A new algorithm for computing the Geronimus transformation with large shifts

M. I. Bueno · A. Deaño · E. Tavernetti

Abstract A 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. The basic Geronimus transformation with shift α transforms the monic Jacobi matrix associated with a measure $d\mu$ into the monic Jacobi matrix associated with $d\mu/(x-\alpha) + C\delta(x-\alpha)$, for some constant C. In this paper we examine the algorithms available to compute this transformation and we propose a more accurate algorithm, estimate its forward errors, and prove that it is forward stable. In particular, we show that for $C = 0$ the problem is very ill-conditioned, and we present a new algorithm that uses extended precision.

Keywords Geronimus transformation · accuracy · roundoff error analysis · orthogonal polynomials · three-term recurrence relations.

Mathematics Subject Classification (2000) 15A21 · 15A23 · 05A05 · 05B25

Maria I. Bueno
Department of Mathematics and College of Creative Studies.
University of California, Santa Barbara, CA.
E-mail: mbueno@math.ucsb.edu

Alfredo Deaño
DAMTP, Centre for Mathematical Sciences.
University of Cambridge, UK.
E-mail: ad495@cam.ac.uk

Edward Tavernetti
Department of Mathematics
University of California, Davis, CA.
E-mail: etavernetti@math.ucdavis.edu

1 Introduction

Given a measure μ, with supp $\mu \subset \mathbb{R}$, one can define a linear functional $\mathscr{L}$ on the space $\mathbb{P}$ of polynomials with real coefficients in the following way:

$$\mathscr{L}(p) = \int p(x)d\mu(x), \qquad p \in \mathbb{P}, \tag{1}$$

which is well defined provided that the moments $\mathscr{L}_n := \mathscr{L}(x^n)$ are finite, $n = 0, 1, 2, \ldots$ In that case, we say that $\mathscr{L}$ is a moment functional. Moreover, if the leading principal submatrices of the Hankel matrix $M = (\mathscr{L}_{i+j})_{i,j=0}^{\infty}$ are nonsingular, then $\mathscr{L}$ is said to be *quasi-definite*, and there exists a sequence of polynomials $\{P_n\}_{n=0}^{\infty}$ orthogonal with respect to μ, that is, [4]

1. $\deg(P_n) = n$ for all $n \geq 0$.
2. $\mathscr{L}(P_n P_m) = K_n \delta_{n,m}$, where $K_n \neq 0$ and $\delta_{n,m}$ is the "Kronecker delta" defined by

$$\delta_{n,m} = \begin{cases} 0, \text{ if } m \neq n, \\ 1, \text{ if } m = n. \end{cases}$$

In particular, $\{P_n\}_{n=0}^{\infty}$ is said to be a monic sequence of orthogonal polynomials (MOPS) if the leading coefficient of each polynomial is equal to one. Every MOPS satisfies a three-term recurrence relation (TTRR):

$$xP_n(x) = P_{n+1}(x) + B_{n+1}P_n(x) + G_n P_{n-1}(x), \tag{2}$$

$$P_{-1}(x) \equiv 0, \quad P_0(x) \equiv 1, \quad B_n, G_n \in \mathbb{R}, \quad G_0 = \mathscr{L}_0, \quad G_n \neq 0 \quad \text{for all } n \geq 0.$$

The previous set of equations can be written in matrix notation as

$$xp = Jp,$$

where $p = [P_0(x), \ P_1(x), \ P_2(x), \ \ldots]^T$ and

$$J = \begin{bmatrix} B_1 & 1 & 0 & \ldots \\ G_1 & B_2 & 1 & \ldots \\ 0 & G_2 & B_3 & \ldots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

This semi-infinite tridiagonal matrix J is called the monic Jacobi matrix associated with the functional $\mathscr{L}$. It is very unusual to denote the entries of a matrix by capital letters, but since the algorithms to compute the Geronimus transformation involve two monic Jacobi matrices, for the sake of clarity, we denote by capital letters the entries in the input matrix and by the same lowercase letters the entries in the output matrix.

For a moment functional $\mathscr{L}$, a polynomial π, and a real number α, let $\pi\mathscr{L}$ and $(x-\alpha)^{-1}\mathscr{L}$ be the moment functionals defined by

$$(\pi\mathscr{L})(p) = \mathscr{L}(\pi p),$$

$$\left((x-\alpha)^{-1}\mathscr{L}\right)(p) = \mathscr{L}\left(\frac{p(x) - p(\alpha)}{x - \alpha}\right).$$

In the literature there are numerous results studying the connection between the recurrence relations of polynomials orthogonal with respect to two allied measures [1, 2,7,15,24]. This relationship can be extended to the corresponding Jacobi matrices. Two examples stand out as particularly important:

– Given $\mathscr{L}$ and $\alpha \in \mathbb{R}$, the transformation that gives the monic Jacobi matrix associated with $(x - \alpha)\mathscr{L}$ in terms of the monic Jacobi matrix associated with $\mathscr{L}$ is called the Christoffel transformation or Darboux transformation.

– Given $\mathscr{L}$, we consider the linear functional $\mathscr{G} := (x-\alpha)^{-1}\mathscr{L} + M\delta(x-\alpha)$, where $\alpha \in \mathbb{R}$ is out of the support of the measure that defines $\mathscr{L}$, and M is a nonzero constant. This transformation performs a rational modification of the measure that defines the functional $\mathscr{L}$ and add a Dirac mass in α. Notice that $M = \mathscr{G}_0$, the first moment of $\mathscr{G}$. The transformation that gives the monic Jacobi matrix associated with $\mathscr{G}$ in terms of the monic Jacobi matrix associated with $\mathscr{L}$ is called the Geronimus transformation or Darboux transformation with free parameter.

