Electronic Transactions on Numerical Analysis.
Volume 18, pp. 101-136, 2004.

ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

STABILITY AND SENSITIVITY OF DARBOUX TRANSFORMATION WITHOUT PARAMETER *

M. ISABEL BUENO† AND FROILÁN M. DOPICO‡

Abstract. The monic Jacobi matrix is a tridiagonal matrix which contains the parameters of the three-term recurrence relation satisfied by the sequence of monic polynomials orthogonal with respect to a measure. Darboux transformation without parameter changes a monic Jacobi matrix associated with a measure μ into the monic Jacobi matrix associated with $xd\mu$. This transformation has been used in several numerical problems as in the computation of Gaussian quadrature rules. In this paper, we analyze the stability of an algorithm which implements Darboux transformation without parameter numerically and we also study the sensitivity of the problem. The main result of the paper is that, although the algorithm for Darboux transformation without parameter is not backward stable, it is forward stable. This means that the forward errors are of similar magnitude to those produced by a backward stable algorithm. Moreover, bounds for the forward errors computable with low cost are presented. We also apply the results to some classical families of orthogonal polynomials.

Key words. Darboux transformation, orthogonal polynomials, stability, sensitivity, tridiagonal matrices, LU factorization, LR algorithm.

AMS subject classifications. 65G50, 42C05, 15A23, 65F30, 65F35.

1. INTRODUCTION.

Let μ be an absolutely continuous measure on the real line, that is, $d\mu = \omega(x)dx$, where ω is a weight function. Suppose that

$$\left| \int_R x^n d\mu \right| < \infty, \quad \text{for } n \geq 0,$$

where, if μ is a positive measure, then $\omega(x)$ is a monotonically increasing function while, if μ is a signed measure, then $\omega(x)$ is a function of bounded variation. Let us consider a sequence of monic polynomials $\{P_n\}$ orthogonal with respect to the measure μ, i.e.,

1. $degree(P_n) = n, \text{ for } n \geq 0.$
2. $\int_R P_n P_m d\mu = K_{n,m}\delta_{nm}, \quad K_{n,n} \neq 0.$

This sequence of monic orthogonal polynomials $\{P_n\}$ satisfies a three-term recurrence relation

$$xP_n(x) = P_{n+1}(x) + B_{n+1}P_n(x) + G_nP_{n-1}(x), \quad n \geq 0,$$

$$P_{-1}(x) = 0, \; P_0(x) = 1.$$

In matrix notation,

$$xP = JP,$$

*Received December 19, 2003. Accepted for publication July 17, 2004. Recommended by L. Reichel. 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).

†Departamento de Matemáticas, Universidad Carlos III de Madrid, Avda. de la Universidad, 30. 28911 Leganés, Spain. E-mail: mbueno@math.uc3m.es, mibueno@math.wm.edu.

‡Departamento de Matemáticas, Universidad Carlos III de Madrid. E-mail: dopico@math.uc3m.es.

ETNA
Kent State University
etna@mcs.kent.edu

where $P = \begin{bmatrix} P_0(x) \\ P_1(x) \\ P_2(x) \\ \vdots \end{bmatrix}$ and

$$J = \begin{bmatrix} B_1 & 1 & 0 & 0 & \dots \\ G_1 & B_2 & 1 & 0 & \dots \\ 0 & G_2 & B_3 & 1 & \dots \\ 0 & 0 & G_3 & B_4 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \tag{1.1}$$

The matrix J is said to be the *monic Jacobi matrix* associated with $\{P_n\}$.

In the context of applications, it is interesting to study how the monic Jacobi matrix J changes when the measure μ is multiplied by a polynomial, i.e., $\pi(x)d\mu(x)$, with $\pi(x)$ a polynomial. The Darboux transformation without parameter, in the case of semi-infinite tridiagonal matrices, is the process that allows us to obtain the monic Jacobi matrix associated with $xd\mu$. It has been shown [8, 9, 3] that the Darboux transform of a semi-infinite tridiagonal matrix J can be computed as follows: 1)compute the LU factorization without pivoting of J, where the main diagonal elements of L are one, 2) multiply the factors L and U in reverse order, i.e., UL. Notice, as pointed out in [8, 17], that this algorithm is an infinite version of one step of the LR algorithm. But the finite version of our algorithm presents a significant difference: once the factors U and L have been multiplied, a one rank matrix must be added. In practice, we will not add this one rank matrix but delete the last row and column of the output matrix.

More generally, the Darboux transformation without parameter and with shift multiplies the measure by any polynomial of degree one, $(ax + b)d\mu$. The Darboux transformation (with parameter) can also be considered [13, 14]. It is just the inverse process of Darboux transformation without parameter. We will not study these kinds of transformations in this paper although in fact they are very interesting.

In the fields of Numerical Analysis, Approximation Theory, Differential Equations and Orthogonal Polynomials, the Darboux process has been used in different ways. For instance, Golub and Kautsky [12, 17], as well as Galant [9], applied Darboux transformation with shift to the calculation of Gaussian quadratures, in particular, to those with multiple free and fixed knots. Taking into account that the eigenvalues of any leading principal submatrix of a monic Jacobi matrix coincide with the zeros of the orthogonal polynomial of degree equal to the order of the submatrix, the knots of the quadratures can be calculated as the eigenvalues of some Jacobi matrices. A. Grünbaum and L. Haine have used Darboux transformation without parameter to obtain Krall polynomials from classical families of orthogonal polynomials in the context of the bispectral problem ([13] and [14]). It is important to mention a recent paper by Gautschi [11] in which the relation of the Darboux process with orthogonal polynomials and Gaussian quadratures is treated in a modern and affable manner.

Considering the importance given in the literature to the Darboux transformation without parameter, it seems to be necessary to give an answer to the following questions: 1) Is the natural algorithm previously described for computing the Darboux transform without parameter of a monic Jacobi matrix numerically stable?; 2) what can we say about the conditioning of the problem? In fact, the answers to these questions are not trivial. Consider, for instance, that Darboux transformation without parameter is related to LU factorization without pivoting which is not a backward stable algorithm. To our knowledge, no formal error analysis of any algorithm to compute the Darboux transformation without parameter has been presented so far. However, it should be noticed that in references [8], [10] and [17] some assertions on the stability of the computation of shifted Darboux transformation have been made. These assertions are based on a few numerical experiments and are not conclusive.

The aim of this work is to develop a general and formal analysis of the stability of the usual algorithm to compute the *Darboux transformation without parameter in its unshifted version*, as well as to give bounds on the forward errors computable with low cost. The study of the sensitivity of the unshifted Darboux transformation without parameter will play a key role in this analysis. The main conclusion we present is that the usual algorithm to compute the Darboux transformation without parameter is *forward stable*, i.e., the forward errors are of similar magnitude to those produced by a backward stable algorithm. No need to say that this result does not imply that the forward errors are small, therefore it is important to introduce computable bounds for the forward errors. We have chosen to develop a componentwise error analysis because in many cases small componentwise forward errors are obtained in the computation of Darboux transformation without parameter. A normwise error analysis is also possible. We have not included it to keep the paper concise. The stability properties of Darboux transformation with shift are different and more difficult to understand. The authors are presently studying this question.

The structure of the paper is the following. In Section 2, some basic notions about orthogonal polynomials, kernel polynomials and Darboux transformation without parameter are given. This section is addressed to those who are not specialists in these aspects. In Section 3, a description of the algorithm is presented as well as the notation used in the paper and some numerical experiments with classical orthogonal polynomials. The analysis of stability is done in Section 4 where Theorem 4.3 is the most important result. The conditioning of the problem is analyzed in Section 5 where accurate bounds are given for two componentwise condition numbers of the unshifted Darboux transformation without parameter: with respect to small relative perturbations of each entry of the matrix and perturbations associated with the backward error. Moreover, a relation between both condition numbers is proven from which the forward stability can be deduced. Furthermore, it is shown that these condition numbers for $n \times n$ monic Jacobi matrices can be computed in $14n$ flops. This result cannot be significantly improved since Darboux process has $2(n-1)$ inputs and $2n-3$ outputs. Some special cases are studied in Section 6. In particular, diagonally dominant tridiagonal matrices and tridiagonal matrices with positive subdiagonal are considered. In Section 7, we study from a theoretical point of view the uniparametric families of classical orthogonal polynomials (Laguerre and Bessel) in terms of their condition numbers for the unshifted Darboux transformation without parameter. We prove that the condition number of Laguerre polynomials is bounded by 3 independently on the order of the monic Jacobi matrix or the parameter α which defines the particular sequence of polynomials. In the case of Bessel polynomials, we show that the corresponding condition number is bounded by a quadratic polynomial in n when the parameter is positive. In Section 8, symmetric Darboux transformation is studied and compared with the monic one. It is shown that in the case of positive measures supported in $(0, \infty)$ both transformations produce backward and forward errors of similar magnitude. Finally, in Section 9, a wide sample of numerical experiments showing the reliability of the stability and sensitivity analysis is presented.

2. DARBOUX TRANSFORMATION AND ORTHOGONAL POLYNOMIALS. In this section, we give some basic notions about Darboux transformation without parameter and its relation with orthogonal polynomials. Readers specialized in orthogonal polynomials can skip straight to Section 3.

Consider a linear functional with real values $\mathbf{L}$ defined on the linear space of polynomials with real coefficients. Then, $\mathbf{L}$ is said to be *quasi-definite* if there exists a sequence of monic polynomials $\{P_n\}$ orthogonal with respect to $\mathbf{L}$, i.e.,

- If $P_n(x) = a_n x^n + a_{n-1}x^{n-1} + ... + a_1 x + a_0$, then $a_n = 1$,
- $degree(P_n) = n, \quad$ for $n \geq 0$,
- $\mathbf{L}(P_n, P_m) = K_{n,m}\delta_{nm}, \quad K_{n,n} \neq 0.$

Because of Boas's and Duran's Theorems [1, 7], we can assure that $\mathbf{L}$ has an integral representation, i.e., there exist a Borel measure μ and a weight function ω such that

$$\mathbf{L}(p) = \int_R p\, d\mu = \int_R p(x)\, \omega(x) dx.$$

Furthermore, $\{P_n\}$ satisfies a three-term recurrence relation,

$$P_{n+1}(x) = (x - B_{n+1})P_n(x) - G_n P_{n-1}(x), \quad n \geq 0, \tag{2.1}$$

$$P_{-1}(x) = 0,\ P_0(x) = 1,\ G_n \neq 0,\ \forall n.$$

In particular, $G_n > 0$ when μ is a positive measure.

Recall from (1.1) that, associated with the sequence of monic polynomials $\{P_n\}$, there exists the monic Jacobi matrix, J . But, if we consider a positive measure μ, it is possible to construct a sequence of orthonormal polynomials with respect to μ. In this case, the corresponding tridiagonal matrix is also *symmetric* and it is called *Jacobi matrix*. For the sake of clarity, we will refer to the *Jacobi matrix* as *symmetric Jacobi matrix* and we will use the notation J_s to denote it.

$$J_s = \begin{bmatrix} B_1 & \sqrt{G_1} & 0 & 0 & \dots \\ \sqrt{G_1} & B_2 & \sqrt{G_2} & 0 & \dots \\ 0 & \sqrt{G_2} & B_3 & \sqrt{G_3} & \dots \\ 0 & 0 & \sqrt{G_3} & B_4 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \tag{2.2}$$

It is interesting to point out the following properties [5].

PROPOSITION 2.1. *The eigenvalues of the leading principal submatrix of J of order n are the zeros of the orthogonal polynomial P_n. The same holds for J_s.*

PROPOSITION 2.2. *Let μ be a positive measure supported in a subset of the real line containing infinitely many points. If $\{P_n\}$ denotes the sequence of monic polynomials orthogonal with respect to μ, then the zeros of P_n are in the convex hull of the support of the measure for any n.*

COROLLARY 2.3. *Let J_s be a Jacobi matrix associated with a positive measure μ supported in an interval contained in $(0, \infty)$. Then, J_s is a symmetric positive definite matrix.*

2.1. Darboux transformation and kernel polynomials. In the sequel, we will consider the modification $\mathbf{xL}$ of the linear functional $\mathbf{L}$.

Given a quasi-definite linear functional $\mathbf{L}$, the linear functional $\mathbf{xL}$ [5] is given by

$$(\mathbf{xL})(p) := \mathbf{L}(xp) = \int_R x\, p(x)\, \omega(x) dx.$$

If $\{P_n\}$ is the sequence of monic polynomials orthogonal with respect to $\mathbf{L}$ and $P_n(0) \neq 0$, for all $n \geq 0$, then $\mathbf{xL}$ is a quasi-definite linear functional. The polynomials orthogonal with respect to $\mathbf{xL}$ are the so-called *kernel polynomials* associated with $\{P_n\}$ [5].

There is a close relation between the monic Jacobi matrix associated with $\mathbf{xL}$ and the monic Jacobi matrix associated with $\mathbf{L}$. In fact, the first one can be obtained from the second one by the application of the so-called Darboux transformation without parameter.

DEFINITION 2.4. *Given a monic Jacobi matrix J, consider the tridiagonal matrix J_1 such that*

$$J = LU,$$

$$J_1 = UL,$$

where $J = LU$ denotes the LU factorization without pivoting of J in such a way that the elements of the main diagonal of L are ones. We say that J_1 is the Darboux transform of J and the process that generates J_1 from J is the so-called Darboux transformation without parameter.

In the sequel, for the sake of simplicity, we will refer to Darboux transformation without parameter as *Darboux transformation*.

THEOREM 2.5. *[3] Let $\mathbf{L}$ be a quasi-definite linear functional and $\{P_n\}$ the corresponding sequence of monic orthogonal polynomials. If $P_n(0) \neq 0$, for all $n \geq 0$, then the Darboux transform of J is the monic Jacobi matrix associated with $\mathbf{xL}$.*

The previous result was already known by Galant and Gautschi although, in [9], Galant gives a simpler result which is valid only for the case when the linear functional is given in terms of a positive measure. In [10], Gautschi presents the same result but he did not formulate it in matrix form.

In order to compute the Darboux transformation in practice, it is necessary to give an equivalent result to Theorem 2.5 in the finite case. Recall that a monic Jacobi matrix is a semi-infinite matrix. The next result is a trivial consequence of Theorem 2.5. We introduce the following shorthand notation: for any matrix A, A_n denotes the $n \times n$ leading principal submatrix of A.

