)} = \frac{w_{2k}^{(n)}}{d_k^{(n)}} > 0, \quad k = 1, 2, \dots, m-1, \quad (3.4)$$

where $s_0^{(n)} := 0$.

Proof. It is obvious that

$$d_1^{(n)} = \lim_{\delta \rightarrow \infty} u_1^{(n)} = \lim_{\delta \rightarrow \infty} w_1^{(n)} = \frac{w_1^{(n)}}{1 + s_0^{(n)}} > 0. \quad (3.5)$$

From (3.5), it follows that

$$\begin{aligned} \lim_{\delta \rightarrow \infty} u_2^{(n)} &= \lim_{\delta \rightarrow \infty} \frac{w_2^{(n)}}{1 + \delta u_1^{(n)}} = \frac{w_2^{(n)}}{1 + d_1^{(n)} \lim_{\delta \rightarrow \infty} \delta} = 0, \\ s_1^{(n)} &= \lim_{\delta \rightarrow \infty} \delta u_2^{(n)} = \lim_{\delta \rightarrow \infty} \frac{\delta w_2^{(n)}}{1 + \delta u_1^{(n)}} = \frac{w_2^{(n)}}{\lim_{\delta \rightarrow \infty} \frac{1}{\delta} + d_1^{(n)}} = \frac{w_2^{(n)}}{d_1^{(n)}} > 0. \end{aligned}$$

Let us assume that $s_{k-1}^{(n)} = \lim_{\delta \rightarrow \infty} \delta u_{2k-2}^{(n)} > 0$. Then we get

$$d_k^{(n)} = \lim_{\delta \rightarrow \infty} u_{2k-1}^{(n)} = \frac{w_{2k-1}^{(n)}}{1 + \lim_{\delta \rightarrow \infty} \delta u_{2k-2}^{(n)}} = \frac{w_{2k-1}^{(n)}}{1 + s_{k-1}^{(n)}} > 0. \quad (3.6)$$

From (3.6), we derive

$$\begin{aligned} \lim_{\delta \rightarrow \infty} u_{2k}^{(n)} &= \frac{w_{2k}^{(n)}}{1 + d_k^{(n)} \lim_{\delta \rightarrow \infty} \delta} = 0, \\ s_k^{(n)} &= \lim_{\delta \rightarrow \infty} \delta u_{2k}^{(n)} = \lim_{\delta \rightarrow \infty} \frac{\delta w_{2k}^{(n)}}{1 + \delta u_{2k-1}^{(n)}} = \frac{w_{2k}^{(n)}}{\lim_{\delta \rightarrow \infty} \frac{1}{\delta} + d_k^{(n)}} = \frac{w_{2k}^{(n)}}{d_k^{(n)}} > 0. \end{aligned}$$

By induction for $k = 1, 2, \dots, m$, we have (3.2) – (3.4). $\square$

Lemma 3.1.1 leads to the following theorem for the dLV similarity transformation in the case where $\delta \rightarrow \infty$.

Theorem 3.1.1. *The dLV similarity transformation with $\delta \rightarrow \infty$ generates the evolution from $w_1^{(n)} > 0, w_2^{(n)} > 0, \dots, w_{2m-1}^{(n)} > 0$ to $w_1^{(n+1)} > 0, w_2^{(n+1)} > 0, \dots, w_{2m-1}^{(n+1)} > 0$, respectively, as follows.*

$$\begin{cases} w_{2k-1}^{(n+1)} = d_k^{(n)}(1 + s_k^{(n)}), & k = 1, 2, \dots, m, \\ w_{2k}^{(n+1)} = s_k^{(n)} d_{k+1}^{(n)}, & k = 1, 2, \dots, m-1, \\ d_k^{(n)} = \frac{w_{2k-1}^{(n)}}{1 + s_{k-1}^{(n)}}, & k = 1, 2, \dots, m, \\ s_k^{(n)} = \frac{w_{2k}^{(n)}}{d_k^{(n)}}, & k = 1, 2, \dots, m-1, \\ s_0^{(n)} := 0, & s_m^{(n)} := 0. \end{cases} \quad (3.7)$$

Proof. From Lemma 3.1.1, it turns out that $d_1^{(n)}, s_1^{(n)}, d_2^{(n)}, s_2^{(n)}, \dots, d_{m-1}^{(n)}, s_{m-1}^{(n)}, d_m^{(n)}$ are recursively given by $d_k^{(n)} = w_{2k-1}^{(n)} / (1 + s_{k-1}^{(n)})$ for $k = 1, 2, \dots, m$ and $s_k^{(n)} = w_{2k}^{(n)} / d_k^{(n)}$ for $k = 1, 2, \dots, m-1$. Let us recall here that $w_{2k-1}^{(n+1)} = u_{2k-1}^{(n)}(1 + \delta u_{2k}^{(n)})$ for $k = 1, 2, \dots, m$ and $w_{2k}^{(n+1)} = u_{2k}^{(n)}(1 + \delta u_{2k+1}^{(n)})$ for $k = 1, 2, \dots, m-1$. By combining it with Lemma 3.1.1, we also get

$$\begin{aligned} w_{2k-1}^{(n+1)} &= \lim_{\delta \rightarrow \infty} [u_{2k-1}^{(n)}(1 + \delta u_{2k}^{(n)})] = \lim_{\delta \rightarrow \infty} u_{2k-1}^{(n)} \lim_{\delta \rightarrow \infty} (1 + \delta u_{2k}^{(n)}) \\ &= d_k^{(n)}(1 + s_k^{(n)}), \quad k = 1, 2, \dots, m, \\ w_{2k}^{(n+1)} &= \lim_{\delta \rightarrow \infty} [u_{2k}^{(n)}(1 + \delta u_{2k+1}^{(n)})] = \lim_{\delta \rightarrow \infty} u_{2k}^{(n)} + \lim_{\delta \rightarrow \infty} (\delta u_{2k}^{(n)}) u_{2k+1}^{(n)} \\ &= s_k^{(n)} d_{k+1}^{(n)}, \quad k = 1, 2, \dots, m-1. \end{aligned}$$

Since $d_1^{(n)} > 0, d_2^{(n)} > 0, \dots, d_m^{(n)} > 0$ and $s_1^{(n)} > 0, s_2^{(n)} > 0, \dots, s_{m-1}^{(n)} > 0$, it is obvious that $w_1^{(n+1)} > 0, w_2^{(n+1)} > 0, \dots, w_{m-1}^{(n+1)} > 0$. $\square$

In order to distinguish the dLV similarity transformation with bounded $\delta > 0$, the similarity transformation given by (3.7) is called the dLV_∞ similarity transformation.

Next, we clarify that the dLV_∞ similarity transformation is regarded as the well-known LR transformations. By taking account of the proof for Theorem 3.1.1, we get a lemma for the dLV system (1.2) with $\delta \rightarrow \infty$.

Lemma 3.1.2. *The dLV system (1.2) with $\delta \rightarrow \infty$ can be rewritten in terms of $d_1^{(n)}, s_1^{(n)}, d_2^{(n)}, s_2^{(n)}, \dots, d_{m-1}^{(n)}, s_{m-1}^{(n)}, d_m^{(n)}$ and $d_1^{(n+1)}, s_1^{(n+1)}, d_2^{(n+1)}, s_2^{(n+1)}, \dots, d_{m-1}^{(n+1)}, s_{m-1}^{(n+1)}, d_m^{(n+1)}$ as*

$$\begin{cases} d_k^{(n+1)}(1 + s_{k-1}^{(n+1)}) = d_k^{(n)}(1 + s_k^{(n)}), & k = 1, 2, \dots, m, \\ d_k^{(n+1)}s_k^{(n+1)} = s_k^{(n)}d_{k+1}^{(n)}, & k = 1, 2, \dots, m-1, \\ s_0^{(n)} := 0, \quad s_m^{(n)} := 0, & n = 0, 1, 2, \dots \end{cases}$$

Theorem 3.1.1 and Lemma 3.1.2 bring an expression the dLV_∞ similarity transformation without using $d_1^{(n)}, d_2^{(n)}, \dots, d_m^{(n)}$ and $s_1^{(n)}, s_2^{(n)}, \dots, s_{m-1}^{(n)}$ as in the following lemma.

Lemma 3.1.3. *The dLV_∞ similarity transformation can be given by the recursion formula*

$$\begin{cases} w_{2k-2}^{(n+1)} + w_{2k-1}^{(n+1)} = w_{2k-1}^{(n)} + w_{2k}^{(n)}, & k = 1, 2, \dots, m, \\ w_{2k-1}^{(n+1)}w_{2k}^{(n+1)} = w_{2k}^{(n)}w_{2k+1}^{(n)}, & k = 1, 2, \dots, m-1, \\ w_0^{(n)} \equiv 0, \quad w_{2m}^{(n)} \equiv 0, & n = 0, 1, 2, \dots \end{cases} \quad (3.8)$$

Proof. From Theorem 3.1.1 and Lemma 3.1.2, we derive

$$\begin{aligned} w_{2k-1}^{(n)} + w_{2k}^{(n)} &= d_k^{(n)}(1 + s_{k-1}^{(n)}) + d_k^{(n)}s_k^{(n)} = s_{k-1}^{(n)}d_k^{(n)} + d_k^{(n)}(1 + s_k^{(n)}) \\ &= d_{k-1}^{(n+1)}s_{k-1}^{(n+1)} + d_k^{(n+1)}(1 + s_{k-1}^{(n+1)}) \\ &= w_{2k-2}^{(n+1)} + w_{2k-1}^{(n+1)}, \quad k = 1, 2, \dots, m, \\ w_{2k}^{(n)}w_{2k+1}^{(n)} &= (d_k^{(n)}s_k^{(n)})[d_{k+1}^{(n)}(1 + s_k^{(n)})] = [d_k^{(n)}(1 + s_k^{(n)})](s_k^{(n)}d_{k+1}^{(n)}) \\ &= [d_k^{(n+1)}(1 + s_{k-1}^{(n+1)})](d_k^{(n+1)}s_k^{(n+1)}) \\ &= w_{2k-1}^{(n+1)}w_{2k}^{(n+1)}, \quad k = 1, 2, \dots, m-1. \end{aligned}$$

$\square$

For given $w_1^{(n)}, w_2^{(n)}, \dots, w_{2m-1}^{(n)}$, it holds that $w_1^{(n+1)}, w_2^{(n+1)}, \dots, w_{2m-1}^{(n+1)}$ computed by (3.8) coincide with those by (3.7). Equation (3.8) also becomes just the recursion formula of the qd algorithm [27, 30] by the replacement $w_{2k-1}^{(n)} = q_k^{(n)}$, $w_{2k}^{(n)} = e_k^{(n)}$, $w_{2k+1}^{(n)} = q_{k+1}^{(n)}$ and $w_{2k-2}^{(n+1)} = q_{k-1}^{(n+1)}$, $w_{2k-1}^{(n+1)} = q_k^{(n+1)}$, $w_{2k}^{(n+1)} = q_k^{(n+1)}$. So, the dLV_∞ similarity transformation is an LR transformation from $Y^{(n)}$ to $Y^{(n+1)}$ as follows.

$$Y^{(n)} = L^{(n)} R^{(n)}, \quad Y^{(n+1)} = R^{(n)} L^{(n)}, \quad (3.9)$$

$$L^{(n)} = \begin{pmatrix} w_1^{(n)} & & & 0 \\ & 1 & & \\ & & w_3^{(n)} & \\ & & \ddots & \ddots \\ & & & 1 & w_{2m-1}^{(n)} \end{pmatrix}, \quad R^{(n)} = \begin{pmatrix} 1 & w_2^{(n)} & & 0 \\ & 1 & \ddots & \\ & & \ddots & w_{2m-2}^{(n)} \\ & & & 1 \end{pmatrix},$$

$$w_{2k-1}^{(n)} = (1 + s_{k-1}^{(n)}) d_k^{(n)}, \quad w_{2k}^{(n)} = d_k^{(n)} s_k^{(n)}.$$

Of course, the dLV_∞ similarity transformation is closely related to the LR transformation based on the recursion formula of the dqd algorithm [11, 27, 30] without cancellation by subtraction, which is a famous algorithm in symmetric tradingonal eigenvalue and bidiagonal singular value problems.