These transformations can be considered as reciprocal in the following sense:

Lemma 1 *[25] Let $\mathscr{L}$ and $\mathscr{G}$ be two linear functionals and α a real number. Then,*

$$(x - \alpha)\mathscr{G} = \mathscr{L} \quad \text{if and only if} \quad \mathscr{G} = (x - \alpha)^{-1}\mathscr{L} + \mathscr{G}_0\delta(x - \alpha).$$

If the functional $\mathscr{L}$ is expressed in integral form as in (1), then

$$\mathscr{G}(p(x)) = \left[(x - \alpha)^{-1}\mathscr{L} + \mathscr{G}_0\delta(x - \alpha)\right](p(x)) = \int p(x)\frac{d\mu}{x - \alpha} + Cp(\alpha),$$

where $C = \mathscr{G}_0 - \mu_0$ and $\mu_0 = \int \frac{d\mu}{x-\alpha}$. Therefore, this transformation depends on two free parameters α and C. From now on we call the transformation that gives the monic Jacobi matrix associated with the functional $\mathscr{G}$ in terms of the monic Jacobi matrix associated with $\mathscr{L}$ the Geronimus transformation with shift α and parameter C.

The Geronimus transformation was first studied by Geronimus in 1940. Among numerous papers by Geronimus on orthogonal polynomials there are two [14,13] which contain ideas that anticipated many investigations in modern mathematical physics. The main contribution by Geronimus was a deep investigation of both Darboux transformations. The first non-trivial application of these transformations was proposed by Geronimus himself in [13]. This application is connected to the problem of classifying all sequences of orthogonal polynomials such that its derivatives form another set of orthogonal polynomials. In the last two decades, these transformations have attracted the interest of various specialists in different branches of mathematics and mathematical physics for their applications to different topics such as Discrete Integrable Systems [20,22,23], Quantum Mechanics, Bispectral Transformations in Orthogonal Polynomials [18,16,17], and Numerical Analysis [5,7,8,12,10].

The problem of the numerical computation of the Geronimus transformation with shift α and parameter C of a Jacobi matrix J has been extensively studied when $C = 0$ and the shift α is close to the support of the measure μ [5,10,8]. However, we have not found any papers on the case $C \neq 0$, or when $C = 0$ and the shift is not close to the support of the measure.

The objectives of this paper are the following:

– to investigate the numerical behavior of the available algorithms to compute the Geronimus transformation,

– to present a new algorithm which is more accurate than the previous ones (Algorithm 3), specially when the shift moves away from the support of the measure,

– to explain why the numerical behavior of any algorithm to compute the Geronimus transformation is considerably different for $C = 0$ and $C \neq 0$.

We also estimate the forward errors (Theorem 2) produced by the new algorithm with $\mathcal{O}(n)$ cost, and prove that this algorithm is componentwise forward stable (Theorem 8), which means that the magnitude of the errors produced by this algorithm are the best ones that can be expected because they reflect the sensitivity of the problem to perturbations in the input data (Theorem 2). No need to say that forward stability does not imply small forward errors when the problem is ill-conditioned.

We also show that this algorithm is more accurate than the previous ones, specially when the shift moves away from the support of the measure and $C \neq 0$ (Section 4.4). We prove that the problem of computing the Geronimus transformation is extremely ill-conditioned when $C = 0$ (Subsection 4.5) and therefore, a significant loss of accuracy can be expected in this case. However, we also show that by computing a few outputs with extended precision, the algorithm becomes much more accurate (See Table 7).

The paper is structured as follows: in Section 2 we give a brief account of the main theoretical results needed. In Section 3 we analyze the available forward and backward algorithms, and in Section 4 we introduce a new algorithm. We present a backward error analysis of this algorithm (Theorem 4.1) and provide a condition number for the problem (Definition 2 and Theorem 3) that allows us to estimate the forward errors produced by the new algorithm in $\mathcal{O}(n)$ flops. Finally, we show several numerical experiments to illustrate the performance of this new method and we prove that it is componentwise forward stable.

2 Theoretical results on the Geronimus transformation

Throughout this section, $\mathscr{L}$ is a quasi-definite moment functional, $\{P_n\}$ the sequence of monic polynomials orthogonal with respect to $\mathscr{L}$, J the monic Jacobi matrix associated with $\{P_n\}$, and α a real number outside the support of the measure that defines $\mathscr{L}$.

Let $J - \alpha I = UL$ denote a decomposition of $J - \alpha I$ as a product of an upper triangular matrix U and a unit lower triangular matrix L, where

$$U = \begin{pmatrix} u_1 & 1 & 0 & \dots \\ 0 & u_2 & 1 & \dots \\ 0 & 0 & u_3 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \ L = \begin{pmatrix} 1 & 0 & 0 & \dots \\ l_1 & 1 & 0 & \dots \\ 0 & l_2 & 1 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \tag{3}$$

whenever it is possible. It is easy to check that whenever the UL factorization of $J - \alpha I$ exists, it is not unique. In fact, the entry u_1 can be considered a free parameter. Then, given α and u_1, we say that $\tilde{J} = LU + \alpha I$ is the Geronimus transform of J with shift α and parameter u_1.

Necessary and sufficient conditions for the existence of the Geronimus transform with shift α and parameter u_1 of a monic Jacobi matrix J are given in [1] and [25].

It is also clear that $\tilde{J}$ is a tridiagonal semi-infinite matrix. By Favard's theorem [4], $\tilde{J}$ generates a new sequence of monic orthogonal polynomials if and only if the entries of $\tilde{J}$ in positions $(i + 1, i)$ for $i \geq 1$ are all nonzero. In this case, the MOPS associated with J and $\tilde{J}$, respectively, can be related through the matrix L, as we next show.

