Mediterr. j. math. 99 (9999), 1–14
1660-5446/99000-0, DOI 10.1007/s00009-003-0000

Mediterranean Journal of Mathematics

Numerical properties of shifted tridiagonal LU factorizations.

C. Brittin and M. I. Bueno

Abstract. In this paper, we consider shifted tridiagonal matrices. We prove that the standard algorithm to compute the LU factorization in this situation is mixed forward-backward stable and, therefore, componentwise forward stable. Moreover, we give a formula to compute the corresponding condition number in O(n) flops.

Mathematics Subject Classification (2000). 65F35, 65F50, 15A12, 15A23, 65G50.

Keywords. Stability, LU factorization, tridiagonal matrix.

1. Introduction

Let A be any n-by-n matrix. This matrix is said to have an LU factorization if there exists a lower triangular matrix L and an upper triangular matrix U such that $A = LU$. The LU factorization is one of the more important factorizations in Matrix Analysis and Numerical Analysis. Traditionally it has been used in the solution of linear systems of equations. In this situation, the backward error analysis is what matters. However, lately the LU factorization has been considered to solve spectral problems related with structured matrices [4, 6]. Here, the goal is to compute the factors L and U with small forward errors [4]. In order to place a bound on the forward error it is necessary to combine a backward error analysis with an appropriate perturbation theory for the LU factorization.

This research has been partially supported by Dirección General de Investigación (Ministerio de Ciencia y Tecnología) of Spain through grants BFM2003-06335-C03-02 and MTM2006-06671 as well as by the Postdoctoral Fellowship EX2004-0658 provided by Ministerio de Educación y Ciencia of Spain.

In this paper we study the LU factorization of shifted tridiagonal matrices. Consider the n-by-n tridiagonal matrix

$$J(c,a,b) = \begin{bmatrix} a_1 & b_1 & 0 & \cdots & 0 \\ c_1 & a_2 & b_2 & \cdots & 0 \\ 0 & c_2 & a_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & c_{n-1} & a_n \end{bmatrix}. \tag{1.1}$$

In the notation $J(c,a,b)$, the variables c, a, b denote, respectively, the following vectors

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

Let α be a real number such that the shifted matrix $J(c,a,b) - \alpha I$ has a unique LU factorization. Let $J(c,a,b) - \alpha I = LU$ be the unique LU factorization of $J(c,a,b) - \alpha I$, where L is a unit lower triangular matrix. Notice that the factors L and U are both bidiagonal matrices.

$$L = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_1 & 1 & 0 & \cdots & 0 \\ 0 & l_2 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & l_{n-1} & 1 \end{bmatrix}, \quad U = \begin{bmatrix} u_1 & b_1 & 0 & \cdots & 0 \\ 0 & u_2 & b_2 & \cdots & 0 \\ 0 & 0 & u_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ddots & u_n \end{bmatrix}. \tag{1.2}$$

In the sequel, we denote by l and u the vectors $[l_1, ..., l_{n-1}]$ and $[u_1, ..., u_n]$, respectively.

In [2], the stability and sensitivity of the tridiagonal LU factorization without pivoting was studied. It was proven that the standard algorithm to compute the LU factorization of a tridiagonal matrix is stable in the mixed forward-backward sense [5], and also componentwise forward stable, i.e., the forward errors are of similar magnitude to those produced by a componentwise backward stable method. Moreover, a formula to compute the condition number associated with the problem in O(n) flops was presented. Moreover, the same results can proven for the LU factorization of Hessenberg matrices.

Here we extend the previous results to shifted tridiagonal matrices, namely, we prove that the LU factorization of shifted tridiagonal matrices is stable and componentwise forward stable. By performing the subtraction $J(c,a,b)-\alpha I$ before computing the corresponding LU factorization, we introduce one more arithmetic operation into the problem. The stability of the problem for shifted tridiagonal matrices may seem to naturally follow from the stability in the tridiagonal case. However we have noticed that the stability of the problem for shifted Hessenberg matrices, for instance, does not correspond to the stability in the Hessenberg case. Hence, the stability of the standard algorithm for computing the LU factorization of shifted tridiagonal matrices is worth exploring.

In this work, we use a different technique from that used in [2] to compute the condition number associated with the problem. The outputs u and l of the algorithm are rational functions of the inputs c, a, b and α, and, as a consequence, u and l are differentiable functions of c, a, b and α whenever the denominators are not zero. Therefore, the condition number can be expressed in terms of partial derivatives [3].

The paper is organized as follows: In Section 2, we give the algorithm to compute the LU factorization of shifted tridiagonal matrices (Algorithm 2.1) and we present the backward error analysis of this algorithm (Theorem 2.2). We show that this algorithm is not backward stable but it is stable in the mixed forward-backward sense (Theorem 2.4). In Section 3, the relative componentwise condition number of the problem is defined (Definition 3.8). There we give a formula to compute the condition number in $O(n)$ flops (Theorem 3.7) and we also give a bound for the forward error in terms of this condition number (Theorem 3.1). Moreover, we prove that Algorithm 2.1 is componentwise forward stable in two different ways: 1) considering Theorem 2.2, 2) considering Theorem 2.4.

2. Backward Error Analysis

Let $J(c,a,b)$ be a tridiagonal matrix as in (1.1). Let us assume that α is any real number such that $J(c,a,b)-\alpha I$ has a unique LU factorization. Next we give the pseudocode that computes the matrices L and U from the entries of $J(c,a,b)$ and α.