Corollary 3.1.1. *Equation (3.7), which generates the dLV_∞ similarity transformation, is algebraically equivalent to the recursion formula of the dqd algorithm,*

$$\begin{cases} q_k^{(n+1)} = d_k^{(n)} + e_k^{(n)}, & k = 1, 2, \dots, m, \\ e_k^{(n+1)} = e_k^{(n)} \frac{q_{k+1}^{(n)}}{q_k^{(n+1)}}, & k = 1, 2, \dots, m-1, \\ d_1^{(n)} = q_1^{(n)}, \quad d_{k+1}^{(n)} = d_k^{(n)} \frac{q_{k+1}^{(n)}}{q_k^{(n+1)}}, & k = 1, 2, \dots, m-1. \end{cases} \quad (3.10)$$

Proof. From the 3rd and the 4th equations of (3.7), it follows that

$$w_{2k-1}^{(n+1)} = d_k^{(n)} (1 + s_k^{(n)}) = d_k^{(n)} + w_{2k}^{(n)}, \quad k = 1, 2, \dots, m-1. \quad (3.11)$$

From the 1st and the 3rd of (3.7), we derive

$$d_{k+1}^{(n)} = \frac{w_{2k+1}^{(n)}}{1 + s_k^{(n)}} = d_k^{(n)} \frac{w_{2k+1}^{(n)}}{w_{2k-1}^{(n+1)}}, \quad k = 1, 2, \dots, m-1. \quad (3.12)$$

By combining the 2nd and the 4th of (3.7) with (3.12), we also get

$$w_{2k}^{(n+1)} = s_k^{(n)} d_{k+1}^{(n)} = w_{2k}^{(n)} \frac{w_{2k+1}^{(n)}}{w_{2k-1}^{(n+1)}}, \quad k = 1, 2, \dots, m-1. \quad (3.13)$$

Let $w_{2k}^{(n)} = e_k^{(n)}$, $w_{2k+1}^{(n)} = q_{k+1}^{(n)}$ and $w_{2k-1}^{(n+1)} = q_k^{(n+1)}$, $w_{2k}^{(n+1)} = e_k^{(n+1)}$ in (3.11) – (3.13). Then (3.11) – (3.13) are just equal to (3.10). $\square$

Moreover, according to [16], the tridiagonal matrix $Y^{(n)} = L^{(n)}R^{(n)}$ is symmetrized as

$$Y_s^{(n)} = (G^{(n)})^{-1}Y^{(n)}G^{(n)} \quad (3.14)$$

$$= \begin{pmatrix} w_1^{(n)} & \sqrt{w_1^{(n)}w_2^{(n)}} & & & \\ \sqrt{w_1^{(n)}w_2^{(n)}} & w_2^{(n)} + w_3^{(n)} & \ddots & & \\ & \ddots & \ddots & \ddots & \sqrt{w_{2m-3}^{(n)}w_{2m-2}^{(n)}} \\ & & & \sqrt{w_{2m-3}^{(n)}w_{2m-2}^{(n)}} & w_{2m-2}^{(n)} + w_{2m-1}^{(n)} \end{pmatrix},$$

$$G^{(n)} := \text{diag} \left(\prod_{i=1}^{m-1} \sqrt{w_{2i-1}^{(n)}w_{2i}^{(n)}}, \prod_{i=2}^{m-1} \sqrt{w_{2i-1}^{(n)}w_{2i}^{(n)}}, \dots, \sqrt{w_{2m-3}^{(n)}w_{2m-2}^{(n)}}, 1 \right).$$

From (3.9) and (3.14), we see that the dLV_∞ similarity transformation takes the form of similarity transformation

$$Y_s^{(n+1)} = Q^{(n)}Y_s^{(n)}(Q^{(n)})^{-1}, \quad Q^{(n)} := (G^{(n+1)})^{-1}R^{(n)}G^{(n)}. \quad (3.15)$$

It is noted here that $Y_s^{(n)}$ can be decomposed into

$$Y_s^{(n)} = (B^{(n)})^\top B^{(n)}, \quad (3.16)$$

$$B^{(n)} := \begin{pmatrix} \sqrt{w_1^{(n)}} & \sqrt{w_2^{(n)}} & & & \\ & \sqrt{w_3^{(n)}} & \ddots & & \\ & & \ddots & \sqrt{w_{2m-2}^{(n)}} & \\ & & & \sqrt{w_{2m-1}^{(n)}} & \\ 0 & & & & \end{pmatrix}. \quad (3.17)$$

Equation (3.16) implies that the singular values of $B^{(n)}$ are equal to the positive square roots of the eigenvalues of $Y_s^{(n)}$. Thus, by combining it with (3.15), we get a lemma with respect to the transformation from $B^{(n)}$ to $B^{(n+1)}$.

Lemma 3.1.4. *The singular values of $B^{(n)}$ coincide with those of $B^{(n+1)}$ given through the dLV_∞ similarity transformation.*

As is discussed in the proof of Corollary 3.1.1, $w_{2k-1}^{(n)}$, $w_{2k}^{(n)}$ and $d_k^{(n)}$ correspond to the dqd variables, while $s_k^{(n)}$ does not appear in the recursion formula of the dqd algorithm.

Finally, we clarify the convergence of the dLV_∞ variables $w_{2k-1}^{(n)}$, $w_{2k}^{(n)}$, $d_k^{(n)}$ and $s_k^{(n)}$ as $n \rightarrow \infty$.

Theorem 3.1.2. *Let us assume that every bidiagonal entry of $B^{(0)}$ is nonzero. Then it holds that*

$$\lim_{n \rightarrow \infty} w_{2k-1}^{(n)} = c_k, \quad k = 1, 2, \dots, m, \quad (3.18)$$

$$\lim_{n \rightarrow \infty} w_{2k}^{(n)} = 0, \quad k = 1, 2, \dots, m-1, \quad (3.19)$$

$$\lim_{n \rightarrow \infty} d_k^{(n)} = c_k, \quad k = 1, 2, \dots, m, \quad (3.20)$$

$$\lim_{n \rightarrow \infty} s_k^{(n)} = 0, \quad k = 1, 2, \dots, m-1, \quad (3.21)$$

where $c_1, c_2, \dots, c_m$ are some positive constants such that $c_1 > c_2 > \dots > c_m$. More precisely, $\sqrt{c_1}, \sqrt{c_2}, \dots, \sqrt{c_m}$ become the singular values of $B^{(0)}$.

Proof. It is obvious that $w_1^{(0)} > 0, w_2^{(0)} > 0, \dots, w_{2m-1}^{(0)} > 0$. From Theorem 3.1.1, it turns out that $w_1^{(n)} > 0, w_2^{(n)} > 0, \dots, w_{2m-1}^{(n)} > 0$ for $n = 1, 2, \dots$. It is of significance to note that $w_1^{(n)} + w_2^{(n)} + \dots + w_{2m-1}^{(n)} = \text{trace}((B^{(n)})^\top B^{(n)}) = (\sigma_1(B^{(n)}))^2 + (\sigma_2(B^{(n)}))^2 + \dots + (\sigma_m(B^{(n)}))^2$ and $w_1^{(n)} w_3^{(n)} \dots w_{2m-1}^{(n)} = \det((B^{(n)})^\top B^{(n)}) = (\sigma_1(B^{(0)})\sigma_2(B^{(0)}) \dots \sigma_m(B^{(0)}))^2$. From Lemma 3.1.4, it holds that $\sigma_1(B^{(n)}) = \sigma_1(B^{(0)}), \sigma_2(B^{(n)}) = \sigma_2(B^{(0)}), \dots, \sigma_m(B^{(n)}) = \sigma_m(B^{(0)})$. So, there exist some positive constants K_1 and K_2 such that $w_1^{(n)}, w_2^{(n)}, \dots, w_{2m-1}^{(n)} < K_1$ and $w_1^{(n)} w_3^{(n)} \dots w_{2m-1}^{(n)} < K_2$.

Since it is shown in Lemma 3.1.1 that $s_1^{(n)} > 0$, it follows that $w_1^{(n+1)} = d_1^{(n)}(1 + s_1^{(n)}) = w_1^{(n)}(1 + s_1^{(n)}) > w_1^{(n)}$. This implies that the sequence $\{w_1^{(0)}, w_1^{(1)}, \dots\}$ is monotonically increasing. Thus, by combining $w_1^{(n)} < K_1$ with it, we simultaneously get $w_1^{(n)} \rightarrow c_1$ and $s_1^{(n)} \rightarrow 0$ as $n \rightarrow \infty$. Let us assume that

$$\lim_{n \rightarrow \infty} w_{2j-1}^{(n)} = c_j, \quad j = 1, 2, \dots, k-1. \quad (3.22)$$

The 1st and the 3rd equations of (3.7) lead to $w_{2j-1}^{(n+1)} = [(1 + s_j^{(n)})/(1 + s_{j-1}^{(n)})]w_{2j-1}^{(n)}$, and then it holds that

$$\begin{aligned} w_1^{(n+1)} w_3^{(n+1)} \dots w_{2k-1}^{(n+1)} &= (1 + s_1^{(n)}) w_1^{(n)} \left(\frac{1 + s_2^{(n)}}{1 + s_1^{(n)}} w_3^{(n)} \right) \dots \left(\frac{1 + s_k^{(n)}}{1 + s_{k-1}^{(n)}} w_{2k-1}^{(n)} \right) \\ &= (1 + s_k^{(n)}) w_1^{(n)} w_3^{(n)} \dots w_{2k-1}^{(n)}. \end{aligned} \quad (3.23)$$

By taking account that $s_k^{(n)} > 0$ and $w_1^{(n)} w_3^{(n)} \dots w_{2m-1}^{(n)} < K_2$ in (3.23), we derive $w_1^{(0)} w_3^{(0)} \dots w_{2k-1}^{(0)} < w_1^{(1)} w_3^{(1)} \dots w_{2k-1}^{(1)} < \dots < K_2$. Thus, $w_1^{(n)} w_3^{(n)} \dots w_{2k-1}^{(n)}$ converges to some positive constant as $n \rightarrow \infty$. Under the assumption (3.22), it is observed that

$$\begin{aligned} \lim_{n \rightarrow \infty} w_{2k-1}^{(n)} &= c_k, \\ \lim_{n \rightarrow \infty} s_k^{(n)} &= 0. \end{aligned}$$

Moreover, it follows from $s_{m-1}^{(n)} > 0$ that $w_{2m-1}^{(n+1)} < w_{2m-1}^{(n)}/(1 + s_{m-1}^{(n)}) < w_{2m-1}^{(n)}$, namely, $\{w_{2m-1}^{(0)}, w_{2m-1}^{(1)}, \dots\}$ is monotonically decreasing. So, by taking account that $w_{2m-1}^{(n)} > 0$, we see that $w_{2m-1}^{(n)}$ converges to some positive constant as $n \rightarrow \infty$. From $\det((B^{(0)})^\top B^{(0)}) = w_1^{(0)} w_3^{(0)} \dots w_{2m-1}^{(0)} = (\sigma_1(B^{(0)}))^2 (\sigma_2(B^{(0)}))^2 \dots (\sigma_m(B^{(0)}))^2$, it turns out that $\sigma_k(B^{(0)}) \neq 0$ for $k = 1, 2, \dots, m$ if $w_{2k-1}^{(0)} \neq 0$ for $k = 1, 2, \dots, m$. The limit $w_{2m-1}^{(n)} \rightarrow 0$ as $n \rightarrow \infty$ contradicts that $\det((B^{(0)})^\top B^{(0)}) \neq 0$. Thus, it is concluded that

$$\lim_{n \rightarrow \infty} w_{2m-1}^{(n)} = c_m.$$

Consequently, we have (3.18) and (3.21).

Let us recall that $d_k^{(n)} = w_{2k-1}^{(n)}/(1 + s_{k-1}^{(n)})$ and $w_{2k}^{(n)} = d_k^{(n)} s_k^{(n)} = s_k^{(n)} w_{2k-1}^{(n)}/(1 + s_{k-1}^{(n)})$. Then, from (3.18) and (3.21), we easily obtain (3.19) and (3.20).

The ratio of $w_{2k}^{(n+1)}$ to $w_{2k}^{(n)}$ as $n \rightarrow \infty$ is given as

$$\lim_{n \rightarrow \infty} \frac{w_{2k}^{(n+1)}}{w_{2k}^{(n)}} = \lim_{n \rightarrow \infty} \frac{s_k^{(n)} d_{k+1}^{(n)}}{w_{2k}^{(n)}} = \lim_{n \rightarrow \infty} \frac{d_{k+1}^{(n)}}{d_k^{(n)}} = \frac{c_{k+1}}{c_k}.$$

The positivity $w_{2k}^{(0)} > 0$, $w_{2k}^{(1)} > 0$ and the limit (3.19) imply that the ratio $w_{2k}^{(n+1)}/w_{2k}^{(n)}$ is smaller than 1 as $n \rightarrow \infty$. So, it is realized that $c_1 > c_2 > \dots > c_m$.

From (3.18) – (3.21), it is obvious that $(B^{(n)})^\top B^{(n)} \rightarrow \text{diag}(c_1, c_2, \dots, c_m)$. So, it turns out that $\sqrt{c_1}, \sqrt{c_2}, \dots, \sqrt{c_m}$ are the singular values of $B^{(n)}$ as $n \rightarrow \infty$. As is shown in the above, the singular values of $B^{(n)}$ coincide with those of $B^{(0)}$. Therefore it is concluded that $\sqrt{c_1}, \sqrt{c_2}, \dots, \sqrt{c_m}$ are the singular values of $B^{(0)}$. $\square$

Theorem 3.1.2 claims that the dLV_∞ similarity transformation is applicable for computing singular values of the bidiagonal matrix of the form of (3.17), in the same way as the original dLV similarity transformation.

3.2 Mixed forward-backward stability of the dLV_∞ similarity transformation

In this section, we evaluate the perturbations of singular values appearing in the dLV_∞ similarity transformation through the mixed forward-backward error analysis developed by Fernando-Parlett [11, 27] in the absence of the overflow and the underflow. The error analysis in this section is different from that in [18], since the dLV_∞ similarity transformation has the auxiliary variables $d_k^{(n)}$ and $s_k^{(n)}$ which do not appear in the original dLV one.

