Stability and sensitivity of tridiagonal LU factorization without pivoting

M. ISABEL BUENO and FROILÁN M. DOPICO *

Department of Mathematics, Universidad Carlos III de Madrid, Avda. de la Universidad, 30. 28911 Leganés, Spain. emails: mbueno@math.uc3m.es, dopico@math.uc3m.es

Abstract.

In this paper the accuracy of LU factorization of tridiagonal matrices without pivoting is considered. Two types of componentwise condition numbers for the L and U factors of tridiadonal matrices are presented and compared. One type is a condition number with respect to small relative perturbations of each entry of the matrix. The other type is a condition number with respect to small componentwise perturbations of the kind appearing in the backward error analysis of the usual algorithm for the LU factorization. We show that both condition numbers are of similar magnitude. This means that the algorithm is componentwise forward stable, i.e., the forward errors are of similar magnitude to those produced by a componentwise backward stable method. Moreover the presented condition numbers can be computed in $O(n)$ flops, which allows to estimate with low cost the forward errors.

AMS subject classification: 65F35, 65F50, 15A12, 15A23, 65G50.

Key words: tridiagonal matrices, LU factorization, condition numbers, error analysis.

1 Introduction.

The LU factorization is one of the most important matrix factorizations appearing in Numerical Analysis [11]. Traditionally, the LU factorization has been used to solve linear systems of equations, while in solving spectral problems orthogonal factorizations have been preferred because of their excellent stability properties [11]. However in the last decade the LU factorization has been employed to solve structured spectral problems [5], [13]. For most of the applications related to the solution of linear systems it is the backward error and not the forward error of the LU factorization that matters, and for the application of LU to the computation of singular value decomposition with high relative accuracy what is needed is to compute the LU factors with small forward errors [5]. The question of how large are the forward errors may be answered by combining backward errors with an adequate perturbation theory for the LU factorization.

*This research has been partially supported by the Ministerio de Ciencia y Tecnología of Spain through grants BFM2003-06335-C03-02 (M. I. Bueno) and BFM2000-0008 (F. M. Dopico).

This paper presents a highly structured componentwise perturbation theory for the LU factorization of tridiagonal matrices without pivoting. Although the use of pivoting strategies is common to stabilize the usual Gaussian algorithm for the LU factorization, in the case of tridiagonal matrices it may be necessary to preserve the structure of the problem in some settings. For instance to solve the nonsymmetric tridiagonal eigenvalue problem with *qd* algorithms [13], or in some problems related to orthogonal polynomials [8, 9, 10, 2]. This prevents the use of pivoting.

The sensitivity of the LU factorization of general matrices has been studied by other authors. A normwise analysis was presented by Barrlund [1], a componentwise analysis was given by Sun [16], and a first-order perturbation expansion for the LU factorization along with bounds on the second order terms was introduced by Stewart [14, 15]. These authors obtained upper bounds for the perturbation of the LU factors. Expressions for normwise condition numbers, i.e., optimum first order upper perturbation bounds, of the LU factors have been presented by Chang and Page [3]. In this paper we improve and extend previous results in the case of tridiagonal matrices by taking advantage of the structure of the problem. Moreover, for the first time in this problem, we deal with two different types of componentwise perturbations of the tridiagonal matrix: small relative perturbations of each entry of the matrix, and small entrywise perturbations of the kind appearing in the backward error analysis of the usual algorithm for the LU factorization. We show that the condition numbers with respect these two types of perturbations are of similar magnitude, which implies that the usual algorithm to compute the LU factorization is *forward stable* according to the definition appearing in [12, p. 9]. We find explicit expressions of these condition numbers which can be computed in $6n$ flops for an $n \times n$ matrix.

The reference [4, §4.2] does not deal with the LU factorization, but it is related to this work because it deals with the conditioning of Cholesky factorization of tridiagonal positive definite matrices. In particular, the normwise condition numbers under general and tridiagonal perturbations are shown to be of similar magnitude. The differences with this paper are that LU factorization is not considered, componentwise condition numbers are not analyzed, and that we compare two types of structured condition numbers. The results of this paper can be extended to normwise condition numbers and to the Cholesky factorization. We will not undertake this task to keep the paper concise.

The paper is organized as follows: In Section 2 the notation used throughout the paper is introduced, along with Algorithm 2.1, the classical tridiagonal LU algorithm. In Section 3, we present two theorems concerning backward errors of Algorithm 2.1: the first one, Theorem 3.1, is an slight improvement of the usual backward result for tridiagonal LU [12, § 9.6]; the second one, Theorem 3.2, is a mixed forward-backward error result, i.e., a theorem in which both input and output have to be perturbed to get an exact LU factorization. This latter result is new, as far as we know, but similar results have been proved for the symmetric LDL^T factorization [6, Theorem 4.4.5], or for one step of the *dqds* algorithm [7, 13]. In Section 4, we analyze the relative componentwise change of the LU

factors under the two kinds of componentwise perturbations that we consider. In consequence two condition numbers of the tridiagonal LU factorization are defined (Definition 4.1), expressed in a explicit computable way (see Theorem 4.7 and Definitions 4.2 and 4.3), shown to be of similar magnitude (see Theorem 4.9), and, finally, proved to be invariant under multiplication by diagonal matrices (Theorem 4.10). Section 5 runs parallel to Section 4, the difference is that the change of the LU factors is measured in norm (see Definition 5.1, and Theorems 5.3 and 5.4), although we consider the same two kinds of componentwise perturbations as in the previous section. In this case the two condition numbers are also of the same magnitude, but they are not invariant under diagonal scalings. Sections 4 and 5 contain the most important results of the paper. In Section 6 an important class of tridiagonal matrices for which the LU factorization can be computed with small componentwise forward and backward errors is considered. The paper finish with some numerical experiments in Section 7.

2 Notation and Algorithm.

Let us consider the tridiagonal $n \times n$ matrix,

$$T = \begin{bmatrix} a_1 & b_1 & 0 & \cdots & 0 \\ c_1 & a_2 & b_2 & \cdots & 0 \\ 0 & c_2 & a_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & c_{n-1} & a_n \end{bmatrix}.$$

For the sake of simplicity, we frequently use the following notation

$$\text{tridiag}[c, a, b] := T,$$

$$c = [c_1, ..., c_{n-1}]^T,\ a = [a_1, ..., a_n]^T,\ b = [b_1, ..., b_{n-1}]^T.$$

The first $n-1$ leading principal submatrices of T are nonsingular if and only if T has a unique LU factorization $T = LU$ where

$$L = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ l_1 & 1 & \cdots & 0 & 0 \\ 0 & l_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & l_{n-1} & 1 \end{bmatrix}, \quad U = \begin{bmatrix} u_1 & b_1 & 0 & \cdots & 0 \\ 0 & u_2 & b_2 & \cdots & 0 \\ 0 & 0 & u_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & u_n \end{bmatrix}.$$

We will also use a simplified notation for the bidiagonal matrices L and U:

$$\text{bidiagl}[l, ones] := L, \quad \text{bidiagu}[u, b] := U,$$

$$l = [l_1, ..., l_{n-1}]^T, \quad u = [u_1, ..., u_n]^T, \quad b = [b_1, ..., b_{n-1}]^T.$$

The MATLAB code that computes the matrices L and U is:

ALGORITHM 2.1. *Given the tridiagonal matrix* $T = \text{tridiag}[c, a, b]$, *this algorithm computes the LU factorization without pivoting of* T.

```
u(1)=a(1)
for i=1:n-1
   l(i)=c(i)/u(i)
   u(i+1)=a(i+1)-l(i)*b(i)
end
```

The computational cost of Algorithm 2.1 is $3(n-1)$ flops.

We use the conventional error model of floating point arithmetic:

$$\mathrm{fl}(a \odot b) = (a \odot b)(1+\delta) = \frac{a \odot b}{1+\eta}, \quad |\delta| \leq \mathbf{u},\ |\eta| \leq \mathbf{u},$$

where a and b are floating point numbers, $\odot \in \{+, -, \times, /\}$, and $\mathbf{u}$ is the unit roundoff of the machine. We will assume the absence of overflow, underflow, or division by zero.