Algorithm 2.1. *Given the tridiagonal matrix $J(c,a,b)$ and the real number α, this algorithm computes the unique LU factorization of $J(c,a,b)-\alpha I$.*

$u_1 = a_1 - \alpha;$
for $i = 1 : n-1$
    $l_i = c_i / u_i;$
    $u_{i+1} = a_{i+1} - l_i * b_i - \alpha;$
end

The computational cost of Algorithm 2.1 is $4n-3$ flops.

In this section we present the backward error analysis of Algorithm 2.1. It has been done using the standard model of floating point arithmetic [5]:

$$fl(x\ op\ y) = (x\ op\ y)(1+\delta) = \frac{x\ op\ y}{1+\eta}, \quad |\delta|, |\eta| \le \epsilon,$$

where x and y are floating point numbers, $op = +, -, *, /$, and ϵ is the unit roundoff of the machine. From now on, given a vector v, $|v|$ denotes the vector whose entries are the absolute values of the entries of v.

We develop our error analysis in the most general setting. For this purpose, we assume that the shift α is a real number, and we denote by $\hat{\alpha}$, the nearest floating point number to α. Moreover, we assume that the input parameters $a_1, ..., a_n$, $b_1, ..., b_{n-1}$ and $c_1, ..., c_{n-1}$, are respectively affected by small relative errors $(1+\epsilon_{a_1}), ..., (1+\epsilon_{a_n}), (1+\epsilon_{b_1}), ..., (1+\epsilon_{b_{n-1}}), (1+\epsilon_{c_1}), ..., (1+\epsilon_{c_{n-1}})$, where $\max_{1\leq i\leq n-1}\{|\epsilon_{a_i}|, |\epsilon_{a_n}|, |\epsilon_{b_i}|, |\epsilon_{c_i}|\} \leq \epsilon$. These errors on the inputs may come from rounding errors committed by storing them in the computer.

Theorem 2.2. *Let $J(c, a, b)$ be an $n\times n$ tridiagonal matrix and let α be a real number such that $J(c, a, b) - \alpha I$ has a unique LU factorization. Let $\hat{\alpha}$ be the nearest floating point number to α. If $\hat{L}, \hat{U}$ are the factors obtained by applying Algorithm* 2.1 *to the matrix with floating entries $J(\hat{c}, \hat{a}, \hat{b})$ where*

$$\hat{a}_i = a_i(1+\epsilon_{a_i}), \quad 1 \leq i \leq n,$$
$$\hat{b}_i = b_i(1+\epsilon_{b_i}), \quad \hat{c}_i = c_i(1+\epsilon_{c_i}), \quad 1 \leq i \leq n-1,$$

and

$$\max_{1\leq i\leq n-1}\{|\epsilon_{a_i}|, |\epsilon_{a_n}|, |\epsilon_{b_i}|, |\epsilon_{c_i}|\} \leq \epsilon,$$

then

$$J(c+\Delta c, a+\Delta a, b+\Delta b) - \hat{\alpha} I = \hat{L}\hat{U},$$

$$|\Delta a| \leq (3\epsilon + 3\epsilon^2 + \epsilon^3)\left[|a| + diag(|\hat{L}||\hat{U}|)\right], \quad |\Delta b| \leq \epsilon\,|b|$$
$$|\Delta c| \leq (2\epsilon + \epsilon^2)\,|c|, \quad |\hat{\alpha} - \alpha| \leq \epsilon|\alpha|.$$

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

Proof. For the computed quantities, we have

$$\hat{l}_i = \frac{c_i}{\hat{u}_i}(1+\epsilon_{c_i})(1+\varepsilon_{l_i}), \quad |\epsilon_{c_i}|, |\varepsilon_{l_i}| \leq \epsilon.$$

Hence $|c_i - \hat{l}_i\hat{u}_i| \leq (2\epsilon + \epsilon^2)|c_i|$, which proves the theorem for the entries $(i+1, i)$. Moreover, for $i \geq 2$,

$$\hat{u}_{i+1}(1+\nu_i) = \Big(a_{i+1}(1+\epsilon_{a_{i+1}}) - \hat{l}_i b_i(1+\epsilon_{b_i})(1+\beta_i)\Big)(1+\gamma_i) - \hat{\alpha},$$
$$|\nu_i|, |\epsilon_{a_{i+1}}|, |\epsilon_{b_i}|, |\beta_i|, |\gamma_i| \leq \epsilon.$$

Hence

$$|a_{i+1} - \hat{u}_{i+1} - \hat{l}_i b_i - \hat{\alpha}| \leq (3\epsilon + 3\epsilon^2 + \epsilon^3)\left[|a_{i+1}| + |\hat{u}_{i+1}| + |\hat{l}_i b_i|\right]$$

and the result follows. □

The previous result shows that the LU factorization of shifted tridiagonal matrices would be backward stable if $|\hat{u}_{i+1}| + |\hat{l}_i b_i| = O(|a_{i+1}|)$ for $i \geq 2$. Unfortunately, we cannot assure that this is the case as the following simple numerical experiment shows. Just consider the 2-by-2 tridiagonal matrix

$$J(c, a, b) = \begin{bmatrix} 0 & 1 \\ 1 & 1 - 10^{-16} \end{bmatrix}.$$

Let $\alpha = -10^{-16}$. Then, $|\hat{u}_2| + |\hat{l}_1 b_1| = 2 \cdot 10^{16}$ while $a_2 = 1 - 10^{-16}$. Hence, the condition for backward stability does not hold in general. However, next we prove that the LU factorization of shifted tridiagonal matrices is stable in the mixed forward-backward sense.