COROLLARY 2.6. *[3] Let $\mathbf{L}$ be a quasi-definite linear functional and $\{P_n\}$ the corresponding sequence of monic orthogonal polynomials. If $P_n(0) \neq 0$, for all $n \geq 0$ and $J_n = L_n U_n$ denotes the LU factorization without pivoting of the leading principal submatrix of order n of J, then $(J_1)_{n-1} = (U_n L_n)_{n-1}$ is the leading principal submatrix of order $n-1$ of the monic Jacobi matrix associated with $\mathbf{xL}$.*

REMARK 2.1. *Notice that from a monic Jacobi matrix of order n, Darboux transformation only provides a monic Jacobi matrix of order $n-1$.*

3. ALGORITHM AND NUMERICAL EXPERIMENTS. In the next section, we will analyze the stability of an algorithm which implements Darboux transformation numerically. Our analysis is based on the special structure of monic Jacobi matrices. In this section we start with a careful description of the algorithm that implements Darboux transformation and also introduce the notation used in the rest of the paper. In Subsection 3.1 some important remarks on the input and output of this algorithm are given. Finally some numerical experiments are presented in Subsection 3.2. From now on all the results refer to leading

ETNA
Kent State University
etna@mcs.kent.edu

principal submatrices of monic Jacobi matrices or symmetric Jacobi matrices. Since we are interested in the numerical analysis of the algorithm that implements Darboux transformation and symmetric Darboux transformation, we have to restrict ourselves to finite matrices.

Let us consider the $n \times n$ monic Jacobi matrix,

$$J_n = \begin{bmatrix} B_1 & 1 & 0 & \cdots & 0 \\ G_1 & B_2 & 1 & \cdots & 0 \\ 0 & G_2 & B_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & G_{n-1} & B_n \end{bmatrix}. \tag{3.1}$$

For the sake of simplicity, we use the following notation

$$J(B,G) := J_n,$$

$$B = [B_1, ..., B_n]^T,\ G = [G_1, ..., G_{n-1}]^T.$$

The same kind of notation is considered for symmetric Jacobi matrices,

$$J_s(B, \sqrt{G}) := (J_s)_n = \begin{bmatrix} B_1 & \sqrt{G_1} & 0 & \cdots & 0 \\ \sqrt{G_1} & B_2 & \sqrt{G_2} & \cdots & 0 \\ 0 & \sqrt{G_2} & B_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sqrt{G_{n-1}} & B_n \end{bmatrix}. \tag{3.2}$$

In the rest of the paper it is assumed that all the monic or symmetric Jacobi matrices we consider have a unique LU factorization without pivoting, and that $G_i \neq 0$, for all i, according to the results summarized in the previous section.

Taking into account Corollary 2.6, the finite version of Darboux transformation includes the following steps.

Matrix description of Darboux transformation of order n.

1. LU factorization without pivoting of $J(B,G)$.

$$J(B,G) = LU$$

where

$$L = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ l_1 & 1 & \cdots & 0 & 0 \\ 0 & l_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & l_{n-1} & 1 \end{bmatrix},\ U = \begin{bmatrix} u_1 & 1 & \cdots & 0 & 0 \\ 0 & u_2 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & u_{n-1} & 1 \\ 0 & 0 & \cdots & 0 & u_n \end{bmatrix}. \tag{3.3}$$

2. Multiplication of U times L.

$$\tilde{J} = UL$$

3. Deletion of the last row and column of $\tilde{J}$ which gives the matrix $(J_1)_{n-1}$.

Notice that $(J_1)_{n-1}$ is an $(n-1)\times(n-1)$ monic Jacobi matrix. In the sequel, $J(b,g) := (J_1)_{n-1}$, where

$$J(b,g) = \begin{bmatrix} b_1 & 1 & 0 & \cdots & 0 & 0 \\ g_1 & b_2 & 1 & \cdots & 0 & 0 \\ 0 & g_2 & b_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & b_{n-2} & 1 \\ 0 & 0 & 0 & \cdots & g_{n-2} & b_{n-1} \end{bmatrix} \tag{3.4}$$

and

$$b = [b_1, ..., b_{n-1}]^T, \; g = [g_1, ..., g_{n-2}]^T.$$

The following MATLAB code computes the LU factorization without pivoting of $J(B,G)$:

ALGORITHM 3.1. *Given a monic Jacobi matrix $J(B,G) \in R^{n\times n}$, this algorithm computes the LU factorization without pivoting of $J(B,G)$.*

```
u(1)=B(1)
for i=1:n-1
   l(i)=G(i)/u(i)
   u(i+1)=B(i+1)-l(i)
end
```

Although the previous algorithm is not our main goal and it appears implicitly in the algorithm that computes Darboux transformation, for the sake of clarity, we give it explicitly since we will refer to it.

The following algorithm computes the Darboux transform of a monic Jacobi matrix, $J(B,G)$:

ALGORITHM 3.2. *Given a monic Jacobi matrix $J(B,G) \in R^{n\times n}$, this algorithm computes its Darboux transform of order $n-1$, $J(b,g) \in R^{(n-1)\times(n-1)}$.*

```
u(1)=B(1)
for i=1:n-2
    l(i)=G(i)/u(i)
    b(i)=u(i)+l(i)
    u(i+1)=B(i+1)-l(i)
    g(i)=u(i+1)l(i)
end
l(n-1)=G(n-1)/u(n-1)
b(n-1)=u(n-1)+l(n-1)
```

The computational cost of Algorithm 3.2 is $4n-6$ flops.

3.1. Input, output and matrix notation. In order to compute the $(n-1)\times(n-1)$ leading principal submatrix of the Darboux transform of $J(B,G)$, Algorithm 3.2 only needs the vectors $B(1:n-1)$ and G, i.e, the element B_n is not needed. Therefore, Algorithm 3.2 has $2(n-1)$ input parameters

$$B' = [B_1, B_2, ..., B_{n-1}], \quad G = [G_1, G_2, ..., G_{n-1}]$$

ETNA
Kent State University
etna@mcs.kent.edu

and $2n-3$ output parameters

$$b = [b_1, b_2, ..., b_{n-1}], \quad g = [g_1, g_2, ..., g_{n-2}].$$

We also observe that, in order to compute $J(b, g)$, we do not need to compute the complete LU factorization of $J(B, G)$. In fact, we just need to compute $[u_1, u_2, ..., u_{n-1}]$ and $[l_1, l_2, ..., l_{n-1}]$. Therefore, it can be said that Darboux transformation is a function depending on the parameters B' and G. Hence, in order to study the backward error or the perturbation theory related with Darboux transformation, we only have to consider these parameters.

Although this is precise from a mathematical point of view, we think that it is not intuitive. Therefore, we will frequently use the matrix description of the Darboux process in the statements of the theorems appearing in the backward error analysis and in the perturbation theory of Darboux transformation as if the input parameters were B and G. It is clear from the discussion above that, in these theorems, B_n and u_n do not play any role.

3.2. Numerical Experiments. Now we show a few numerical experiments in which we apply Algorithm 3.2 for computing the Darboux transformation to some classical families of orthogonal polynomials. These experiments present a typical behaviour which illustrates why the structured stability analysis presented in this paper is needed. Recall [5] that a sequence of orthogonal polynomials is said to be classical if it satisfies a linear second order differential equation with polynomial coefficients

$$a_2(x)y'' + a_1(x)y' + a_0(x)y + \lambda y = 0,$$

where $a_0(x)$, $a_1(x)$ and $a_2(x)$ are polynomials of degrees at most 0, 1 and 2, respectively and $\lambda \neq 0$, i.e., for each integer $n \geq 0$, the corresponding classical orthogonal polynomial P_n is an eigenfunction of a second order differential operator $a_2(x)D'' + a_1(x)D' + a_0(x)I$ with $\lambda = \lambda_n$, the corresponding eigenvalue.

In particular, we consider Jacobi, Laguerre and Bessel polynomials. Both Laguerre and Bessel polynomials are uniparametric families while Jacobi polynomials are a biparametric family, i.e., we can obtain different sequences of Laguerre or Bessel polynomials by varying the value of one parameter while the family of Jacobi polynomials depends on two parameters. The differences among these three families are related to the measures with respect to which they are orthogonal as well as to the features of the corresponding symmetric Jacobi matrices (3.2).

1. Jacobi polynomials are orthogonal with respect to a positive measure supported in $[-1, 1]$. The corresponding symmetric Jacobi matrix is not positive definite.
2. Laguerre polynomials are orthogonal with respect to a positive measure supported in $(0, \infty)$. The corresponding symmetric Jacobi matrix is positive definite.
3. Bessel polynomials are orthogonal with respect to a signed measure. Therefore, the symmetric Jacobi matrix does not exist.

Notice that Bessel polynomials are the only ones among the classical families that are orthogonal with respect to a signed measure [5].

In the following experiments we compare the output of Algorithm 3.2 in the floating point arithmetic of MATLAB 5.3 ($\mathbf{u} = 1.11 \times 10^{-16}$) with the output computed by MAPLE. More precisely, exact values of the outputs b_i and g_i (see (3.4)) are calculated by MAPLE, and then these expressions are rounded numerically to 32 decimal digits of precision.

In this section, $\hat{b} = [\hat{b}_1, ..., \hat{b}_{n-1}]$ and $\hat{g} = [\hat{g}_1, ..., \hat{g}_{n-2}]$ denote the quantities computed by MATLAB and $b = [b_1, ..., b_{n-1}]$, $g = [g_1, ..., g_{n-2}]$ denote the quantities computed by

MAPLE. In the next tables the following quantities are shown: the **componentwise forward errors** of b and g

$$forb = \max_{k=1..n-1} \left\{ \left| \frac{b_k - \hat{b}_k}{b_k} \right| \right\}, \quad forg = \max_{k=1..n-2} \left\{ \left| \frac{g_k - \hat{g}_k}{g_k} \right| \right\}, \tag{3.5}$$

and the **normwise forward errors** of b and g,

$$FORb = \frac{||b - \hat{b}||}{||J(b,g)||}, \quad FORg = \frac{||g - \hat{g}||}{||J(b,g)||}. \tag{3.6}$$

REMARK 3.1. *In the previous definition, we use the max norm both for vectors and matrices, i.e.,*

$$||v|| = \max_i \{|v(i)|\}, \text{ if } v \text{ is a vector},$$

$$||A|| = \max_{i,j} \{|A(i,j)|\}, \text{ if } A \text{ is a matrix}.$$

In the next tables J denotes the monic Jacobi matrix which is the input of Algorithm 3.2, while L and U denote the factors of the LU factorization of J. We include the results obtained along with the spectral condition numbers of the matrices involved in the process[1]. We also include $maxL := \max_i\{|l_i|\}$ where l_i denotes the entry of L in the position $(i+1, i)$.

1. Results for Laguerre Polynomials with parameter $1/10$.

	n=10	n=50	n=100
forb	$1.5\ 10^{-16}$	$1.5\ 10^{-16}$	$1.56\ 10^{-16}$
forg	$2.11\ 10^{-16}$	$2.15\ 10^{-16}$	$2.25\ 10^{-16}$
FORb	$1.95\ 10^{-17}$	$5.79\ 10^{-18}$	$2.87\ 10^{-18}$
FORg	$1.56\ 10^{-16}$	$1.85\ 10^{-16}$	$1.83\ 10^{-16}$
maxL	10	50	100
$\kappa_2(J)$	$5.4\ 10^{7}$	$4.46\ 10^{64}$	$1.39\ 10^{158}$
$\kappa_2(L)$	$5.7\ 10^{7}$	$> 10^{64}$	$> 10^{158}$
$\kappa_2(U)$	11.9	53.84	106.29

2. Results for Jacobi polynomials with parameters $1, -1/2$.

	n=10	n=50	n=100
forb	$1.84\ 10^{-16}$	$3.17\ 10^{-16}$	$4.74\ 10^{-16}$
forg	$7.16\ 10^{-16}$	$7.9\ 10^{-16}$	$1.27\ 10^{-15}$
FORb	$1.63\ 10^{-16}$	$3.11\ 10^{-16}$	$4.62\ 10^{-16}$
FORg	$6.51\ 10^{-16}$	$7.8\ 10^{-16}$	$1.23\ 10^{-15}$
maxL	1.15	1.19	1.2
$\kappa_2(J)$	$2.86\ 10^{3}$	$3.17\ 10^{15}$	$2.5\ 10^{30}$
$\kappa_2(L)$	3.91	4.36	4.4
$\kappa_2(U)$	$6.03\ 10^{3}$	$7.07\ 10^{15}$	$> 10^{30}$

[1]The spectral condition numbers of the matrices J, L and U have been computed making use of the variable precision arithmetic of the Symbolic Math Toolbox of MATLAB.

3. Results for Bessel polynomials with parameter $1/2$.

	n=10	n=50	n=100
forb	$2.7\ 10^{-15}$	$2.02\ 10^{-14}$	$1.05\ 10^{-13}$
forg	$4.96\ 10^{-16}$	$1.23\ 10^{-15}$	$1.65\ 10^{-15}$
FORb	$1.11\ 10^{-16}$	$1.11\ 10^{-16}$	$1.11\ 10^{-16}$
FORg	$1.39\ 10^{-17}$	$1.39\ 10^{-17}$	$1.39\ 10^{-17}$
maxL	0.23	0.23	0.23
$\kappa_2(J)$	$1.52\ 10^{9}$	$6.44\ 10^{78}$	$1.56\ 10^{187}$
$\kappa_2(L)$	1.34	1.34	1.34
$\kappa_2(U)$	$1.59\ 10^{9}$	$6.7\ 10^{78}$	$1.58\ 10^{187}$

From the previous results, the following remarks arise. The numerical computation of normwise forward errors shows that Algorithm 3.2 applied to the previous specific examples produces relative forward errors $O(\mathbf{u})$. This is also the case for componentwise forward errors except in the case of Bessel polynomials, where a moderate dependence on the dimension of the problem appears. On the other hand, it is well known that LU factorization without pivoting is not backward stable, in general, especially when the growth factor is large [16, Lemma 9.6]. But, even for small growth factors, large forward errors may appear in the L, U factors, because the usual bound of the normwise condition number of LU factorization is (see [16, section 9.11])

$$\chi(J) = \|L^{-1}\|_2 \|U^{-1}\|_2 \|J\|_2.$$

This bound, $\chi(J)$, on the condition number of LU factorization can be improved in an almost optimal way [19] to $(\min_{\text{D diagonal}} \kappa_2(LD))\kappa_2(U)$ for the L factor and $\kappa_2(L)(\min_{\text{D diagonal}} \kappa_2(DU))$ for the U factor. The values of $\kappa_2(J)$, $\kappa_2(L)$ and $\kappa_2(U)$ appearing in the three examples above show that all of them have one of the factors L or U ill-conditioned and, then, large forward errors should be expected in the L or U factor. Moreover, in two of the examples (Laguerre and Jacobi), since not all the entries l_i of the matrix L are, in absolute value, less than one, pivoting is neccesary to ensure the backward stability of LU factorization. Finally, the second step of the algorithm (multiplication UL) is not backward stable either.