By recalling (2.7), in floating point arithmetic, the recursion formula for the dLV_∞ similarity transformation is mathematically written as

$$\begin{cases} \hat{d}_k^{(n)} = \text{fl}(w_{2k-1}^{(n)} / \text{fl}(1 + \hat{s}_{k-1}^{(n)})), & k = 1, 2, \dots, m, \\ \hat{s}_k^{(n)} = \text{fl}(w_{2k}^{(n)} / \hat{d}_k^{(n)}), & k = 1, 2, \dots, m-1, \\ \hat{w}_{2k-1}^{(n+1)} = \text{fl}(\hat{d}_k^{(n)} * \text{fl}(1 + \hat{s}_k^{(n)})), & k = 1, 2, \dots, m, \\ \hat{w}_{2k}^{(n+1)} = \text{fl}(\hat{s}_k^{(n)} * \hat{d}_{k+1}^{(n)}), & k = 1, 2, \dots, m-1. \end{cases} \quad (3.24)$$

Let $\hat{\Phi} : W^{(n)} \rightarrow \hat{W}^{(n+1)}$ be the mapping given by (3.24). Moreover, let $\Phi_1 : W^{(n)} \rightarrow \tilde{W}^{(n)}$, $\Phi_2 : \tilde{W}^{(n)} \rightarrow \check{W}^{(n+1)}$ and $\Phi_3 : \check{W}^{(n+1)} \rightarrow \hat{W}^{(n+1)}$ be the mappings such that the sets $\tilde{W}^{(n)} := \{\tilde{w}_1^{(n)}, \tilde{w}_2^{(n)}, \dots, \tilde{w}_{2m-1}^{(n)}\}$, $\check{W}^{(n+1)} := \{\check{w}_1^{(n+1)}, \check{w}_2^{(n+1)}, \dots, \check{w}_{2m-1}^{(n+1)}\}$ and $\hat{W}^{(n+1)} := \{\hat{w}_1^{(n+1)}, \hat{w}_2^{(n+1)}, \dots, \hat{w}_{2m-1}^{(n+1)}\}$ satisfy

$$\begin{cases} \tilde{w}_1^{(n)} = w_1^{(n)}, \\ \tilde{w}_{2k-1}^{(n)} = w_{2k-1}^{(n)}(1 + \varepsilon_{1,k})(1 + \varepsilon_{2,k}), \\ \quad = \frac{w_{2k-1}^{(n)}}{(1 + \varepsilon_{3,k})(1 + \varepsilon_{4,k})}, & k = 2, 3, \dots, m, \\ \tilde{w}_{2k}^{(n)} = w_{2k}^{(n)}(1 + \varepsilon_{5,k}), \\ \quad = \frac{w_{2k}^{(n)}}{1 + \varepsilon_{6,k}}, & k = 1, 2, \dots, m-1, \end{cases} \quad (3.25)$$

$$\begin{cases} \hat{d}_k^{(n)} = \frac{\tilde{w}_{2k-1}^{(n)}}{1 + \hat{s}_{k-1}^{(n)}}, & k = 1, 2, \dots, m, \\ \hat{s}_k^{(n)} = \frac{\tilde{w}_{2k}^{(n)}}{\hat{d}_k^{(n)}}, & k = 1, 2, \dots, m-1, \\ \check{w}_{2k-1}^{(n+1)} = \hat{d}_k^{(n)}(1 + \hat{s}_k^{(n)}), & k = 1, 2, \dots, m, \\ \check{w}_{2k}^{(n+1)} = \hat{s}_k^{(n)}\hat{d}_{k+1}^{(n)}, & k = 1, 2, \dots, m-1, \end{cases} \quad (3.26)$$

$$\begin{cases} \hat{w}_{2k-1}^{(n+1)} = \check{w}_{2k-1}^{(n+1)}(1 + \varepsilon_{7,k})(1 + \varepsilon_{8,k}), \\ \quad = \frac{\check{w}_{2k-1}^{(n+1)}}{(1 + \varepsilon_{9,k})(1 + \varepsilon_{10,k})}, & k = 1, 2, \dots, m-1, \\ \hat{w}_{2k}^{(n+1)} = \check{w}_{2k}^{(n+1)}(1 + \varepsilon_{11,k}), \\ \quad = \frac{\check{w}_{2k}^{(n+1)}}{1 + \varepsilon_{12,k}}, & k = 1, 2, \dots, m-1, \\ \hat{w}_{2m-1}^{(n+1)} = \check{w}_{2m-1}^{(n+1)}, \end{cases} \quad (3.27)$$

where all the absolute values of ε 's are equal to or smaller than ε . Then we get a lemma with respect to the relationship of $\hat{\Phi}$ to Φ_1 , Φ_2 and Φ_3 .

$$\begin{array}{ccc}
W^{(n)} & \xrightarrow{\hat{\Phi}} & \hat{W}^{(n+1)} \\
\Phi_1 \downarrow & & \uparrow \Phi_3 \\
\tilde{W}^{(n)} & \xrightarrow{\Phi_2} & \check{W}^{(n+1)}
\end{array}$$

Figure 3.1: The dLV_∞ similarity transformation in floating point arithmetic.

Lemma 3.2.1. *The mapping $\hat{\Phi}$ is equal to the composition of Φ_1 , Φ_2 and Φ_3 , namely, $\hat{\Phi} = \Phi_3 \circ \Phi_2 \circ \Phi_1$, as is shown in Fig. 3.1.*

Proof. The 1st and the 2nd equations of (3.24) lead to

$$\hat{d}_k^{(n)} = \frac{w_{2k-1}^{(n)}(1 + \varepsilon_{1,k})(1 + \varepsilon_{2,k})}{1 + \hat{s}_{k-1}^{(n)}} = \frac{w_{2k-1}^{(n)}}{(1 + \hat{s}_{k-1}^{(n)})(1 + \varepsilon_{3,k})(1 + \varepsilon_{4,k})}, \quad (3.28)$$

$$\hat{s}_k^{(n)} = \frac{w_{2k}^{(n)}(1 + \varepsilon_{5,k})}{\hat{d}_k^{(n)}} = \frac{w_{2k}^{(n)}}{\hat{d}_k^{(n)}(1 + \varepsilon_{6,k})}. \quad (3.29)$$

Then, by combining the 1st and the 2nd of (3.26) with (3.28) and (3.29), we get (3.25). Similarly, from the 3rd and the 4th of (3.24) and (3.26), we have (3.27). $\square$

Lemma 3.2.1 suggests that the relative differences $\tilde{W}^{(n)}$ and $\hat{W}^{(n)}$ are small relatively different from $W^{(n)}$ and $\check{W}^{(n)}$, respectively.

Let $\tilde{B}^{(n)} = (\tilde{b}_{i,j}^{(n)})$, $\check{B}^{(n+1)} = (\check{b}_{i,j}^{(n+1)})$ and $\hat{B}^{(n+1)} = (\hat{b}_{i,j}^{(n+1)})$ be the upper bidiagonal matrices associated with $\tilde{W}^{(n)}$, $\check{W}^{(n+1)}$, and $\hat{W}^{(n+1)}$, respectively in the same way as $B^{(n)}$ associated with $W^{(n)}$. By combining Theorem 2.2.2 with Lemma 3.2.1, we get a theorem for the perturbations among $\sigma_i(B^{(n)})$, $\sigma_i(\tilde{B}^{(n)})$, $\sigma_i(\check{B}^{(n+1)})$ and $\sigma_i(\hat{B}^{(n+1)})$.

Theorem 3.2.1. *The singular values $\sigma_k(B^{(n)})$, $\sigma_k(\tilde{B}^{(n)})$, $\sigma_k(\check{B}^{(n+1)})$ and $\sigma_k(\hat{B}^{(n+1)})$ satisfy*

$$\sigma_k(\tilde{B}^{(n)}) = \sigma_k(B^{(n)}) \exp \left[\frac{3}{2}(m-1)\tilde{\varepsilon}_k \right], \quad k = 1, 2, \dots, m, \quad (3.30)$$

$$\sigma_k(\check{B}^{(n+1)}) = \sigma_k(\tilde{B}^{(n)}), \quad k = 1, 2, \dots, m, \quad (3.31)$$

$$\sigma_k(\hat{B}^{(n+1)}) = \sigma_k(\check{B}^{(n+1)}) \exp \left[\frac{3}{2}(m-1)\hat{\varepsilon}_k \right], \quad k = 1, 2, \dots, m, \quad (3.32)$$

where $|\tilde{\varepsilon}_k|, |\hat{\varepsilon}_k| \leq \varepsilon$.

Proof. It is obvious from Lemma 3.2.1 that

$$\sqrt{\tilde{w}_{2k-1}^{(n)}} = \sqrt{w_{2k-1}^{(n)}} \sqrt{(1 + \varepsilon_{1,k})(1 + \varepsilon_{2,k})}, \quad \sqrt{\tilde{w}_{2k}^{(n)}} = \sqrt{w_{2k}^{(n)}} \sqrt{1 + \varepsilon_{3,k}}.$$

It is noted here that $\max(1 + \varepsilon_{1,k}, (1 + \varepsilon_{1,k})^{-1}), \max(1 + \varepsilon_{2,k}, (1 + \varepsilon_{2,k})^{-1}), \max(1 + \varepsilon_{3,k}, (1 + \varepsilon_{3,k})^{-1}) \leq 1 + \varepsilon$. With the help of Theorem 2.2.2, we can estimate the relative difference in any singular value between $B^{(n)}$ and $\tilde{B}^{(n)}$ as follows.

$$\exp \left[-\frac{3}{2}(m-1)\varepsilon \right] \leq \frac{\sigma_k(\tilde{B}^{(n)})}{\sigma_k(B^{(n)})} \leq \exp \left[\frac{3}{2}(m-1)\varepsilon \right], \quad k = 1, 2, \dots, m. \quad (3.33)$$

Let $\tilde{\varepsilon}_k$ satisfy $|\tilde{\varepsilon}_k| \leq \varepsilon$. Then (3.33) leads to (3.30).

Similarly, the relative difference in any singular value between $\tilde{B}^{(n+1)}$ and $\hat{B}^{(n+1)}$ is estimated as

$$\exp \left[-\frac{3}{2}(m-1)\varepsilon \right] \leq \frac{\sigma_k(\hat{B}^{(n+1)})}{\sigma_k(\tilde{B}^{(n+1)})} \leq \exp \left[\frac{3}{2}(m-1)\varepsilon \right], \quad k = 1, 2, \dots, m. \quad (3.34)$$

By taking account of suitable $\hat{\varepsilon}_k$ such that $|\hat{\varepsilon}_k| \leq \varepsilon$, from (3.34), we derive (3.32). $\square$

Incidentally, the replacement of $3/2$ with 2 in (3.30) and (3.32) brings to the perturbation theorem in [18] concerning the original dLV similarity transformation. In other words, the perturbations of singular values in the dLV $_{\infty}$ similarity transformation is slightly smaller than those in the original one in the absence of the overflow and the underflow.

3.3 Forward stability of the dLV $_{\infty}$ similarity transformation

Let us introduce three sets $W^{(n)} := \{w_1^{(n)}, w_2^{(n)}, \dots, w_{2m-1}^{(n)}\}$, $W^{(n+1)} := \{w_1^{(n+1)}, w_2^{(n+1)}, \dots, w_{2m-1}^{(n+1)}\}$ and $\hat{W}^{(n+1)} := \{\hat{w}_1^{(n+1)}, \hat{w}_2^{(n+1)}, \dots, \hat{w}_{2m-1}^{(n+1)}\}$. Then $\Phi : W^{(n)} \rightarrow W^{(n+1)}$ and $\hat{\Phi} : W^{(n)} \rightarrow \hat{W}^{(n+1)}$ are the mappings given by the dLV $_{\infty}$ similarity transformations in exact arithmetic and floating point one, respectively. See also Figure 3.2 for the relationship of two mappings Φ and $\hat{\Phi}$. The following proposition suggests the relative forward errors in the auxiliary variables in floating point arithmetic.

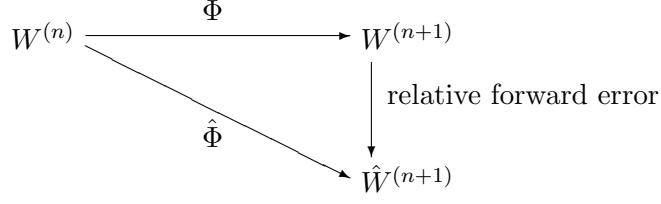


Figure 3.2: The dLV_∞ similarity transformations in exact arithmetic and floating point one.

Proposition 3.3.1. *The auxiliary variables $\hat{d}_k^{(n)}$ and $\hat{s}_k^{(n)}$ in floating point arithmetic are representable in terms of $d_k^{(n)}$ and $s_k^{(n)}$ in exact one as*

$$\hat{d}_k^{(n)} = d_k^{(n)}(1 + \xi_k), \quad |\xi_k| \leq (3k - 3)\varepsilon + O(\varepsilon^2), \quad k = 1, 2, \dots, m, \quad (3.35)$$

$$\hat{s}_k^{(n)} = s_k^{(n)}(1 + \eta_k), \quad |\eta_k| \leq (3k - 2)\varepsilon + O(\varepsilon^2), \quad k = 1, 2, \dots, m - 1. \quad (3.36)$$

Proof. From $\hat{s}_0^{(n)} = 0$, it is obvious that $\varepsilon_{3,1} = 0$, $\varepsilon_{4,1} = 0$ and $\hat{d}_1^{(n)} = w_1^{(n)} = d_1^{(n)}$. Equation (3.29) with $k = 1$ leads to $\hat{s}_1^{(n)} = s_1^{(n)}(1 + \varepsilon_{5,1})$ since $\hat{d}_1^{(n)} = d_1^{(n)}$ and $w_2^{(n)}/d_1^{(n)} = s_1^{(n)}$. It is worth finding here that

$$\frac{1}{1 + \hat{s}_1^{(n)}} = \frac{1}{1 + s_1^{(n)}} \frac{1}{1 + \varepsilon'_1}, \quad (3.37)$$

where $|\varepsilon'_1| \leq |\varepsilon_{5,1}| \leq \varepsilon$. So, by taking account that $w_3^{(n)}/(1 + s_1^{(n)}) = d_2^{(n)}$ in (3.28) with $k = 2$, we easily get $\hat{d}_2^{(n)} = d_2^{(n)}/[(1 + \varepsilon'_1)(1 + \varepsilon'_2)(1 + \varepsilon'_3)]$ where $\varepsilon'_2 = \varepsilon_{3,2}$ and $\varepsilon'_3 = \varepsilon_{4,2}$.