Lemma 2 *Let J be a monic Jacobi matrix and let $\alpha \in \mathbb{R}$ be such that $J - \alpha I$ has an UL factorization. Let $u_1 \in \mathbb{R}$ and let $\tilde{J}$ be the Geronimus transform with shift α and parameter u_1 of J. Assume that $\{P_n\}$ and $\{Q_n\}$ are, respectively, the MOPS associated*

with J and $\tilde{J}$. If $J - \alpha I = UL$ is the UL factorization of $J - \alpha I$ such that $\tilde{J} = LU + \alpha I$, then L is the change of basis matrix from $\{P_n\}$ to $\{Q_n\}$, i.e. $Q = LP$, where Q and P are, respectively, the column vectors containing the polynomials in $\{P_n\}$ and $\{Q_n\}$.

Proof. Multiply $J - \alpha I = UL$ by L on the left to get

$$L(J - \alpha I) = (LU)L. \tag{4}$$

Replace LU by $\tilde{J} - \alpha I$ in (4) and multiply by L^{-1} on the right to get

$$L(J - \alpha I)L^{-1} = \tilde{J} - \alpha I$$

Thus, $J - \alpha I$ is similar to $\tilde{J} - \alpha I$. Considering the relation $xQ = \tilde{J}Q$, we have

$$(x - \alpha)Q = (\tilde{J} - \alpha I)Q = L(J - \alpha I)L^{-1}Q$$

and multiplying by L^{-1} on the left we have

$$(x - \alpha)L^{-1}Q = (J - \alpha I)L^{-1}Q$$

and hence $x(L^{-1}Q) = J(L^{-1}Q)$, and $L^{-1}Q$ is a MOPS p satisfying $xp = Jp$. By uniqueness $L^{-1}Q = P$, which implies the result. □

It can be proven [1, 25] that if the matrix $J - \alpha I = UL$, with U and L as in (3), then the Geronimus transform with shift α and parameter u_1 is the Jacobi matrix associated with a functional $\mathscr{G}$ given by

$$\mathscr{G} = (x - \alpha)^{-1}\mathscr{L} + \mathscr{G}_0 \delta(x - \alpha),$$

where $\mathscr{G}_0$ is the first moment of the functional $\mathscr{G}$. Next we show the relationship between $\mathscr{G}_0$ and the parameter u_1 involved in the UL factorization of J.

Lemma 3 *Let $\mathscr{L}$ be a quasi-definite moment functional, and J the corresponding Jacobi matrix. Then, the Geronimus transform of J with shift α and parameter u_1 is associated with the moment functional*

$$\mathscr{G} = (x - \alpha)^{-1}\mathscr{L} + \frac{\mathscr{L}_0}{u_1}\delta(x - \alpha),$$

where $\mathscr{L}_0$ is the first moment of the functional $\mathscr{L}$. Moreover, if the integral representation of $\mathscr{L}$ is given by

$$\mathscr{L}(p) = \int p(x)d\mu(x),$$

then the Geronimus transform of J is associated with the moment functional with integral representation

$$\mathscr{G}(p) = \int p(x)\frac{d\mu(x)}{x - \alpha} + \left(\frac{\mathscr{L}_0}{u_1} - \mu_0\right)p(\alpha),$$

where $\mu_0 = \int \frac{d\mu(x)}{x-\alpha}$ and $p \in \mathbb{P}$.

Proof. By Lemma 1, $(x - \alpha)\mathscr{G} = \mathscr{L}$. Let $\{P_n\}_{n=0}^{\infty}$ and $\{Q_n\}_{n=0}^{\infty}$ be the MOPS with respect to $\mathscr{L}$ and $\mathscr{G}$, respectively. Then, if we denote $P = [P_1(x), P_2(x), ...]^t$, and $Q = [Q_1(x), Q_2(x), ...]^t$, we get

$$((x - \alpha)\mathscr{G}) \left(QQ^t \right) = \mathscr{L} \left(QQ^t \right).$$

Taking into account Lemma 2,

$$\mathscr{G} \left((x - \alpha)QQ^t \right) = \mathscr{L} \left(LPP^t L^t \right).$$

Considering the recurrence relation that $\{Q_n\}$ satisfies and the linearity of $\mathscr{L}$ and $\mathscr{G}$,

$$\mathscr{G} \left((\tilde{J} - \alpha I)QQ^t \right) = L\mathscr{L} \left(PP^t \right) L^t$$

$$(\tilde{J} - \alpha I)\mathscr{G} \left(QQ^t \right) = LD_p L^t$$

where D_p is the diagonal matrix whose diagonal elements are given by $(D_p)_{ii} = \mathscr{L}(P_i^2)$ for all i. Thus,

$$(\tilde{J} - \alpha I) = L \left(D_p L^t D_q^{-1} \right) = LU,$$

where D_q is defined similarly to D_p. Notice that D_p and D_q are invertible matrices by definition of orthogonal polynomials.

Finally, this implies that $u_1 = \mathscr{L}_0/\mathscr{G}_0$, and the result follows. The last part of the lemma is obtained by considering the integral representation of $\mathscr{L}$, that is,

$$\mathscr{G}(p) = \int p(x) \frac{d\mu(x)}{x - \alpha} - p(\alpha) \int \frac{d\mu(x)}{x - \alpha} + \frac{\mathscr{L}_0}{u_1} p(\alpha).$$

□

From the point of view of the algorithms that we will present in the next section, we need to use the fact that the MOPS $\{P_n(x)\}_{n=0}^{\infty}$ obeys a three term recurrence relation of the form:

$$y_{n+1} = (\alpha - B_{n+1})y_n - G_n y_{n-1}, \quad n \geq 0. \tag{5}$$

Now it is important to note that if $\alpha \notin \operatorname{supp} \mu$, then the functions $\{\rho_n(\alpha, C)\}_{n=-1}^{\infty}$ defined by

$$\rho_n(\alpha, C) = -\mathscr{G}(P_n(x)) = - \left(\int P_n(x) \frac{d\mu}{x - \alpha} + CP_n(\alpha) \right), \quad n \geq 0, \quad \rho_{-1}(\alpha, C) = 1$$

obey the same TTRR satisfied by the sequence of orthogonal polynomials $P_n(x)$ for every value of C. Moreover, when $C = 0$ it turns out that $\{\rho_n(\alpha, C)\}_{n=-1}^{\infty}$ is the minimal solution of this recursion, which means that

$$\lim_{n\to\infty} \frac{\rho_n(\alpha, 0)}{g_n} = 0,$$

for any other solution of the TTRR, say g_n, which is independent of $\rho_n(\alpha, 0)$. The solution g_n is called dominant, see [10] for more details on the general theory.