These facts imply that large errors should be expected, but they are not observed in the experiments we have shown for Laguerre, Jacobi and Bessel polynomials. Therefore, our goal is to develop a structured analysis of stability that explains these and other examples. It is important to stress that, although in the three specific examples discussed in this section, and in many others, the forward errors of Darboux transformation are of the order of machine precision, this is not always the case. For instance, in Section 9 some examples are presented for which the Darboux transformation of Bessel polynomials is computed with very large forward errors. Therefore, accurate bounds which allow us to estimate the magnitude of the forward error have to be developed.

The previous remarks show that the analysis of the stability of Algorithm 3.2 is not trivial, one of the main reasons being that the structure of the mathematical problem does not allow the use of pivoting.

4. BACKWARD ERROR ANALYSIS. In this section we assume that the elements of the monic Jacobi matrix $J(B, G)$ are real floating point numbers. This assumption may seem unsuitable for the problems we are dealing with. Consider, for instance, the three examples discussed in the previous section. In these examples the elements of $J(B, G)$ are computed by using well known formulas for the classical families of orthogonal polynomials (see Sec.

7). Therefore, some rounding errors are neccesarily present in the input parameters B, G of Algorithm 3.2. If these errors on the input change significantly the exact value of Darboux transformation, then the errors produced specifically by Algorithm 3.2 may not govern the overall forward errors. However, as we will show in Section 5 (see Remark 5.5), the forward errors coming from small relative componentwise perturbations of B and G, are of the same magnitude as the forward errors coming from Algorithm 3.2 applied to floating point numbers. Thus our assumption on the input being floating point numbers can be done without spoiling the applicability of our analysis and simplifies somewhat the subsequent developments.

In the stability analysis, we use the standard model of floating point arithmetic:

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

where x and y are floating point numbers , op $= +, -, *, /$ and $\mathbf{u}$ is the unit roundoff of the machine. Moreover, we assume that

$$fl(\sqrt{x}) = \sqrt{x}(1+\delta) = \frac{\sqrt{x}}{1+\eta}, \quad |\delta|, |\eta| \le \mathbf{u}.$$

In the sequel, given a matrix A, then $|A|$ denotes the matrix whose entries are the absolute values of the entries of A.

We start giving a stability result for LU factorization without pivoting of monic Jacobi matrices. Although more general results for tridiagonal matrices are given in [16], the following theorem improves slightly those results taking into account that the elements in the superdiagonal are ones.

THEOREM 4.1. *Let Algorithm 3.1 be applied to the monic Jacobi matrix $J(B,G)$ of order n. Then the computed factors $\hat{L}$, $\hat{U}$ satisfy*

$$J(B+\Delta B, G+\Delta G) = \hat{L}\hat{U}, \quad |\Delta B| \le \mathbf{u}|diag(\hat{U})|, \quad |\Delta G| \le \mathbf{u}|G|.$$

Proof. For the computed quantities, we have

$$\hat{l}_i = \frac{G_i}{\hat{u}_i}(1+\delta_i), \quad |\delta_i| \le \mathbf{u}.$$

Therefore, $|G_i - \hat{u}_i\hat{l}_i| \le |G_i|\mathbf{u}$, which proves the theorem for the entries in the subdiagonal, G. On the other hand,

$$\hat{u}_i(1+\epsilon_i) = B_i - \hat{l}_{i-1}, \quad |\epsilon_i| \le \mathbf{u}.$$

Then, $|B_i - \hat{u}_i - \hat{l}_{i-1}| \le |\hat{u}_i|\mathbf{u}$, which proves the theorem. □

Next we consider the second step of Darboux transformation, i.e., multiplication of U and L. The proof of the following theorem is immediate.

THEOREM 4.2. *Let U and L be matrices as those appearing in (3.3). Let $J(b,g) = UL$ be the exact product of U times L, where the notation in Section 3 is used. Let $J(\hat{b},\hat{g})$ be the computed product of U times L, then*

$$|b-\hat{b}| \le \mathbf{u}|\hat{b}|, \quad \textit{and} \quad |g-\hat{g}| \le \mathbf{u}|\hat{g}|,$$

which, in matrix notation, implies:

$$J(\hat{b}+\Delta\hat{b}, \hat{g}+\Delta\hat{g}) = UL, \quad \textit{with} \quad |\Delta\hat{b}| \le \mathbf{u}|\hat{b}| \quad \textit{and} \quad |\Delta\hat{g}| \le \mathbf{u}|\hat{g}|.$$

ETNA
Kent State University
etna@mcs.kent.edu

Next we give the stability result for Darboux transformation.

THEOREM 4.3. *Given a monic Jacobi matrix of order n, $J(B,G)$, let $J(b,g)$ be its Darboux transform of order $n-1$. If $J(\hat{b},\hat{g})$, $\hat{L}$ and $\hat{U}$ are the matrices computed by Algorithm 3.2, then*

$$J(B+\Delta B, G+\Delta G) = \hat{L}\hat{U}, \quad |\Delta B| \leq \mathbf{u}|diag(\hat{U})|, \quad |\Delta G| \leq \mathbf{u}|G|,$$

$$J(\hat{b}+\Delta\hat{b}, \hat{g}+\Delta\hat{g}) = (\hat{U}\hat{L})_{n-1}, \quad |\Delta\hat{b}| \leq \mathbf{u}|\hat{b}|,\ |\Delta\hat{g}| \leq \mathbf{u}|\hat{g}|.$$

Proof. It is enough to consider Theorems 4.1 and 4.2. □

Notice that Theorem 4.3 asserts that the computed Darboux transform $J(\hat{b},\hat{g})$ is almost the exact Darboux transform of $J(B+\Delta B, G+\Delta G)$. Therefore, we conclude that the Algorithm 3.2 will be *componentwise stable in the mixed forward-backward sense* [16, p. 7] if $|\hat{u}_i| = O(|B_i|)$, $1 \leq i \leq n$.

In the particular case when the monic Jacobi matrix $J(B,G)$ is associated with a positive measure supported on $(0,\infty)$, Algorithm 3.2 is componentwise stable. The next lemma proves this assertion.

LEMMA 4.4. *Let $J(B,G)$ be a monic Jacobi matrix of order n associated with a positive measure supported on $(0,\infty)$. Then, the LU factorization of $J(B,G)$ is componentwise backward stable.*

Proof. Since $J(B,G)$ is associated with a positive measure, $G_i > 0$, $i = 1:n-1$. Then, there exists a diagonal matrix D such that $J_s(B,\sqrt{G}) = D\ J(B,G)\ D^{-1}$. Moreover, $J_s(B,G)$ is positive definite because the measure is supported in $(0,\infty)$. If $J(B,G) = LU$ denotes the unique LU factorization of $J(B,G)$, taking into account that

$$J_s(B,\sqrt{G}) = (DLD^{-1})diag(U)(DLD^{-1})^T,$$

we have $u_i > 0$ and, subsequently, $l_i > 0$, for all $i \geq 1$. Therefore, since $B_i = u_i + l_{i-1}$, $u_i \leq B_i$. □

REMARK 4.1. *Laguerre polynomials are orthogonal with respect to a positive measure supported on $(0,\infty)$ but not Jacobi and Bessel polynomials. Next we show that the LU factorization of the monic Jacobi matrix associated with* ***Jacobi polynomials*** *is not componentwise backward stable. From Theorem 4.1,*

$$\frac{|\Delta B_i|}{|B_i|} \leq \mathbf{u}\frac{|\hat{u}_i|}{|B_i|}, \quad \frac{|\Delta G_i|}{|G_i|} \leq \mathbf{u}.$$

Then, if "errback" *denotes* $\max_{i=1:n}\{1, \frac{|\hat{u}_i|}{|B_i|}\}$, *for Jacobi polynomials with parameters* 1, *and* $-1/2$, *considered in previous section, we get*

	n=10	*n=50*	*n=100*	*n=400*	*n=1000*
errback	$7.07\ 10^2$	$1.64\ 10^4$	$6.5\ 10^4$	$1.03\ 10^6$	$6.44\ 10^6$

The previous results show that Darboux transformation is not componentwise stable for Jacobi polynomials. However, the forward errors obtained for parameters $1, -1/2$ are of order $\mathbf{u}$ *(see table in Subsection 3.2). Therefore, to explain these errors it is necessary to find the componentwise condition number with respect to the kind of perturbations suggested by the backward error analysis.*

ETNA
Kent State University
etna@mcs.kent.edu

5. CONDITIONING. From now on we analyze the sensitivity of Darboux transformation under perturbations of the initial data, i.e., the monic Jacobi matrix $J(B,G)$. We consider two kinds of perturbations:

- Perturbations associated with the backward error found in Theorem 4.3.
- Relative componentwise perturbations in B and G, i.e., $|\Delta B| \leq \epsilon|B|$ and $|\Delta G| \leq \epsilon|G|$, with small ϵ.

In both cases, the superdiagonal of ones stays unperturbed.

We measure the sensitivity of the problem by the notion of componentwise relative condition number. In fact, since we consider two different kinds of perturbations, we define two different condition numbers.

DEFINITION 5.1. *Let $J(b,g)$ be the Darboux transform of order $(n-1)$ of a monic Jacobi matrix of order n, $J(B,G)$, and $J(b+\Delta b, g+\Delta g)$ be the Darboux transform of order $(n-1)$ of the monic Jacobi matrix of order n, $J(B+\Delta B, G+\Delta G)$. Let $J(B,G) = LU$ be the unique LU factorization of $J(B,G)$. The componentwise relative condition number of the Darboux transformation of the monic Jacobi matrix $J(B,G)$ with respect to perturbations associated with backward errors is defined as*

$$cond_B(J(B,G)) := \lim_{\epsilon\to 0} \sup \left\{ \max_k \left\{ \left|\frac{\Delta b_k}{\epsilon b_k}\right|, \left|\frac{\Delta g_k}{\epsilon g_k}\right| \right\} : \ |\Delta B| \leq \epsilon |diag(U)|, \ |\Delta G| \leq \epsilon|G| \right\}.$$

DEFINITION 5.2. *Let $J(b,g)$ be the Darboux transform of order $(n-1)$ of a monic Jacobi matrix of order n, $J(B,G)$, and $J(b+\Delta b, g+\Delta g)$ be the Darboux transform of order $(n-1)$ of the monic Jacobi matrix of order n, $J(B+\Delta B, G+\Delta G)$. The componentwise relative condition number of the Darboux transformation of the monic Jacobi matrix $J(B,G)$ with respect to perturbations in components is defined as*

$$cond_C(J(B,G)) := \lim_{\epsilon\to 0} \sup \left\{ \max_k \left\{ \left|\frac{\Delta b_k}{\epsilon b_k}\right|, \left|\frac{\Delta g_k}{\epsilon g_k}\right| \right\} : \ |\Delta B| \leq \epsilon |B|, \ |\Delta G| \leq \epsilon|G| \right\}.$$

REMARK 5.1. *In the two previous definitions* $\max_k \left\{ \left|\frac{\Delta b_k}{\epsilon b_k}\right|, \left|\frac{\Delta g_k}{\epsilon g_k}\right| \right\}$ *denotes*

$$\max \left\{ \left|\frac{\Delta b_1}{\epsilon b_1}\right|, \ldots, \left|\frac{\Delta b_{n-1}}{\epsilon b_{n-1}}\right|, \left|\frac{\Delta g_1}{\epsilon g_1}\right|, \ldots, \left|\frac{\Delta g_{n-2}}{\epsilon g_{n-2}}\right| \right\}.$$

It is well known that the forward errors produced by an algorithm can be bounded by the backward error times the condition number. Taking into account Definition 5.1 and Theorem 4.3, it is easy to get the following bound for the forward error produced by the algorithm that implements Darboux transformation.

LEMMA 5.3. *Let $J(b,g)$ and $J(\hat{b},\hat{g})$ be, respectively, the exact and the computed Darboux transform of $J(B,G)$ from Algorithm 3.2, then*

$$\max_k \left\{ \left|\frac{b_k - \hat{b}_k}{b_k}\right|, \left|\frac{g_k - \hat{g}_k}{g_k}\right| \right\} \leq \mathbf{u}(1 + cond_B(J(B,G))) + O(\mathbf{u}^2).$$

Notice that one has to be added to $cond_B(J(B,G))$ because Theorem 4.3 is a mixed forward-backward error result.

ETNA
Kent State University
etna@mcs.kent.edu

In the next subsection we give some auxiliary results that are necessary to find explicit expressions for reliable bounds for the condition numbers given by Definitions 5.1 and 5.2. In Subsection 5.2 we obtain those expressions as well as a relation between both bounds, which essentially shows that $cond_C(J(B,G))$ and $cond_B(J(B,G))$ are of similar magnitude.

5.1. Auxiliary results. This section contains some lemmas that are necessary to prove some of the most important results of this paper. Readers who are not interested in technical details can skip straight to Subsection 5.2.

Let $J(B,G)$ and $J(B+\Delta B, G+\Delta G)$ be monic Jacobi matrices such that the corresponding LU factorization without pivoting exist and let

$$J(B,G) = LU,$$

$$J(B+\Delta B, G+\Delta G) = (L+\Delta L)(U+\Delta U).$$

Moreover, the corresponding Darboux transforms are denoted by $J(b,g)$ and $J(b+\Delta b, g+\Delta g)$.

Taking into account Algorithm 3.1 and Subsection 3.1, it must be remembered that, for the perturbed matrix,

$$u_1 + \Delta u_1 = B_1 + \Delta B_1, \tag{5.1}$$

$$l_k + \Delta l_k = \frac{G_k + \Delta G_k}{u_k + \Delta u_k}, \quad k = 1 : n-1, \tag{5.2}$$

$$u_{k+1} + \Delta u_{k+1} = B_{k+1} + \Delta B_{k+1} - l_k - \Delta l_k, \quad k = 1 : n-2. \tag{5.3}$$

Moreover, taking into account Algorithm 3.2

$$b_k + \Delta b_k = u_k + \Delta u_k + l_k + \Delta l_k, \quad k = 1 : n-1, \tag{5.4}$$

$$g_k + \Delta g_k = (u_{k+1} + \Delta u_{k+1})(l_k + \Delta l_k), \quad k = 1 : n-2. \tag{5.5}$$

Our first goal is to find expressions for the elements Δu_k and Δl_k in terms of the elements of ΔB and ΔG. The following two lemmas are particular instances of lemmas given in [2]. The proofs of both results are trivial. In the rest of the section second order terms in ϵ are not considered because the condition numbers $cond_B(J(B,G))$ and $cond_C(J(B,G))$ are defined in the limit $\epsilon \to 0$.