Let us assume that

$$\hat{d}_{k-1}^{(n)} = \frac{d_{k-1}^{(n)}}{(1 + \varepsilon''_1)(1 + \varepsilon''_2) \cdots (1 + \varepsilon''_{3(k-1)-3})}, \quad (3.38)$$

where $|\varepsilon''_j| \leq \varepsilon$ for $j = 1, 2, \dots, 3(k-1)-3$. Then, by using (3.38) and $w_{2k-2}^{(n)}/d_{k-1}^{(n)} = s_{k-1}^{(n)}$ after replacing k with $k-1$ in (3.29), we get

$$\hat{s}_{k-1}^{(n)} = s_{k-1}^{(n)}(1 + \varepsilon''_1)(1 + \varepsilon''_2) \cdots (1 + \varepsilon''_{3(k-1)-2}), \quad (3.39)$$

where $1 + \varepsilon''_{3(k-1)-2} = 1/(1 + \varepsilon_{6,k-1})$ and $|\varepsilon''_{3(k-1)-2}| \leq \varepsilon$. Moreover, by taking account that

$$\frac{1}{1 + \hat{s}_{k-1}^{(n)}} = \frac{1}{1 + s_{k-1}^{(n)}} \frac{1}{(1 + \varepsilon'''_1)(1 + \varepsilon'''_2) \cdots (1 + \varepsilon'''_{3(k-1)-2})},$$

where $|\varepsilon_j'''| \leq \varepsilon$ for $j = 1, 2, \dots, 3(k-1) - 2$ and $w_{2k-1}^{(n)}/(1 + s_{k-1}^{(n)}) = d_k^{(n)}$ in (3.28), we derive

$$\hat{d}_k^{(n)} = \frac{d_k^{(n)}}{(1 + \varepsilon_1''')(1 + \varepsilon_2''') \cdots (1 + \varepsilon_{3k-3}''')}, \quad (3.40)$$

where $\varepsilon_{3k-4}''' = \varepsilon_{3,k}$ and $\varepsilon_{3k-3}''' = \varepsilon_{4,k}$. From (3.29) with (3.40) and $w_{2k}^{(n)}/d_k^{(n)} = s_k^{(n)}$, it follows that

$$\hat{s}_k^{(n)} = s_k^{(n)}(1 + \varepsilon_1''')(1 + \varepsilon_2''') \cdots (1 + \varepsilon_{3k-2}'''), \quad (3.41)$$

where $\varepsilon_{3k-2}''' = \varepsilon_{6,k}$. Thus, by induction for k , from (3.40) and (3.41) we have (3.35) and (3.36). $\square$

Proposition 3.3.1 enable us to give the main theorem concerning the relative forward error of the dLV $_{\infty}$ similarity transformation in floating point arithmetic as is shown in Figure 3.2.

Theorem 3.3.1. *The relative forward errors in $\hat{w}_1^{(n+1)}, \hat{w}_2^{(n+1)}, \dots, \hat{w}_{2m-1}^{(n+1)}$ are bounded as follow.*

$$\max_k \left\{ \left| \frac{w_k^{(n+1)} - \hat{w}_k^{(n+1)}}{w_k^{(n+1)}} \right| \right\} \leq (6m - 4)\varepsilon + O(\varepsilon^2). \quad (3.42)$$

Proof. By combining Proposition 3.3.1 with

$$\begin{aligned} \hat{w}_{2k-1}^{(n+1)} &= \hat{d}_k^{(n)}(1 + \hat{s}_k^{(n)})(1 + \varepsilon_{7,k})(1 + \varepsilon_{8,k}), \\ \hat{w}_{2k}^{(n+1)} &= \hat{s}_k^{(n)} \hat{d}_{k+1}^{(n)}(1 + \varepsilon_{11,k}), \end{aligned}$$

where $|\varepsilon_{7,k}|, |\varepsilon_{8,k}|, |\varepsilon_{11,k}| \leq \varepsilon$, we get

$$\begin{aligned} \hat{w}_{2k-1}^{(n+1)} &= w_{2k-1}^{(n+1)}(1 + \mu_k), \quad |\mu_k| \leq (6k - 3)\varepsilon + O(\varepsilon^2), \quad k = 1, 2, \dots, m-1, \\ \hat{w}_{2k}^{(n+1)} &= w_{2k}^{(n+1)}(1 + \nu_k), \quad |\nu_k| \leq (6k - 1)\varepsilon + O(\varepsilon^2), \quad k = 1, 2, \dots, m-1, \\ \hat{w}_{2m-1}^{(n+1)} &= w_{2m-1}^{(n+1)}(1 + \mu_m), \quad |\mu_m| \leq (6m - 4)\varepsilon + O(\varepsilon^2) \end{aligned}$$

which implies (3.42). $\square$

Theorem 3.3.1 shows that the upper bound of relative forward error in the dLV $_{\infty}$ similarity transformation is sufficiently small in floating point arithmetic, regardless of the entries of $W^{(n)}$. Therefore it is concluded that the dLV $_{\infty}$ similarity transformation is always forward stable in the absence of the overflow and the underflow.

The dLV_∞ similarity transformation will be more useful for computing the singular values of B through introducing a shift of the origin into it. It is however remarkable that not only the zero-shift dLV_∞ similarity transformations but also nonzero-shift dLV_∞ ones surely run since the settings of the shift often fail in floating point arithmetic. The forward stability of the zero-shift dLV_∞ similarity transformation is still a key point for computing the singular values of B with high relative accuracy in floating point arithmetic.

3.4 Numerical examples

In this section, we present some examples in order to numerically confirm the resulting theorems for the error analysis of the dLV_∞ similarity transformation. We first show the relative errors in the computed singular values of B by the sequence of the dLV_∞ similarity transformations. Simultaneously, with respect to the relative errors in the computed singular values of B , we compare the sequence of the dLV_∞ similarity transformations with those of the original dLV with $\delta^{(n)} = 1$ and the dqd similarity transformations. In this section, we simply call the sequences of the dLV_∞ , the dLV with $\delta^{(n)} = 1$ and the dqd similarity transformations as the dLV_∞ , the dLV and the dqd, respectively. We next investigate the relative forward errors in the dLV_∞ variables after the dLV_∞ similarity transformation. We use our computer with CPU: Intel Core i5 2.67GHz, RAM: 4GB, OS: Windows 7 and Compiler: gcc version 4.5.3.

We prepare three 100-by-100 bidiagonal matrices B_1 , B_2 and B_3 with the (i, j) entries

$$b_{i,j} = \begin{cases} 2.001, & i = j, \\ 2, & i + 1 = j, \end{cases}$$

$$b_{i,j} = \begin{cases} 1, & i = j, \\ 10, & i + 1 = j, \end{cases}$$

$$b_{i,j} = \begin{cases} 1, & i = j = 1, \\ 0.001, & i + 1 = j = 2, \\ 2, & i = j \neq 1, \\ 0.002, & i + 1 = j \neq 2, \end{cases}$$

respectively. In quadruple-precision arithmetic, the computed singular values

by the dqd is as follows.

$$\left\{ \begin{array}{l} \sigma_1(B_1) = 4.00051130686550 \times 10^0, \\ \sigma_2(B_1) = 3.99904534684389 \times 10^0, \\ \vdots \\ \sigma_{99}(B_1) = 9.40106762784246 \times 10^{-2}, \\ \sigma_{100}(B_1) = 3.19067259533378 \times 10^{-2}, \\ \sigma_1(B_2) = 1.09995522237261 \times 10^1, \\ \sigma_2(B_2) = 1.09982092254914 \times 10^1, \\ \vdots \\ \sigma_{99}(B_2) = 9.00054946951601 \times 10^0, \\ \sigma_{100}(B_2) = 9.89999999999999 \times 10^{-100}, \\ \sigma_1(B_3) = 2.00199901409324 \times 10^0, \\ \sigma_2(B_3) = 2.00199605734307 \times 10^0, \\ \vdots \\ \sigma_{99}(B_3) = 1.99800098785371 \times 10^0, \\ \sigma_{100}(B_3) = 9.99999833333097 \times 10^{-1}. \end{array} \right.$$

Hereinafter, we regard them as the exact singular values. The matrix B_1 does not have any special feature with respect to the distribution of singular values. While B_2 and B_3 have densely arranged singular values around 1.0×10^1 and 2.0×10^0 , respectively. The matrix B_2 differs from B_3 in that the condition number is considerably large where the condition number is given by the ratio of the maximal singular value to the minimal one. Figure 3.3 shows the relative errors $r_k(B_j) = |\hat{\sigma}_k(B_j) - \sigma_k(B_j)|/\sigma_k(B_j)$ in the computed singular values $\hat{\sigma}_k(B_j)$ by not only the dLV_∞ but also the dLV and the dqd in double-precision arithmetic. The circles, the triangles and the crosses denote the relative errors $r_1(B_j), r_2(B_j), \dots, r_{100}(B_j)$ in the cases where the dLV_∞ , the dLV and the dqd, respectively. Table 3.1 describes the iteration number $N(B_j)$, which is the minimal positive integer such that $\max_k w_{2k}^{(N(B_j))}$ in the dLV_∞ , the dLV or $\max_k e_k^{(N(B_j))}$ in the dqd is smaller than the machine epsilon $\varepsilon = 2^{-52}$, the maximal relative errors $r_{\max}(B_j) := \max_k r_k(B_j)$ and the average relative error $r_{\text{ave}}(B_j) := \sum_{k=1}^{100} r_k(B_j)/100$ in double-precision arithmetic.

It is observed that $r_{\max}(B_1)$ is smaller than the product of $N(B_1)$ shown in Table 3.1 and $\exp(3 \times 99 \times \varepsilon) - 1$ estimated by Theorem 3.2.1 since $\exp(3 \times 99 \times 2^{-52}) - 1 \approx 6.6 \times 10^{-14}$. It turns out from Table 3.1 and Fig. 3.3 that the noticeable difference of the relative errors do not appear, among the dLV_∞ , the dLV and the dqd. With respect to the iteration number $N(B_1)$ in Table 3.1, the dLV_∞ and the dqd are better than the dLV. Similarly to the case of B_1 ,

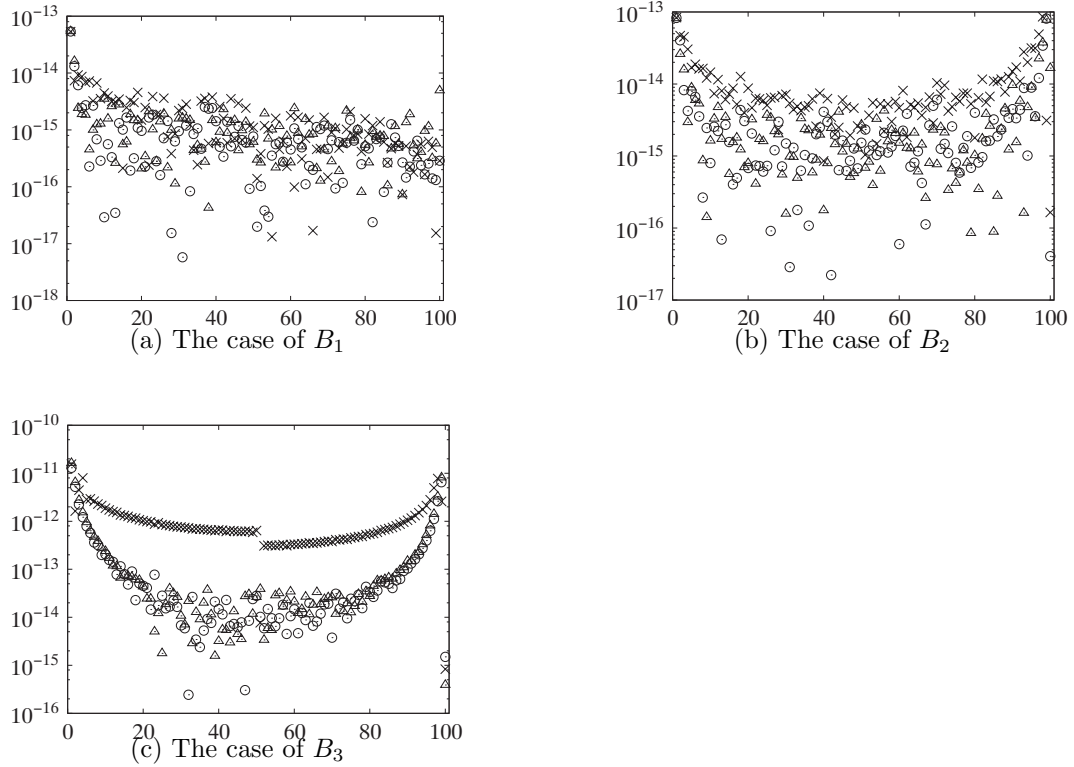


Figure 3.3: Graphs of the subscripts $1, 2, \dots, 100$ of the computed singular values $\hat{\sigma}_1(B_j), \hat{\sigma}_2(B_j), \dots, \hat{\sigma}_{100}(B_j)$ (x-axis) and the relative errors $r_1(B_j), r_2(B_j), \dots, r_{100}(B_j)$ in $\hat{\sigma}_1(B_j), \hat{\sigma}_2(B_j), \dots, \hat{\sigma}_{100}(B_j)$ in double-precision arithmetic. Circle: the dLV_∞ , triangle: the dLV and cross: the dqd.

in the case of B_2 , we can easily check that Theorem 3.2.1 holds. It is obvious from Table 3.1 and Fig. 3.3 that the dLV_∞ is slightly the best with respect to the accuracy of the computed singular values of B_2 . More precisely, the relative errors appearing in the dLV_∞ are almost equal to those in the dLV, and are explicitly different from the dqd, compared with the case of B_1 . With respect to the iteration number $N(B_2)$, the dLV_∞ is the same as the dqd, and is better than the dLV. In the case of B_3 , the dLV_∞ and the dLV generate more accurate singular values than the dqd, and Theorem 3.2.1 is numerically verified. The iteration number $N(B_3)$ in the dLV_∞ algorithm is the almost same as that of the dqd, and is smaller than that of the dLV.