Definition 2.3. [5] A method for computing $y = f(x)$ is called mixed forward-backward stable (or numerically stable) if, for any x, it produces a computed $\hat{y}$ satisfying

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

Informally, a mixed forward-backward stable algorithm produces almost the right answer for almost the right data.

Theorem 2.4. *Let $J(c, a, b)$ be any $n \times n$ tridiagonal matrix and let α be a real number such that $J(c, a, b) - \alpha I$ has a unique LU factorization. If $\hat{L}, \hat{U}$ are the factors obtained by applying Algorithm* 2.1 *to $J(c, a, b) - \alpha I$, then the following diagram commutes:*

$$\begin{array}{ccc} \{a, b, c, \alpha\} & \xrightarrow{\textit{Computed LU}} & \{\hat{l}, \hat{u}\} \\ \textit{Relative change of at most } 3\epsilon \Big\downarrow & & \Big\uparrow \textit{Relative change of at most } \epsilon \\ \{\tilde{a}, \tilde{b}, \tilde{c}, \tilde{\alpha}\} & \xrightarrow{\textit{Exact LU}} & \{\hat{l}, \tilde{\hat{u}}\} \end{array}$$

where, for all i, $\tilde{a}_i$, $\tilde{b}_i$ $\tilde{c}_i$ and $\tilde{\alpha}$ are obtained, respectively, from a_i, b_i, c_i and α by a relative change smaller than 2ϵ, 3ϵ, 3ϵ and ϵ, and $\tilde{\hat{u}}_i$ is obtained from $\hat{u}_i$ (resp. $\hat{l}_i$) by a relative change smaller than ϵ.

Remark 2.5. In this theorem, $O(\epsilon^2)$ terms are ignored for simplicity.

Proof. The computed quantities satisfy

$$\hat{l}_i = \frac{c_i(1 + \epsilon_{c_i})(1 + \varepsilon_i)}{\hat{u}_i}, \quad |\epsilon_{c_i}|, |\varepsilon_i| \le \epsilon, \tag{2.1}$$

$$\hat{u}_{i+1}(1 + \nu_{i+1}) = [a_{i+1}(1 + \epsilon_{a_{i+1}}) - \hat{l}_i b_i(1 + \epsilon_{b_i})(1 + \beta_i)](1 + \gamma_i) - \alpha(1 + \epsilon_\alpha) \tag{2.2}$$

$$|\nu_{i+1}|, |\epsilon_{a_{i+1}}|, |\epsilon_{b_i}|, |\beta_i|, |\gamma_i| \le \epsilon. \tag{2.3}$$

By defining,

$$\tilde{a}_i = a_i(1 + \epsilon_{a_i})(1 + \gamma_i), \quad \tilde{b}_i = b_i(1 + \epsilon_{b_i})(1 + \beta_i)(1 + \gamma_i),$$
$$\tilde{c}_i = c_i(1 + \epsilon_{c_i})(1 + \nu_i)(1 + \varepsilon_i), \qquad \tilde{\alpha} = \alpha(1 + \epsilon_\alpha), \qquad \text{and } \tilde{\hat{u}}_i = \hat{u}_i(1 + \nu_i),$$

then following exact relations follow from (2.1) and (2.2)

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

□

The result obtained from Theorem 2.4 can also be expressed in the following way taking into account the following notation: $L(l, 1)$ and $U(u, b)$ denote, respectively, a unit bidiagonal lower triangular matrix and a bidiagonal upper triangular matrix as in (1.2).

Theorem 2.6. *Let $J(c,a,b)$ be an $n \times n$ tridiagonal matrix and let α be a real number such that $J(c,a,b) - \alpha I$ has a unique LU factorization. If $\hat{L}, \hat{U}$ are the factors obtained by applying Algorithm 2.1 to $J(c,a,b)$, then there exist vectors Δc, Δa, Δb, $\Delta \hat{l}$ and $\Delta \hat{u}$ such that*

$$J(c+\Delta c, a+\Delta a, b+\Delta b) - \tilde{\alpha} = L(\hat{l},1)U(\hat{u}+\Delta\hat{u}, b+\Delta b), \tag{2.4}$$

where

$$|\Delta a| \leq 2\epsilon|a|, \quad |\Delta b| \leq 3\epsilon|b|, \quad |\Delta c| \leq 3\epsilon|c|, \quad |\tilde{\alpha}-\alpha| \leq \epsilon|\alpha|, \quad |\Delta\hat{u}| \leq \epsilon|\hat{u}|.$$

This shows that Algorithm 2.1 is *componentwise stable in the mixed forward-backward sense* or just stable. This also implies, as we will show in next sections, that Algorithm 2.1 is *componentwise forward stable*, which means that the obtained forward errors are of similar magnitude to those produced by a backward stable algorithm. Roughly speaking, this ensures that the forward errors obtained from this algorithm are the best one can expect from the sensitivity of the problem.

3. A bound for the componentwise forward errors

In order to estimate the magnitude of the forward errors, we need to compute a condition number for this problem. It is well known that the forward errors produced by a backward stable algorithm are bounded by the product of the backward error and the condition number of the problem. However, Algorithm 2.1 is not backward stable. When the algorithm is stable in the mixed forward-backward sense, a bound can be found in a similar way. In order to give such a bound we need first to define a condition number for the problem we are studying.