REMARK 5.2. *In the sequel, we assume that any term containing u_0, l_0, G_0 or B_0 is zero. Moreover, $\Delta u_0 = \Delta l_0 = \Delta B_0 = \Delta G_0 = 0$.*

LEMMA 5.4. *The following result is correct to first order*

$$\Delta l_k = l_k \left(\frac{\Delta G_k}{G_k} - \frac{\Delta u_k}{u_k} \right), \quad k = 1 : n-1, \tag{5.6}$$

$$\Delta u_k = \Delta B_k - \Delta l_{k-1}, \quad k = 1 : n-1. \tag{5.7}$$

LEMMA 5.5. *The following recurrence relation is obtained to first-order:*

$$\Delta u_k = \Delta B_k - \frac{\Delta G_{k-1}}{u_{k-1}} + \frac{l_{k-1}}{u_{k-1}}\Delta u_{k-1}, \quad k = 1:n-1.$$

Proof. The result is obtained immediately by substituting (5.6) into (5.7). □

In Lemma 5.6 an explicit expression for Δu_k is obtained from the recurrence relation given in the previous lemma.

LEMMA 5.6. *The following expression is correct to first order.*

$$\Delta u_k = \Delta B_k - \frac{\Delta G_{k-1}}{u_{k-1}} + \sum_{i=1}^{k-1}\left(\Delta B_i - \frac{\Delta G_{i-1}}{u_{i-1}}\right)\prod_{j=i}^{k-1}\frac{l_j}{u_j}, \quad k = 1:n-1.$$

Proof. The result is obtained by induction on the expression given in Lemma 5.5. □

Our next aim is to find expressions of the elements Δb_k and Δg_k in terms of Δu_k.
LEMMA 5.7. *The following equations are correct to first order.*

$$\Delta b_k = \Delta u_k + \Delta l_k, \quad k = 1:n-1,$$

$$\Delta g_k = l_k \Delta u_{k+1} + u_{k+1}\Delta l_k, \quad k = 1:n-2.$$

Proof. From (5.4), the first result is obtained straightforwardly. On the other hand, taking into account (5.5), the second result is obtained to first order. □

From Lemma 5.7 it is possible to obtain an expression for $\frac{\Delta b_k}{b_k}$ and $\frac{\Delta g_k}{g_k}$ in terms of $\frac{\Delta u_k}{u_k}$. In the sequel, for any number a,

$$\delta a := \frac{\Delta a}{a}.$$

LEMMA 5.8.

$$\delta b_k = \frac{l_k}{u_k + l_k}\delta G_k + \frac{u_k - l_k}{u_k + l_k}\delta u_k, \quad k = 1:n-1, \tag{5.8}$$

$$\delta g_k = \left(1 + \frac{l_k}{u_{k+1}}\right)\delta B_{k+1} + \left(1 - \frac{l_k}{u_{k+1}}\right)(\delta G_k - \delta u_k), \quad k = 1:n-2. \tag{5.9}$$

Proof. From Lemma 5.7 and taking into account that $b_k = u_k + l_k$,

$$\delta b_k = \frac{u_k}{u_k + l_k}\delta u_k + \frac{l_k}{u_k + l_k}\delta l_k. \tag{5.10}$$

ETNA
Kent State University
etna@mcs.kent.edu

Then, considering the expression for Δl_k in terms of Δu_k given in Lemma 5.4

$$\delta b_k = \frac{u_k}{u_k + l_k}\delta u_k + \frac{l_k}{u_k + l_k}(\delta G_k - \delta u_k),$$

and (5.8) follows in a straightforward way.

From Lemma 5.7 again and taking into account that $g_k = l_k u_{k+1}$, we get

$$\delta g_k = \delta u_{k+1} + \delta l_k = \delta u_{k+1} + \delta G_k - \delta u_k. \tag{5.11}$$

Then, from Lemma 5.5, it is possible to express δg_k only in terms of δu_k,

$$\delta g_k = \delta G_k + \left(1 + \frac{l_k}{u_{k+1}}\right)\delta B_{k+1} - \frac{l_k}{u_{k+1}}(\delta G_k - \delta u_k) - \delta u_k$$

and the result follows. □

In order to find an explicit expression for $cond_B(J(B,G))$ and $cond_C(J(B,G))$, it is necessary to find bounds for $|\delta b_k|$ and $|\delta g_k|$. We first consider the perturbations associated with the backward error appearing in the definition of $cond_B(J(B,G))$.

LEMMA 5.9. *Let us assume that* $|\Delta B_k| \leq \epsilon |u_k|$, *and* $|\Delta G_k| \leq \epsilon |G_k|$ *hold for* $k = 1 : n-1$. *Then, to first order,*

$$|\delta u_k| \leq \epsilon \left[1 + \left|\frac{l_{k-1}}{u_k}\right| + \sum_{i=1}^{k-1}\left(1 + \left|\frac{l_{i-1}}{u_i}\right|\right)\prod_{j=i}^{k-1}\left|\frac{l_j}{u_{j+1}}\right|\right]. \tag{5.12}$$

Or equivalently

$$|\delta u_k| \leq \epsilon \left[1 + 2\sum_{i=1}^{k-1}\prod_{j=i}^{k-1}\left|\frac{l_j}{u_{j+1}}\right|\right]. \tag{5.13}$$

Proof. From Lemma 5.6, we get

$$\frac{\Delta u_k}{u_k} = \frac{\Delta B_k}{u_k} - \frac{\Delta G_{k-1}}{u_k u_{k-1}} + \sum_{i=1}^{k-1}\left(\frac{\Delta B_i}{u_i} - \frac{\Delta G_{i-1}}{u_i u_{i-1}}\right)\prod_{j=i}^{k-1}\frac{l_j}{u_{j+1}},$$

which implies (5.12). Some extra calculations lead us to the second result. □

DEFINITION 5.10. *For k=1:n-1,*

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

Now we get bounds for δb_k and δg_k taking into account the perturbations associated with the backward error given in Theorem 4.3.

LEMMA 5.11. *Let us assume that* $|\Delta B_k| \leq \epsilon |u_k|$, *and* $|\Delta G_k| \leq \epsilon |G_k|$ *hold for* $k = 1 : n-1$. *Then, to first order,*

$$|\delta b_k| \leq \epsilon \left\{\left|\frac{l_k}{u_k + l_k}\right| + \left|\frac{u_k - l_k}{u_k + l_k}\right| cond_B(u_k)\right\}, \quad k = 1 : n-1,$$

ETNA
Kent State University
etna@mcs.kent.edu

$$|\delta g_k| \le \epsilon \left\{ 1 + \left| 1 - \frac{l_k}{u_{k+1}} \right| (1 + cond_B(u_k)) \right\}, \quad k = 1 : n-2.$$

Proof. The results are easily obtained from (5.13) and Lemma 5.8. □

REMARK 5.3. *For the sake of simplicity, we denote by $cond_B(b_k)$ and $cond_B(g_k)$ the bounds for $|\delta b_k|$ and $|\delta g_k|$ divided by ϵ, i.e.,*

$$cond_B(b_k) := \left| \frac{l_k}{u_k + l_k} \right| + \left| \frac{u_k - l_k}{u_k + l_k} \right| cond_B(u_k), \tag{5.14}$$

$$cond_B(g_k) := 1 + \left| 1 - \frac{l_k}{u_{k+1}} \right| (1 + cond_B(u_k)). \tag{5.15}$$

We consider now perturbations in components as in Definition 5.2 and obtain the corresponding lemmas similar to Lemmas 5.9 and 5.11.

DEFINITION 5.12. *For k=1:n-1,*

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

LEMMA 5.13. *Let us assume that $|\Delta B_k| \le \epsilon |B_k|$, and $|\Delta G_k| \le \epsilon |G_k|$ hold for $k = 1 : n-1$. Then, to first order,*

$$|\delta u_k| \le \epsilon \; cond_C(u_k).$$

Proof. The proof is similar to the proof of Lemma 5.9. □

LEMMA 5.14. *Let us assume that $|\Delta B_k| \le \epsilon |B_k|$ and $|\Delta G_k| \le \epsilon |G_k|$ hold for $k = 1 : n-1$. Then, to first order,*

$$|\delta b_k| \le \epsilon \left\{ \left| \frac{l_k}{u_k + l_k} \right| + \left| \frac{u_k - l_k}{u_k + l_k} \right| cond_C(u_k) \right\},$$

$$|\delta g_k| \le \epsilon \left\{ \left| 1 + \frac{l_k}{u_{k+1}} \right| + \left| 1 - \frac{l_k}{u_{k+1}} \right| [1 + cond_C(u_k)] \right\}.$$

Proof. The proof is similar to the proof of Lemma 5.11. □

REMARK 5.4. *Considering the bounds obtained in the previous lemma, we denote by $cond_C(b_k)$ and $cond_C(g_k)$ the bounds for $|\delta b_k|$ and $|\delta g_k|$ divided by ϵ.*

$$cond_C(b_k) := \left| \frac{l_k}{u_k + l_k} \right| + \left| \frac{u_k - l_k}{u_k + l_k} \right| cond_C(u_k), \tag{5.16}$$

$$cond_C(g_k) := \left| 1 + \frac{l_k}{u_{k+1}} \right| + \left| 1 - \frac{l_k}{u_{k+1}} \right| [1 + cond_C(u_k)]. \tag{5.17}$$

ETNA
Kent State University
etna@mcs.kent.edu

5.2. Condition numbers and relation between their magnitudes. In this subsection, we give explicit expressions for accurate bounds for the condition numbers $cond_B(J(B,G))$ and $cond_C(J(B,G))$, prove that both bounds are of similar magnitude and, finally, we also show how to compute the bounds of the condition numbers with cost O(n) flops. From the fact that both bounds for the condition numbers are of similar magnitude, the forward stability of Algorithm 3.2 is also deduced.

Taking into account Definitions 5.1 and 5.2 and the results of Subsection 5.1, given a monic Jacobi matrix $J(B,G)$,

$$cond_B(J(B,G)) \leq \max\{cond_B(b_{n-1}), \max_{k=1:n-2}\{cond_B(b_k), cond_B(g_k)\}\}, \tag{5.18}$$

$$cond_C(J(B,G)) \leq \max\{cond_C(b_{n-1}), \max_{k=1:n-2}\{cond_C(b_k), cond_C(g_k)\}\}, \tag{5.19}$$

where $cond_B(b_k)$ and $cond_B(g_k)$ have been defined in (5.14) and (5.15), respectively, as well as $cond_C(b_k)$ and $cond_C(g_k)$ have been defined in (5.16) and (5.17), respectively.

We have not been able to prove that the inequalities appearing in (5.18) and (5.19) are sharp, i.e., that there exist perturbations fulfilling the conditions of Definitions 5.1 and 5.2 for which the inequalities in Lemmas 5.11 and 5.14 become equalities. Therefore, we have not found expressions for the true condition numbers of Darboux transformation, $cond_B(J(B,G))$ and $cond_C(J(B,G))$. We will denote the bounds appearing in (5.18) and (5.19) by

$$bcond_B(J(B,G)) := \max\{cond_B(b_{n-1}), \max_{k=1:n-2}\{cond_B(b_k), cond_B(g_k)\}\} \tag{5.20}$$

$$bcond_C(J(B,G)) := \max\{cond_C(b_{n-1}), \max_{k=1:n-2}\{cond_C(b_k), cond_C(g_k)\}\} \tag{5.21}$$

Numerical experiments in Section 9 show that $bcond_B(J(B,G))$ is an accurate upper bound for $cond_B(J(B,G))$ and that it really reflects the sensitivity of the problem. Therefore it is fair to think of $bcond_B(J(B,G))$ and $bcond_C(J(B,G))$ as condition numbers. We would also like to comment that in some cases in which the signs of u_k and l_k have special relations, it is easy to see that $cond_B(J(B,G)) = bcond_B(J(B,G))$ and $cond_C(J(B,G)) = bcond_C(J(B,G))$. Finally, it is interesting to note that explicit expressions for the true condition numbers of the general tridiagonal LU factorization without pivoting have been found in [2].

The following Lemma will allow us to prove that the bounds $bcond_B(J(B,G))$ and $bcond_C(J(B,G))$ are of similar magnitude.

LEMMA 5.15.

$$\frac{1}{2} cond_C(b_k) \leq cond_B(b_k) \leq 2\, cond_C(b_k), \quad k = 1 : n-1,$$

$$\frac{1}{2} cond_C(g_k) \leq cond_B(g_k) \leq 2\, cond_C(g_k), \quad k = 1 : n-2.$$

Proof. In order to prove that $cond_C(b_k) \leq 2\, cond_B(b_k)$, just apply the triangular inequality to $cond_C(b_k)$. On the other hand, for any number a, $|1+a| \geq 1 - |a|$. Apply this inequality to $cond_C(b_k)$ to prove that $cond_B(b_k) \leq 2\, cond_C(b_k)$.

ETNA
Kent State University
etna@mcs.kent.edu

From Lemma 5.11, an alternative expression for $cond_B(g_k)$ can be obtained.

$$cond_B(g_k) = 1 + \left|1 - \frac{l_k}{u_{k+1}}\right| + \left|1 - \frac{l_k}{u_{k+1}}\right| cond_B(u_k). \tag{5.22}$$

For any number a, $|1+a| \leq 2 + |1-a|$. Apply this result as well as the triangular inequality to $cond_C(g_k)$ to prove that $cond_C(g_k) \leq 2\ cond_B(g_k)$. Compare the bound of $cond_C(g_k)$ with (5.22).

We can also get the following alternative expression for $cond_C(g_k)$

$$cond_C(g_k) = \left|1 + \frac{l_k}{u_{k+1}}\right| + \left|1 - \frac{l_k}{u_{k+1}}\right| + \left|1 - \frac{l_k}{u_{k+1}}\right| cond_C(u_k). \tag{5.23}$$

Then, taking into account that for any a, $|1+a| + |1-a| \geq 2$ and $|1+a| \geq 1 - |a|$ as well as (5.23), it is easy to prove that $2\ cond_C(g_k) \geq cond_B(g_k)$. □

As a consequence of the previous lemma, we obtain the following theorem which is one of the most important results in this paper.