As a consequence of this situation, when $C = 0$, it is not recommended to use the three-term recurrence relation in the forward direction (for increasing n) to generate $\{\rho_n(\alpha, C)\}_{n=-1}^{\infty}$, due to numerical instability. However, the TTRR can be used in the

backward direction, and the process can be reformulated in terms of the associated continued fraction

$$\frac{y_n}{y_{n-1}} = \frac{G_n}{\alpha - B_{n+1}-}\frac{G_{n+1}}{\alpha - B_{n+2}-}\frac{G_{n+2}}{\alpha - B_{n+3}-}..., \quad n = 0, 1, 2, ...$$

which converges to the ratio of minimal solutions according to Pincherle's theorem [11].

Let us define the following quantities:

$$r_{n-1} := \frac{\rho_n(\alpha, C)}{\rho_{n-1}(\alpha, C)} = \frac{G_n}{\alpha - B_{n+1}-}\frac{G_{n+1}}{\alpha - B_{n+2}-}\frac{G_{n+2}}{\alpha - B_{n+3}-}..., \quad n = 0, 1, 2, ... \quad (6)$$

Note that, in particular, $r_{-1} = \rho_0(\alpha, C) = -(\mu_0 + C)$. The importance of these variables in the Geronimus transformation will be given in Lemma 4, which expresses the quantities r_k defined in (6) in terms of the entries in the subdiagonal of the matrix L in the UL factorization of $J - \alpha I$.

Lemma 4 *Let $\{P_n\}$ be the sequence of monic polynomials orthogonal with respect to the linear functional $\mathscr{L}(p) = \int p\, d\mu$. Let $C, \alpha \in \mathbb{R}$, and $\alpha \notin \operatorname{supp}\mu$. Assume that $J - \alpha I = UL$ is the UL factorization of $J - \alpha I$ such that $\tilde{J} = LU + \alpha I$ is the monic Jacobi matrix associated with $\mathscr{G}(p) = \int p(x)/(x - \alpha)d\mu + Cp(\alpha)$. Then,*

$$r_{k-1} := \frac{\rho_k(\alpha, C)}{\rho_{k-1}(\alpha, C)} = -l_k, \quad \text{for all } k \geq 1, \quad (7)$$

where $l_k = L(k+1, k)$.

Proof. The result can be proven by induction. After dividing by $\rho_{k-1}(\alpha, C)$ the TTRR

$$\rho_k(\alpha, C) = (\alpha - B_k)\rho_{k-1}(\alpha, C) - G_{k-1}\rho_{k-2}(\alpha, C), \quad k \geq 1,$$

consider the expression for l_k given in Algorithm 1. □

3 Algorithms for computing the Geronimus transformation and numerical experiments

In this section we examine the currently available algorithms for numerically generating a Geronimus transform of a monic Jacobi matrix J. First we present the standard algorithm which can be derived from the matrix version of the Geronimus transformation given in (8). Then, we present other algorithms used in the literature.

When $C = 0$ and the shift α is close to the support of the measure, researchers [5, 10] recommend a split strategy, that is, to use a "forward algorithm" when the shift α approaches the support of the measure, and a "backward algorithm" when the shift moves away from the support.

When $C \neq 0$, we can still use the "forward algorithms". However, the "backward algorithm" does not converge and is not useful as we explain below. In this section, we also show, through numerical experiments, that the "forward algorithms" and the "backward algorithm" (when available) become less accurate as the shift moves away from the support of the measure.

From now on all the results refer to leading principal submatrices of monic Jacobi matrices. Since we are interested in the numerical analysis of algorithms that implement the Geronimus transformation, we can only consider finite matrices. We denote by

$J_n(B,G)$ the $n \times n$ leading principal submatrix of J, where $B = [B_1, ..., B_n]^T$, and $G = [G_1, ..., G_{n-1}]^T$ and $\tilde{J}_n(b,g)$ is the $n \times n$ leading principal submatrix of $\tilde{J}$, $b = [b_1, ..., b_n]^T$ being the elements on the main diagonal of $\tilde{J}_n(b,g)$, and $g = [g_1, ..., g_{n-1}]^T$ the elements on the first lower subdiagonal. Then, the finite version of the Geronimus transformation with shift α and parameter u_1 is given by

$$J_n(B,G) - \alpha I_n = U_n L_n + l_n e_n e_n^t, \quad \tilde{J}_n(b,g) = L_n U_n + \alpha I_n, \tag{8}$$

where e_n denotes the n-th column of the n-by-n identity matrix and M_n denotes the leading principal submatrix of order n of any matrix M.

Since we can only consider a finite leading principal submatrix of the initial monic Jacobi matrix as input for any algorithm to compute the Geronimus transformation, in order to determine the appropriate value of the free parameter u_1, the parameters C, μ_0, and $\mathscr{L}_0$ need to be known (as Lemma 3 shows). Thus, in all the algorithms in this paper we consider as inputs B, G, α, C, μ_0, and $\mathscr{L}_0$.