Theorem 3.1. *Let $J(c,a,b) - \alpha I = L(l,1)U(u,b)$ be the exact LU factorization of the shifted tridiagonal matrix $J(c,a,b) - \alpha I$, where $\alpha \in \mathbb{R}$. Let $L(\hat{l},1)$, and $U(\hat{u},b)$ be the factors computed by Algorithm 2.1. Then,*

$$\max_k \left\{ \left| \frac{l_k - \hat{l}_k}{l_k} \right|, \left| \frac{u_k - \hat{u}_k}{u_k} \right| \right\} \leq \frac{\epsilon}{1-\epsilon}\left(1 + 3\, condC(J(c,a,b),\alpha)\right) + O(\epsilon^2),$$

where $condC(J(c,a,b),\alpha)$ denotes the condition number of the problem (See Definition 3.8).

Proof. By definition of the condition number and taking into account (2.4), we get

$$\left| \frac{u_i - \hat{u}_i - \Delta\hat{u}_i}{u_i} \right| \leq \max_{i=1:n-1} \left\{ \left|\frac{\Delta a_i}{a_i}\right|, \left|\frac{\Delta a_n}{a_n}\right|, \left|\frac{\Delta b_i}{b_i}\right|, \left|\frac{\Delta c_i}{c_i}\right|, \left|\frac{\Delta \alpha}{\alpha}\right| \right\} condC(J(c,a,b),\alpha).$$

This implies that

$$\left| \frac{u_i - \hat{u}_i}{u_i} \right| - \left| \frac{\Delta\hat{u}_i}{u_i} \right| \leq 3\epsilon\, condC(J(c,a,b),\alpha),$$

or equivalently, taking into account (2.4) again,

$$\left|\frac{u_i-\hat{u}_i}{u_i}\right| \leq 3\epsilon\; condC(J(c,a,b),\alpha) + \left|\frac{\Delta\hat{u}_i}{u_i}\right| \leq 3\epsilon\; condC(J(c,a,b),\alpha) + \epsilon\left|\frac{\hat{u}_i}{u_i}\right|,$$

Therefore,

$$\left|\frac{u_i-\hat{u}_i}{u_i}\right| \leq 3\epsilon\; condC(J(c,a,b),\alpha) + \epsilon\left[1+\left|\frac{u_i-\hat{u}_i}{u_i}\right|\right],$$

and the result follows for the entries in u. A similar bound can be found for the forward errors corresponding to the entries of l. Taking into account both bounds, the result follows. □

The bound obtained in Theorem 3.1 for the forward error is expressed in terms of the condition number of the problem $condC(J(c,a,b),\alpha)$. Next we define this condition number and give an explicit expression for it. We study the sensitivity of the shifted LU factorization of tridiagonal matrices with respect to perturbations of the initial data, i.e., the parameters of the tridiagonal matrix $J(c,b,a)$, and the shift α. We give the definition of the relative componentwise condition number of the shifted tridiagonal LU factorization with respect to relative componentwise perturbations in c, a, b and α, i.e., $|\Delta c| \leq \mathbf{u}|c|$, $|\Delta a| \leq \mathbf{u}|a|$, $|\Delta b| \leq \mathbf{u}|b|$ and $|\Delta\alpha| \leq \mathbf{u}|\alpha|$, with small $\mathbf{u}$.

Definition 3.2. Let $L(l,1)$ and $U(u,b)$ be the matrices obtained from the exact LU factorization of $J(c,b,a)-\alpha I$, where $J(c,b,a)$ is a $n\times n$ tridiagonal matrix and α is a real number. Let $L(l+\Delta l,1)$ and $U(u+\Delta u, b+\Delta b)$ be the factors obtained from the LU factorization of $J(c+\Delta c, b+\Delta b, a+\Delta a)-(\alpha+\Delta\alpha)I$. Let us define

$$DC = \max\left\{\max_{1\leq i\leq n}\left\{\frac{|\Delta a_i|}{|a_i|}\right\}, \max_{1\leq i\leq n-1}\left\{\frac{|\Delta c_i|}{|c_i|},\frac{|\Delta b_i|}{|b_i|}\right\}, \frac{|\Delta\alpha|}{|\alpha|}\right\},$$

where any quotient has to be understood as zero if the corresponding denominator is equal to zero. Then the relative componentwise condition number of the shifted tridiagonal LU factorization with respect to small componentwise relative perturbations of c, a, b and α is defined as

$$condC(J(c,b,a),\alpha) = \lim_{\mathbf{u}\to 0}\sup_{0\leq DC\leq\mathbf{u}} \frac{\max\left\{\max\limits_{1\leq i\leq n}\left\{\left|\frac{\Delta u_i}{u_i}\right|\right\}, \max\limits_{1\leq i\leq(n-1)}\left\{\left|\frac{\Delta l_i}{l_i}\right|\right\}\right\}}{DC}.$$

The condition number $condC(J(c,a,b),\alpha)$ is infinite if some of the denominators appearing in the relative changes of the outputs l_i, u_n, i.e, $\frac{|\Delta l_i|}{|l_i|}$, $\frac{|\Delta u_n|}{|u_n|}$ is zero. In these cases, other condition numbers have to be considered. For instance, measuring absolute changes in the corresponding components of b (resp. u_n), or measuring relative normwise changes of b (resp. u_n). We do not consider these particular situations in this work. Notice that we have not considered the situation $u_i = 0$ for $i = 1,...,n-1$ because $u_i = det(J(c,b,a)([1,...,i],[1,...,i]))$, which is nonzero since $J(c,b,a)$ has an LU factorization. Here $J(c,b,a)([1,...,i],[1,...,i])$ denotes the leading principal submatrix of $J(c,b,a)$ of order i.