Finally, the entrywise absolute values of a matrix A or a vector b are denoted by $|A|$ or $|b|$. And the norm $\|A\|$ of a matrix A denotes the "max norm": $\|A\| = \max_{ij} |a_{ij}|$. It is well known that this norm is not consistent, but for sparse matrices it is a simple and proper choice.

3 Backward Error Analysis.

In this section we assume that the elements of the matrix T are real floating point numbers. The same results hold for complex matrices T at the cost of increasing the bounds by a small integer factor [12, § 3.6]. The following Theorem improves slighty the backward error analysis presented in [12, § 9.6] by using that the entries of U in the positions $(i, i+1)$ are just b_i. Moreover, this Theorem remarks that the relative componentwise backward error on the positions $(i+1, i)$ is bounded just by $\mathbf{u}$.

THEOREM 3.1. *If Algorithm 2.1 is applied to the tridiagonal $n \times n$ matrix $T = \mathrm{tridiag}[c, a, b]$ then the computed LU factors, $\widehat{L} = \mathrm{bidiagl}[\hat{l}, ones]$ and $\widehat{U} = \mathrm{bidiagu}[\hat{u}, b]$, satisfy*

$$\mathrm{tridiag}[c + \Delta c, a + \Delta a, b] = \widehat{L}\widehat{U}, \quad |\Delta c| \leq \mathbf{u}\,|c|,\ |\Delta a| \leq \mathbf{u}\,\mathrm{diag}(|\widehat{L}||\widehat{U}|),$$

where $\mathrm{diag}(|\widehat{L}||\widehat{U}|)$ *denotes the main diagonal of* $|\hat{L}||\hat{U}|$.

PROOF. For the computed quantities, we have

$$\hat{l}_i = \frac{c_i}{\hat{u}_i}(1+\varepsilon_i), \quad |\varepsilon_i| \leq \mathbf{u}.$$

Hence $|c_i - \hat{u}_i \hat{l}_i| \leq |c_i|\mathbf{u}$, which proves the theorem for the entries $(i+1, i)$.

Moreover,

$$\hat{u}_{i+1}(1+\delta_i) = a_{i+1} - \hat{l}_i b_i (1+\eta_i), \quad |\delta_i| \leq \mathbf{u},\ |\eta_i| \leq \mathbf{u}.$$

Hence $|a_{i+1} - \hat{u}_{i+1} - \hat{l}_i b_i| \leq (|\hat{u}_{i+1}| + |\hat{l}_i b_i|)\mathbf{u}$, which proves the theorem. □

As the usual result in LU factorization, the previous theorem does not imply the ideal result $|\Delta c| \leq \mathbf{u}\,|c|$, $|\Delta a| \leq \mathbf{u}\,|a|$ (see [12, § 9.6], for special types of matrices for which these ideal relations hold). However, a result of this type can be obtained if the output of Algorithm 2.1 is also perturbed. This is done in Theorem 3.2 which proves that Algorithm 2.1 is *mixed forward-backward stable* or, in the terminology of [12, p. 7], *numerically stable*. Notice that Algorithm 2.1 uses as input $\{c, a, b\}$, the three diagonals of the tridiagonal matrix T, and produces the floating point output $\{\hat{u}, \hat{l}\}$. In Theorem 3.2, we introduce three ideal vectors $\tilde{c}$, $\tilde{\hat{u}}$ and $\tilde{\hat{l}}$, such that in exact arithmetic Algorithm 2.1 maps $\{\tilde{c}, a, b\}$ into $\{\tilde{\hat{u}}, \tilde{\hat{l}}\}$. Then, we prove that $\tilde{c}$, $\tilde{\hat{u}}$ and $\tilde{\hat{l}}$ are componentwise tiny relative perturbations of, respectively, c, $\hat{u}$ and $\hat{l}$.

THEOREM 3.2. *Let* $\widehat{L} = \text{bidiagl}[\hat{l}, ones]$ *and* $\widehat{U} = \text{bidiagu}[\hat{u}, b]$ *be the LU factors computed by Algorithm 2.1 applied to the tridiagonal* $n \times n$ *matrix* $T = \text{tridiag}[c, a, b]$, *then the following diagram commutes:*

$$\begin{array}{ccc} \{c, a, b\} & \xrightarrow{Computed\ LU} & \{\hat{u}, \hat{l}\} \\ {\scriptstyle Relative\ change\ 3\mathbf{u}\ in\ c}\downarrow & & \uparrow{\scriptstyle Relative\ change\ \mathbf{u}\ in\ \tilde{\hat{u}}\ and\ \tilde{\hat{l}}} \\ \{\tilde{c}, a, b\} & \xrightarrow{exact\ LU} & \{\tilde{\hat{u}}, \tilde{\hat{l}}\} \end{array}$$

Where, for all i, $\tilde{c}_i$ *is obtained from* c_i *by a relative change smaller than* $3\mathbf{u}$, *and* $\tilde{\hat{u}}_i$ *(resp.* $\tilde{\hat{l}}_i$*) is obtained from* $\hat{u}_i$ *(resp.* $\hat{l}_i$*) by a relative change smaller than* $\mathbf{u}$.

REMARK 3.1. *In this theorem,* $O(\mathbf{u}^2)$ *terms are ignored for simplicity.*

PROOF. The computed quantities satisfy

$$\hat{l}_i = \frac{c_i}{\hat{u}_i(1+\epsilon_i)}, \quad |\epsilon_i| \leq \mathbf{u}, \tag{3.1}$$

$$\hat{u}_{i+1}(1+\delta_i) = a_{i+1} - b_i\hat{l}_i(1+\eta_i), \quad |\delta_i| \leq \mathbf{u}, \quad |\eta_i| \leq \mathbf{u}. \tag{3.2}$$

By defining

$$\tilde{c}_i := c_i\frac{(1+\delta_{i-1})(1+\eta_i)}{1+\epsilon_i},$$

$$\tilde{\hat{l}}_i := \hat{l}_i(1+\eta_i) \quad \text{and} \quad \tilde{\hat{u}}_i := \hat{u}_i(1+\delta_{i-1}),$$

the following exact relations follows from (3.1) and (3.2)

$$\tilde{\hat{l}}_i = \frac{\tilde{c}_i}{\tilde{\hat{u}}_i}, \quad \text{and} \quad \tilde{\hat{u}}_{i+1} = a_{i+1} - b_i\tilde{\hat{l}}_i.$$

□

Theorem 3.2 can be rewritten in a more familiar way which shows that Algorithm 2.1 is componentwise stable in the mixed forward-backward sense or just stable:

$$T + \Delta T = (\hat{L} + \Delta\hat{L})(\hat{U} + \Delta\hat{U}), \quad |\Delta T| \leq 3\mathbf{u}\,|T|,\ |\Delta\hat{L}| \leq \mathbf{u}\,|\hat{L}|,\ |\Delta\hat{U}| \leq \mathbf{u}\,|\hat{U}|.$$

Theorem 3.2 implies that although Algorithm 2.1 is not backward stable as shown by Theorem 3.1, it produces outputs with errors of similar magnitudes to those produced by a backward stable method, i.e., Algorithm 2.1 is *forward stable* according to [12, p. 9]. However, we cannot still estimate the magnitude of the forward errors. To do this it is necessary to multiply the backward error by a proper *condition number*.

4 Conditioning: components vs. components.

One of the most useful rules of thumb in Numerical Linear Algebra says that the forward error produced by an algorithm can be bounded by the backward error times the condition number. In the previous section we have analyzed the backward errors, now we will study the sensitivity of tridiagonal LU factorization without pivoting under perturbations of the tridiagonal matrix $T = \text{tridiag}[c, a, b]$. We will consider two kinds of perturbations:

- Perturbations associated with the backward error found in Theorem 3.1. This implies that the vector b is not perturbed.
- Relative componentwise perturbations in c and a, i.e., $|\Delta c| \leq \epsilon |c|$ and $|\Delta a| \leq \epsilon |a|$ with small ϵ, and unperturbed b. The cause of this is that Theorems 3.1 and 3.2 keep b fixed[1].