The following pseudocode gives the standard algorithm to compute the Geronimus transform with shift α and parameters C, μ_0, and $\mathscr{L}_0$ of an $n \times n$ monic Jacobi matrix $J_n(B,G)$. This algorithm is obtained from (8). Notice that

$$UL = \begin{bmatrix} u_1 + l_1 & 1 & 0 & 0 \cdots \\ u_2 l_1 & u_2 + l_2 & 1 & 0 \cdots \\ 0 & u_3 l_2 & u_3 + l_3 & 1 \cdots \\ \vdots & \vdots & \vdots & \vdots \ddots \end{bmatrix}, \quad LU = \begin{bmatrix} u_1 & 1 & 0 & 0 \cdots \\ u_1 l_1 & l_1 + u_2 & 1 & 0 \cdots \\ 0 & u_2 l_2 & l_2 + u_3 & 1 \cdots \\ \vdots & \vdots & \vdots & \vdots \ddots \end{bmatrix}.$$

Algorithm 1 *Given an $n \times n$ monic Jacobi matrix $J_n(B,G)$, this algorithm computes its Geronimus transform $\tilde{J}_n(b,g)$ of order n with shift α and parameters C, μ_0, and $\mathscr{L}_0$.*

$u_1 = \mathscr{L}_0/(C + \mu_0)$
$b_1 = u_1 + \alpha$
$for \quad i = 1 : n - 1$
$\quad l_i = B_i - u_i - \alpha$
$\quad g_i = u_i * l_i$
$\quad u_{i+1} = G_i/l_i$
$\quad b_{i+1} = u_{i+1} + l_i + \alpha$
end

The computational cost of Algorithm 1 is $6n - 2$ flops. This algorithm is closely related to the qd-algorithm proposed by Rutishauser. In [21], Rutishauser introduces σ-degree monic polynomials $p_\sigma^{(\nu)}(x)$, depending on an additional integer parameter ν and with the initial condition $p_0^{(\nu)}(x) \equiv 1$. These polynomials satisfy two basic relations:

$$p_\sigma^{(\nu)}(x) = x p_{\sigma-1}^{(\nu+1)}(x) - q_\sigma^{(\nu)} p_{\sigma-1}^{(\nu)}(x) \tag{9}$$

and

$$p_\sigma^{(\nu+1)}(x) - p_\sigma^{(\nu)}(x) = -e_\sigma^{(\nu)} p_{\sigma-1}^{(\nu+1)}(x). \tag{10}$$

Compatibility of these relations yields a three-term recurrence relation for the polynomials $p_\sigma^{(\nu)}$ (i.e. $p_{\sigma-1}^{(\nu+1)}$ are orthogonal polynomials), together with nonlinear relations for the coefficients $q_\sigma^{(\nu)}$ and $e_\sigma^{(\nu)}$ (i.e. the qd-algorithm). Relation (9) is equivalent to the

Christoffel transformation from polynomials $p_\sigma^{(\nu)}(x)$ to polynomials $p_\sigma^{(\nu+1)}(x)$, while relation (10) is equivalent to Geronimus transformation from polynomials $p_\sigma^{(\nu+1)}(x)$ to polynomials $p_\sigma^{(\nu)}(x)$. These transformations are not shifted, that is, $\alpha = 0$. However, already in the classical book [6, pg. 460], a generic scheme of the shifted Rutishauser algorithm is presented.

If we denote by $\{P_n\}$ and $\{Q_n\}$ the sequence of polynomials whose Jacobi matrix is J and $\tilde{J}$, respectively, condition (10) is equivalent in our notation to

$$P_n - Q_n = -l_n P_{n-1},$$

(see (3) and Lemma 2).

Next we present an algorithm slightly different than Algorithm 1 that can be obtained by replacing l_k by $-r_{k-1}$, using Lemma 4, and eliminating the variables u_k in Algorithm 1.

Algorithm 2 (Forward algorithm) *Given an $n \times n$ monic Jacobi matrix $J_n(B, G)$, this algorithm computes its Geronimus transform $\tilde{J}_n(b, g)$ of order n with shift α and parameters C, μ_0, and $\mathscr{L}_0$.*
$r_{-1} = -(\mu_0 + C)$
$G_0 = \mathscr{L}_0$
for $k = 0 : n - 2$
$\quad r_k = -B_{k+1} + \alpha - G_k/r_{k-1}$
end
$b_1 = B_1 + r_0$
$g_1 = \mathscr{L}_0 * r_0/r_{-1}$
for $k = 2 : n - 1$
$\quad b_k = B_k + r_{k-1} - r_{k-2}$
$\quad g_k = G_{k-1}r_{k-1}/r_{k-2}$
end
$\quad b_n = B_n + r_{n-1} - r_{n-2}$

The computational cost of this algorithm is $7n - 3$ flops.

Notice that both Algorithms 1 and 2 are "forward algorithms" since they compute l_n and r_n, respectively, for increasing values of n. However we call Algorithm 2 "Forward Algorithm" because this is the algorithm proposed by W. Gautschi [10] in the split strategy for $C = 0$.

W. Gautschi also proposes an alternative algorithm when $C = 0$, in which the quantities r_k are computed backwards. Namely, given an initial value $m \geq n$:

$$r_m = 0, \qquad r_{i-1} = \frac{G_i}{\alpha - B_{i+1} - r_i}, \qquad i = m, m-1, \ldots, 1,$$

together with $r_{-1} = \mathscr{L}_0/(\alpha - B_1 - r_0)$. Observe that this is equivalent to (6). The quantities b_k and g_k are then computed in the same way as in the forward algorithm.

In [10] Gautschi studies the properties of Algorithm 2 and the backward method. He states that the forward algorithm is better when α is very close to the support of the measure and the order n of $J_n(B, G)$ is not too large; otherwise, the backward algorithm is advised.