Next we deduce a recursive expression for $condC(J(c,a,b),\alpha)$. The entries of the vectors u and l are rational functions of the inputs c, a, b, and α, and, as a consequence, the entries of u and l are differentiable functions of c, a, b, and α whenever the denominators are different from zero. Therefore, $condC(J(c,a,b),\alpha)$ can be expressed in terms of partial derivatives [3]. More precisely:

$$condC(J(c,a,b),\alpha) = \max\{\max_{1\le k\le n}\{condC(u_k)\}, \max_{1\le k\le n-1}\{condC(l_k)\}\}, \tag{3.1}$$

where

$$condC(u_k) = \sum_{i=1}^{k}\left|\frac{a_i}{u_k}\frac{\partial u_k}{\partial a_i}\right| + \sum_{i=1}^{k-1}\left|\frac{c_i}{u_k}\frac{\partial u_k}{\partial c_i}\right| + \sum_{i=1}^{k-1}\left|\frac{b_i}{u_k}\frac{\partial u_k}{\partial b_i}\right| + \left|\frac{\alpha}{u_k}\frac{\partial u_k}{\partial \alpha}\right|, \tag{3.2}$$

$$condC(l_k) = \sum_{i=1}^{k}\left|\frac{a_i}{l_k}\frac{\partial l_k}{\partial a_i}\right| + \sum_{i=1}^{k}\left|\frac{c_i}{l_k}\frac{\partial l_k}{\partial c_i}\right| + \sum_{i=1}^{k-1}\left|\frac{b_i}{l_k}\frac{\partial l_k}{\partial b_i}\right| + \left|\frac{\alpha}{l_k}\frac{\partial l_k}{\partial \alpha}\right|. \tag{3.3}$$

In the previous expressions $l_0 := 0$ and $b_0 := 0$.

Notice that, according to Lemma 3.4 in the next subsection,

$$\left|\frac{c_k}{l_k}\frac{\partial l_k}{\partial c_k}\right| = \left|\frac{c_k}{l_k}\frac{1}{u_k}\right| = 1.$$

Taking into account (3.1) and (3.3), we deduce that $condC(J(c,a,b),\alpha) \ge 1$. Considering Theorem 3.1, we get the following result.

The next theorem gives a bound for the forward errors produced by Algorithm 2.1.

Theorem 3.3. *Let $J(c,a,b) - \alpha I = L(l,1)U(u,b)$ be the exact LU factorization of the shifted tridiagonal matrix $J(c,a,b) - \alpha I$, where $\alpha \in \mathbb{R}$. Let $L(\hat{l},1)$, and $U(\hat{u},b)$ be the factors computed by Algorithm* 2.1*. Then,*

$$\max_k \left\{\left|\frac{l_k - \hat{l}_k}{l_k}\right|, \left|\frac{u_k - \hat{u}_k}{u_k}\right|\right\} \le \frac{4\epsilon}{1-\epsilon}\, condC(J(c,a,b),\alpha) + O(\epsilon^2).$$

This means that Algorithm 2.1 *is componentwise forward stable.*

3.1. A recursive formula for the condition number and another proof of the forward stability

In this subsection, we derive a recursive formula to compute the condition number $condC(J(c,a,b),\alpha)$ in $O(n)$ flops. Using this expression, we give an alternative proof of the forward stability of Algorithm 2.1 taking into account the stability result in Theorem 2.2.

Considering Algorithm 2.1 and the expression for l_k, it is easy to check that l_k is function of $a_1,...,a_k,c_1,...,c_k,b_1,...,b_{k-1},\alpha$.

Lemma 3.4. *Let $J(c, b, a)$ be a $n \times n$ tridiagonal matrix. If α is a real number such that $J(c, a, b) - \alpha I$ has a unique LU factorization, then l_k has the following partial derivatives with respect to $a_1, ..., a_k, c_1, ..., c_k, b_1, ..., b_{k-1}, \alpha$.*

$$\frac{\partial l_k}{\partial a_i} = \begin{cases} \dfrac{l_k b_{k-1}}{u_k} \dfrac{\partial l_{k-1}}{\partial a_i}, & 1 \le i < k, \\ -\dfrac{l_k}{u_k}, & i = k. \end{cases}$$

$$\frac{\partial l_k}{\partial c_i} = \begin{cases} \dfrac{l_k b_{k-1}}{u_k} \dfrac{\partial l_{k-1}}{\partial c_i}, & 1 \le i < k, \\ \dfrac{1}{u_k}, & i = k. \end{cases}$$

$$\frac{\partial l_k}{\partial b_i} = \begin{cases} \dfrac{l_k b_{k-1}}{u_k} \dfrac{\partial l_{k-1}}{\partial b_i}, & 1 \le i < k-1, \\ \dfrac{l_k l_{k-1}}{u_k}, & i = k-1. \end{cases}$$

$$\frac{\partial l_k}{\partial \alpha} = \begin{cases} \dfrac{l_k}{u_k} \left(1 + b_{k-1} \dfrac{\partial l_{k-1}}{\partial \alpha}\right), & 2 \le k, \\ \dfrac{l_1}{u_1}, & k = 1. \end{cases}$$

Proof. First we consider $\partial l_k / \partial a_i$. Taking into account the l_k expression that follows from Algorithm 2.1, the case for $i = k$ is easy to check. Consider the case for $1 \le i < k$. By the chain rule

$$\frac{\partial l_k}{\partial a_i} = \frac{\partial l_k}{\partial u_k} \cdot \frac{\partial u_k}{\partial l_{k-1}} \cdot \frac{\partial l_{k-1}}{\partial a_i}$$

By Algorithm 2.1

$$\frac{\partial l_k}{\partial u_k} = -\frac{l_k}{u_k}, \text{ and } \frac{\partial u_k}{\partial l_{k-1}} = -b_{k-1}.$$

Hence, the expression for $\partial l_k / \partial a_i$ follows. Likewise, we can derive the expressions for $\partial l_k / \partial c_i$ and $\partial l_k / \partial b_i$ in a similar fashion.