THEOREM 5.16. *Given a monic Jacobi matrix $J(B,G)$,*

$$\frac{1}{2} bcond_C(J(B,G)) \leq bcond_B(J(B,G)) \leq 2\ bcond_C(J(B,G)).$$

REMARK 5.5. *As we have already remarked, the numerical experiments in Section 9 show that the bounds $bcond_B(J(B,G))$ and $bcond_C(J(B,G))$ really reflect the sensitivity of the Darboux transformation. Therefore, from Theorem 5.16, we can deduce the forward stability of Algorithm 3.2 to compute the unshifted Darboux transform of a monic Jacobi matrix $J(B,G)$. From Lemma 5.3*

$$\max_k \left\{ \left|\frac{b_k - \hat{b}_k}{b_k}\right|, \left|\frac{g_k - \hat{g}_k}{g_k}\right| \right\} \leq \mathbf{u}[bcond_B(J(B,G)) + 1] + O(\mathbf{u}^2) \leq$$

$$\leq \mathbf{u}[2\ bcond_C(J(B,G)) + 1] + O(\mathbf{u}^2).$$

The meaning of the above inequality is that the componentwise forward errors produced by Algorithm 3.2 are of the same order of magnitude as those produced by a componentwise backward stable algorithm. According to the definition appearing in [16, p.9], this means that Algorithm 3.2 is forward stable. Thus, the forward errors in Algorithm 3.2 are the "best one could expect" from the sensitivity of the problem.

Moreover, Theorem 5.16 guarantees that the stability analysis we have developed remains valid when the inputs of Algorithm 3.2, B and G, are not floating point numbers (recall the comments in the first paragraph of Section 4) but they are computed with componentwise errors of order $\mathbf{u}$. In this case, the first order overall relative componentwise forward errors in the computation of the Darboux transformation are the sum of two terms: $\mathbf{u}\ bcond_C(J(B,G))$ coming from the errors in B and G, plus $\mathbf{u}(1 + bcond_B(J(B,G)))$ coming from the errors produced by Algorithm 3.2. Thus a sensible overall error bound is $3\mathbf{u}(1 + bcond_B(J(B,G)))$, which implies that, up to inessential numerical factors, $bcond_B(J(B,G))$ is the quantity governing the forward errors.

The argument above can be trivially extended to the case in which B and G are computed with componentwise forward error bounded by a known constant times $\mathbf{u}$.

For practical purposes, it is important to prove that $bcond_B(J(B,G))$ can be computed with low cost.

LEMMA 5.17. *The quantities $cond_B(u_k)$, $k = 1 : n-1$, defined in Definition 5.10 satisfy the following recurrence relation*

$$cond_B(u_1) = 1, \quad cond_B(u_k) = 1 + \left|\frac{l_{k-1}}{u_k}\right|(1 + cond_B(u_{k-1})).$$

Proof. It suffices to substitute the last expression appearing in Definition 5.10 in the previous equation. □

Then, taking into account (5.14), (5.15) and (5.20) as well as Lemma 5.17, the cost to compute $bcond_B(J(B,G))$ is $14n - 22$ flops. Notice that, since the Algorithm 3.2 has $2(n-1)$ input parameters and $2n-3$ output, it is not possible to find a method to estimate the forward errors with computational cost less than $O(n)$.

6. PARTICULAR CASES. Next we apply the results obtained in Section 5 to two special kinds of monic Jacobi matrices:

1. Monic Jacobi matrices with positive G, i.e., corresponding to polynomials orthogonal with respect to a positive measure.
2. Monic Jacobi matrices diagonally dominant by rows and columns.

For the monic Jacobi matrices with positive G, we prove that the condition number $cond_B(J(B,G))$ only depends on the quotients $\left|\frac{l_j}{u_{j+1}}\right|$ since $\left|\frac{u_j-l_j}{u_j+l_j}\right| \leq 1$ and $\left|\frac{l_j}{u_j+l_j}\right| \leq 1$ for all j. For monic Jacobi matrices diagonally dominant by rows and columns, however, we prove that $\left|\frac{l_j}{u_{j+1}}\right| \leq 1$ for any j and the condition number depends on the other two kinds of quotients. Then, we conclude that for monic Jacobi matrices diagonally dominant by rows and columns with positive G, Darboux transformation is very well conditioned and small forward errors are obtained. Moreover, from Lemma 4.4, Darboux transformation is also componentwise mixed forward-backward stable in this case.

6.1. Monic Jacobi matrix associated with a positive measure. Consider the special case when a monic Jacobi matrix $J(B,G)$ is associated with a positive measure. Then, $G_k > 0$, $\forall k \geq 1$. Since $G_k = u_k l_k$, u_k and l_k have the same sign. Therefore,

$$\left|\frac{u_k - l_k}{u_k + l_k}\right| \leq 1, \quad \left|\frac{l_k}{u_k + l_k}\right| \leq 1,$$

and the expressions for $cond_B(b_k)$ and $cond_B(g_k)$ can be bounded in the following way:

$$cond_B(b_k) \leq 2 + 2\sum_{i=1}^{k-1}\prod_{j=i}^{k-1}\left|\frac{l_j}{u_{j+1}}\right|, \tag{6.1}$$

$$cond_B(g_k) 1 + \left|1 - \frac{l_k}{u_{k+1}}\right|\left(2 + 2\sum_{i=1}^{k-1}\prod_{j=i}^{k-1}\left|\frac{l_j}{u_{j+1}}\right|\right). \tag{6.2}$$

Equations (6.1) and (6.2) show that in the case of positive measures, $cond_B(J(B,G))$ is bounded by a function of the ratios $\left|\frac{l_j}{u_{j+1}}\right|$. Thus, if $\left|\frac{l_j}{u_{j+1}}\right| \leq 1,\ \forall j$,

$$cond_B(J(B,G)) \leq \max\{2n-2, 4n-7\}.$$

6.2. Diagonally dominant monic Jacobi matrices. Assume that the monic Jacobi matrix $J(B,G)$ is diagonally dominant by rows and columns simultaneously.

Then,

$$|B_j| > 1,\ |B_j| > |G_{j-1}|, \quad \text{for all } j \geq 1, \tag{6.3}$$

$$|u_j| > 1,\ |u_j| > |G_j|, \quad \text{for all } j \geq 1, \tag{6.4}$$

As an immediate consequence of (6.4)

$$|l_j| = \left|\frac{G_j}{u_j}\right| < 1, \quad j \geq 1. \tag{6.5}$$

Moreover, from (6.4) and (6.5),

$$\left|\frac{l_j}{u_{j+1}}\right| < 1, \quad j \geq 1. \tag{6.6}$$

Then,

$$cond_B(u_k) \leq 2k-1 \quad \text{and} \quad cond_B(g_k) \leq 4k+1.$$

Therefore, the elements $g_k,\ k = 1 : n-2$ are computed by Algorithm 3.2 with a "small" forward error. On the other hand,

$$cond_B(b_k) = \left|\frac{l_k}{u_k + l_k}\right| + (2k-1)\left|\frac{u_k - l_k}{u_k + l_k}\right|. \tag{6.7}$$

In the special case when l_k and u_k have the same sign for any k, i.e., when the measure is positive,

$$cond_B(b_k) \leq 2k.$$

Then, under this restraint, the whole matrix $J(\hat{b},\hat{g})$ is computed by Algorithm 3.2 with a "small" forward error. This also means that Darboux transformation is very well-conditioned for matrices associated with positive measures that are diagonally dominant by rows and columns.

7. BOUNDS FOR THE CONDITION NUMBER OF SOME FAMILIES OF CLASSICAL ORTHOGONAL POLYNOMIALS. This section has a marked theoretical orientation. Some analytic considerations relative to the value of the condition number of Darboux transformation of classical families of orthogonal polynomials are presented. This allows us to explain the good results obtained in the numerical experiments appearing in Subsection 3.2 for Laguerre and Bessel polynomials. We show that for Laguerre polynomials, independently of the value of the parameter α or the order n of the corresponding monic

ETNA
Kent State University
etna@mcs.kent.edu

Jacobi matrix, the condition number $cond_B(J(B,G))$ is bounded by 3. This means that Darboux transformation for Laguerre polynomials is stable (see Lemma 4.4) and besides small forward errors are produced by Algorithm 3.2. For Bessel polynomials and positive values of the parameter α, we prove that the condition number of Darboux transformation is bounded by a quadratic polynomial in n. Moreover, for sufficiently large values of $|\alpha|$, we prove that $bcond_B(J(B,G))$ tends to 3. We also make some remarks about numerical experiments applied to Bessel polynomials with negative values of α in which $bcond_B(J(B,G))$ takes large values. The analytical behaviour of the condition number of Jacobi polynomials has not been studied. Based on numerical experiments, we can say that $cond_B(J(B,G))$ is not large. For example, for monic Jacobi matrices of order 100, we have not found examples in which $bcond_B(J(B,G))$ is larger than 10^5 by using direct search methods [16, Chapter 26]. Moreover, for sufficiently large values of one of the parameters defining the Jacobi polynomials it can be shown that $bcond_B(J(B,G))$ tends to 3 as it occurs with Bessel polynomials. We have also found an example for which the corresponding monic Jacobi matrix has no LU factorization, ($\alpha = 17$ and $\beta = 40$).

The following lemma gives an expression for the elements u_k in terms of the values of the orthogonal polynomials evaluated at zero. Later on we will apply this result to classical families of orthogonal polynomials.

LEMMA 7.1. *Let $J = LU$ be the LU factorization without pivoting of the semi-infinite monic Jacobi matrix J associated with the sequence of monic orthogonal polynomials $\{P_n\}$. Then*

$$u_k = -\frac{P_k(0)}{P_{k-1}(0)}, \quad \textit{for all } k.$$

Proof. The result is obtained by induction on k, taking into account the three-term recurrence relation that $\{P_m\}$ satisfies (see (2.1)). Since $P_1(0) = -B_1$ and $P_0(0) = 1$,

$$u_1 = B_1 = -\frac{P_1(0)}{P_0(0)}.$$

Assume that $u_k = -\frac{P_k(0)}{P_{k-1}(0)}$ for $k \leq n$. Then, taking into account the three-term recurrence relation again

$$P_{n+1}(0) = -B_{n+1}P_n(0) - G_nP_{n-1}(0).$$

Dividing the previous expression by $P_n(0)$ and applying the induction hypotheses,

$$-\frac{P_{n+1}(0)}{P_n(0)} = B_{n+1} - \frac{G_n}{u_n} = B_{n+1} - l_n = u_{n+1}.$$

□

7.1. Laguerre Polynomials. Laguerre polynomials constitute a uniparametric family. For every value of the parameter, the corresponding sequence of polynomials is orthogonal with respect to a positive measure supported in $(0, \infty)$. Therefore, the results obtained in Section 6.1 can be applied to them.

ETNA
Kent State University
etna@mcs.kent.edu

Monic Laguerre polynomials of parameter $\alpha > -1$ satisfy the following three-term recurrence relation:

$$L_{n+1}^{\alpha}(x) = (x - B_{n+1}^{\alpha})L_n^{\alpha}(x) - G_n^{\alpha} L_{n-1}^{\alpha}(x), \quad n \geq 0, \tag{7.1}$$

$$L_0^{\alpha}(x) = 1, \quad L_{-1}^{\alpha}(x) = 0,$$

where

$$B_n^{\alpha} = 2n + \alpha - 1, \quad G_n^{\alpha} = n(n+\alpha). \tag{7.2}$$

LEMMA 7.2.

$$L_n^{\alpha}(0) = (-1)^n \prod_{i=1}^{n} (i + \alpha), \quad \textit{for all } n.$$

Proof. The result is obtained by induction on n and taking into account (7.1) and (7.2). □

Taking into account Lemmas 7.1 and 7.2 as well as the expression of l_j in terms of G_j and u_j, in the next lemma we give an explicit expression of u_j and l_j in terms of j and the parameter α.

LEMMA 7.3. *For Laguerre polynomials*

$$u_j = j + \alpha, \quad l_j = j, \quad \textit{for all } j.$$

Our aim in this section is to give a tight bound of $cond_B(J(B,G))$ when $J(B,G)$ is the $n \times n$ monic Jacobi matrix associated with Laguerre polynomials of parameter α. We give an explicit expression of $cond_B(b_k)$ and $cond_B(g_k)$ (recall Definition 5.10, (5.14), (5.15) and (5.20)) taking into account Lemma 7.3 that will allow us to obtain that bound. It is straightforward to obtain

$$\frac{l_j}{u_{j+1}} = \frac{j}{j+1+\alpha}, \quad \frac{l_j}{u_j + l_j} = \frac{j}{2j+\alpha}, \quad \frac{u_j - l_j}{u_j + l_j} = \frac{\alpha}{2j+\alpha}. \tag{7.3}$$

LEMMA 7.4. *For* $k \geq 1$,

$$cond_B(u_k) = \frac{2k+\alpha}{2+\alpha}.$$

Proof. By induction on k, we prove that

$$\sum_{i=1}^{k-1} \prod_{j=i}^{k-1} \left| \frac{l_j}{u_{j+1}} \right| = \frac{k-1}{2+\alpha}.$$

Then, taking into account Definition 5.10, the result is obtained. □

LEMMA 7.5.

$$cond_B(b_k) = \frac{k}{2k+\alpha} + \frac{|\alpha|}{2+\alpha}, \quad cond_B(g_k) = \frac{3\alpha+4}{2+\alpha}.$$

Proof. It suffices to consider (5.14), (5.15), Lemma 7.4 as well as (7.3) to obtain the results. □

THEOREM 7.6. *Let $J(B,G)$ be the monic Jacobi matrix of order n associated with Laguerre polynomials of parameter α. Then*

$$cond_B(J(B,G)) \leq 3.$$

Proof. Taking into account Lemma 7.5, it can be shown in a straightforward way that $cond_B(b_k) \leq 3$ and $cond_B(g_k) \leq 3$ for all k and α. Then, considering (5.18), the result is obtained. □

As a consequence of the developments in this section, we can assert that the errors produced by Algorithm 3.2 are optimal in the case of Laguerre polynomials. In the first place, it is componentwise stable in the mixed forward-backward sense (see Theorem 4.3 and Lemma 4.4) and in the second place, Lemma 5.3 and Theorem 7.6 guarantee that the forward error is less than $4\mathbf{u} + O(\mathbf{u}^2)$. Moreover, these results are independent of n.

7.2. Generalized Bessel polynomials. Bessel polynomials are orthogonal with respect to a signed measure. Bessel polynomials of parameter α satisfy the following three-term recurrence relation:

$$Q^\alpha_{n+1}(x) = (x - B^\alpha_{n+1})Q^\alpha_n(x) - G^\alpha_n Q^\alpha_{n-1}(x), \quad n \geq 1, \tag{7.4}$$

$$Q^\alpha_0(x) = 1, \quad Q^\alpha_1(x) = x + \frac{2}{2+\alpha}.$$

Moreover,

$$B^\alpha_n = \frac{-2\alpha}{(2n+\alpha)(2n+\alpha-2)}, \quad G^\alpha_n = \frac{-4n(n+\alpha)}{(2n+\alpha)^2(2n-1+\alpha)(2n+1+\alpha)}. \tag{7.5}$$

LEMMA 7.7.

$$Q^\alpha_k(0) = \frac{2^k}{\prod_{i=k+1}^{2k}(i+\alpha)}, \quad k \geq 0.$$

Proof. The result is obtained by induction on k and taking into account (7.4) and (7.5). □

In the next lemma we give explicit expressions for u_j and l_j in terms of j and the parameter α. This is again a simple consequence of Lemma 7.1.