Next, we numerically demonstrate Theorem 3.3.1 for estimating the relative errors in the dLV_∞ variables. In the case of B_1 , $w_{2k-1}^{(1)}$ for $k = 1, 2, \dots, 100$ and $w_{2k}^{(1)}$ for $k = 1, 2, \dots, 99$ are exactly expressed in terms of $w_{2k-1}^{(0)} = 2.001^2$

Table 3.1: The iteration number $N(B_j)$, the maximal $r_{\max}(B_j)$ and the average relative error $r_{\text{ave}}(B_j)$ in the dLV_∞ , the dLV and the dqd in double-precision arithmetic.

	the dLV_∞	the dLV	the dqd
$N(B_1)$	35151	37348	35151
$r_{\max}(B_1)$	5.3×10^{-14}	5.4×10^{-14}	5.3×10^{-14}
$r_{\text{ave}}(B_1)$	1.4×10^{-15}	1.7×10^{-15}	2.3×10^{-15}
$N(B_2)$	104789	105655	104789
$r_{\max}(B_2)$	8.3×10^{-14}	8.2×10^{-14}	8.6×10^{-14}
$r_{\text{ave}}(B_2)$	4.5×10^{-15}	4.6×10^{-15}	1.0×10^{-14}
$N(B_3)$	4520413	5648260	4520409
$r_{\max}(B_3)$	1.3×10^{-11}	1.6×10^{-11}	1.5×10^{-11}
$r_{\text{ave}}(B_3)$	3.9×10^{-13}	4.8×10^{-13}	1.2×10^{-12}

for $k = 1, 2, \dots, 100$ and $w_{2k}^{(0)} = 2^2$ for $k = 1, 2, \dots, 99$ as follows.

$$\begin{aligned}
w_{2k-1}^{(1)} &= 2.001^2 \frac{\left(\frac{2000}{2001}\right)^{2k+2} - 1}{\left(\frac{2000}{2001}\right)^{2k} - 1}, \\
w_{2k}^{(1)} &= 2^2 \frac{\left(\frac{2000}{2001}\right)^{2k} - 1}{\left(\frac{2000}{2001}\right)^{2k+2} - 1}, \quad k = 1, 2, \dots, 99, \\
w_{199}^{(1)} &= 2.001^2 \frac{\left(\frac{2000}{2001}\right)^2 - 1}{\left(\frac{2000}{2001}\right)^{200} - 1}.
\end{aligned}$$

Similarly, the exact values $w_{2k-1}^{(1)}$ for $k = 1, 2, \dots, 100$ and $w_{2k}^{(1)}$ for $k = 1, 2, \dots, 99$ are given by using the values of $w_{2k-1}^{(0)}$ for $k = 1, 2, \dots, 100$ and $w_{2k}^{(0)}$ for $k = 1, 2, \dots, 99$ in the cases of B_2 and B_3 . Fig. 3.4 shows the forward errors $|\hat{w}_k^{(1)} - w_k^{(1)}|/w_k^{(1)}$ for $k = 1, 2, \dots, 199$, where $\hat{w}_k^{(1)}$ denotes the computed value by the dLV_∞ similarity transformation in double-precision arithmetic. It turns out from Fig. 3.4 that the forward errors in $\hat{w}_k^{(1)}$ are sufficiently small. The

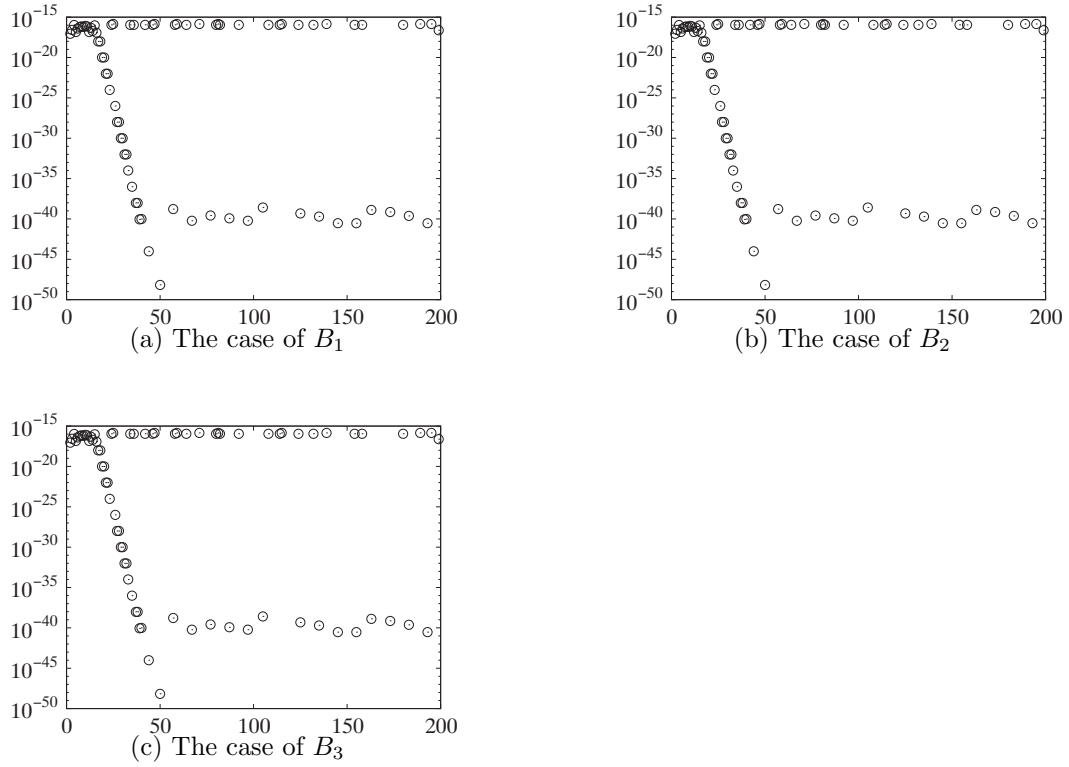


Figure 3.4: Graphs of the subscripts $1, 2, \dots, 199$ of $\hat{w}_1^{(1)}, \hat{w}_2^{(1)}, \dots, \hat{w}_{199}^{(1)}$ (x-axis) and the relative errors in $\hat{w}_1^{(1)}, \hat{w}_2^{(1)}, \dots, \hat{w}_{199}^{(1)}$ (y-axis) after the dLV_∞ similarity transformation. All the plots below the x-axis are omitted.

values of $r_{\max}^{(1)}(B_j) := \max_k |\hat{w}_k^{(1)} - w_k^{(1)}|/w_k^{(1)}$ in double-precision arithmetic are

$$r_{\max}^{(1)}(B_1) = 1.7 \times 10^{-16}, \quad r_{\max}^{(1)}(B_2) = 1.4 \times 10^{-16}, \quad r_{\max}^{(1)}(B_3) = 1.3 \times 10^{-16}.$$

This suggests that, in the cases of B_1 , B_2 and B_3 , the forward errors in $\hat{w}_k^{(1)}$ for $k = 1, 2, \dots, 199$ are smaller than $3m\varepsilon = 6.7 \times 10^{-14}$ shown as in Theorem 3.3.1.

Chapter 4

Toward Accurate Singular Vectors of Bidiagonal Matrices

In Chapter 1, we give a brief explanation about the condition number of matrix, by which we measure the condition of linear system $A\mathbf{x} = \mathbf{b}$, where $A \in \mathbf{R}^{m \times m}$ and $\mathbf{x}, \mathbf{b} \in \mathbf{R}^m$. As a first step for obtaining more accurate singular vectors from linear system (1.3), we discuss a preconditioner such that the preconditioned system is more well-conditioned than the original system in this chapter. More precisely, we confirm the behavior of the condition number or extended condition number of coefficient matrix in preconditioning.

Though we assume that the coefficient matrix A of an original linear system is M -tridiagonal, it is not required that A is symmetric, in this chapter. It aims at applications of results in this chapter not only to singular vector computation but also to wider class of linear systems whose coefficient matrix is M -tridiagonal. Actually, in the case when we solve linear systems with Richardson's iteration, we do not need symmetry of coefficient matrix and the maximal and minimal eigenvalue of coefficient matrix involves convergence.[31]. Thus, we express the elements of A as (1.10).

This chapter is organized as follows. In Section 4.1, we explain of the arbitrary-stride reduction method. Next, in Section 4.2, we clarify the row and column permutations for transforming the M -tridiagonal matrix into a tridiagonal matrix without changing the eigenvalues of the matrix. In Section 4.3, we define arbitrary-stride reduction in terms of a combination of 2-stride reduction and row and column permutations. Using the findings of Wang *et al.* [38], in Section 4.4, we describe a remarkable property of coefficient matrices in linear systems involving their condition numbers or extended condition numbers following stride reduction, and then present two numerical examples.

4.1 Arbitrary-stride reduction

The 2-stride, 3-stride and 4-stride reductions for tridiagonal linear systems are presented by Evans [10]. Thus, we will formulate the arbitrary-stride reduction along the same lines. It is easy to expand this reduction to $2M$ -stride reduction for an arbitrary positive integer M . In this section, we define $2M$ -stride reduction for transforming M -tridiagonal linear systems into $2M$ -tridiagonal systems.

We consider the M -tridiagonal linear system $A\mathbf{x} = \mathbf{y}$ with a unique solution $\mathbf{x} = (x_1, x_2, \dots, x_m)^\top \in \mathbf{R}^m$ and given $\mathbf{y} = (y_1, y_2, \dots, y_m)^\top \in \mathbf{R}^m$. By focusing on the $(i - M)$ th, i th and $(i + M)$ th entries of both sides of $A\mathbf{x} = \mathbf{y}$, we easily derive the following:

$$c_{i-M}x_{i-2M} + b_{i-M}x_{i-M} + a_{i-M}x_i = y_{i-M}, \quad (4.1)$$

$$c_i x_{i-M} + b_i x_i + a_i x_{i+M} = y_i, \text{ and} \quad (4.2)$$

$$c_{i+M}x_i + b_{i+M}x_{i+M} + a_{i+M}x_{i+2M} = y_{i+M}. \quad (4.3)$$

By multiplying both sides of (4.1) by $-c_i/b_{i-M}$ and adding them to both sides of (4.2), we derive

$$-\frac{c_i}{b_{i-M}}c_{i-M}x_{i-2M} + \left(b_i - \frac{c_i}{b_{i-M}}a_{i-M}\right)x_i + a_i x_{i+M} = y_i - \frac{c_i}{b_{i-M}}y_{i-M}. \quad (4.4)$$

We note that the term with respect to x_{i-M} does not appear in (4.4). Similarly, it follows from (4.2) and (4.3) that:

$$c_i x_{i-M} + \left(b_i - \frac{a_i}{b_{i+M}}c_{i+M}\right)x_i - \frac{a_i}{b_{i+M}}a_{i+M}x_{i+2M} = y_i - \frac{a_i}{b_{i+M}}y_{i+M}, \quad (4.5)$$

which does not contain the term with respect to x_{i+M} . Moreover, by adding (4.4) to (4.5) and by subtracting (4.2), we obtain:

$$\tilde{c}_i x_{i-2M} + \tilde{b}_i x_i + \tilde{a}_i x_{i+2M} = \tilde{y}_i, \text{ where} \quad (4.6)$$

$$\begin{cases} \tilde{a}_i = -\frac{a_i}{b_{i+M}}a_{i+M}, \\ \tilde{b}_i = b_i - \frac{c_i}{b_{i-M}}a_{i-M} - \frac{a_i}{b_{i+M}}c_{i+M}, \\ \tilde{c}_i = -\frac{c_i}{b_{i-M}}c_{i-M}, \\ \tilde{y}_i = y_i - \frac{c_i}{b_{i-M}}y_{i-M} - \frac{a_i}{b_{i+M}}y_{i+M}. \end{cases} \quad (4.7)$$