Clearly, $\frac{\partial l_1}{\partial \alpha} = \frac{l_1}{u_1}$. Then, for $k > 1$ we obtain the following

$$\frac{\partial l_k}{\partial \alpha} = \frac{\partial}{\partial \alpha}\left(\frac{c_k}{u_k}\right) = -\frac{l_k}{u_k} \cdot \frac{\partial u_k}{\partial \alpha} = \frac{l_k}{u_k}\left(1 + b_{k-1} \frac{\partial l_{k-1}}{\partial \alpha}\right).$$

□

Now we compute the partial derivatives for u_k. We note, based on Algorithm 2.1, that u_k is a function of $a_1, ..., a_k,\ c_1, ..., c_{k-1},\ b_1, ..., b_{k-1},\ \alpha$.

Lemma 3.5. *Let $J(c, b, a)$ be a $n \times n$ tridiagonal matrix. If α is a real number such that $J(c, a, b) - \alpha I$ has a unique LU factorization, then u_k has the following partial derivatives with respect to $a_1, ..., a_k, c_1, ..., c_{k-1}, b_1, ..., b_{k-1}, \alpha$.*

$$\frac{\partial u_k}{\partial a_i} = \begin{cases} \dfrac{l_{k-1}b_{k-1}}{u_{k-1}}\dfrac{\partial u_{k-1}}{\partial a_i}, & 1 \le i < k, \\ 1, \quad i = k. \end{cases}$$

$$\frac{\partial u_k}{\partial c_i} = \begin{cases} \dfrac{l_{k-1}b_{k-1}}{u_{k-1}}\dfrac{\partial u_{k-1}}{\partial c_i}, & 1 \le i < k-1, \\ -\dfrac{b_{k-1}}{u_{k-1}}, \quad i = k-1. \end{cases}$$

$$\frac{\partial u_k}{\partial b_i} = \begin{cases} \dfrac{l_{k-1}b_{k-1}}{u_{k-1}}\dfrac{\partial u_{k-1}}{\partial b_i}, & 1 \le i < k-1, \\ -l_{k-1}, \quad i = k-1. \end{cases}$$

$$\frac{\partial u_k}{\partial \alpha} = \begin{cases} -1 + \dfrac{l_{k-1}b_{k-1}}{u_{k-1}}\dfrac{\partial u_{k-1}}{\partial \alpha}, & 2 \le k, \\ -1, \quad k = 1. \end{cases}$$

Proof. First we consider $\partial u_k/\partial a_i$. Taking into account the u_k expression that follows from Algorithm 2.1, the case for $i = k$ is easy to check. Consider the case for $1 \le i < k$. By the chain rule

$$\frac{\partial u_k}{\partial a_i} = \frac{\partial u_k}{\partial l_{k-1}} \cdot \frac{\partial l_{k-1}}{\partial u_{k-1}} \cdot \frac{\partial u_{k-1}}{\partial a_i}$$

By Algorithm 2.1

$$\frac{\partial u_k}{\partial l_{k-1}} = -b_{k-1}, \text{ and } \quad \frac{\partial l_{k-1}}{\partial u_{k-1}} = -\frac{l_{k-1}}{u_{k-1}}.$$

Hence, the expression for $\partial u_k/\partial a_i$ follows. Likewise, we can derive the expressions for $\partial u_k/\partial c_i$ and $\partial u_k/\partial b_i$ in a similar fashion.

Finally, $\frac{\partial u_1}{\partial \alpha} = -1$. For $k > 1$ we obtain the following expression:

$$\frac{\partial u_k}{\partial \alpha} = \frac{\partial}{\partial \alpha}(a_k - \alpha - l_{k-1}b_{k-1}) = -1 - b_{k-1}\frac{\partial l_{k-1}}{\partial \alpha} = -1 + \frac{l_{k-1}b_{k-1}}{u_{k-1}}\frac{\partial u_{k-1}}{\partial \alpha}. \quad (3.4)$$

□

Next we define some auxiliary quantities that will be useful to give a recursive formula for the condition number $condC(J(c, b, a), \alpha)$. Let us call

$$condC_{abc}(l_k) := \sum_{i=1}^{k}\left|\frac{a_i}{l_k}\frac{\partial l_k}{\partial a_i}\right| + \sum_{i=1}^{k}\left|\frac{c_i}{l_k}\frac{\partial l_k}{\partial c_i}\right| + \sum_{i=1}^{k-1}\left|\frac{b_i}{l_k}\frac{\partial l_k}{\partial b_i}\right|, \quad (3.5)$$

$$condC_{abc}(u_k) := \sum_{i=1}^{k}\left|\frac{a_i}{u_k}\frac{\partial u_k}{\partial a_i}\right| + \sum_{i=1}^{k-1}\left|\frac{c_i}{u_k}\frac{\partial u_k}{\partial c_i}\right| + \sum_{i=1}^{k-1}\left|\frac{b_i}{u_k}\frac{\partial u_k}{\partial b_i}\right|, \quad (3.6)$$

These quantities can be computed recursively as the following lemma shows.