The sensitivity of a problem is measured by the notion of condition number: the ratio between the relative change in the solution and the relative change in the data. In our case, the backward error analysis motivates to measure componentwise the change in the data (matrix T), but we can measure the relative change of the LU factors component or normwise. We will use components in this section and a norm in the next one.

We consider two different kinds of perturbations, therefore we also define two different condition numbers:

DEFINITION 4.1. *Let*

$$T = \text{tridiag}[c, a, b] = \text{bidiagl}[l, ones]\,\text{bidiagu}[u, b] = LU$$

and

$$\text{tridiag}[c + \Delta c, a + \Delta a, b] = \text{bidiagl}[l + \Delta l, ones]\,\text{bidiagu}[u + \Delta u, b]$$

be the unique LU factorizations of two $n \times n$ tridiagonal matrices. We define the condition numbers

$$cond_B(T) := \lim_{\epsilon \to 0} \sup \left\{ \max_k \left\{ \frac{|\Delta u_k|}{\epsilon |u_k|}, \frac{|\Delta l_k|}{\epsilon |l_k|} \right\} : |\Delta a| \leq \epsilon\, \text{diag}(|L||U|),\ |\Delta c| \leq \epsilon |c| \right\}$$

[1] At the cost of complicating somewhat the analysis, it is possible to consider perturbations $|\Delta b| \leq \epsilon |b|$ of the vector b. This can increase the condition numbers we are going to study by a factor 2 at most. Notice that if T is a matrix of real numbers that has to be stored in a computer, then roundoff errors of magnitude $\mathbf{u}$ appear in b.

and

$$cond_C(T) := \lim_{\epsilon \to 0} \sup \left\{ \max_k \left\{ \frac{|\Delta u_k|}{\epsilon |u_k|}, \frac{|\Delta l_k|}{\epsilon |l_k|} \right\} \, : \, |\Delta a| \leq \epsilon \, |a|, \, |\Delta c| \leq \epsilon \, |c| \right\},$$

where any quotient $x/0$ is interpreted as zero if $x = 0$ and infinity otherwise.

REMARK 4.1. *It should be understood that in the previous definition*

$$\max_k \left\{ \frac{|\Delta u_k|}{\epsilon |u_k|}, \frac{|\Delta l_k|}{\epsilon |l_k|} \right\} = \max \left\{ \frac{|\Delta u_1|}{\epsilon |u_1|}, \ldots, \frac{|\Delta u_n|}{\epsilon |u_n|}, \frac{|\Delta l_1|}{\epsilon |l_1|}, \ldots, \frac{|\Delta l_{n-1}|}{\epsilon |l_{n-1}|} \right\}.$$

This shorthand notation will be frequently used in the rest of the paper.

REMARK 4.2. *The assumption that T has a unique LU factorization implies $u_k \neq 0$, for $k = 1 : n-1$. Notice that $l_k = 0$ if and only if $c_k = 0$, which implies $\Delta c_k = 0$, $l_k + \Delta l_k = 0$ and $\Delta l_k = 0$. This fact complicates somewhat the expressions we will obtain for the condition numbers. The convention that any quotient $x/0$ is interpreted as zero if $x = 0$ and infinity otherwise is followed throughout this section.*

REMARK 4.3. *It is clear from Definition 4.1 how to define separate condition numbers for L and U. It will also be clear from the following developments how to get explicit expressions and how to compute these separate condition numbers.*

In the condition numbers defined in Definition 4.1 the B in $cond_B$ stands for "backward error", and the C in $cond_C$ stands for "components". Notice that $cond_B(T)$ and $cond_C(T)$ are both "local" condition numbers that can be used to estimate the forward errors to first order in ϵ. Thus, $\mathbf{u}\, cond_B(T)$ is by Theorem 3.1 a first order upper bound for the maximum relative error in components of the output of Algorithm 2.1. The same essentially holds for $\mathbf{u}\, cond_C(T)$ by Theorem 3.2, because the perturbation in the output appearing in this Theorem changes at most the last digit. The previous remark suggests, although does not prove, that $cond_B(T)$ and $cond_C(T)$ should be of similar magnitude, otherwise one of the previous error bounds would be much larger than the other. This will be proved in Theorem 4.9. Finally, notice that $|\Delta a| \leq \mathbf{u}\, \mathrm{diag}(|\widehat{L}||\widehat{U}|)$ appears in Theorem 3.1, while $|\Delta a| \leq \epsilon\, \mathrm{diag}(|L||U|)$ appears in the definition of $cond_B(T)$, this makes no difference because in this definition $\epsilon \to 0$.

This section is organized as follows: In subsection 4.1 we will give some auxiliary perturbation results. In subsection 4.2 the individual condition numbers for l_k and u_k are introduced. This will lead us, in subsection 4.3, to the explicit expression of both condition numbers, $cond_B(T)$ and $cond_C(T)$, as well as, to prove that they are of similar magnitudes. This and the fact that the condition numbers can be computed in $6n$ flops are some of the most important results in this paper. In subsection 4.4 we will prove that both condition numbers are invariant under diagonal transformations.

4.1 Auxiliary results.

This section contains some lemmas that are necessary to prove the main results of the paper. Readers who are not interested in technical details can skip straight to subsection 4.3, where the main results are presented.

The notation introduced in Definition 4.1 will be used throughout this section. The first goal is to get expressions for the components of the vectors Δl and Δu in function of the components of Δa and Δc. Taking into account that the condition numbers are defined in the limit $\epsilon \to 0$, second order terms in ϵ are not considered. It should be remembered that:

$$u_1 + \Delta u_1 = a_1 + \Delta a_1, \tag{4.1}$$

$$l_i + \Delta l_i = \frac{c_i + \Delta c_i}{u_i + \Delta u_i}, \quad i = 1 : n-1, \tag{4.2}$$

$$u_{i+1} + \Delta u_{i+1} = a_{i+1} + \Delta a_{i+1} - (l_i + \Delta l_i) b_i, \quad i = 1 : n-1. \tag{4.3}$$

These expressions are just the Algorithm 2.1 for the perturbed matrix.

REMARK 4.4. *In the sequel, we assume that any term containing u_0, l_0, c_0, a_0 or b_0 is zero. Moreover, $\Delta u_0 = \Delta l_0 = \Delta c_0 = \Delta a_0 = 0$.*

In the following Lemma, and in the rest of the results of this section, we focus on the change of u_k. The change of l_k is obtained from the change of u_k and the change of the data by using (4.2).

LEMMA 4.1. *The following recurrence relation is obtained to first-order:*

$$\Delta u_k = \Delta a_k - \frac{b_{k-1}}{u_{k-1}} (\Delta c_{k-1} - l_{k-1} \Delta u_{k-1}), \quad 1 \leq k \leq n, \tag{4.4}$$

and moreover

$$\Delta l_k = l_k \left(\frac{\Delta c_k}{c_k} - \frac{\Delta u_k}{u_k} \right), \quad 1 \leq k \leq n-1. \tag{4.5}$$

PROOF. Since $u_1 = a_1$ then $\Delta u_1 = \Delta a_1$. Moreover to first order,

$$l_k + \Delta l_k = \frac{c_k + \Delta c_k}{u_k + \Delta u_k} = \frac{c_k}{u_k} \left(1 + \frac{\Delta c_k}{c_k} - \frac{\Delta u_k}{u_k} \right) = l_k \left(1 + \frac{\Delta c_k}{c_k} - \frac{\Delta u_k}{u_k} \right).$$

And we get,

$$\Delta l_k = l_k \left(\frac{\Delta c_k}{c_k} - \frac{\Delta u_k}{u_k} \right) = \frac{\Delta c_k}{u_k} - \frac{l_k}{u_k} \Delta u_k. \tag{4.6}$$

On the other hand,

$$\Delta u_{k+1} = \Delta a_{k+1} - b_k \Delta l_k \tag{4.7}$$

follows from (4.3). If we plug (4.6) into (4.7), we get

$$\Delta u_{k+1} = \Delta a_{k+1} - \frac{b_k}{u_k} \Delta c_k + \frac{b_k l_k}{u_k} \Delta u_k.$$

□

In Lemma 4.1 a recurrence relation for Δu_k has been presented. In the next Lemma 4.2 an explicit expression for Δu_k is obtained from the recurrence relation.