From (4.6) and (4.7), we found that the M -tridiagonal linear system $A\mathbf{x} = \mathbf{y}$ can be transformed into a $2M$ -tridiagonal linear system

$$\tilde{A}\mathbf{x} = \tilde{\mathbf{y}}, \quad (4.8)$$

$$\tilde{A} := \begin{pmatrix} & \tilde{b}_1 & & \tilde{a}_1 & & \\ & & \ddots & & \ddots & \\ \tilde{c}_{2M+1} & & & \tilde{b}_{2M+1} & & \\ & & \ddots & & \ddots & \\ & & & & \ddots & \tilde{a}_{m-2M} \\ & & & & & \ddots \\ & & & & \tilde{c}_m & \\ & & & & & \tilde{b}_m \end{pmatrix}, \quad \tilde{\mathbf{y}} := \begin{pmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \vdots \\ \tilde{y}_m \end{pmatrix}. \quad (4.9)$$

Since a stride of $2M$ appears in the subscripts in (4.6), this transformation is referred to as $2M$ -stride reduction. The $2M$ -tridiagonal linear system $\tilde{A}\mathbf{x} = \tilde{\mathbf{y}}$ can also be found by multiplying both sides of the M -tridiagonal linear system $A\mathbf{x} = \mathbf{y}$ by a M -tridiagonal matrix with $\alpha_i := -a_i/b_{i+M}$ and $\beta_i := -c_i/b_{i-M}$:

$$S := \begin{pmatrix} 1 & & \alpha_1 & & \\ & \ddots & & \ddots & \\ \beta_{M+1} & & 1 & & \ddots \\ & \ddots & & \ddots & \alpha_{m-M} \\ & & \ddots & & \ddots \\ & & & \beta_m & 1 \end{pmatrix}, \quad (4.10)$$

namely, $SA = \tilde{A}$ and $S\mathbf{y} = \tilde{\mathbf{y}}$. It is noted here that S is nonsingular if $b_i \neq 0$ for $i = 1, 2, \dots, m$ and A is nonsingular.

Proposition 4.1.1. *Let $b_i \neq 0$ for $i = 1, 2, \dots, m$, A be nonsingular and D , A_b be*

$$D := \text{diag} \left((-1)^{\lceil \frac{1}{M} \rceil}, (-1)^{\lceil \frac{2}{M} \rceil}, \dots, (-1)^{\lceil \frac{m}{M} \rceil} \right),$$

$$A_b := \text{diag} (b_1, b_2, \dots, b_m),$$

where $\lceil \cdot \rceil$ denotes the ceiling function. Then,

$$S = DADA_b^{-1}$$

and S is nonsingular.

Proof. Let us now introduce an m -dimensional unit column vector $\mathbf{e}_j$ whose j th entry is 1. By considering the product $A((-1)^{\lceil j/M \rceil} \mathbf{e}_j)$, we obtain

$$A \left((-1)^{\lceil \frac{j}{M} \rceil} \mathbf{e}_j \right) = (-1)^{\lceil \frac{j}{M} \rceil} \mathbf{A}_{*,j},$$

where $\mathbf{A}_{*,j}$ is the j th column vector of A . Moreover, by considering the product $((-1)^{\lceil i/M \rceil} \mathbf{e}_i)^\top (-1)^{\lceil j/M \rceil} \mathbf{A}_{*,j}$, we derive

$$\left((-1)^{\lceil \frac{i}{M} \rceil} \mathbf{e}_i \right)^\top (-1)^{\lceil \frac{j}{M} \rceil} \mathbf{A}_{*,j} = (-1)^{\lceil \frac{i}{M} \rceil + \lceil \frac{j}{M} \rceil} A_{i,j},$$

where $A_{i,j}$ is the (i,j) th entry of A . For $i = j - M$, by taking into account that $\lceil x - i \rceil = \lceil x \rceil - i$ for arbitrary real number x and arbitrary integer i , we have

$$(-1)^{\lceil \frac{i}{M} \rceil + \lceil \frac{j}{M} \rceil} A_{i,j} = (-1)^{2\lceil \frac{j}{M} \rceil - 1} A_{j-M,j} = -a_{j-M}.$$

For $i = j$, we have

$$(-1)^{\lceil \frac{i}{M} \rceil + \lceil \frac{j}{M} \rceil} A_{i,j} = (-1)^{2\lceil \frac{j}{M} \rceil} A_{j,j} = b_j.$$

For $i = j + M$, by taking into account that $\lceil x + i \rceil = \lceil x \rceil + i$ for arbitrary real number x and arbitrary integer i , we have

$$(-1)^{\lceil \frac{i}{M} \rceil + \lceil \frac{j}{M} \rceil} A_{i,j} = (-1)^{2\lceil \frac{j}{M} \rceil + 1} A_{j+M,j} = -c_{j+M}.$$

Let

$$D := \begin{pmatrix} \left((-1)^{\lceil \frac{1}{M} \rceil} \mathbf{e}_1 \right)^\top \\ \left((-1)^{\lceil \frac{2}{M} \rceil} \mathbf{e}_2 \right)^\top \\ \vdots \\ \left((-1)^{\lceil \frac{m}{M} \rceil} \mathbf{e}_m \right)^\top \end{pmatrix}.$$

Then, by recalling that $A_{i,j} = 0$ for $i \neq j - M, j, M + j$, we observe that:

$$DA \left((-1)^{\lceil \frac{j}{M} \rceil} \mathbf{e}_j \right) = \begin{pmatrix} \left((-1)^{\lceil \frac{1}{M} \rceil} \mathbf{e}_1 \right)^\top \\ \left((-1)^{\lceil \frac{2}{M} \rceil} \mathbf{e}_2 \right)^\top \\ \vdots \\ \left((-1)^{\lceil \frac{m}{M} \rceil} \mathbf{e}_m \right)^\top \end{pmatrix} A \left((-1)^{\lceil \frac{j}{M} \rceil} \mathbf{e}_j \right) = \begin{pmatrix} \mathbf{0} \\ b_j \\ \mathbf{0} \\ -c_{j+M} \\ \mathbf{0} \end{pmatrix},$$

$$j = 1, 2, \dots, M,$$

$$DA \left((-1)^{\lceil \frac{j}{M} \rceil} \mathbf{e}_j \right) = \begin{pmatrix} \mathbf{0} \\ -a_{j-M} \\ \mathbf{0} \\ b_j \\ \mathbf{0} \\ -c_{j+M} \\ \mathbf{0} \end{pmatrix}, \quad j = M + 1, M + 2, \dots, m - M,$$

$$DA \left((-1)^{\lceil \frac{j}{M} \rceil} \mathbf{e}_j \right) = \begin{pmatrix} \mathbf{0} \\ -a_{j-M} \\ \mathbf{0} \\ b_j \\ \mathbf{0} \end{pmatrix}, \quad j = m - M + 1, m - M + 2, \dots, m.$$

Thus, by noting that $DA((-1)^{\lceil 1/M \rceil} \mathbf{e}_1 (-1)^{\lceil 2/M \rceil} \mathbf{e}_2 \dots (-1)^{\lceil m/M \rceil} \mathbf{e}_m) = DAD$, we find that:

$$DAD = \begin{pmatrix} b_1 & & -a_1 & & & & \\ & \ddots & & \ddots & & & \\ -c_{M+1} & & b_{M+1} & & \ddots & & \\ & \ddots & & \ddots & & -a_{m-M} & \\ & & \ddots & & \ddots & & \\ & & & -c_m & & b_m & \end{pmatrix},$$

$$DADA_b^{-1} = \begin{pmatrix} 1 & & -\frac{a_1}{b_{M+1}} & & & & \\ & \ddots & & \ddots & & & \\ -\frac{c_{M+1}}{b_1} & & 1 & & \ddots & & \\ & \ddots & & \ddots & & -\frac{a_{m-M}}{b_m} & \\ & & \ddots & & \ddots & & \\ & & & -\frac{c_m}{b_{m-M}} & & 1 & \end{pmatrix} = S. \quad (4.11)$$

In order to show that S is nonsingular, we consider $\det(S)$. By taking into account that $\det(D) = (-1)^{\sum_{i=1}^m \lceil i/M \rceil}$, namely, $\det(D) = -1$ or 1 , from (4.11), we derive

$$\det(S) = (\det(D))^2 \det(A) \det(A_b^{-1}) = \det(A) \det(A_b^{-1}).$$

Since A is nonsingular and $b_i \neq 0$ for $i = 1, 2, \dots, m$, we see that $\det(A) \neq 0$ and $\det(A_b^{-1}) = \prod_{i=1}^m (1/b_i) \neq 0$. Therefore, we have $\det(S) \neq 0$, that is, S is nonsingular. $\square$

The case when $M = 1$ in Proposition 4.1.1 corresponds to Lemma 2.1 in [38].

4.2 Block diagonalization with tridiagonal blocks

In this section, we show that the M -tridiagonal matrix $A \in \mathbf{R}^{m \times m}$ can be transformed into a tridiagonal matrix without changing the eigenvalues. More precisely, we clarify that a similarity transformation for A leads to a block diagonal matrix with M diagonal blocks that are all tridiagonal. It is expected that $P^\top A P$ for a permutation matrix P becomes not only tridiagonal but also block diagonal. This is because A has $(m - M)$ entries in the upper band, m entries in the diagonal band and $(m - M)$ entries in the lower band, except at obviously zero-valued bands. The remainder of this section actually presents a block diagonalization $P^\top A P$ using a permutation matrix P . $P^\top A P$ is also called a *symmetric permutation* of A [13]. The block diagonalization is similar to that given by Jia *et al.* [21] and Sogabe and El-Mikkawy [33], but differs in the arrangement of diagonal blocks. Jia *et al.* [21] also made use of block diagonalization to present an explicit formula for the inverse of an M -tridiagonal Toeplitz matrix. Our research requires block diagonalization to consider an equivalent transformation of $2M$ -stride reduction. It is worth noting here that $m = kM + \ell = \ell(k + 1) + (M - \ell)k$ for some positive integer k and some nonnegative integer ℓ . This suggests that M diagonal blocks can be divided into ℓ diagonal blocks of order $(k + 1)$ and $(M - \ell)$ diagonal blocks of order k .

Let us now introduce an m -dimensional unit column vector $\mathbf{e}_j$ whose j th entry is 1. By considering the product $A\mathbf{e}_j$, we obtain the j th column vector of A , namely, $A\mathbf{e}_j = \mathbf{A}_{*,j}$. Moreover, by considering the product $(\mathbf{e}_i)^\top \mathbf{A}_{*,j}$, we derive the entry in the i th row of $\mathbf{A}_{*,j}$, which is simply the (i, j) th entry

of A , namely, $(\mathbf{e}_i)^\top \mathbf{A}_{*,j} = A_{i,j}$. Let $E_j := (\mathbf{e}_j \ \mathbf{e}_{M+j} \ \mathbf{e}_{2M+j} \ \cdots \ \mathbf{e}_{kM+j})$ for $j = 1, 2, \dots, \ell$. Then, by recalling that $A_{-M+j,j} = a_{-M+j}$, $A_{j,j} = b_j$, $A_{M+j,j} = c_{M+j}$ and $A_{i,j} = 0$ for $i \neq -M+j, j, M+j$, we observe that:

$$\begin{aligned}
(E_j)^\top A \mathbf{e}_j &= (E_j)^\top \mathbf{A}_{*,j} = \begin{pmatrix} A_{j,j} \\ A_{M+j,j} \\ A_{2M+j,j} \\ \vdots \\ A_{kM+j,j} \end{pmatrix} = \begin{pmatrix} b_j \\ c_{M+j} \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad j = 1, 2, \dots, \ell, \\
(E_j)^\top A \mathbf{e}_{iM+j} &= (E_j)^\top \mathbf{A}_{*,iM+j} = \begin{pmatrix} A_{j,iM+j} \\ \vdots \\ A_{(i-2)M+j,iM+j} \\ A_{(i-1)M+j,iM+j} \\ A_{iM+j,iM+j} \\ A_{(i+1)M+j,iM+j} \\ A_{(i+2)M+j,iM+j} \\ \vdots \\ A_{kM+j,iM+j} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ a_{(i-1)M+j} \\ b_{iM+j} \\ c_{(i+1)M+j} \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \\
i &= 1, 2, \dots, k-1, \quad j = 1, 2, \dots, \ell, \\
(E_j)^\top A \mathbf{e}_{kM+j} &= (E_j)^\top \mathbf{A}_{*,kM+j} = \begin{pmatrix} A_{j,kM+j} \\ \vdots \\ A_{(k-2)M+j,kM+j} \\ A_{(k-1)M+j,kM+j} \\ A_{kM+j,kM+j} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ a_{(k-1)M+j} \\ b_{kM+j} \end{pmatrix}, \\
j &= 1, 2, \dots, \ell.
\end{aligned}$$

Therefore, by noting that $(E_j)^\top A (\mathbf{e}_j \ \mathbf{e}_{M+j} \ \cdots \ \mathbf{e}_{kM+j}) = (E_j)^\top A E_j$, we find that:

$$\begin{aligned}
(E_j)^\top A E_j &= T_j, \quad j = 1, 2, \dots, \ell, \\
T_j &:= \begin{pmatrix} b_j & a_j & & & \\ c_{M+j} & b_{M+j} & a_{M+j} & & \\ & c_{2M+j} & b_{2M+j} & \ddots & \\ & & \ddots & \ddots & a_{(k-1)M+j} \\ & & & c_{kM+j} & b_{kM+j} \end{pmatrix} \in \mathbf{R}^{(k+1) \times (k+1)}.
\end{aligned}$$

Moreover, let $E_j := (\mathbf{e}_j \ \mathbf{e}_{M+j} \ \mathbf{e}_{2M+j} \ \cdots \ \mathbf{e}_{(k-1)M+j})$ for $j = \ell+1, \ell+2, \dots, M$.