Lemma 3.6.

$$condC_{abc}(u_k) = \left|\frac{a_k}{u_k}\right| + \left|\frac{l_{k-1}b_{k-1}}{u_k}\right|(2 + condC_{abc}(u_{k-1})), \quad for\ k \geq 2,$$

where $condC_{a,b,c}(u_1) = \left|\frac{a_1}{u_1}\right|$.

$$condC_{abc}(l_k) = 1 + \left|\frac{a_k}{u_k}\right| + \left|\frac{l_{k-1}b_{k-1}}{u_k}\right|(1 + condC_{abc}(l_{k-1})), \quad for\ k \geq 2,$$

where $condC_{a,b,c}(l_1) = 1 + \left|\frac{a_1}{u_1}\right|$.

We can now explicitly compute the condition number $condC(J(c, a, b), \alpha)$. We present recursion formulas which have been derived from (3.2), (3.3), and Lemmas 3.4, 3.5 and 3.6.

Theorem 3.7.

$$condC(J(c, a, b), \alpha) = \max\{\max_{1\leq k\leq n}\{condC(u_k)\}, \max_{1\leq k\leq n-1}\{condC(l_k)\}\},$$

where

$$condC(u_k) = \left|\frac{a_k}{u_k}\right| + \left|\frac{l_{k-1}b_{k-1}}{u_k}\right|(2 + condC_{a,b,c}(u_{k-1})) + \left|\frac{\alpha}{u_k}\cdot\frac{\partial u_k}{\partial\alpha}\right|, \tag{3.7}$$

$$condC(l_k) = 1 + \left|\frac{a_k}{u_k}\right| + \left|\frac{l_{k-1}b_{k-1}}{u_k}\right|(1 + condC_{a,b,c}(l_{k-1})) + \left|\frac{\alpha}{l_k}\cdot\frac{\partial l_k}{\partial\alpha}\right|. \tag{3.8}$$

The cost to compute $condC(J(c, a, b), \alpha)$ is $17n - 20$. Therefore, we have got an expression for the condition number that can be computed in $O(n)$ flops.

In the sequel we give an alternative proof of the forward stability of Algorithm 2.1. First we define a new condition number for the LU factorization of shifted tridiagonal matrices. Now we consider a different kind of perturbation of the initial data, perturbations associated with the backward error found in Theorem 2.2.

Definition 3.8. Let $L(l, 1)$ and $U(u, b)$ be the matrices obtained from the exact LU factorization of $J(c, b, a) - \alpha I$, where $J(c, b, a)$ is a $n\times n$ tridiagonal matrix and α is a real number. Let $L(l + \Delta l, 1)$ and $U(u + \Delta u, b + \Delta b)$ be the factors obtained from the LU factorization of $J(c+\Delta c, b+\Delta b, a+\Delta a) - (\alpha+\Delta\alpha)I$. Let us define

$$DB = \max\left\{\max_{1\leq i\leq n}\left\{\frac{|\Delta a_i|}{|a_i| + |u_i| + |l_{i-1}b_{i-1}|}\right\}, \max_{1\leq i\leq n-1}\left\{\frac{|\Delta c_i|}{|c_i|}, \frac{|\Delta b_i|}{|b_i|}\right\}, \frac{|\Delta\alpha|}{|\alpha|}\right\},$$

where any quotient has to be understood as zero if the corresponding denominator is equal to zero. Then the relative componentwise condition number of the shifted tridiagonal LU factorization with respect to perturbations associated to the backward errors in Theorem 2.2 is defined as

$$condB(J(c, b, a), \alpha) = \lim_{\mathbf{u}\to 0}\sup_{0\leq DB\leq\mathbf{u}}\frac{\max\left\{\max_{1\leq i\leq n}\left\{\left|\frac{\Delta u_i}{u_i}\right|\right\}, \max_{1\leq i\leq (n-1)}\left\{\left|\frac{\Delta l_i}{l_i}\right|\right\}\right\}}{DB}.$$

A recursive expression for this new condition number can be found similarly to how $condC(J(c,a,b),\alpha)$ was computed.

Theorem 3.9.

$$condB(J(c,a,b),\alpha) = \max\{\max_{1\leq k\leq n}\{condB(u_k)\}, \max_{1\leq k\leq n-1}\{condB(l_k)\}\},$$

where

$$condB(u_k) = 1 + \left|\frac{a_k}{u_k}\right| + \left|\frac{l_{k-1}b_{k-1}}{u_k}\right|(3 + condB_{a,b,c}(u_{k-1})) + \left|\frac{\alpha}{u_k}\cdot\frac{\partial u_k}{\partial\alpha}\right|, \quad (3.9)$$

$$condB(l_k) = 2 + \left|\frac{a_k}{u_k}\right| + \left|\frac{l_{k-1}b_{k-1}}{u_k}\right|(2 + condB_{a,b,c}(l_{k-1})) + \left|\frac{\alpha}{l_k}\cdot\frac{\partial l_k}{\partial\alpha}\right|. \quad (3.10)$$

Notice that, taking into account Theorem 2.2,

$$\max_k\left\{\left|\frac{l_k-\hat{l}_k}{l_k}\right|, \left|\frac{u_k-\hat{u}_k}{u_k}\right|\right\} \leq 3\epsilon\, condB(J(c,a,b),\alpha) + O(\epsilon^2),$$

and therefore, Algorithm 2.1 is forward stable if $condB(J(c,a,b),\alpha)$ has the same order of magnitude as $condC(J(c,a,b),\alpha)$.

The proof of the following lemma is straightforward.