LEMMA 4.2. *The following expression is obtained to first-order:*

$$\Delta u_k = \Delta a_k - \frac{b_{k-1}}{u_{k-1}}\Delta c_{k-1} + \sum_{i=1}^{k-1}\left(\Delta a_i - \frac{b_{i-1}}{u_{i-1}}\Delta c_{i-1}\right)\prod_{j=i}^{k-1}\frac{b_j l_j}{u_j}, \quad k \geq 1, \tag{4.8}$$

where we assume that the summation in the expression for Δu_1 is zero.

PROOF. Lemma 4.1 produces the result for $k = 1, 2$, and the proof follows easily by induction. □

The expression for Δu_k obtained in Lemma 4.2 as well as (4.5) are the starting point to get expressions for the condition numbers appearing in Definition 4.1. In the rest of this subsection we will deduce the same formulas for Δu_k and Δl_k by a matrix formulation in order to highlight the relation between the condition numbers and the matrices L^{-1} and U^{-1}. In this sense, the following approach has the advantage that can be compared with the perturbation results obtained in [1, 3, 14, 16], all of them written as functions of L^{-1} and U^{-1}. However, the following developments will also show that the approach previously presented in Lemmas 4.1 and 4.2 is better adapted to the structured problem we are dealing with since the calculations are simpler and shorter.

Let $T = \text{tridiag}[c, a, b]$ be a tridiagonal matrix with unique LU factorization $T = LU$. For simplicity, we will consider that T is nonsingular, although, in general, only the existence of a unique LU factorization is needed. Let us consider a perturbation $T + \Delta T$ of T like the one appearing in Definition 4.1, and such that the corresponding LU factorization without pivoting exists and it is unique. Then,

$$T + \Delta T = (L + \Delta L)(U + \Delta U), \tag{4.9}$$

where ΔL is a strictly lower triangular matrix with all the entries in positions different from $(i + 1, i)$, $i = 1 : (n - 1)$, equal to zero, and ΔU is a diagonal matrix. From (4.9), we obtain to first order

$$\Delta T = \Delta L\, U + L\Delta U. \tag{4.10}$$

Let us denote $F := L^{-1}\Delta T\, U^{-1}$. Then, taking into account (4.10), we get

$$F = L^{-1}\Delta L + \Delta U\, U^{-1}.$$

Notice that $F^U := \Delta U\, U^{-1}$ is an upper triangular matrix and $F^L := L^{-1}\,\Delta L$ is a strictly lower triangular matrix. In fact, F^U is the upper triangular part of F, and F^L the strict lower triangular part of F. Therefore, since $\Delta U = F^U U$ and $\Delta L = L F^L$, it is easy to prove that

$$\Delta u_k = F_{kk}\, u_k = (L^{-1}\,\Delta T\, U^{-1})_{kk}\; u_k, \quad k = 1 : n, \tag{4.11}$$

$$\Delta l_k = F_{k+1,k} = (L^{-1}\,\Delta T\,U^{-1})_{k+1,k}, \quad k = 1:n-1. \tag{4.12}$$

The rest of the elements of ΔL and ΔU are equal to zero.

Taking into account that

$$(L^{-1})_{ij} = \begin{cases} 1 & \text{if } i = j, \\ (-1)^{i+j}\prod_{r=j}^{i-1} l_r & \text{if } i > j, \\ 0 & \text{if } i < j. \end{cases},$$

$$(U^{-1})_{ij} = \begin{cases} \frac{1}{u_i} & \text{if } i = j, \\ \frac{(-1)^{i+j}\prod_{r=i}^{j-1} b_r}{\prod_{r=i}^{j} u_r} & \text{if } i < j, \\ 0 & \text{if } i > j. \end{cases},$$

$$(\Delta T)_{ij} = \begin{cases} \Delta a_i & \text{if } i = j, \\ \Delta c_{i-1} & \text{if } j = i-1, \\ 0 & \text{in any other case.} \end{cases},$$

tedious but straightforward computations show that (4.11) and (4.12) are precisely (4.8) and (4.5), respectively. Moreorever, using $\Delta U = F^U U$ and $\Delta L = L\,F^L$, it can be checked that $(\Delta U)_{ij} = 0$ if $i \neq j$, and $(\Delta L)_{ij} = 0$ if $i \neq (j+1)$.

When T is a singular matrix with unique LU factorization, a similar reasoning to the previous one is possible. In fact, it suffices to consider $F := L^{-1}\Delta T\,\tilde{U}^{-1}$, where $\tilde{U} = U + \alpha e_n e_n^t$ for some $\alpha \neq 0$. This kind of approach was introduced for general matrices in [3, Theorem 2.1].

4.2 Condition numbers of u_k and l_k

In order to find an explicit expression for $cond_B(T)$ and $cond_C(T)$ it is necessary to find a bound of $|\frac{\Delta u_k}{u_k}|$ and $|\frac{\Delta l_k}{l_k}|$. We first consider the perturbations associated with the backward error appearing in the definition of $cond_B(T)$. To simplify the statement of some results, we define the following quantity:

DEFINITION 4.2. *For $k = 1:n$*

$$cond_B(u_k) := 1 + \frac{2|l_{k-1}b_{k-1}|}{|u_k|} + \sum_{i=1}^{k-1}\left(1 + \frac{2|l_{i-1}b_{i-1}|}{|u_i|}\right)\prod_{j=i}^{k-1}\frac{|b_j l_j|}{|u_{j+1}|}.$$

It will be shown in Remark 4.7 that the quantities $cond_B(u_k)$, $k = 1:n$, are condition numbers for u_k. These numbers will appear in the explicit expression of $cond_B(T)$.

LEMMA 4.3. *If $|\Delta a_k| \leq \epsilon(|u_k| + |l_{k-1}b_{k-1}|)$ and $|\Delta c_k| \leq \epsilon|c_k|$ hold for $k \geq 1$, then to first order*

$$\left|\frac{\Delta u_k}{u_k}\right| \leq \epsilon\, cond_B(u_k), \quad 1 \leq k \leq n,$$

$$\left|\frac{\Delta l_k}{l_k}\right| \leq \begin{cases} \epsilon\,(1 + cond_B(u_k)) & \text{if } c_k \neq 0 \\ 0 & \text{if } c_k = 0 \end{cases} \qquad 1 \leq k \leq n-1.$$

PROOF. It is enough to take absolute values in the expression for Δu_k appearing in Lemma 4.2, apply the triangular inequality and divide by $|u_k|$. The result for Δl_k follows from (4.5), and the fact that $c_k = 0$ implies $l_k = 0$ and $\Delta l_k = 0$. □

REMARK 4.5. *It is easy to prove that $cond_B(u_k)$ can be written in the following compact way:*

$$cond_B(u_k) = 1 + 3\sum_{i=1}^{k-1}\prod_{j=i}^{k-1}\frac{|b_j l_j|}{|u_{j+1}|}. \tag{4.13}$$

The recurrence relation for $cond_B(u_k)$ appearing in the next Lemma will be used to estimate the cost of computing $cond_B(T)$.