LEMMA 7.8.

$$u_j = \frac{-2(j+\alpha)}{(2j-1+\alpha)(2j+\alpha)}, \quad l_j = \frac{2j}{(2j+\alpha)(2j+1+\alpha)}, \quad \textit{for all } j.$$

In order to estimate a bound for $cond_B(J(B,G))$, where $J(B,G)$ is the monic Jacobi matrix of order n associated with Bessel polynomials of parameter α, it is necessary to bound the quotients

$$\left|\frac{l_k}{u_{k+1}}\right|, \quad \left|\frac{l_k}{u_k+l_k}\right|, \quad \left|\frac{u_k-l_k}{u_k+l_k}\right|.$$

This can easily be done for positive values of the parameter α.

LEMMA 7.9. *If* $\alpha > 0$, *then* $\left|\frac{l_k}{u_{k+1}}\right| \leq 1$ *for any* $k \geq 1$.

Proof. Considering Lemma 7.8, we get

$$\left|\frac{l_k}{u_{k+1}}\right| = \left|\frac{-k(2k+2+\alpha)}{(2k+\alpha)(k+1+\alpha)}\right|. \tag{7.6}$$

Then, $\left|\frac{l_k}{u_{k+1}}\right| \leq 1$ if and only if

$$\frac{\alpha^2+\alpha(4k+1)+4k(k+1)}{(2k+\alpha)(k+1+\alpha)} \geq 0 \quad \text{and} \quad \frac{\alpha(\alpha+2k+1)}{(2k+\alpha)(k+1+\alpha)} \geq 0.$$

But it is easy to see that both inequalities are true for any $k \geq 1$ and $\alpha > 0$. ☐

Then, taking into account Lemma 7.9, if $J(b,g)$ denotes the Darboux transform of order $n-1$ of the monic Jacobi matrix of order n associated with Bessel polynomials of parameter $\alpha > 0$,

$$cond_B(b_k) \leq \left|\frac{l_k}{u_k+l_k}\right| + \left|\frac{u_k-l_k}{u_k+l_k}\right|(2k-1), \quad cond_B(g_k) \leq 4k+1. \tag{7.7}$$

LEMMA 7.10. *If* $\alpha > 0$, *then*

$$\left|\frac{l_k}{u_k+l_k}\right| \leq k.$$

Proof. Taking into account Lemma 7.8

$$\frac{l_k}{u_k+l_k}\frac{k(2k-1+\alpha)}{-\alpha^2-(2k+1)\alpha-2k}. \tag{7.8}$$

Then, it is easy to prove the lemma. ☐

LEMMA 7.11. *If* $\alpha > 0$, *then*

$$\left|\frac{u_k-l_k}{u_k+l_k}\right| \leq 2k.$$

Proof. In a similar way to the proof of the previous lemma, we get

$$\left|\frac{u_k-l_k}{u_k+l_k}\right|\frac{\alpha^2+(4k+1)\alpha+4k^2}{\alpha^2+(2k+1)\alpha+2k}. \tag{7.9}$$

Then, the result follows straightforwardly. ☐

Considering Lemmas 7.10 and 7.11 as well as (7.7) , it is easy to get a bound for $cond_B(J(B,G))$ when $J(B,G)$ is the $n \times n$ monic Jacobi matrix associated with Bessel polynomials of parameter α.

ETNA
Kent State University
etna@mcs.kent.edu

THEOREM 7.12. *Let $J(B,G)$ be the monic Jacobi matrix of order n associated with Bessel polynomials of parameter $\alpha > 0$. Then,*

$$cond_B(J(B,G)) \le 4n^2 - 9n + 5.$$

When the parameter α is negative, it is possible to find examples in which $bcond_B(J(B,G))$ is very large. For example, for $\alpha = -94/7$ and $n = 100$, $bcond_B(J(B,G)) = 6.32 \cdot 10^{18}$. For more details, see Section 9.

Things are completely different if we consider $|\alpha|$ sufficiently large.

THEOREM 7.13. *Let $J(B,G)$ be the monic Jacobi matrix of order n associated with Bessel polynomials of parameter α. Then,*

$$\lim_{\alpha\to\infty} bcond_B(J(B,G)) = 3, \qquad \lim_{\alpha\to-\infty} bcond_B(J(B,G)) = 3$$

Proof. Notice from (7.6), (7.8) and (7.9), that

$$\lim_{\alpha\to\infty}\left|\frac{l_j}{u_{j+1}}\right| = 0, \quad \lim_{\alpha\to\infty}\left|\frac{l_j}{u_j + l_j}\right| = 0, \quad \lim_{\alpha\to\infty}\left|\frac{u_j - l_j}{u_j + l_j}\right| = 1,$$

and the same results are obtained when considering the limit $\alpha \to -\infty$. Then, taking into account (5.14), (5.15) and Definition 5.10,

$$\lim_{\alpha\to\infty} cond_B(b_k) = 1, \quad \lim_{\alpha\to\infty} cond_B(g_k) = 3$$

and the result is obtained in a straightforward way. □

8. DARBOUX TRANSFORMATION FOR SYMMETRIC POSITIVE DEFINITE JACOBI MATRICES. In this section we analyze the symmetric Darboux transformation. If we consider a positive measure μ supported in $(0,\infty)$, apart from the corresponding monic Jacobi matrix J (see (1.1)), we can consider the symmetric Jacobi matrix J_s (see (2.2)). Moreover, this matrix is positive definite and, therefore, its Cholesky factorization exists. In [17], it is proven that the following transformation (symmetric Darboux transformation) computes the symmetric Jacobi matrix associated with $xd\mu$,

$$J_s = LL^t, \quad (J_1)_s = L^t L,$$

where LL^t denotes the Cholesky factorization of J_s.

It is well known that the usual algorithm to compute the Cholesky factorization is a normwise backward stable algorithm and that, for tridiagonal matrices, it is componentwise backward stable [16]. On the other hand, numerical experiments show that, for finite matrices, the spectral condition number of $J_s(B,\sqrt{G})$, $\kappa_2(J_s(B,\sqrt{G}))$, can be much smaller than $\kappa_2(J(B,G))$. Here the notation introduced in Section 3 is used. For instance, in the next table we show both condition numbers for Laguerre Polynomials with parameter $1/10$; n denotes the order of the matrix.

	n=10	n=50	n=100
$\kappa_2(J)$	$5.4\ 10^7$	$4.46\ 10^{64}$	$1.39\ 10^{158}$
$\kappa_2(J_s)$	$3.58\ 10^2$	$8.3\ 10^3$	$3.3\ 10^4$

ETNA
Kent State University
etna@mcs.kent.edu

Then, it is natural to ask whether, in the case of positive measures supported in $(0,\infty)$, we can expect to compute the corresponding symmetric Jacobi matrix associated with $xd\mu$ with higher accuracy than the monic Jacobi matrix. In this section we show that this is not true. In fact, both algorithms are mixed forward-backward stable and the corresponding condition numbers have similar magnitudes (Theorem 8.15) which implies that similar forward errors must be expected.

Let $J(B,G)$ (see (3.1)) be the $n \times n$ monic Jacobi matrix associated with a positive measure μ. Consider the corresponding symmetric Jacobi matrix, $J_s(B,\sqrt{G})$. Both matrices are similar. In fact, there exists a diagonal matrix D such that

$$J(B,G) = D \cdot J_s(B,\sqrt{G}) \cdot D^{-1},$$

where

$$D(i,i) = \begin{cases} 1 & i = 1, \\ \sqrt{\prod_{j=1}^{i-1} G_j}, & i \geq 2. \end{cases}$$

Assume that the measure μ is supported in $(0,\infty)$. Then, $J_s(B,\sqrt{G})$ is a positive definite matrix and, therefore, the corresponding Cholesky factorization exists. Moreover, let us consider the symmetric Darboux transformation applied to the section of order n of $J_s(B,\sqrt{G})$, i.e.,

$$J_s(B,\sqrt{G}) = LL^t,$$

$$J_s(b,\sqrt{g}) = (L^tL)_{n-1}.$$

It is obvious that $J_s(b,\sqrt{g})$ is also a positive definite tridiagonal matrix. This matrix corresponds to the $(n-1)\times(n-1)$ leading principal submatrix of the symmetric Jacobi matrix associated with $xd\mu$ [12, 17]. In the sequel, we denote the symmetric Jacobi matrices by $J_s(B,C)$ assuming that the entries of C are all positive.

The MATLAB code that computes Cholesky factorization of $J_s(B,C)$ is

ALGORITHM 8.1. *Given a symmetric positive definite Jacobi matrix $J_s(B,C) \in R^{n\times n}$, this algorithm computes its Cholesky factorization.*

```
l(1,1)=sqrt(B(1))
for i=1:n-1
   l(i+1,i)=C(i)/l(i,i)
   l(i+1,i+1)=sqrt(B(i+1)-l(i+1,i)^2)
end
```

The following algorithm computes the symmetric Darboux transform of a symmetric Jacobi matrix, $J_s(B,C)$.

ALGORITHM 8.2. *Given a positive definite symmetric Jacobi matrix $J_s(B,C) \in R^{n\times n}$, this algorithm computes its symmetric Darboux transform of order $n-1$, $J_s(b,c) \in R^{(n-1)\times(n-1)}$.*

```
l(1,1)=sqrt(B(1))
for i=1:n-2
     l(i+1,i)=C(i)/l(i,i)
     b(i)=l(i,i)^2 + l(i+1,i)^2
     l(i+1,i+1)=sqrt(B(i+1)-l(i+1,i)^2)
```

ETNA
Kent State University
etna@mcs.kent.edu


```
    c(i)=l(i+1,i+1)l(i+1,i)
end
l(n,n-1)=C(n-1)/l(n-1,n-1)
b(n-1)=l(n-1,n-1)^2 + l(n,n-1)^2
```

The computational cost of Algorithm 8.2 is $7n-11$ flops which is almost twice the cost of Algorithm 3.2.

The next lemma states the relation between the elements of a monic Jacobi matrix and the associated symmetric Jacobi matrix. This lemma will be used when proving that the condition numbers of Darboux transformation and symmetric Darboux transformation are of similar magnitude.

LEMMA 8.1. *Let $J(B,G)=\tilde{L}\tilde{U}$ be the LU factorization without pivoting of $J(B,G)$, and let $J_s(B,C)=LL^t$ ($C:=\sqrt{G}$) be the Cholesky factorization of $J_s(B,C)$. Then,*

$$l_{ii}=\sqrt{\tilde{u}_i},\quad i=1:n,\quad l_{i+1,i}=\sqrt{\tilde{l}_i},\quad i=1:n-1,$$

where $\tilde{u}_i$ denotes the element of $\tilde{U}$ in position (i,i), $\tilde{l}_i$ denotes the element of $\tilde{L}$ in position $(i+1,i)$, and l_{ii} and $l_{i+1,i}$ denote the elements of L in positions (i,i) and $(i+1,i)$, respectively.

8.1. Backward error analysis. The following three results are equivalent to Theorems 4.1, 4.2 and 4.3 obtained for Darboux transformation, but since the matrix $J_s(B,C)$ is positive definite, perfect componentwise stability is obtained. Recall that a similar result was obtained for monic Jacobi matrices associated with positive measures supported in $(0,\infty)$ (see Lemma 4.4 and Theorem 4.3). In this section, it is assumed that the elements of $J_s(B,C)$ are real floating point numbers. Similar remarks to those appearing at the beginning of Section 4 and Remark 5.5 remain valid in this case.

THEOREM 8.2. *Let $J_s(B,C)$ be a positive definite Jacobi matrix of order n. If $\hat{L}$ is the factor computed by Algorithm 8.1, then*

$$J_s(B+\Delta B, C+\Delta C)=\hat{L}\hat{L}^t,\quad |\Delta B|\le\frac{3\mathbf{u}+3\mathbf{u}^2+\mathbf{u}^3}{1-3\mathbf{u}-3\mathbf{u}^2-\mathbf{u}^3}|B|,\quad |\Delta C|\le\mathbf{u}|C|.$$

Proof. For the computed quantities, we get

$$\hat{l}_{11}(1+\epsilon_1)=\sqrt{B_1},\quad |\epsilon_1|\le\mathbf{u},$$

and then,

$$|B_1-\hat{l}_{11}^2|\le(2\mathbf{u}+\mathbf{u}^2)\hat{l}_{11}^2.$$

For $i=2:n-1$,

$$\hat{l}_{ii}(1+\epsilon_i)=\sqrt{\frac{B_i-\hat{l}_{i,i-1}^2(1+\delta_i)}{(1+\eta_i)}},\quad |\epsilon_i|,|\delta_i|,|\eta_i|\le\mathbf{u},$$

$$\hat{l}_{ii}^2(1+2\epsilon_i+\epsilon_i^2)(1+\eta_i)=B_i-\hat{l}_{i,i-1}^2(1+\delta_i),$$

$$B_i-\hat{l}_{i,i-1}^2-\hat{l}_{ii}^2=\hat{l}_{i,i-1}^2\delta_i+\hat{l}_{ii}^2(2\epsilon_i+\eta_i+2\epsilon_i\eta_i+\epsilon_i^2+\epsilon_i^2\eta_i).$$

Then, we get

$$|B_i - \hat{l}_{i,i-1}^2 - \hat{l}_{ii}^2| \leq \hat{l}_{i,i-1}^2 \mathbf{u} + \hat{l}_{ii}^2(3\mathbf{u}+3\mathbf{u}^2+\mathbf{u}^3) \leq (3\mathbf{u}+3\mathbf{u}^2+\mathbf{u}^3)(\hat{l}_{i,i-1}^2+\hat{l}_{ii}^2).$$

Finally, for $i = 1 : n-1$

$$\hat{l}_{i+1,i} = \frac{C_i}{\hat{l}_{ii}}(1+\delta_i), \quad \text{and} \quad |C_i - \hat{l}_{ii}\hat{l}_{i+1,i}| \leq \mathbf{u}|C_i|.$$

Previous results show that

$$J_s(B+\Delta B, C+\Delta C) = \hat{L}\hat{L}^t, \quad |\Delta B| \leq (3\mathbf{u}+3\mathbf{u}^2+\mathbf{u}^3)|diag(\hat{L}\hat{L}^t)|, \quad |\Delta C| \leq \mathbf{u}|C|.$$

But

$$|diag(\hat{L}\hat{L}^t)| \leq |B| + |\Delta B| \leq |B| + (3\mathbf{u}+3\mathbf{u}^2+\mathbf{u}^3)|diag(\hat{L}\hat{L}^t)|.$$

Therefore,

$$|diag(\hat{L}\hat{L}^t)| \leq \frac{|B|}{1-3\mathbf{u}-3\mathbf{u}^2-\mathbf{u}^3}$$

and the result follows easily. □

For the second step of symmetric Darboux transformation, i.e., $J_s(b,c) = (L^tL)_{n-1}$, we get the following result.