Lemma 3.10. *For $k = 1,...,n-1$*

$$condC(l_k) \leq condB(l_k) \leq 2\; condC(l_k),$$

Notice that $condC(u_k) \leq condB(u_k)$ is also a trivial result. Now we must prove that $condB(u_k) \leq R\; condC(u_k)$ for some moderate constant R. However this is not a straightforward result. Let us define $cond_\alpha(u_k) := \left|\frac{\alpha}{u_k}\cdot\frac{\partial u_k}{\partial\alpha}\right|$. Then,

$$condC(u_k) = condC_{a,b,c}(u_k) + cond_\alpha(u_k).$$

The next lemma is the key to proving the remaining inequality.

Lemma 3.11. *For all $k = 1,...,n$,*

$$condC(u_k) \geq 1.$$

Proof.

$$condC_{abc}(u_k) = condC_a(u_k) + condC_b(u_k) + condC_c(u_k),$$

where $condC_a(u_k) = \sum_{i=1}^{k}\left|\frac{a_i}{u_k}\frac{\partial u_k}{\partial a_i}\right|$, $condC_c(u_k) = \sum_{i=1}^{k-1}\left|\frac{c_i}{u_k}\frac{\partial u_k}{\partial c_i}\right|$ and $condC_b(u_k) = \sum_{i=1}^{k-1}\left|\frac{b_i}{u_k}\frac{\partial u_k}{\partial b_i}\right|$. It can easily been proven that

$$condC_a(u_k) = \left|\frac{a_k}{u_k}\right| + \sum_{i=1}^{k-1}\left|\frac{a_i}{u_i}\right|\prod_{j=i}^{k-1}\left|\frac{l_jb_j}{u_{j+1}}\right|.$$

$$condC_b(u_k) = condC_c(u_k) = \sum_{i=1}^{k-1}\prod_{j=i}^{k-1}\left|\frac{l_jb_j}{u_{j+1}}\right|.$$

$$cond_\alpha(u_k) = \left| \frac{\alpha}{u_k} + \sum_{i=1}^{k-1} \frac{\alpha}{u_k} \prod_{j=i}^{k-1} \frac{l_j b_j}{u_j} \right|.$$

Taking into account that $\alpha = a_i - u_i - l_{i-1}b_{i-1}$ for $i = 1, ..., n$, we get

$$\begin{aligned} cond_\alpha(u_k) &= \left| -1 + \frac{a_k}{u_k} - \frac{l_{k-1}b_{k-1}}{u_k} + \sum_{i=1}^{k-1} \frac{a_i - u_i - l_{i-1}b_{i-1}}{u_i} \prod_{j=i}^{k-1} \frac{l_j b_j}{u_{j+1}} \right| \\ &= \left| 1 - \frac{a_k}{u_k} - \sum_{i=1}^{k-1} \frac{a_i}{u_i} \prod_{j=i}^{k-1} \frac{l_j b_j}{u_{j+1}} + 2 \sum_{i=1}^{k-1} \prod_{j=i}^{k-1} \frac{l_j b_j}{u_{j+1}} \right| \\ &\geq 1 - condC_a(u_k) - 2\, condC_b(u_k) \end{aligned}$$

and the result follows. □

Lemma 3.12. *For $k \geq 1$,*

$$condC(u_k) \leq condB(u_k) \leq 3\, condC(u_k).$$

Proof. The first inequality is obvious. Let us prove the second one. Notice that

$$condB(u_k) = 1 + \left| \frac{a_k}{u_k} \right| + \sum_{i=1}^{k-1} \left(4 + \left| \frac{a_i}{u_i} \right| \right) \prod_{j=i}^{k-1} \left| \frac{l_j b_j}{u_{j+1}} \right| + cond_\alpha(u_k).$$

$$condC(u_k) = \left| \frac{a_k}{u_k} \right| + \sum_{i=1}^{k-1} \left(2 + \left| \frac{a_i}{u_i} \right| \right) \prod_{j=i}^{k-1} \left| \frac{l_j b_j}{u_{j+1}} \right| + cond_\alpha(u_k).$$

Then, taking into account Lemma 3.11,

$$condB(u_k) \leq condC(u_k) + 2\, condC(u_k),$$

and the result follows. □

Theorem 3.13. *Let $J(c, a, b)$ be a tridiagonal matrix and let α be a real number such that $J(c, a, b) - \alpha I$ has a unique LU factorization. Then,*

$$condC(J(c, a, b), \alpha) \leq condB(J(c, a, b), \alpha) \leq 3\, condC(J(c, a, b), \alpha).$$

This result implies that the LU factorization of shifted tridiagonal matrices is componentwise forward stable.

Proof. The proof can be obtained using the definition of $condC(J(c, a, b), \alpha)$ and $condB(J(c, a, b), \alpha)$, and taking into account Lemmas 3.10 and 3.12. □

References

[1] C. Brittin, and M. I. Bueno, *A note on the stability of the LU factorization of Hessenberg matrices*, preprint.

[2] M. I. Bueno, and F. M. Dopico, *Stability and sensitivity of tridiagonal LU factorization without pivoting*, BIT 44(2004), 651-673.

[3] F. Chaitin-Chatelin, and V. Frayssé, *Lectures on finite precision computations*, SIAM, Philadelphia (1996).

[4] 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), 21-80.

[5] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd. ed., SIAM, Philadelphia (2002).

[6] B. N. Parlett, *The new qd algorithms*, in Acta Numerica 1995, Acta Numer., Cambridge University Press, Cambridge, UK 1995, 459-491.

C. Brittin
100 Lord Pelham Way
Yorktown, VA 23693
USA
e-mail: cabrittin@gmail.com

M. I. Bueno
Mathematics Department,
University of California,
South Hall, Room 6607
Santa Barbara, CA 93106 USA
e-mail: mbueno@math.ucsb.edu