LEMMA 4.4.

$$cond_B(u_1) = 1, \quad cond_B(u_k) = 1 + \frac{|b_{k-1}l_{k-1}|}{|u_k|}(2 + cond_B(u_{k-1})), \quad 2 \le k \le n.$$

PROOF. Taking into account (4.13)

$$1 + \frac{|b_{k-1}l_{k-1}|}{|u_k|}(2 + cond_B(u_{k-1})) = 1 + \frac{|b_{k-1}l_{k-1}|}{|u_k|}\left(3 + 3\sum_{i=1}^{k-2}\prod_{j=i}^{k-2}\frac{|l_j b_j|}{|u_{j+1}|}\right)$$

$$= 1 + 3\frac{|b_{k-1}l_{k-1}|}{|u_k|} + 3\sum_{i=1}^{k-2}\prod_{j=i}^{k-1}\frac{|l_j b_j|}{|u_{j+1}|} = 1 + 3\sum_{i=1}^{k-1}\prod_{j=i}^{k-1}\frac{|l_j b_j|}{|u_{j+1}|} = cond_B(u_k).$$

□

Now, we consider the componentwise perturbations appearing in the definition of $cond_C(T)$. We begin with the following definition:

DEFINITION 4.3. *For $k = 1 : n$*

$$\begin{aligned} cond_C(u_k) \quad := \quad & \left|1 + \frac{l_{k-1}b_{k-1}}{u_k}\right| + \left|\frac{l_{k-1}b_{k-1}}{u_k}\right| \\ & + \sum_{i=1}^{k-1}\left(\left|1 + \frac{l_{i-1}b_{i-1}}{u_i}\right| + \left|\frac{l_{i-1}b_{i-1}}{u_i}\right|\right)\prod_{j=i}^{k-1}\left|\frac{l_j b_j}{u_{j+1}}\right|. \end{aligned}$$

It will be shown in Remark 4.7 that the quantities $cond_C(u_k)$, $k = 1 : n$, are condition numbers for u_k. These numbers will appear in the explicit expression of $cond_C(T)$.

LEMMA 4.5. *If $|\Delta a_k| \le \epsilon|a_k|$ and $|\Delta c_k| \le \epsilon|c_k|$, hold for $k \ge 1$, then to first order*

$$\left|\frac{\Delta u_k}{u_k}\right| \le \epsilon\, cond_C(u_k), \quad 1 \le k \le n,$$

$$\left|\frac{\Delta l_k}{l_k}\right| \leq \begin{cases} \epsilon\,(1+cond_C(u_k)) & \text{if } c_k \neq 0 \\ 0 & \text{if } c_k = 0 \end{cases} \qquad 1 \leq k \leq n-1.$$

PROOF. The proof is similar to that of Lemma 4.3. To get the final expressions remember that $a_k = u_k + l_{k-1}b_{k-1}$. □

Next, we give the corresponding recurrence relation for $cond_C(u_k)$.

LEMMA 4.6.

$$cond_C(u_1) = 1,$$

$$cond_C(u_k) = \left|1 + \frac{l_{k-1}b_{k-1}}{u_k}\right| + \left|\frac{l_{k-1}b_{k-1}}{u_k}\right| (1 + cond_C(u_{k-1})), \quad 2 \leq k \leq n.$$

PROOF. The proof is analogous to the proof of Lemma 4.4. □

4.3 Condition numbers and relation between their magnitudes.

Our aim in this subsection is to find explicit expressions for the condition numbers $cond_B(T)$ and $cond_C(T)$, and establish that their magnitudes are similar.

We will need to distinguish between the case $c_k \neq 0$ and $c_k = 0$, thus the following definition is introduced

$$\overline{\delta}_{c_k} = \begin{cases} 1 & \text{if } c_k \neq 0 \\ 0 & \text{if } c_k = 0 \end{cases} \qquad 1 \leq k \leq n-1.$$

From Definition 4.1 and Lemmas 4.3 and 4.5, it is obvious that

$$cond_B(T) \leq \max\{\max_{k=1:n-1}\{\overline{\delta}_{c_k} + cond_B(u_k)\},\, cond_B(u_n)\}, \tag{4.14}$$

$$cond_C(T) \leq \max\{\max_{k=1:n-1}\{\overline{\delta}_{c_k} + cond_C(u_k)\},\, cond_C(u_n)\}. \tag{4.15}$$

Notice that $cond_B(u_1) = cond_C(u_1) = 1$ therefore the previous bounds for both condition numbers are greater than 1. In fact, we are going to prove that these bounds are precisely the condition numbers.

THEOREM 4.7. *Let*

$$T = \text{tridiag}[c, a, b] = \text{bidiagl}[l, ones]\,\text{bidiagu}[u, b] = LU$$

be the unique LU factorization of the tridiagonal $n \times n$ matrix T, then

$$cond_B(T) = \max\{\max_{k=1:n-1}\{\overline{\delta}_{c_k} + cond_B(u_k)\},\, cond_B(u_n)\},$$

$$cond_C(T) = \max\{\max_{k=1:n-1}\{\overline{\delta}_{c_k} + cond_C(u_k)\},\, cond_C(u_n)\},$$

where $\overline{\delta}_{c_k} = 1$ if $c_k \neq 0$ and $\overline{\delta}_{c_k} = 0$ otherwise.

PROOF. The result is proven for $cond_B(T)$. The proof for $cond_C(T)$ is analogous step by step. We have to prove that it is possible to choose perturbations, Δc_i, for $i = 1 : n-1$, and Δa_i, for $i = 1 : n$, such that the inequality in (4.14)

becomes an equality, and such that the perturbations are of the type appearing in Definition 4.1 for $cond_B(T)$. In fact, the perturbations are going to be chosen in a way such that all the inequalities appearing in Lemma 4.3 become equalities to first order. This implies the equality in (4.14).

The upper bounds for $|\Delta u_k/u_k|$ and $|\Delta l_k/l_k|$ were obtained in Lemma 4.3 from Lemma 4.2 and (4.5), by using the triangular inequality and setting $|\Delta a_k| = \epsilon(|u_k| + |l_{k-1}b_{k-1}|)$ and $|\Delta c_k| = \epsilon|c_k|$, for $k \geq 1$. Therefore, the absolute value of the perturbations is fixed, we only need to fix their signs in a way such that no cancelations occur in the equation of Lemma 4.2 and in (4.5). Taking into account that Lemma 4.2 is equivalent to the recurrence relation (4.4), we look for no cancelations in (4.4) and (4.5).

The sign of $\Delta a_1 = \Delta u_1$ is randomly chosen. Assume that the signs of Δa_j, for $j = 1 : (k-1)$, and Δc_j, for $j = 1 : (k-2)$, have been chosen such that no cancelations occur in Δu_j, for $j = 1 : (k-1)$. Notice that the sign of Δu_{k-1} is fixed by this selection, thus it is clear from (4.4) that the signs of Δa_k and Δc_{k-1} can be selected to avoid cancelation in Δu_k. This iterative procedure gives the signs of Δa_j, for $j = 1 : n$, and Δc_j, for $j = 1 : (n-1)$, in a way that no cancelation appears in getting Δu_j, for $j = 1 : n$. In particular, there is no cancelation in the expressions $(\Delta c_j - l_j \Delta u_j)$ for $j = 1 : n-1$, which implies that there is no cancelation in the expressions (4.6) for Δl_j, $j = 1 : n-1$. □

REMARK 4.6. *The previous theorem, along with Definitions 4.2 and 4.3, gives explicit expressions for $cond_B(T)$ and $cond_C(T)$ that can be useful for theoretical purposes. However to compute these condition numbers Lemmas 4.4 and 4.6 are preferred. Thus, $cond_B(T)$ and $cond_C(T)$ can be computed with cost $6n-7$ and $7n-8$ flops, respectively, and $(n-1)$ comparisons to determine the maximum.*

REMARK 4.7. *Notice that $cond_B(u_k)$ and $cond_C(u_k)$ are relative condition numbers for u_k. This follows from the proof of Theorem 4.7. The corresponding condition numbers for l_k are $1 + cond_B(u_k)$ and $1 + cond_C(u_k)$, if $c_k \neq 0$.*

The expressions we have obtained for $cond_B(T)$ and $cond_C(T)$ are written in function of the elements of L and U. It would be nicer to write these condition numbers by using only the data of the problem, i.e., the elements of T. However, this does not seem possible, and, it should be noticed that all the perturbation bounds obtained so far for the LU factorization of general matrices involve L and U [1, 3, 14, 15, 16]. The condition numbers can be seen as functions of the quantities $l_{k-1}b_{k-1}/u_k$, for $k = 2 : n$, and these quantities can be written as

$$\frac{l_{k-1}b_{k-1}}{u_k} = \frac{a_k}{u_k} - 1,$$

which implies that if $0 \leq (a_k/u_k) \leq 2$ for all k then the value of $cond_B(T)$ is bounded by $3n-2$, by Remark 4.5, and $cond_C(T)$ is of similar magnitude. However, $cond_B(T)$ and $cond_C(T)$ can be moderate although $0 \leq (a_k/u_k) \leq 2$ is not fulfilled for some k. Notice also, that an important element growth of u_k with respect to a_k does not imply a large value of the condition numbers.