Similarly, it follows that

$$(E_j)^\top A E_j = T_j, \quad j = \ell + 1, \ell + 2, \dots, M,$$

$$T_j := \begin{pmatrix} b_j & a_j & & & \\ c_{M+j} & b_{M+j} & a_{M+j} & & \\ & c_{2M+j} & b_{2M+j} & \ddots & \\ & & \ddots & \ddots & a_{(k-2)M+j} \\ & & & c_{(k-1)M+j} & b_{(k-1)M+j} \end{pmatrix} \in \mathbf{R}^{k \times k}.$$

For $i \neq j - M, j, j + M$, it holds that $(e_i)^\top A e_j = A_{i,j} = 0$. This property immediately leads to $(E_i)^\top A E_j = 0$ for $i, j = 1, 2, \dots, M$ with $i \neq j$. Thus, by letting $P := (E_1, E_2, \dots, E_\ell, E_{\ell+1}, E_{\ell+2}, \dots, E_M)$, we derive $P^\top A P = \text{diag}(T_1, T_2, \dots, T_\ell, T_{\ell+1}, T_{\ell+2}, \dots, T_M)$. Since P is a permutation matrix for A , it is clear that $P^\top = P^{-1}$. Taking this into account, we find:

$$P^{-1} A P = \text{diag}(T_1, T_2, \dots, T_\ell, T_{\ell+1}, T_{\ell+2}, \dots, T_M), \quad (4.12)$$

which implies that the similarity transformation for A using the permutation matrix P generates a block diagonal matrix with ℓ tridiagonal blocks of order $(k + 1)$ and $(M - \ell)$ blocks of order k .

4.3 An equivalent transformation

In this section, we describe the transformation from A to $\tilde{A} = S A$ in a $2M$ -stride reduction for the linear system $A \mathbf{x} = \mathbf{y}$, in terms of the 2-stride reduction with block diagonalization.

We begin by considering a linear system with the coefficient matrix given by the similarity transformation by P for $\tilde{A}$, namely, $P^{-1} \tilde{A} P = (P^{-1} S P)(P^{-1} A P)$. It is worth noting that S is also M -tridiagonal. Therefore, along the same

lines in Section 4.2, we easily derive:

$$P^{-1}SP = \text{diag}(\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_\ell, \mathcal{T}_{\ell+1}, \mathcal{T}_{\ell+2}, \dots, \mathcal{T}_M), \quad (4.13)$$

$$\mathcal{T}_j := \begin{pmatrix} 1 & \alpha_j & & & \\ \beta_{M+j} & 1 & \alpha_{M+j} & & \\ & \beta_{2M+j} & 1 & \ddots & \\ & & \ddots & \ddots & \alpha_{(k-1)M+j} \\ & & & \beta_{kM+j} & 1 \end{pmatrix}, \quad j = 1, 2, \dots, \ell,$$

$$\mathcal{T}_j := \begin{pmatrix} 1 & \alpha_j & & & \\ \beta_{M+j} & 1 & \alpha_{M+j} & & \\ & \beta_{2M+j} & 1 & \ddots & \\ & & \ddots & \ddots & \alpha_{(k-2)M+j} \\ & & & \beta_{(k-1)M+j} & 1 \end{pmatrix},$$

$$j = \ell + 1, \ell + 2, \dots, M.$$

By combining (4.12) with (4.13), we obtain:

$$P^{-1}\tilde{A}P = \text{diag}(\mathcal{T}_1T_1, \mathcal{T}_2T_2, \dots, \mathcal{T}_\ell T_\ell, \mathcal{T}_{\ell+1}T_{\ell+1}, \mathcal{T}_{\ell+2}T_{\ell+2}, \dots, \mathcal{T}_MT_M). \quad (4.14)$$

A linear system with the coefficient matrix $P^{-1}\tilde{A}P$ can be regarded as a union of M linear systems. The 2-stride reduction is applicable to each of the smaller linear systems. Let B be the coefficient matrix in the resulting linear system after 2-stride reduction for each smaller linear equation. It is remarkable that $\mathcal{T}_j$ differs from T_j only in the symbols of the tridiagonal entries. This is the same relationship that S has to A with $M = 1$ and $m = k, k + 1$. Thus, $\mathcal{T}_jT_j$ is also given by 2-stride reduction for the linear system with the coefficient matrix T_j . Using (4.14), we find $P^{-1}\tilde{A}P = B$. Therefore, we conclude that $\tilde{A}$ can be derived using the similarity transformation by P for B , namely, $\tilde{A} = PBP^{-1}$. See also Figure 4.1 for the relationship of coefficient matrices.

4.4 Extended condition numbers of coefficient matrices

In this section, with respect to extended condition numbers, we investigate an interesting property of coefficient matrices in linear systems appearing in the stride reduction method.

We consider the case where A is positive definite. It is clear that $\Lambda(A) = \Lambda(P^{-1}AP)$, where $\Lambda(\cdot)$ denotes the set of matrix eigenvalues. Using (4.12), we

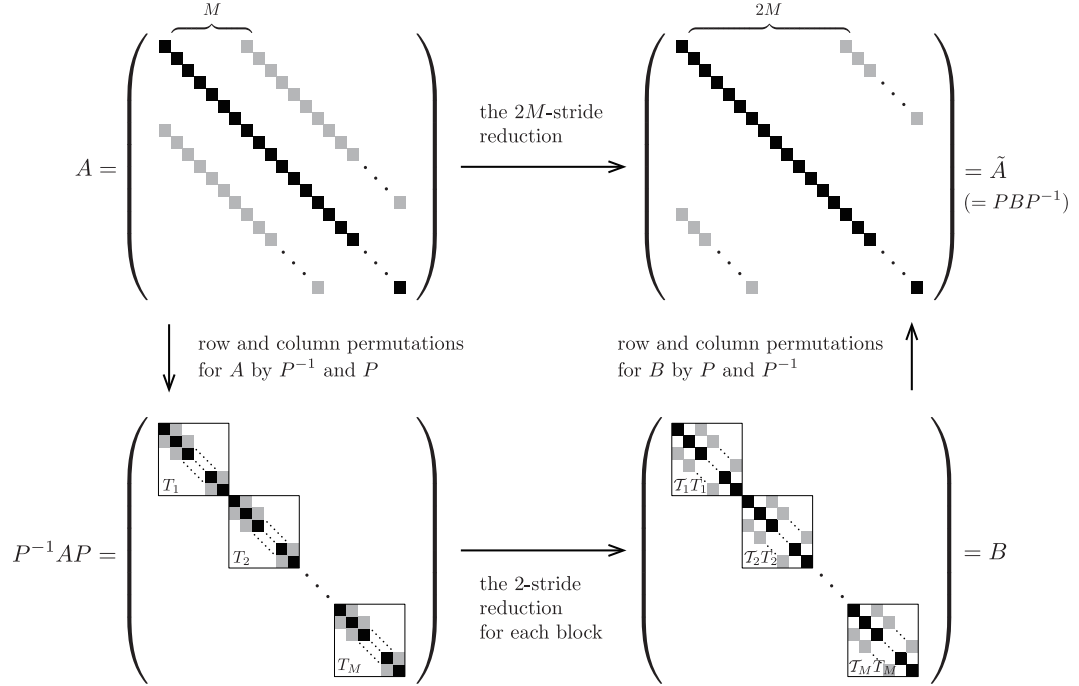


Figure 4.1: Coefficient matrices in the $2M$ -stride reduction and its equivalent transformation.

obtain:

$$\Lambda(A) = \bigcup_{j=1}^M \Lambda(T_j), \quad (4.15)$$

which implies that every T_j is also positive definite. Let us assume that all pairs of symmetrical parts in A have the same signs, namely, $a_i c_{M+i} > 0$ for $i = 1, 2, \dots, m-M$. Then, by observing the entries of every T_j , we can see that they have the same sign as appearing in the entries of A . We recall that $\mathcal{T}_j T_j$ is the coefficient matrix in the resulting linear system after 2-stride reduction of a linear system with the coefficient matrix T_j . Thus, according to Theorem 2.4 given by Wang *et al.* [38], any eigenvalues of $\mathcal{T}_j T_j$, denoted by $\lambda(\mathcal{T}_j T_j)$, satisfy

$$\min(\Lambda(T_j)) < \lambda(\mathcal{T}_j T_j) < \max(\Lambda(T_j)), \quad j = 1, 2, \dots, M, \quad (4.16)$$

where $\min(\Lambda(\cdot))$ and $\max(\Lambda(\cdot))$ denote the minimal and maximal elements of $\Lambda(\cdot)$, respectively. From (4.15) and (4.16), it follows that

$$\lambda_{\min}(A) < \lambda(\mathcal{T}_j T_j) < \lambda_{\max}(A), \quad j = 1, 2, \dots, M. \quad (4.17)$$

As shown in Section 4.3, it holds that $\tilde{A} = PBP^{-1}$ for $B = \text{diag}(\mathcal{T}_1T_1, \mathcal{T}_2T_2, \dots, \mathcal{T}_MT_M)$. This similarity immediately leads to $\Lambda(\tilde{A}) = \cup_{j=1}^M \Lambda(\mathcal{T}_jT_j)$. By combining this with (4.17), we develop the following theorem for the relationship of $\lambda(\tilde{A})$ to $\lambda(A)$.

Theorem 4.4.1. *Let $A \in \mathbf{R}^{m \times m}$ be an M -tridiagonal matrix and let $\tilde{A}$ be the $2M$ -tridiagonal matrix obtained through $2M$ -stride reduction of the linear system $A\mathbf{x} = \mathbf{y}$. If A is positive definite and has the same sign in each pair of symmetric entries, it holds that:*

$$\lambda_{\min}(A) < \lambda(\tilde{A}) < \lambda_{\max}(A). \quad (4.18)$$

In other words, the extended condition number of $\tilde{A}$ is smaller than that of A .

Here, we take notice of the entries $\tilde{a}_i$ and $\tilde{c}_{i+2M}$ for $i = 1, 2, \dots, m - 2M$ in $\tilde{A}$ under the assumption that $a_i c_{i+M} > 0$ for $i = 1, 2, \dots, m - M$ in A . By recalling that $\tilde{a}_i = -a_i a_{i+M} / b_{i+M}$ and $\tilde{c}_{i+2M} = -c_{i+M} c_{i+2M} / b_{i+M}$ for $i = 1, 2, \dots, m - 2M$, we easily derive

$$\tilde{c}_{i+2M} \tilde{a}_i = \frac{(c_{i+2M} a_{i+M})(c_{i+M} a_i)}{(b_{i+M})^2} > 0.$$

This positivity implies that Theorem 4.4.1 also holds in $4M$ -stride reduction of the linear system $\tilde{A}\mathbf{x} = \tilde{\mathbf{y}}$. Moreover, it is clear from (4.7) that the symmetry of $\tilde{A}$ follows that of A in the $2M$ -stride reduction of the linear system $A\mathbf{x} = \mathbf{y}$. Thus, in the sequence of stride reductions, the ratios of maximum to minimum eigenvalues of the coefficient matrices are simply the condition numbers of the matrices if A is symmetric. Therefore, the condition number or the extended condition number of a coefficient matrix monotonously decreases when stride reductions are recursively applied to the linear system $A\mathbf{x} = \mathbf{y}$, if A is a tridiagonal positive-definite matrix with $a_i c_{i+M} > 0$ for $i = 1, 2, \dots, m - M$.

We present two examples by which we numerically verify the condition numbers or the extended condition numbers of coefficient matrices in the stride reduction method. In an original linear system $A\mathbf{x} = \mathbf{y}$, let us consider two cases where A is a symmetric tridiagonal matrix of order 59 with $a_i = c_i = -15.03759398496241$ and $b_i = 31.07518796992481$, and where A is a 5-tridiagonal matrix of order 100 with $a_i = -4$, $b_i = 8$, and $c_i = 1$. The former case appears in implicit schemes for solving models of one-dimensional heat conduction with respect to homogeneous materials [41] and the latter is an example given by Jia *et al.* [21]. Moreover, let $A^{(n)}$ denote the coefficient matrix of the resulting linear system after stride reductions have been performed n

Table 4.1: Condition numbers of coefficient matrices after stride reductions of an original linear system with a symmetric tridiagonal coefficient matrix.

Iteration number n	Matrix form of $A^{(n)}$	Condition number of $A^{(n)}$
0	tridiagonal	58.69012794456805
1	2-tridiagonal	15.17679164630569
2	4-tridiagonal	5.616092489770868
3	8-tridiagonal	3.162913589084670
4	16-tridiagonal	2.544195550363177

Table 4.2: Extended condition numbers of coefficient matrices after stride reductions of an original linear system with a 5-tridiagonal coefficient matrix.