LEMMA 8.3. *Let L be a bidiagonal lower triangular matrix, and let $J_s(b,c) = L^tL$ be the exact product of L^t times L. Let $J_s(\hat{b},\hat{c})$ be the computed product of L^t times L. Then,*

$$|b-\hat{b}| \leq \frac{2\mathbf{u}}{1-\mathbf{u}}|\hat{b}|, \quad \textit{and} \quad |c-\hat{c}| \leq \mathbf{u}|\hat{c}|,$$

which in matrix notation implies that

$$J_s(\hat{b}+\Delta\hat{b}, \hat{c}+\Delta\hat{c}) = (L^tL), \quad |\Delta\hat{b}| \leq \frac{2\mathbf{u}}{1-\mathbf{u}}|\hat{b}|, \quad |\Delta\hat{c}| \leq \mathbf{u}|\hat{c}|.$$

Proof.

$$\hat{b}_k(1+\epsilon_k) = l_{kk}^2(1+\delta_k) + l_{k+1,k}^2(1+\eta_k), \quad |\epsilon_k|, |\delta_k|, |\eta_k| \leq \mathbf{u}.$$

Then,

$$|\hat{b}_k - l_{kk}^2 - l_{k+1,k}^2| \leq \mathbf{u}(|\hat{b}_k| + l_{kk}^2 + l_{k+1,k}^2),$$

which means that

$$|b_k - \hat{b}_k| \leq \mathbf{u}(|\hat{b}_k| + b_k) \leq \mathbf{u}(2|\hat{b}_k| + |b_k - \hat{b}_k|)$$

and the result follows in a straightforward way.

On the other hand,

$$\hat{c}_k(1+\zeta_k) = l_{k+1,k}l_{k+1,k+1}, \quad |\zeta_k| \leq \mathbf{u}.$$

ETNA
Kent State University
etna@mcs.kent.edu

$$|\hat{c}_k - l_{k+1,k} l_{k+1,k+1}| \le \mathbf{u} |\hat{c}_k|.$$

□

Therefore, taking into account Theorem 8.2 and Lemma 8.3, it is easy to prove the main stability result for the symmetric Darboux transformation.

THEOREM 8.4. *Given a positive definite symmetric Jacobi matrix of order n, $J_s(B,C)$, let $J_s(b,c)$ be its symmetric Darboux transform of order $n-1$. If $J_s(\hat{b},\hat{c})$ and $\hat{L}$ are the matrices computed by Algorithm 8.2, then*

$$J_s(B+\Delta B, C+\Delta C) = \hat{L}\hat{L}^t, \quad |\Delta B| \le \frac{3\mathbf{u}+3\mathbf{u}^2+\mathbf{u}^3}{1-3\mathbf{u}-3\mathbf{u}^2-\mathbf{u}^3}|B|, \quad |\Delta C| \le \mathbf{u}|C|,$$

$$J_s(\hat{b}+\Delta\hat{b}, \hat{c}+\Delta\hat{c}) = (\hat{L}^t\hat{L})_{n-1}, \quad |\Delta\hat{b}| \le \frac{2\mathbf{u}}{1-\mathbf{u}}|\hat{b}|, \quad |\Delta\hat{c}| \le \mathbf{u}|\hat{c}|.$$

Notice that we have shown that the symmetric Darboux transformation is componentwise stable in the mixed forward-backward sense.

8.2. Conditioning of symmetric Darboux transformation. In this section we define the corresponding componentwise relative condition number with respect to perturbations in components associated with the symmetric Darboux transformation and a sharp bound for it is obtained. Notice that, in this case, Theorem 8.4 guarantees that this is also the condition number associated with the backward errors. This fact simplifies the analysis compared with Section 5.

DEFINITION 8.5. *Let $J_s(b,c)$ be the symmetric Darboux transform of order $n-1$ of a symmetric Jacobi matrix of order n, $J_s(B,C)$, and $J_s(b+\Delta b, c+\Delta c)$ be the Darboux transform of order $n-1$ of the symmetric Jacobi matrix of order n, $J_s(B+\Delta B, C+\Delta C)$. The componentwise relative condition number of the symmetric Darboux transformation of the matrix $J_s(B,C)$ with respect to perturbations in components is defined as*

$$cond_s(J_s(B,C)) := \lim_{\epsilon\to 0}\sup\left\{\max_k\left\{\left|\frac{\Delta b_k}{\epsilon\, b_k}\right|, \left|\frac{\Delta c_k}{\epsilon\, c_k}\right|\right\} : \ |\Delta B| \le \epsilon|B|,\ |\Delta C| \le \epsilon|C|\right\}.$$

Considering the previous definition and Theorem 8.4 it is possible to bound the forward error in terms of the condition number.

LEMMA 8.6. *Let $J_s(b,c)$ and $J_s(\hat{b},\hat{c})$ be, respectively, the exact and the computed symmetric Darboux transform of $J_s(B,C)$ from Algorithm 8.2. Then,*

$$\max_k\left\{\left|\frac{b_k-\hat{b}_k}{b_k}\right|, \left|\frac{c_k-\hat{c}_k}{c_k}\right|\right\} \le 3\mathbf{u}(1+cond_s(J_s(B,C))) + O(\mathbf{u}^2).$$

To get an explicit expression for $cond_s(J_s(B,C))$, similar developments to those in Section 5 are presented. Consider a positive definite perturbation of a symmetric positive definite Jacobi matrix $J_s(B,C)$, $J_s(B+\Delta B, C+\Delta C)$, and suppose that L and $L+\Delta L$ are the corresponding Cholesky factors. We omit most of the proofs since they are similar to those in Section 5.

ETNA
Kent State University
etna@mcs.kent.edu

LEMMA 8.7. *The following results are correct to first order.*

$$\Delta l_{k+1,k} = l_{k+1,k}\left(\frac{\Delta C_k}{C_k} - \frac{\Delta l_{kk}}{l_{kk}}\right), \quad k = 1 : n-1.$$

$$\Delta l_{kk} = \frac{\Delta B_k}{2 l_{kk}} - \frac{l_{k,k-1}}{l_{kk}} \Delta l_{k,k-1}, \quad k = 1 : n-1.$$

From the previous lemma, we get to first order,

$$\frac{\Delta l_{kk}}{l_{kk}} = \frac{\Delta B_k}{2 l_{kk}^2} - \frac{l_{k,k-1}^2}{l_{kk}^2}\left(\frac{\Delta C_{k-1}}{C_{k-1}} - \frac{\Delta l_{k-1,k-1}}{l_{k-1,k-1}}\right). \tag{8.1}$$

LEMMA 8.8. *The following expressions for relative variations are correct to first order.*

$$\delta l_{11} = \frac{\delta B_1}{2},$$

$$\delta l_{kk} = \frac{\Delta B_k}{2 l_{kk}^2} - \frac{l_{k,k-1}^2}{l_{kk}^2}\delta C_{k-1} + \sum_{i=1}^{k-1}\left(\frac{\Delta B_i}{2 l_{i,i}^2} - \frac{l_{i,i-1}^2}{l_{ii}^2}\delta C_{i-1}\right)\prod_{j=i}^{k-1}\frac{l_{j+1,j}^2}{l_{j+1,j+1}^2}, \quad k = 1 : n-1.$$

LEMMA 8.9. *Let us assume that* $|\Delta B_k| \leq \epsilon |B_k|$ *and,* $|\Delta C_k| \leq \epsilon |C_k|$ *hold for* $k = 1 : n-1$*. Then, to first order,*

$$|\delta l_{kk}| \leq \epsilon\left[\frac{1}{2} + \frac{3 l_{k,k-1}^2}{2 l_{kk}^2} + \sum_{i=1}^{k-1}\left(\frac{1}{2} + \frac{3 l_{i,i-1}^2}{2 l_{ii}^2}\right)\prod_{j=i}^{k-1}\frac{l_{j+1,j}^2}{l_{j+1,j+1}^2}\right].$$

REMARK 8.1. *The bound for* $|\delta l_{kk}|$ *given in the previous lemma can also be expressed in the following way:*

$$|\delta l_{kk}| \leq \epsilon\left(\frac{1}{2} + 2\sum_{i=1}^{k-1}\prod_{j=i}^{k-1}\frac{l_{j+1,j}^2}{l_{j+1,j+1}^2}\right). \tag{8.2}$$

DEFINITION 8.10. *For* $k = 1 : n-1$

$$cond_s(l_{kk}) := \frac{1}{2} + 2\sum_{i=1}^{k-1}\prod_{j=i}^{k-1}\frac{l_{j+1,j}^2}{l_{j+1,j+1}^2}.$$

The following lemma gives a recursive formula to compute $cond_s(l_{kk})$.

LEMMA 8.11. *For* $k = 1 : n-1$,

$$cond_s(l_{11}) = \frac{1}{2}, \quad cond_s(l_{kk}) = \frac{1}{2}\left(1 + \frac{l_{k,k-1}^2}{l_{kk}^2}\right) + \frac{l_{k,k-1}^2}{l_{kk}^2}[1 + cond_s(l_{k-1,k-1})].$$

LEMMA 8.12. *The following equations are correct to first order,*

$$\Delta b_k = 2 l_{kk}\Delta l_{kk} + 2 l_{k+1,k}\Delta l_{k+1,k}, \quad k = 1 : n-1,$$

ETNA
Kent State University
etna@mcs.kent.edu

$$\Delta c_k = l_{k+1,k}\Delta l_{k+1,k+1} + l_{k+1,k+1}\Delta l_{k+1,k}, \quad k = 1 : n-2.$$

LEMMA 8.13.

$$\delta b_k = \frac{2l_{k+1,k}^2}{l_{kk}^2 + l_{k+1,k}^2}\delta C_k + 2\frac{l_{kk}^2 - l_{k+1,k}^2}{l_{kk}^2 + l_{k+1,k}^2}\delta l_{kk}, \quad k = 1 : n-1,$$

$$\delta c_k = \frac{l_{k+1,k+1}^2 + l_{k+1,k}^2}{2l_{k+1,k+1}^2}\delta B_{k+1} + \left(1 - \frac{l_{k+1,k}^2}{l_{k+1,k+1}^2}\right)(\delta C_k - \delta l_{kk}), \quad k = 1 : n-2.$$

Then, taking into account Remark 8.1 and Lemma 8.13, we obtain the following result,

THEOREM 8.14. *Let us assume that* $|\Delta B_k| \le \epsilon |B_k|$, $|\Delta C_k| \le \epsilon |C_k|$ *hold for* $k = 1 : n-1$. *Then, to first order,*

$$|\delta b_k| \le \epsilon \left[\frac{2l_{k+1,k}^2}{l_{kk}^2 + l_{k+1,k}^2} + 2\frac{|l_{kk}^2 - l_{k+1,k}^2|}{l_{kk}^2 + l_{k+1,k}^2}\, cond_s(l_{kk})\right], \quad k = 1 : n-1,$$

$$|\delta c_k| \le \epsilon \left[\frac{1}{2} + \frac{l_{k+1,k}^2}{2l_{k+1,k+1}^2} + \left|1 - \frac{l_{k+1,k}^2}{l_{k+1,k+1}^2}\right| (1 + cond_s(l_{kk}))\right], \quad k = 1 : n-2.$$

Let us define

$$cond_s(b_k) := \frac{2l_{k+1,k}^2}{l_{kk}^2 + l_{k+1,k}^2} + 2\frac{|l_{kk}^2 - l_{k+1,k}^2|}{l_{kk}^2 + l_{k+1,k}^2}\, cond_s(l_{kk}),$$

i.e., $cond_s(b_k)$ is the bound for $|\delta b_k|$ divided by ϵ. In a similar way,

$$cond_s(c_k) := \frac{1}{2} + \frac{l_{k+1,k}^2}{2l_{k+1,k+1}^2} + \left|1 - \frac{l_{k+1,k}^2}{l_{k+1,k+1}^2}\right| (1 + cond_s(l_{kk}))$$

denotes the bound for $|\delta c_k|$ divided by ϵ.

Then, taking into account Definition 8.5,

$$cond_s(J_s(B,C)) \le \max\{cond_s(b_{n-1}), \max_{k=1:n-2}\{cond_s(b_k), cond_s(c_k)\}\}. \tag{8.3}$$

We have not been able to prove that the inequality appearing in (8.3) is sharp, i.e., that there exist perturbations fulfilling the conditions of Definition 8.5 for which the inequalities in Theorem 8.14 become equalities. Therefore, we have not found expressions for the true condition number of the symmetric Darboux transformation, $cond_s(J_s(B,C))$. We will denote the bound appearing in (8.3) by

$$bcond_s(J_s(B,C)) := \max\{cond_s(b_{n-1}), \max_{k=1:n-2}\{cond_s(b_k), cond_s(c_k)\}\}. \tag{8.4}$$

Now, let us compare $bcond_s(J_s(B,C))$ ($C = \sqrt{G}$) and $bcond_B(J(B,G))$ as defined in (5.20). Taking into account Lemmas 5.3 and 8.6, it is obvious that the condition numbers control the forward errors. Then, the comparison of both condition numbers implies the comparison of the respective forward errors obtained from the application of Darboux transformation to a monic Jacobi matrix $J(B,G)$ and the application of the symmetric Darboux transformation to the symmetric Jacobi matrix associated with $J(B,G)$ when the corresponding measure is positive and supported in $(0,\infty)$. We have not expressions for $cond_s(J_s(B,C))$ or $cond_B(J(B,G))$. Therefore, we are forced to compare the bounds $bcond_s(J_s(B,C))$ and $bcond_B(J(B,G))$. We have performed numerical experiments similar to those appearing in Section 9 in the case of the symmetric Darboux transformation. These experiments have shown that $bcond_s(J_s(B,C))$ is an accurate approximation of the true condition number $cond_s(J_s(B,C))$ as well as $bcond_B(J(B,G))$ is an accurate approximation of $cond_B(J(B,G))$.