Now, we undertake the task of comparing the magnitudes of the condition numbers.

LEMMA 4.8. *For* $1 \leq k \leq n$,

$$cond_C(u_k) \leq cond_B(u_k) \leq 3\, cond_C(u_k).$$

PROOF. From Definitions 4.2 and 4.3, we get

$$cond_C(u_k) \leq cond_B(u_k).$$

Notice that $|1+x|+|x| \geq 1-|x|+|x| = 1$ holds for any number x. Therefore

$$cond_C(u_k) \geq 1 + \sum_{i=1}^{k-1} \prod_{j=i}^{k-1} \left| \frac{b_j l_j}{u_{j+1}} \right|,$$

and from Remark 4.5,

$$3\, cond_C(u_k) \geq 3 + 3 \sum_{i=1}^{k-1} \prod_{j=i}^{k-1} \left| \frac{b_j l_j}{u_{j+1}} \right| \geq cond_B(u_k).$$

□

As a consequence of the previous lemma, we get one of the most relevant results of this paper.

THEOREM 4.9. *For any tridiagonal* $n \times n$ *matrix* T *having a unique LU factorization*

$$cond_C(T) \leq cond_B(T) \leq 3\, cond_C(T).$$

We finish by remarking that Theorem 4.9 implies, even in the absence of Theorem 3.2, that Algorithm 2.1 to compute the LU factorization without pivoting of tridiagonal matrices is forward stable in the sense of [12, p. 9]:

$$\max_i \left\{ \frac{|\hat{l}_i - l_i|}{|l_i|}, \frac{|\hat{u}_i - u_i|}{|u_i|} \right\} \leq \mathbf{u}\, cond_B(T) + O(\mathbf{u}^2) \leq 3\, \mathbf{u}\, cond_C(T) + O(\mathbf{u}^2).$$

Therefore the magnitude of the forward errors in the output of Algorithm 2.1 is "the best you can expect".

4.4 Invariance of the condition numbers under diagonal scalings.

The following theorem states that the condition numbers introduced in Definition 4.1 do not change under diagonal scalings.

THEOREM 4.10. *Let* T *be an* $n \times n$ *tridiagonal matrix having a unique LU factorization, and* D_1 *and* D_2 *be nonsingular diagonal matrices, then*

$$cond_B(T) = cond_B(D_1 T D_2) \quad \text{and} \quad cond_C(T) = cond_C(D_1 T D_2).$$

PROOF. Let $T = \text{tridiag}[c, a, b]$ and $T^D = D_1 T D_2 = \text{tridiag}[c^D, a^D, b^D]$ be the tridiagonal matrices we consider, and let $d_k^{(1)}$ and $d_k^{(2)}$ be the elements on the main diagonal of D_1 and D_2 respectively. Notice that $c_k = 0$ if and only if $c_k^D = 0$. By Theorem 4.7 and Definitions 4.2 and 4.3, it is obvious that the theorem is proven if we show that the quantities $l_{k-1} b_{k-1}/u_k$, for $k = 2 : n$, do not change under diagonal scalings. To prove this, observe that if $T = LU$ is the LU factorization of T then $T^D = (D_1 L D_1^{-1})(D_1 U D_2) \equiv L^D U^D$ is the LU factorization of T^D. Therefore the elements of U^D and L^D are

$$u_k^D = u_k d_k^{(1)} d_k^{(2)} \quad \text{and} \quad l_k^D = l_k \frac{d_{k+1}^{(1)}}{d_k^{(1)}}, \quad k \geq 1.$$

Taking into account that $b_k^D = b_k d_k^{(1)} d_{k+1}^{(2)}$, we get

$$\frac{l_{k-1}^D b_{k-1}^D}{u_k^D} = \frac{l_{k-1} b_{k-1}}{u_k}.$$

□

5 Conditioning: norms vs. components.

The backward error analysis of Algorithm 2.1 fixes the type of perturbations we have to consider, but it is well known that for many applications it is enough to have small *normwise* relative forward errors. This leads us to introduce the following condition numbers.

DEFINITION 5.1. *Let*

$$T = \text{tridiag}[c, a, b] = LU$$

and

$$\text{tridiag}[c + \Delta c, a + \Delta a, b] = (L + \Delta L)(U + \Delta U)$$

be the unique LU factorizations of two $n \times n$ tridiagonal matrices. We define the condition numbers

$$ncond_B(T) := \lim_{\epsilon \to 0} \sup \left\{ \max \left\{ \frac{\|\Delta U\|}{\epsilon \|U\|}, \frac{\|\Delta L\|}{\epsilon \|L\|} \right\} : |\Delta a| \leq \epsilon \, \text{diag}(|L||U|), \; |\Delta c| \leq \epsilon |c| \right\},$$

$$ncond_C(T) := \lim_{\epsilon \to 0} \sup \left\{ \max \left\{ \frac{\|\Delta U\|}{\epsilon \|U\|}, \frac{\|\Delta L\|}{\epsilon \|L\|} \right\} : |\Delta a| \leq \epsilon \, |a|, \; |\Delta c| \leq \epsilon \, |c| \right\}.$$

In the previous definition any norm may be used, however we will focus, for the sake of simplicity, in the "max norm" defined in Section 2. This Section runs parallel to Section 4, except for the fact that the condition numbers $ncond_B$ and

$ncond_C$ are not invariant under diagonal scalings. Thus, we will omit most of the comments and proofs. It is important to notice that

$$ncond_B(T) \leq cond_B(T) \quad \text{and} \quad ncond_C(T) \leq cond_C(T),$$

therefore, matrices for which Algorithm 2.1 produces small relative errors in norm but large errors in components may exist. We illustrate this in Section 7.

5.1 Auxiliary results for absolute errors.

We first consider perturbations associated with the backward error.

LEMMA 5.1. *If* $|\Delta a_k| \leq \epsilon(|u_k| + |l_{k-1}b_{k-1}|)$ *and* $|\Delta c_k| \leq \epsilon|c_k|$ *hold for* $k \geq 1$, *then to first order*

$$|\Delta u_k| \leq \epsilon\, ncond_B(u_k), \quad 1 \leq k \leq n,$$

$$|\Delta l_k| \leq \epsilon\,|l_k| \left(1 + \frac{ncond_B(u_k)}{|u_k|}\right) \quad 1 \leq k \leq n-1,$$

where $ncond_B(u_k)$, $k = 1:n$, *is defined by the following recurrence relation*

$$\begin{aligned} ncond_B(u_1) &= |u_1|, \\ ncond_B(u_k) &= |u_k| + |b_{k-1}l_{k-1}| \left(2 + \frac{ncond_B(u_{k-1})}{|u_{k-1}|}\right). \end{aligned} \tag{5.1}$$

Moreover, the following explicit expression holds for $k = 1:n$:

$$ncond_B(u_k) = |u_k| + 2\,|l_{k-1}b_{k-1}| + \sum_{i=1}^{k-1} (|u_i| + 2\,|l_{i-1}b_{i-1}|) \prod_{j=i}^{k-1} \frac{|b_j l_j|}{|u_j|}. \tag{5.2}$$

PROOF. The proof is done by applying triangular inequalities to the expressions of Lemmas 4.1 and 4.2, in a similar way to the proofs of Lemmas 4.3 and 4.4. In fact, if in these two Lemmas $|u_k|cond_B(u_k)$ is replaced by $ncond_B(u_k)$, then Lemma 5.1 follows. □

Now, we state without proof the corresponding result for small componentwise perturbations.