This backward algorithm can produce very accurate Jacobi matrices but, unlike the forward methods, it may require infeasibly large initial matrices $J_m(B, G)$ to produce an output matrix $\tilde{J}_n(b, g)$ of quite moderate dimension. Estimators for determining the

advised initial order m of $J_m(B, G)$ are given in [9] but they are only well-defined for the classical families of orthogonal polynomials.

Elhay and Kautsky [5] also suggest a split strategy in the case $C = 0$, the backward algorithm being the same as the one proposed by Gautschi. However, the forward algorithm they propose, called the Inverse Cholesky algorithm, is more expensive than Algorithm 2 (computational cost of at least $O(n^2)$) and their numerical experiments in [5] show comparable performance.

3.1 Numerical experiments

Here we present some numerical experiments that show the accuracy of the algorithms presented in the previous subsection.

In order to check the accuracy of the algorithms, we have computed the following componentwise forward errors:

$$error\ \mathbf{b} = \max_{k=1\dots n} \left\{ \left| \frac{b_k - \hat{b}_k}{b_k} \right| \right\}, \quad error\ \mathbf{g} = \max_{k=1\dots n-1} \left\{ \left| \frac{g_k - \hat{g}_k}{g_k} \right| \right\}, \quad (11)$$

where $\hat{b}_k$ and $\hat{g}_k$ denote the outputs computed by a given algorithm in standard double precision, i.e., $\mathbf{u} \approx 1.11 \times 10^{-16}$ is the unit roundoff of the finite arithmetic, while b_k and g_k denote the outputs obtained by running the same algorithm with 64 decimal digits of precision.

The experiments have been done using MATLAB 7.6.0 and the variable precision arithmetic of its Symbolic Math Toolbox. In all our tests, theoretical error bounds guarantee that the outputs obtained by running the algorithms with 64 decimal digits of precision have more than 50 significant decimal digits.

We have applied Algorithm 1, Algorithm 2 and the Backward Algorithm to the following Jacobi matrices:

1. The 60-by-60 monic Jacobi matrix corresponding to the Jacobi polynomials with parameters $a = -1/3$ and $b = 1/7$.
2. The 60-by-60 monic Jacobi matrix corresponding to the Laguerre polynomials with parameter $a = -1/3$.

In both cases, we considered a broad range of values for the shift α and two different values for the parameter $C = \{0, 10\}$. For other nonzero values of C, the behavior of the algorithms is similar to that of $C = 10$. The results can be found in Tables 1-4.

Notice that when $C = 0$, the three algorithms lose all their accuracy as the shift α moves away from the support. When $C \neq 0$, the accuracy of the algorithms also decreases as α moves away from the support although in a more moderate way. Notice that the numerical behavior of Algorithm 1 and the Forward Algorithm seems very similar.

4 A new algorithm

In this section we present a new algorithm to compute a Geronimus transform of a monic Jacobi matrix J. We will show that, with this new algorithm, the accuracy increases as α moves away from the support of the measure when $C \neq 0$. In Section

α	Error **b**	Error **g**	Error **b**	Error **g**	Error **b**	Error **g**
-1.0001	$1.4\ 10^{-11}$	$2.2\ 10^{-16}$	$2.5\ 10^{-11}$	$6.7\ 10^{-16}$	$1.3\ 10^{-11}$	$4.4\ 10^{-15}$
-1.1	16.78	1.7	29.26	0.18	25	1.6
-2	2.43	2.16	2.43	2.16	1.4	4.5
-10	43.32	1.57	43.32	1.57	26.3	1.21

Table 1 Algorithm 1–Algorithm 2–Backward algorithm.
Forward errors for Jacobi Polynomials with $a = -1/3$, $b = 1/7$, $n = 60$, $C = 0$.

α	Error **b**	Error **g**	Error **b**	Error **g**
-1.0001	$2.27\ 10^{-12}$	$2.7\ 10^{-16}$	$2.97\ 10^{-12}$	$3.33\ 10^{-16}$
-1.1	$1.5\ 10^{-11}$	$2.5\ 10^{-16}$	$2.15\ 10^{-11}$	$4.44\ 10^{-16}$
-10	$2.05\ 10^{-10}$	$3.38\ 10^{-16}$	$2.74\ 10^{-10}$	$4.2\ 10^{-16}$
-100	$1.06\ 10^{-9}$	$3.35\ 10^{-16}$	$1.16\ 10^{-9}$	$4.44\ 10^{-16}$
-10^6	$1.25\ 10^{-5}$	$3.35\ 10^{-16}$	$7.55\ 10^{-6}$	$2.22\ 10^{-16}$

Table 2 Algorithm 1–Algorithm 2.
Forward Errors for Jacobi Polynomials with $a = -1/3$, $b = 1/7$, $n = 60$, $C = 10$.

α	Error **b**	Error **g**	Error **b**	Error **g**	Error **b**	Error **g**
-0.0001	$2.1\ 10^{-16}$	$3.64\ 10^{-16}$	$1.72\ 10^{-15}$	$4,35\ 10^{-16}$	$4.9\ 10^{-1}$	$4.7\ 10^{-1}$
-0.1	$1.45\ 10^{-15}$	$2.14\ 10^{-15}$	$6.76\ 10^{-15}$	$1.07\ 10^{-14}$	$4.8\ 10^{-16}$	$6.7\ 10^{-16}$
-1	$1.71\ 10^{-6}$	$2.83\ 10^{-6}$	$1.7\ 10^{-6}$	$2.83\ 10^{-6}$	$7\ 10^{-7}$	10^{-6}
-10	2.74	44.65	2.74	44.67	1.4	2.5

Table 3 Algorithm 1–Algorithm 2–Backward algorithm.
Forward Errors for Laguerre Polynomials with $a = -1/3$, $n = 60$, $C = 0$.