THEOREM 8.15. *Let $J(B,G)$ and $J_s(B,C)$ ($C = \sqrt{G}$) be, respectively, the monic and the symmetric Jacobi matrices associated with a family of polynomials orthogonal with respect to a positive measure supported in $(0,\infty)$. Then*

$$\frac{1}{2}\, bcond_B(J(B,G)) \leq bcond_s(J_s(B,C)) \leq 2\, bcond_B(J(B,G)).$$

Proof. Taking into account Lemma 8.1, an alternative expression for $cond_s(b_k)$ and $cond_s(c_k)$ is

$$cond_s(b_k) = \frac{2l_k}{u_k + l_k} + \frac{|u_k - l_k|}{u_k + l_k}\left(1 + 4\sum_{i=1}^{k-1}\prod_{j=i}^{k-1}\frac{l_j}{u_{j+1}}\right),$$

$$cond_s(c_k) = \frac{1}{2} + \frac{l_k}{2u_{k+1}} + \left|1 - \frac{l_k}{u_{k+1}}\right|\left(\frac{3}{2} + 2\sum_{i=1}^{k-1}\prod_{j=i}^{k-1}\frac{l_j}{u_{j+1}}\right).$$

Now we compare the previous expressions with (5.14) and (5.15) and it is trivial to obtain

$$\frac{1}{2} cond_B(b_k) \leq cond_s(b_k) \leq 2\, cond_B(b_k).$$

On the other hand, taking into account that for any number a, $|1+a| \leq 2 + |1-a|$

$$cond_s(c_k) \leq cond_B(g_k).$$

Moreover, since $|1+a| + |1-a| \geq 2,$ for any a,

$$cond_s(c_k) \geq \frac{1}{2} cond_B(g_k).$$

□

REMARK 8.2. *Taking into account the previous theorem as well as Lemmas 5.3 and 8.6, we can say that*

- *The bounds for the condition numbers $bcond_s(J_s(B,C))$ and $bcond_B(J(B,G))$ are of similar magnitude.*
- *The bounds for the componentwise forward errors associated with Darboux transformation and symmetric Darboux transformation are of similar magnitude.*
- *Moreover, both algorithms are stable in the mixed forward-backward sense, as it has been shown in Theorem 4.3, Lemma 4.4, and Theorem 8.4.*

ETNA
Kent State University
etna@mcs.kent.edu

Previous remarks let us assert that, in the case of positive measures supported in $(0, \infty)$*, the accuracy with which Darboux transformation computes* $J(B, G)$ *is the same as the accuracy with which symmetric Darboux transformation computes* $J_s(B, C)$*.*

On the other hand, notice that the computational cost of Algorithm 8.2 is $7n - 11$ *flops while the computational cost of Algorithm 3.2 is* $4n - 6$ *flops. Taking into account Lemma 8.11 and the definition of* $bcond_s(J_s(B, C))$*, the cost of computing* $bcond_s(J_s(B, C))$ *is* $19n - 25$ *flops compared with the* $14n - 22$ *flops of computing* $bcond_B(J(B, G))$*. This leads us to think that, unless one needs specifically the parameters of the recurrence relation that the orthonormal polynomials satisfy, non-symmetric Darboux transformation is more efficient than symmetric.*

9. NUMERICAL EXPERIMENTS. Finally, we conclude with a set of numerical experiments. In Section 5 we studied thoroughly the conditioning of the unshifted Darboux transformation without parameter. The main result of that section states that Algorithm 3.2 for computing Darboux transformation is forward stable based on the fact that $bcond_B(J(B, G))$ and $bcond_C(J(B, G))$ are of similar magnitude (Theorem 5.16 and Remark 5.5). Nevertheless, we just compare the bounds for the true condition numbers and the reader could doubt the reliability of our result. For this reason, we include this section with a variety of numerical experiments supporting our assertion of forward stability. Moreover, these experiments show that $bcond_B(J(B, G))$ is a good approximation of the true condition number because it will be seen that $\mathbf{u}(1 + bcond_B(J(B, G)))$ is a reliable measure of the componentwise forward errors (recall Lemma 5.3).

In the first set of experiments we compare the forward errors obtained in two ways: 1) applying Algorithm 3.2 to certain monic Jacobi matrices and comparing with the exact results, 2) perturbing randomly each entry B_i or G_i to $\tilde{B}_i$ or $\tilde{G}_i$ in such a way that $|B_i - \tilde{B}_i| \leq 5\mathbf{u}|B_i|$, $|G_i - \tilde{G}_i| \leq 5\mathbf{u}|G_i|$, applying the algorithm exactly to the perturbed data and comparing with the exact results. The experiments have been done using MATLAB 5.3 ($\mathbf{u} = 1.11\ 10^{-16}$). In order to simulate an exact application of the algorithm we have used the variable precision arithmetic of the Symbolic Math toolbox of MATLAB using 64 decimal digits of precision. We have analyzed the following cases:

1. Monic Jacobi matrices of dimension 100×100 associated with Bessel polynomials with parameter $\alpha = -101/7 + k^2$, where $k = 1 : 20$. In the table below, this experiment is denoted by “Bessel”.
2. Monic Jacobi matrices of dimension 100×100 associated with Jacobi polynomials with parameters $\alpha = -19/10 + k$, $\beta = (-9 + k)/10$, where $k = 1 : 20$. In the table below, this experiment is denoted by “Jacobi”.
3. Monic Jacobi matrices of dimension 100×100 associated with Laguerre polynomials with parameter $\alpha = -19/10 + k$, where $k = 1 : 20$. In the table below, this experiment is denoted by “Laguerre”.
4. Monic Jacobi matrices of dimension 100×100 whose elements in diagonals B and G are normally distributed random numbers. We have also considered 20 different matrices. In the table below, this experiment is denoted by “Random1”.
5. Monic Jacobi matrices of dimension 100×100 whose elements in diagonal B and G are normally distributed random numbers multiplied by $10^{5*randn}$ where $randn$ denotes a normally distributed random number. We have also considered 20 different matrices. In the table bellow, this experiment is denoted by “Random2”.

In the second set of experiments we compare the forward error produced by Algorithm 3.2 with the bound $\mathbf{u}(1 + bcond_B(J(B, G)))$ in the cases considered in the first set of experiments.

ETNA
Kent State University
etna@mcs.kent.edu

We denote by $vm1$ the vector with the maximum componentwise forward errors obtained in the way 1) and $vm2$ denotes the vector with the forward errors obtained in the way 2). More precisely the maximum componentwise forward error is $\max\{forb, forg\}$ where $forb$ and $forg$ were defined in (3.5).

	Bessel	Jacobi	Laguerre	Random1	Random2
$\max(vm1)$	$2.3\ 10^{2}$	$6.97\ 10^{-13}$	$4.43\ 10^{-16}$	$3.04\ 10^{-11}$	$1.43\ 10^{-13}$
$\min(vm1)$	$3.28\ 10^{-16}$	$8.96\ 10^{-16}$	$2.15\ 10^{-16}$	$2.54\ 10^{-15}$	$2.18\ 10^{-16}$
$\max(vm2)$	$2.59\ 10^{2}$	$1.02\ 10^{-11}$	$8.48\ 10^{-15}$	$4.02\ 10^{-10}$	$6.89\ 10^{-12}$
$\min(vm2)$	$4.76\ 10^{-15}$	$1.07\ 10^{-14}$	$4.22\ 10^{-15}$	$3.51\ 10^{-14}$	$4.38\ 10^{-15}$
$\max(vm1./vm2)$	3.94	5.32	7.003	$2.74\ 10^{-1}$	$1.74\ 10^{-1}$
$\mathrm{mean}(vm1./vm2)$	$3.12\ 10^{-1}$	$3.84\ 10^{-1}$	$4.31\ 10^{-2}$	$6.69\ 10^{-2}$	$5.95\ 10^{-2}$
$\min(vm1./vm2)$	$4.44\ 10^{-2}$	$1.20\ 10^{-2}$	$2.54\ 10^{-2}$	$1.21\ 10^{-2}$	$6.12\ 10^{-3}$

The fifth row of the previous table shows that the forward errors produced by Algorithm 3.2 are, at most, a little larger than the errors produced by perturbing the data, and the sixth and seventh rows show that frequently they are smaller. This means that Algorithm 3.2 is forward stable, as predicted by the theory in Section 5. It is also interesting to note that the experiment with Bessel polynomials offers a wide range of values of the forward errors $[3.28\ \ 10^{-16}, 2.3\ \ 10^{2}]$. In fact, when the parameter takes negative values, i.e., $\{-94/7, -73/7, -38/7\}$, we obtain the following results:

Bessel	α=-94/7	α=-73/7	α=-38/7
$vm1$	$7.03\ 10^{1}$	$2.30\ 10^{2}$	$4.71\ 10^{-4}$
$vm2$	$1.78\ 10^{1}$	$2.59\ 10^{2}$	$7.65\ 10^{-3}$

These results show that Algorithm 3.2 is forward stable even in the presence of large forward errors. This is remarkable because the analysis in Section 5 just leads to first order error bounds (Lemma 5.3). One could think that these results for the componentwise forward errors can be improved noticeably by considering normwise forward errors. However, although the errors in norm are certainly smaller, they are very far from being of order machine precision. In the following table we include normwise forward errors (3.6) denoted by $mn1$ and $mn2$, respectively, for Bessel polynomials with parameters $\{-94/7, -73/7, -38/7\}$:

Bessel	α=-94/7	α=-73/7	α=-38/7
$mn1$	$2.95\ 10^{-3}$	$3.24\ 10^{-3}$	$1.32\ 10^{-8}$
$mn2$	$2.12\ 10^{-3}$	$3.53\ 10^{-3}$	$6.24\ 10^{-7}$

In spite of the results obtained in the above experiments, it still remains to be verified that the bound $bcond_B(J(B,G))$ for the true condition number really reflects the sensitivity of the problem and, therefore, it is a useful bound for the forward error. In the following table we compare the bound for the forward error (obtained from Lemma 5.3) and the real forward errors. Again we consider the examples used above. Here

$$fm := \frac{\max\{forb, forg\}}{\mathbf{u}(1 + bcond_B(J(B,G)))}.$$

	Bessel	Jacobi	Laguerre	Random1	Random2
$\max\{fm\}$	3.69	$1.88\ 10^{-1}$	1.46	$5.098\ 10^{-1}$	$6.14\ 10^{-1}$
$\mathrm{mean}\{fm\}$	$5.86\ 10^{-1}$	$7.29\ 10^{-2}$	$6.18\ 10^{-1}$	$1.66\ 10^{-1}$	$1.74\ 10^{-1}$
$\min\{fm\}$	$2.9\ 10^{-2}$	$1.98\ 10^{-2}$	$5.19\ 10^{-1}$	$5.6\ 10^{-2}$	$1.53\ 10^{-2}$

The results in this table show that $bcond_B(J(B,G))$ is a good approximation of $cond_B(J(B,G))$ since the bound $\mathbf{u}(1 + bcond_B(J(B,G)))$ is just a little larger than the forward errors produced by Algorithm 3.2 in a wide set of examples.

The presence of values of fm slightly larger than 1 can be explained by the fact that the errors coming from computing the input data B and G have not been taken into account, and although they are of the same magnitude (see Remark 5.5), the forward error bound can be increased by a factor 3.

10. Acknowledgements. The authors thank Prof. Juan Manuel Molera and Prof. Francisco Marcellán for their suggestions and comments to improve this paper.

REFERENCES

[1] R. P. BOAS, *The Stieltjes moment problem for functions of bounded variation*, Bull. Amer. Math. Soc., 45 (1939), pp. 399–404.
[2] M. I. BUENO AND F. M. DOPICO, *Stability and sensitivity of tridiagonal LU factorization without pivoting*, to appear in BIT.
[3] M. I. BUENO AND F. MARCELLÁN, *Darboux Transformation and Perturbation of linear functionals*, Linear Algebra Appl., 384 (2004), pp. 215–242.
[4] X. CHANG AND C. C. PAIGE, *On the sensitivity of the LU factorization*, BIT, 38 (1998), pp. 486–501.
[5] T. S. CHIHARA, *An Introduction to Orthogonal Polynomials*, Gordon and Breach, New York, 1978.
[6] G. DARBOUX, *Leçons sur la Théorie Générale des Surfaces*, 2éme partie, Gauthiers-Villars, Paris, 1889.
[7] A. J. DURÁN, *The Stieltjes moment problem for rapidly decreasing functions*, Proc. Amer. Math. Soc., 107 (1989), pp. 731–741.
[8] D. GALANT, *An implementation of Christoffel's Theorem in the Theory of Orthogonal Polynomials* , Math. Comp., 25 (1971), pp. 111–113.
[9] D. GALANT, *Algebraic Methods for modified orthogonal polynomials*, Math. Comp., 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] W. GAUTSCHI, *The interplay between classical analysis and (numerical) linear algebra – A tribute to Gene H. Golub*, Electron. Trans. Numer. Anal., 13 (2002), pp. 119–147. http://etna.mcs.kent.edu/vol.13.2002/pp119-147.dir/pp119-147.pdf.
[12] G. H. GOLUB AND J. KAUTSKY, *Calculation of Gauss Quadratures with Multiple Free and Fixed Knots*, Numer. Math., 41 (1983), pp. 147–163.
[13] F. A. GRÜNBAUM AND L. HAINE, *Orthogonal Polynomials satisfying Differential Equations: the Role of the Darboux Transformation*, CRM Proceedings and Lecture Notes, vol. 9. Amer. Math. Soc., Providence, Rhode Island, 1996, pp. 143–154.
[14] F. A. GRÜNBAUM AND L. HAINE, *Bispectral Darboux Transformations:An Extension of the Krall Polynomials*, Int. Math. Res. Not., 8 (1997), pp. 359–392.
[15] F. A. GRÜNBAUM, L. HAINE, AND E. HOROZOV, *Some functions that generalize the Krall-Laguerre polynomials*, J. Comput. Appl. Math., 106 (1999), pp. 271–297.
[16] N. J. HIGHAM, *Accuracy and Stability of Numerical Algorithms*, 2nd. ed., SIAM, Philadelphia, 2002.
[17] J. KAUTSKY AND G. H. GOLUB, *On the calculation of Jacobi Matrices*, Linear Algebra Appl., 52/53 (1983), pp. 439–455.
[18] V. B. MATVEEV AND M. A. SALLE, *Differential-Difference Evolution Equations. II (Darboux Transformation for the Toda Lattice)* , Lett. Math. Phys., 3 (1979), pp. 425–429.
[19] G. W. STEWART, *On the perturbation of LU and Cholesky factors*, IMA J. Numer. Anal., 17 (1997), pp. 1–6.