LEMMA 5.2. *If* $|\Delta a_k| \leq \epsilon|a_k|$ *and* $|\Delta c_k| \leq \epsilon|c_k|$ *hold for* $k \geq 1$, *then to first order*

$$|\Delta u_k| \leq \epsilon\, ncond_C(u_k), \quad 1 \leq k \leq n,$$

$$|\Delta l_k| \leq \epsilon\,|l_k| \left(1 + \frac{ncond_C(u_k)}{|u_k|}\right) \quad 1 \leq k \leq n-1,$$

where $ncond_C(u_k)$, $k = 1:n$, *is defined by the following recurrence relation*

$$\begin{aligned} ncond_C(u_1) &= |u_1|, \\ ncond_C(u_k) &= |u_k + b_{k-1}l_{k-1}| + |b_{k-1}l_{k-1}| \left(1 + \frac{ncond_C(u_{k-1})}{|u_{k-1}|}\right). \end{aligned} \tag{5.3}$$

Moreover, the following explicit expression holds for $k = 1 : n$:

$$\begin{aligned} ncond_C(u_k) &= |u_k + l_{k-1}b_{k-1}| + |l_{k-1}b_{k-1}| \\ &\quad + \sum_{i=1}^{k-1} \left(|u_i + l_{i-1}b_{i-1}| + |l_{i-1}b_{i-1}|\right) \prod_{j=i}^{k-1} \frac{|b_j l_j|}{|u_j|}. \end{aligned} \tag{5.4}$$

5.2 Condition numbers and their equivalence.

Let us define for simplicity, for $1 \le k \le (n-1)$:

$$ncond_B(l_k) := |l_k| \left(1 + \frac{ncond_B(u_k)}{|u_k|}\right),$$

$$ncond_C(l_k) := |l_k| \left(1 + \frac{ncond_C(u_k)}{|u_k|}\right).$$

The proof of the following theorem is the same as for Theorem 4.7.

THEOREM 5.3. *Let*

$$T = \text{tridiag}[c, a, b] = \text{bidiagl}[l, ones]\,\text{bidiagu}[u, b] = LU$$

be the unique LU factorization of the tridiagonal $n \times n$ matrix T, then

$$ncond_B(T) = \max\left\{ \frac{\max_{k=1:n}\{ncond_B(u_k)\}}{\max_k\{|u_k|, |b_k|\}}, \frac{\max_{k=1:n-1}\{ncond_B(l_k)\}}{\max_k\{|l_k|, 1\}} \right\},$$

$$ncond_C(T) = \max\left\{ \frac{\max_{k=1:n}\{ncond_C(u_k)\}}{\max_k\{|u_k|, |b_k|\}}, \frac{\max_{k=1:n-1}\{ncond_C(l_k)\}}{\max_k\{|l_k|, 1\}} \right\}.$$

REMARK 5.1. *The recurrence relations (5.1) and (5.3) can be used to compute $ncond_B(T)$ and $ncond_C(T)$ with cost $7(n-1)$ and $8(n-1)$, respectively, along with $5n-6$ comparisons to determine the maximums.*

Finally, we show that the magnitudes of $ncond_B(T)$ and $ncond_C(T)$ are similar:

THEOREM 5.4. *For any tridiagonal $n \times n$ matrix T having a unique LU factorization*

$$ncond_C(T) \le ncond_B(T) \le 3\, ncond_C(T).$$

PROOF. As in the case of the Theorem 4.9, if we prove

$$ncond_C(u_k) \le ncond_B(u_k) \le 3\, ncond_C(u_k), \tag{5.5}$$

then Theorem 5.4 follows. If the triangular inequality is applied to (5.4) then we get the first inequality of (5.5). Notice that for any numbers x and y, $|x|+2|y| \le |x+y|+3|y| \le 3(|x+y|+|y|)$ holds. If this inequality is applied to (5.2) then we get the second inequality of (5.5). □

The numerical applications of this result for the relative normwise forward errors produced by Algorithm 2.1 can be discussed in a similar way to that appearing at the end of Subsection 4.3.

6 Tridiagonal LU factorization of diagonally dominant matrices.

It is well known that the usual algorithm to compute the LU factorization without pivoting is normwise backward stable when it is applied on diagonally dominant matrices by rows or columns [12, Theorem 9.9]. In the case of tridiagonal diagonally dominant matrices Algorithm 2.1 is componentwise backward stable [12, Theorem 9.13]. No need to say that this does not imply that the output of Algorithm 2.1 has small forward errors. However this is the case if the matrix is simultaneously diagonally dominant by rows *and* columns, and, besides, the absolute values of the entries in the positions $(i, i+1)$ are in non decreasing order. This is shown in the next Theorem.

THEOREM 6.1. *Let $T = \text{tridiag}[c, a, b]$ be a tridiagonal matrix, and $T = LU$ be its unique LU factorization. If T is diagonally dominant by rows and columns and $\max_i\{\left|\frac{b_{i-1}}{b_i}\right|\} \leq 1$ then*

$$cond_B(T) \leq 3n - 2.$$

PROOF. If T is diagonally dominant by rows then $|u_i| \geq |b_i|$, and if T is diagonally dominant by columns then $|l_i| \leq 1$. Therefore,

$$\left|\frac{b_i l_i}{u_{i+1}}\right| \leq \left|\frac{b_i}{b_{i+1}}\right| \leq 1.$$

Then, from Remark 4.5,

$$cond_B(u_i) \leq 1 + 3(i-1).$$

And the result follows from Theorem 4.7. □

The condition $\max_i\{\left|\frac{b_{i-1}}{b_i}\right|\} \leq 1$ is necessary. It is not difficult to devise diagonally dominant tridiagonal matrices for which this condition does not hold and $cond_B(T)$ takes large values. The set of matrices fulfilling $\max_i\{\left|\frac{b_{i-1}}{b_i}\right|\} \leq 1$ includes the case of tridiagonal matrices whose entries in positions $(i, i+1)$ are all equal to 1. This case appears frequently in problems associated with orthogonal polynomials [8, 9, 2], and in spectral problems [13], because any tridiagonal matrix with $b_i \neq 0$, $i = 1 : n-1$, is similar to one of these matrices.

7 Numerical examples.

The numerical examples that we present in this section have three goals. In the first place, we want to show that the condition numbers we have defined give a reliable measure of the forward errors in the output of Algorithm 2.1. In the second place, some examples illustrate that there is no relation between the size of forward and backward errors: it is possible to have large backward errors and small forward errors, and viceversa. To finish, an example is presented for which the forward errors are large componentwise but small in norm, this justify the developments in Section 5.

In these experiments we compare the output of Algorithm 2.1 in the floating point arithmetic of MATLAB 5.3 ($\mathbf{u} = 1.11 \times 10^{-16}$), with the output computed by the Symbolic Math Toolbox of MATLAB with variable precision arithmetic of 32 significant decimal digits. Moreover, we present separate results for the L and U factors, both for errors and condition numbers. Remember at this point the comments in Remark 4.3.

We will use the following notation: ./ denotes, as in MATLAB, componentwise division, letters with hat denote quantities computed by MATLAB, and letters without hat denote quantities computed by the Symbolic Math Toolbox. The input of the algorithm is the floating point representation of a tridiagonal matrix $T = \text{tridiag}[c, a, b]$, both in MATLAB and in the Symbolic Math Toolbox. The condition numbers are computed by using the computed factors $\widehat{L}$ and $\widehat{U}$. We will consider the following quantities:

- errback= $\mathbf{u} \cdot \max(1, \max(\text{diag}(|\hat{L}||\hat{U}|)./|a|)$. This is by Theorem 3.1 the maximum componentwise theoretical backward error.
- forwardu=$\max_i(|u_i - \hat{u}_i|/|u_i|)$. Componentwise forward error in U.
- forwardl=$\max_i(|l_i - \hat{l}_i|/|l_i|)$. Componentwise forward error in L.
- nforwardu = $\|U - \hat{U}\|/\|U\|$. Normwise forward error in U.
- nforwardl = $\|L - \hat{L}\|/\|L\|$. Normwise forward error in L.
- condu = $\max_{k=1:n}(cond_B(u_k))$. Condition number in components of U.
- condl = $\max_{k=1:n-1}(1 + cond_B(u_k))$. Condition number in components of L. We consider matrices with $c_k \neq 0$ for all k.
- ncondu = $\max_{k=1:n}\{ncond_B(u_k)\}/\max_k\{|u_k|, |b_k|\}$. Condition number in norm of U.
- ncondl = $\max_{k=1:n-1}\{ncond_B(l_k)\}/\max_k\{|l_k|, 1\}$. Condition number in norm of L.