α	Error **b**	Error **g**	Error **b**	Error **g**
-0.0001	$2.01\ 10^{-16}$	$3.32\ 10^{-16}$	$1.73\ 10^{-15}$	$3.86\ 10^{-16}$
-0.1	$1.04\ 10^{-15}$	$2.18\ 10^{-16}$	$1.73\ 10^{-15}$	$4.1\ 10^{-16}$
-1	$2.28\ 10^{-16}$	$2.18\ 10^{-16}$	$2.1\ 10^{-16}$	$4.36\ 10^{-16}$
-10	$3.72\ 10^{-16}$	$4.26\ 10^{-16}$	$6.19\ 10^{-16}$	$4.39\ 10^{-16}$
-100	$3.92\ 10^{-15}$	$2.7\ 10^{-16}$	$2.25\ 10^{-15}$	$2.99\ 10^{-16}$
-10^6	$1.08\ 10^{-10}$	$2.16\ 10^{-16}$	$1.08\ 10^{-10}$	$4.01\ 10^{-16}$

Table 4 Algorithm 1–Algorithm 2.
Forward Errors for Laguerre Polynomials with $a = -1/3$, $n = 60$, $C = 10$

4.6 we will also show that this new algorithm is forward stable. This means that the forward errors we get from this algorithm are the best that can be expected taking into account the conditioning of the problem.

This new algorithm does not improve the accuracy when $C = 0$ because, as we will show in Subsection 4.3, the problem of computing the Geronimus transformation of a monic Jacobi matrix when $C = 0$ is very ill-conditioned. We will also show that the conditioning of the problem depends strongly on the computation of the very first outputs and the accuracy increases notably when computing those outputs with extended accuracy and taking them as new inputs of the same algorithm.

The new algorithm that we present in this section only requires as input a monic Jacobi matrix of the same size as the output matrix. The numerical experiments will also show that the new algorithm do not improve significantly the accuracy when the shift has a moderate size due to the conditioning of the problem.

Let us define new variables $\{t_i\}_{i=1}^{n-1}$ as $t_i := l_i + \alpha$. Then, the following new algorithm to compute the Geronimus transformation with shift α and parameters C, μ_0, and $\mathscr{L}_0$ can be derived. Notice that the variables $l_1, ..., l_{n-1}$ have disappeared since they have been replaced by $t_1, ..., t_{n-1}$.

Algorithm 3 (New algorithm) *Given an $n \times n$ monic Jacobi matrix $J_n(B, G)$, this algorithm computes its Geronimus transform $\tilde{J}_n(b, g)$ of order n with shift α and parameters C, μ_0, and $\mathscr{L}_0$.*

$u_1 = \mathscr{L}_0/(C + \mu_0)$
$b_1 = u_1 + \alpha$
$for\ i = 1 : n - 1$
$\qquad t_i = B_i - u_i$
$\qquad g_i = (t_i - \alpha) * u_i$
$\qquad u_{i+1} = G_i/(t_i - \alpha)$
$\qquad b_{i+1} = u_{i+1} + t_i$
end

The computational cost of Algorithm 3 is $5n - 2$ flops.

A matrix version of this new algorithm is

$$J_n(B, G) - \alpha I_n = U_n\left(T_n - \alpha D_n\right) + l_n e_n e_n^t, \quad \tilde{J}_n(b, g) = \left(T_n - \alpha D_n\right) U_n + \alpha I_n,$$

where

$$U_n = \begin{pmatrix} u_1 & 1 & 0 & ... & 0 \\ 0 & u_2 & 1 & ... & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & ... & 0 & u_n \end{pmatrix}, \quad T_n = \begin{pmatrix} 1 & 0 & 0 & ... & 0 \\ t_1 & 1 & 0 & ... & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & ... & t_{n-1} & 1 \end{pmatrix}, \quad D_n = \begin{pmatrix} 0 & 0 & 0 & ... & 0 \\ 1 & 0 & 0 & ... & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & ... & 1 & 0 \end{pmatrix}.$$

Some numerical results are presented in Tables 5 and 6, namely, the computed forward errors by Algorithm 3. Those tables also include the condition number, which will be defined in Subsection 4.2 and whose explicit expression is given in Theorem 3. Notice that the accuracy of the outputs increases as $|\alpha|$ increases when $C \neq 0$. However, no improvement can be observed when $C = 0$.

Before carrying out a rigorous roundoff error and stability analysis of the algorithm, we can explain why the accuracy of the outputs improves when $C \neq 0$. Notice that the new algorithm is obtained from Algorithm 1 through some, apparently, slight modifications which actually have a significant influence on stability and accuracy.

We have observed that some harmful cancellations in the computation of the outputs b_i by Algorithm 1 may arise. A significant situation where this problem can be clearly understood appears when the shift α is large. It can easily be shown that $\lim_{|\alpha| \to \infty} u_k = 0$ for $k \geq 2$ (see Lemma 5 in Section 4.3), and therefore $l_i = B_i - \alpha - u_i \sim -\alpha$ when $|\alpha| \to \infty$ and $i \geq 2$, and then $b_{i+1} = u_{i+1} + l_i + \alpha \sim (-\alpha) + \alpha$ when $|\alpha| \to \infty$ and $i \geq 2$. The reader should notice that this cancellation is avoided in Algorithm 3.

From Lemma 5 in Section 4.3 we also observe that some harmful cancellations may occur in Algorithm 1 when $C = 0$ in the computation of b_1, l_1, and u_2, but these are not eliminated by Algorithm 3. In fact, these cancellations cannot be eliminated because they reflect the ill-conditioning of the problem.

Table 5 New algorithm
Forward Errors for Jacobi Polynomials
$a = -1/3$, $b = 1/7$, $n = 60$, $C = 0$ (left) and $C = 10$ (right).