Iteration number n	Matrix form of $A^{(n)}$	Extended condition number of $A^{(n)}$
0	5-tridiagonal	2.955816794797333
1	10-tridiagonal	1.321685342470923
2	20-tridiagonal	1.087052791446233
3	40-tridiagonal	1.077378827066963
4	80-tridiagonal	1.077350269185423

times on the original linear system $A\mathbf{x} = \mathbf{y}$ where $A^{(0)} = A$. Tables 4.1 and 4.2 show the matrix forms and condition numbers of the former $A^{(0)}$, $A^{(1)}$, $A^{(2)}$, $A^{(3)}$, $A^{(4)}$ and the matrix forms and extended condition numbers of the latter ones, respectively. From Tables 4.1 and 4.2, we verify that, as n grew larger, the nonzero off-diagonal entries of $A^{(n)}$ move away from the diagonal entries, and the condition numbers or the extended condition numbers of $A^{(n)}$ became smaller.

Chapter 5

Concluding remarks

The main results we have through the thesis, research associated with them and prospect of future work are summarized as follows.

In Chapter 2 [A1], the m2dLVs transformation which is algebraically equivalent to the mdLVs transformation is introduced and we have shown that the m2dLVs transformation is numerically stable. Also, we have estimated the relative perturbations of the singular values and obtained an upper bound of the forward errors, in the m2dLVs transformation. After seeing the some results in [A1], Takata *et al.* has presented a new shift strategy in the m2dLVs algorithm, showed that the new algorithm accurately computes singular values of bidiagonal matrices and given an example which shows the m2dLVs algorithm more accurately computes singular values than the dqds algorithm [37].

In Chapter 3 [A2], we have derived a new transformation, the dLV_∞ similarity transformation from the dLV system as the limit $\delta^{(n)} \rightarrow \infty$, and shown that the dLV_∞ similarity transformation is applicable to singular value computation for bidiagonal matrices. Then, we have shown numerical stability and forward stability of the dLV_∞ similarity transformation, and estimated the relative perturbations of singular values and the forward errors of the variables appearing in the dLV_∞ similarity transformation. Since computational complexity of the dLV_∞ similarity transformation is less than that of the original one, from the point of view that numerical errors accumulate, the perturbations of singular values in the dLV_∞ similarity transformation is slightly smaller than those in the original one in the absence of the overflow and the underflow. Actually, we have given some numerical results which show that the dLV_∞ similarity transformation is slightly better with respect to the accuracy of the computed singular values than the original one. However, we also have obtained some numerical results which show that the dLV_∞ similarity transformation is better with respect to the accuracy of the computed singular

values than the dqd transformation though the dLV_∞ similarity transformation is algebraically equivalent to the dqd transformation. The author might work on this reason in future. Also, in order to accelerate the convergence, the author will introduce a shift of origin into the dLV_∞ similarity transformation and confirm its numerical stability.

In Chapter 4 [A3], we have shown that a preconditioned linear system $\tilde{A}\mathbf{x} = \tilde{\mathbf{y}}$ is derived from the original system $A\mathbf{x} = \mathbf{y}$ by $2M$ -stride reduction, and the extended condition number of $\tilde{A}$ becomes smaller than that of A if A is positive-definite. It implies that the condition of the precondition system is improved than that of the original system by $2M$ -stride reduction. In addition, since M -tridiagonal matrix is block diagonalized by the appropriate symmetric permutation and all of the M block matrices are tridiagonal, we can apply the symmetric permutation to parallel computation for solving each block linear system. $2M$ -stride reduction is useful for increasing the number of block matrices. Moreover, from the main results in Chapter 4, we see that S shown in (4.10) and the permutation matrix P shown in Section 4.2 are applicable to the left preconditioner M_L and the right preconditioner M_R to the original system (1.3). More precisely, in matrix expression, the preconditioned system is written as follows.

$$(P_N)^{-1}S_N \cdots S_2S_1AP_N((P_N)^{-1}\mathbf{x}) = \mathbf{0},$$

where S_k is nonsingular matrix which operates 2^kM -stride reduction for $k = 1, 2, \dots, N$ and P_N is the permutation matrix which block diagonalize $2^N M$ -tridiagonal matrices by symmetric permutation. Therefore, we can expect that the computed singular vectors become more accurate by applying the MR³ algorithm or the twisted factorization based on the dLV type transformation to the preconditioned system.

However, seen from the point of view that accuracy of the MR³ algorithm depends on the relative gaps of $\tilde{A}$, we need to discuss the relation between the distribution of eigenvalues of A and that of $\tilde{A}$. Moreover, in singular vector computation, in general, it does not guarantee that the coefficient matrix of the linear system (1.3) is positive-definite. Hence, we also need to confirm the behavior of the condition number in $2M$ -stride reduction in the case when A is not positive-definite, in order to estimate the condition of the preconditioned system. The author will treat it and the behavior of the distribution of eigenvalues and the condition number in $2M$ -stride reduction, in future work.

Acknowledgements

The author would like to express his special gratitude to his supervisor, Professor Yoshimasa Nakamura, for his helpful advice and various instruction. The author would also like to acknowledge Professor Masashi Iwasaki of Kyoto Prefectural University for his strategic advice and continuous encouragement. Additionally, the author would like to appreciate lots of knowledge of numerical algorithms and their numerical behavior told by Professor Kinji Kimura. Moreover, the author thanks Professor Hiroto Sekido for telling his critical and insightful view to advance Section 4.3. Finally, the author is grateful for the whole members of research group of Applied Mathematical Analysis Laboratory for many helpful discussions and suggestions.

Bibliography

- [1] M.I. Bueno, F.M. Dopico, A more accurate algorithm for computing the Christoffel transformation, *J. Comput. Appl. Math.* **205** (2007) 567–582.
- [2] M.I. Bueno, F.M. Dopico, Stability and sensitivity of Darboux transformation without parameter, *Elec. Trans. Numer. Anal.* **18** (2004) 101–136.
- [3] M.I. Bueno, F. Marcellan, Darboux transformation and perturbation of linear functionals, *Linear Algebra Appl.* **384** (2004) 215–242.
- [4] B.L. Buzbee, G.H. Golub, C.W. Nielson, On direct methods for solving Poisson’s equation, *SIAM J. Numer. Anal.* **7** (1970) 627–656.
- [5] F. Chaitin-Chatelin, V. Frayssé, *Lectures on Finite Precision Computations*, SIAM, Philadelphia, 1996.
- [6] M.T. Chu, Linear algebra algorithms as dynamical systems, *Acta Numer.* **17** (2008) 1–86.
- [7] J. Demmel, W. Kahan, Accurate singular values of bidiagonal matrices, *SIAM J. Sci. Stat. Comput.* **11** (1990) 873–912.
- [8] I.S. Dhillon, B.N. Parlett, Orthogonal Eigenvectors and relative gaps, *SIAM J. Matrix Anal. Appl.* **25** (2004) 858–899.
- [9] B. Einarsson, *Accuracy and Reliability in Scientific Computing*, SIAM Philadelphia, 2005.
- [10] J.D. Evans, Cyclic and stride reduction methods for generalised tridiagonal matrices, *Int. J. Comput. Math.* **73** (2000) 487–492.
- [11] K.V. Fernando, B.N. Parlett, Accurate singular values and differential qd algorithms, *Numer. Math.* **67** (1994) 191–229.
- [12] W. Gander, G.H. Golub, *Cyclic reduction - history and applications*, Proc. Workshop Sci. Computing, 1997.

- [13] G.H. Golub, C.F. Van Loan, *Matrix Computations*, 4th ed., Johns Hopkins, 2012.
- [14] N.J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.
- [15] R. Hirota, Conserved quantities of a random-time Toda equation, *J. Phys. Soc. Japan* **66** (1997) 283–284.
- [16] M. Iwasaki, Y. Nakamura, On the convergence of a solution of the discrete Lotka-Volterra system, *Inverse Probl.* **18** (2002) 1569–1578.
- [17] M. Iwasaki, Y. Nakamura, An application of the discrete Lotka-Volterra system with variable step-size to singular value computation, *Inverse Probl.* **20** (2004) 553–563.
- [18] M. Iwasaki, S. Sakano, Y. Nakamura, Accurate twisted factorization of real symmetric tridiagonal matrices and its application to singular value decomposition (in Japanese), *Transactions of the Japan Society for Industrial and Applied Mathematics* **15** (2005) 461–481.
- [19] M. Iwasaki, Y. Nakamura, Accurate computation of singular values in terms of shifted integrable schemes, *Japan J. Indust. Appl. Math.* **23** (2006) 239–259.
- [20] M. Iwasaki, Y. Nakamura, Positivity of dLV and mdLVs algorithms for computing singular values, *Elec. Trans. Numer. Anal.* **38** (2011) 184–201.
- [21] J.T. Jia, T. Sogabe, M.E.A. El-Mikkawy, Inversion of k -tridiagonal matrices with Toeplitz structure, *Comput. Math. Appl.* **65** (2013) 116–125.
- [22] F. Jiang, R. Kannan, M. L. Littman, S. Vempala, *Efficient singular value decomposition via improved document sampling*, Dept. of Computer Science, Duke University, 1999.
- [23] D.S. Kershaw, The incomplete Cholesky-conjugate gradient method for the iterative solution of systems of linear equations, *J. Comput. Phys.* **26** (1978) 43–65.
- [24] G. D. Luenberger, *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, New York, 1973.

- [25] J.A. Meijerinka, H.A. van der Vorst, An Iterative Solution Method for Linear Systems of which the Coefficient Matrix is a Symmetric M-Matrix, *Math. Comput.* **31** (1977) 148–162.
- [26] Y. Nakamura, *Functionality of Integrable System (in Japanese)*, Kyoritsu Publishing, Tokyo, 2006.
- [27] B.N. Parlett, The new qd algorithm, *Acta Numerica* **4** (1995) 459–491.
- [28] B. N. Parlett, I. S. Dhillon, Fernando’s solution to Wilkinson’s problem: An application of double factorization, *Lin. Alg. Appl.* **267** (1997) 247–279.
- [29] W. K. Pratt, *Digital image processing*, New York, USA, Wiley-Interscience publishing, 1978.
- [30] H. Rutishauser, *Lectures on Numerical Mathematics*, Brinkhäuser, Boston, 1990.
- [31] Y. Saad, *Iterative Methods for Sparse Linear Systems, 2nd ed.*, SIAM, 2003.
- [32] A. van der Sluis, H.A. van der Vorst, The rate of convergence of conjugate gradients, *Numer. Math.* **48** (1986) 543–560.
- [33] T. Sogabe, M.E.A. El-Mikkawy, Fast block diagonalization of k -tridiagonal matrices, *Appl. Math. Comput.* **218** (2011) 2740–2743.
- [34] R.A. Sweet, A generalized cyclic reduction algorithm, *SIAM J. Numer. Anal.* **11** (1974) 506–520.
- [35] M. Takata, K. Kimura, M. Iwasaki, Y. Nakamura, Performance of a new scheme for bidiagonal singular value decomposition of large scale, *Proceedings of IASTED International Conference on Parallel and Distributed Computing and Networks* (2006) 304–309.
- [36] M. Takata, K. Kimura, M. Iwasaki, Y. Nakamura, Implementation of library for high speed singular value decomposition, *Journal of Information Processing Society of Japan* **47** No. SIG7(ACS 14) (2006) 91–104.
- [37] M. Takata, H. Toyokawa, H. Ishigami, K. Kimura, T. Yamashita, M. Iwasaki, Y. Nakamura, New shift strategy in the dqds algorithm and the m2dLVs algorithm for singular values (in Japanese), *IPSJ Transactions on Advanced Computing Systems* **6** (2013) 94–107.

- [38] T. Wang, M. Iwasaki, Y. Nakamura, On condition numbers in the cyclic reduction processes of a tridiagonal matrix, *Int. J. Comput. Math.* **87** (2010) 3079–3093.
- [39] J. H. Wilkinson, *Rounding Errors in Algebraic Processes*, Prentice-Hall, 1963.
- [40] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, 1965.
- [41] M. Žecová, J. Terpák, Heat conduction modeling by using fractional-order derivatives, *Appl. Math. Comput.* **257** (2015) 365–373.
- [42] V. Spiridonov, A. Zhedanov, Discrete-time Volterra chain and classical orthogonal polynomials, *J. Phys. A.:Math. Gen.* **30** (1997) 8727–8737.
- [43] LAPACK : <http://www.netlib.org/lapack/>

List of Author's Papers and Works Related to the Thesis

- [A1] M. Nagata, M. Iwasaki, Y. Nakamura, Error analysis of the mdLVs algorithm for computing bidiagonal singular values, *Numer. Algor.* **61** (2012) 261–274. The final publication is available at Springer via <http://dx.doi.org/10.1007/s11075-012-9607-5>.
- [A2] M. Nagata, M. Iwasaki, Y. Nakamura, Accurate similarity transformation derived from the discrete Lotka-Volterra system for bidiagonal singular values, *Calcolo* **51** (2014) 305–317. The final publication is available at Springer via <http://dx.doi.org/10.1007/s10092-013-0085-5>.
- [A3] M. Nagata, M. Hada, M. Iwasaki, Y. Nakamura, Eigenvalue clustering of coefficient matrices in the iterative stride reductions for linear systems, *Compt. Math. Appl.* **71** (2016) 349–355. The final publication is available at Elsevier via <http://dx.doi.org/10.1016/j.camwa.2015.11.022>.