It is important to bear in mind when inspecting the following experiments that $\mathbf{u}$ times a condition number is a (first order) upper bound of the corresponding forward error. Remember the discussion at the end of Subsection 4.3.

7.1 Example 1.

In this experiment we have generated 100 random tridiagonal matrices of dimension 100×100. The elements of the matrices follow a normal distribution with mean zero and variance ten. For each matrix in this sample we compute the ratios (forwardu/$\mathbf{u}$condu) and (forwardl/$\mathbf{u}$condl). If these ratios are very small then the condition numbers overestimate the actual errors. In our experiment the minimum and mean values for these quantities are: min(forwardu/$\mathbf{u}$condu) = 0.01, mean(forwardu/$\mathbf{u}$condu) = 0.07, min(forwardl/$\mathbf{u}$condl) = 0.01 and

mean(forwardl/$\mathbf{u}$condl) = 0.07. This means that the condition numbers we have study overstimate the maximum componentwise forward error, at most, by a factor 100, and, in mean, by a factor 14.

7.2 Example 2.

We analyze the case of a positive definite symmetric tridiagonal matrix T. This property guarantees the componetwise backward stability of Algorithm 2.1 [12, Theorem 9.12]. In our example $cond_B(T)$ and $ncond_B(T)$ are large.

$$T = \begin{bmatrix} 1 & \sqrt{1-2\cdot 10^{-10}} & 0 \\ \sqrt{1-2\cdot 10^{-10}} & 1 & \sqrt{4\cdot 10^{-10}-10^{-13}} \\ 0 & \sqrt{4\cdot 10^{-10}-10^{-13}} & 2 \end{bmatrix}.$$

We get the following results

errback	1.11e-16
forwardu	3.31e-004
condu	5.998e+013
forwardl	8.27e-008
condl	1.5e+010
nforwardu	1.65e-007
ncondu	3.e+010
nforwardl	8.27e-008
ncondl	1.5e+010

7.3 Example 3.

Now, we consider the case of a tridiagonal matrix with backward stable LU factorization and such that the modulus of the elements of the subdiagonal of L are less than one, i.e. partial pivoting does not produce any row permutation. Even in this case, $cond_B(T)$ and $ncond_B(T)$ are large.

$$T = \begin{bmatrix} 1 & \sqrt{3}\cdot 10^8 & 0 \\ \sqrt{\frac{1}{2}} & \frac{2}{3}+\sqrt{\frac{3}{2}}\cdot 10^8 & 2\cdot 10^9 \\ 0 & \frac{2}{3}\sqrt{\frac{7}{10}} & 2+2\sqrt{\frac{7}{10}}\cdot 10^9 \end{bmatrix}.$$

The exact values of the nontrivial elements of the L factor are: $l_1 = \sqrt{1/2}$ and $l_2 = \sqrt{7/10}$. The results in this case are

errback	1.11e-16
forwardu	3.12e+001
condu	4.61e+017
forwardl	3.73e-008
condl	5.51e+008
nforwardu	3.12e-008
ncondu	4.62e+008
nforwardl	3.12e-008
ncondl	4.62e+008

7.4 Example 4.

Let us take a look at an example of a tridiagonal matrix with large backward error and small condition numbers.

$$T = \begin{bmatrix} -\sqrt{2} & \sqrt{3} & 0 \\ -\sqrt{2}\,(\sqrt{5}+10^{-14}) & \sqrt{3}\cdot 10^{-14} & \sqrt{5} \\ 0 & -\sqrt{3}\,(1+10^{-12}) & \sqrt{5}\,10^{-12} \end{bmatrix}.$$

The results obtained for this matrix are

errback	4.97e-002
forwardu	1.45e-016
condu	7.00
forwardl	1.99e-016
condl	5.00
nforwardu	1.15e-016
ncondu	6.92
nforwardl	1.99e-016
ncondl	2.00

7.5 Example 5.

The last example is a case in which the condition number $cond_B(T)$ is large but $ncond_B(T)$ is small. This fact is reflected in the magnitude of the forward errors. The three diagonals of $T = \text{tridiag}[c, a, b]$ are:

$$c = [10^{15}\cdot\sqrt{1-2\cdot 10^{-10}},\ 2\cdot 10^{-15}\cdot\sqrt{4\cdot 10^{-10}-10^{-13}}]^T,$$

$$a = [10^{15},\ 1,\ \frac{10^{-3}}{2}+4\cdot 10^{-15}-10^{-18}]^T,$$

$$b = [\sqrt{1-2\cdot 10^{-10}},\ \sqrt{4\cdot 10^{-10}-10^{-13}}]^T.$$

The results are

errback	1.11e-016
forwardu	8.27e-008
condu	1.5e+010
forwardl	8.27e-008
condl	1.5e+010
nforwardu	1.66e-032
ncondu	1
nforwardl	1.65e-017
ncondl	3

Acknowledgements.

The authors thank one of the referees for his suggestions to improve the contents of the paper by including the matrix formulation that appears in the last part of Section 4.1.

REFERENCES

1. A. Barrlund, *Perturbation bounds for the LDL^H and LU decompositions*, BIT, 31 (1991), pp. 358-363.
2. M. I. Bueno and F. Marcellán, *Darboux Transformation and Perturbation of linear functionals*, to appear in Linear Algebra Appl.
3. X.-W. Chang and C. C. Paige, *On the sensitivity of the LU factorization*, BIT, 38 (1998), pp. 486-501.
4. X.-W. Chang and C.C. Paige, *Sensitivity analyses for factorizations of sparse or structured matrices*, Linear Algebra Appl., 284 (1998), pp. 53-71.
5. J. Demmel, M. Gu, S. Eisenstat, I. Slapničar, K. Veselić and Z. Drmač, *Computing the singular value decomposition with high relative accuracy*, Linear Algebra Appl., 299 (1999), pp. 21-80.
6. I. S. Dhillon, *A new $O(n^2)$ algorithm for the symmetric tridiagonal eigenvalue/eigenvector problem.* Ph.D. Thesis, Computer Science Division, University of California, Berkeley, 1997.
7. K. V. Fernando and B. N. Parlett, *Accurate singular values and differential qd algorithms*, Numer. Math., 67 (1994), pp. 191-229.
8. D. Galant, *An implementation of Christoffel's Theorem in the Theory of Orthogonal Polynomials* , Math. Comput., 25 (1971), pp. 111-113.
9. D. Galant, *Algebraic Methods for modified orthogonal polynomials*, Math. Comput., 59 (1992), pp. 541-546.
10. W. Gautschi, *An algorithmic implementation of the generalized Christoffel theorem* , Numerical Integration (G. Hämmerlin, ed.), Internat. Ser. Numer. Math., 57 (1982), pp. 89-106.
11. G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, 1996.

12. N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.
13. B. N. Parlett, *The new qd algorithms*, Acta Numerica (1995), pp. 459-491.
14. G. W. Stewart, *On the perturbation of LU, Cholesky and QR factorizations*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 1141-1145.
15. G. W. Stewart, *On the perturbation of LU and Cholesky factors*, IMA J. Numer. Anal., 17 (1997), pp. 1-6.
16. J.-G. Sun, *Componentwise perturbation bounds for some matrix decompositions*, BIT, 32 (1992), pp. 702-714.