α	Error **b**	Error **g**	cond	Error **b**	Error **g**	cond
-1.0001	$7.55\,10^{-12}$	$2.22\,10^{-16}$	$3.46\,10^5$	$1.34\,10^{-12}$	$2.7\,10^{-16}$	$3.59\,10^4$
-1.1	16.04	1.7	$4.87\,10^{16}$	$4.05\,10^{-12}$	$2.5\,10^{-16}$	$1.22\,10^5$
-10	43.32	1.57	$1.83\,10^{17}$	$5.53\,10^{-13}$	$3.38\,10^{-16}$	$1.12\,10^4$
-100	2.89	2.96	$1.33\,10^{17}$	$4.74\,10^{-14}$	$3.35\,10^{-16}$	$1.13\,10^3$
-1000	9.69	10.65	$7.64\,10^{16}$	$8.4\,10^{-15}$	$3.35\,10^{-16}$	113.81
-10^6	0.35	0.53	$8.61\,10^{16}$	$1.64\,10^{-15}$	$3.35\,10^{-16}$	38.4

Table 6 New algorithm
Forward Errors for Laguerre Polynomials
$a = -1/3$, $n = 60$, $C = 0$ (left) and $C = 10$ (right).

α	Error **b**	Error **g**	cond	Error **b**	Error **g**	cond
-0.0001	$2.11\,10^{-16}$	$3.64\,10^{-16}$	4.23	$2.11\,10^{-16}$	$3.32\,10^{-16}$	4.27
-0.1	$1.45\,10^{-15}$	$2.14\,10^{-15}$	$1.19\,10^3$	$1.04\,10^{-15}$	$2.18\,10^{-16}$	54.64
-1	$1.71\,10^{-6}$	$2.83\,10^{-6}$	$2.92\,10^{11}$	$2.1\,10^{-16}$	$2.18\,10^{-16}$	5.89
-10	2.74	44.67	$2.5\,10^{17}$	$1.96\,10^{-16}$	$4.26\,10^{-16}$	4.48
-100	2.24	3.83	$4.97\,10^{17}$	$2.11\,10^{-16}$	$2.7\,10^{-16}$	3
-10^6	1.15	1.21	$4.24\,10^{16}$	$2.2\,10^{-16}$	$2.16\,10^{-16}$	3

4.1 Backward error analysis of Algorithm 3

We use the standard model of floating point arithmetic [19]:

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

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

We develop our error analysis in the most general setting. For this purpose we assume that the shift α and C are real numbers, and we denote by $\hat{\alpha}$ and $\hat{C}$ the nearest floating point numbers to α and C. Similarly, we denote by $\hat{\mathscr{L}}_0$ and $\hat{\mu}_0$ the nearest floating point numbers to $\mathscr{L}_0$ and μ_0. Moreover, we assume that the input parameters $B_1, ..., B_{n-1}$ and $G_1, ..., G_{n-1}$ are each affected respectively by the small relative errors $\epsilon_{B_1}, ..., \epsilon_{B_{n-1}}, \epsilon_{G_1}, ..., \epsilon_{G_{n-1}}$, where $\max_{1 \leq i \leq n-1}\{|\epsilon_{B_i}|, |\epsilon_{G_i}|\} \leq D\mathbf{u}$, D being a moderate constant. These errors in the inputs may come from the rounding process when storing them in the computer. In addition, for the Jacobi matrices associated with families of classical orthogonal polynomials, the inputs B_i and G_i are computed using well-known formulae which may cause further errors.

Theorem 1 *Let $J_n(B, G)$ be a monic Jacobi matrix of order n. Let $\tilde{J}_n(b, g)$ be the Geronimus transform with shift α and parameters C, μ_0, and $\mathscr{L}_0$ of $J_n(B, G)$. Let $\hat{\alpha}$, $\hat{\mu}_0$, and $\hat{C}$ be the nearest floating point numbers to α, μ_0, and C. Consider the application of Algorithm 3 to the matrix with floating point entries $J_n(\hat{B}, \hat{G})$ where*

$$\hat{B}_i = B_i(1 + \epsilon_{B_i}), \quad \hat{G}_i = G_i(1 + \epsilon_{G_i}), \quad 1 \leq i \leq n-1,$$

and

$$\max_{1 \leq i \leq n-1}\{|\epsilon_{B_i}|, |\epsilon_{G_i}|\} \leq D\boldsymbol{u},$$

for a positive integer D such that $D\boldsymbol{u} \ll 1$. If $\tilde{J}_n(\hat{b}, \hat{g})$ is the matrix computed by Algorithm 3, and $\hat{L}, \hat{T}$ are the computed intermediate matrices appearing in Algorithm 3, then

$$J_n(B + \Delta B, G + \Delta G) - \hat{\alpha}I_n = \hat{U}_n(\hat{T}_n - \hat{\alpha}I_n),$$

$$\tilde{J}_n(\hat{b} + \Delta\hat{b}, \hat{g} + \Delta\hat{g}) = (\hat{T}_n - \hat{\alpha}I_n)\hat{U}_n + \hat{\alpha}I_n$$

where this transformation has parameters $\hat{C}$, $\Delta\mathscr{L}_0$, and $\hat{\mu}_0$, and

$$
\begin{aligned}
|\hat{\alpha} - \alpha| &\le \boldsymbol{u}|\alpha| \\
|\Delta\mathscr{L}_0| &\le 3\boldsymbol{u}|\mathscr{L}_0| + O(\boldsymbol{u}^2), \\
|\hat{\mu}_0 - \mu_0| &\le \boldsymbol{u}|\mu_0| \\
|\hat{C} - C| &\le \boldsymbol{u}|C|, \\
|\Delta B_i| &\le (D+1)\boldsymbol{u}(|B_i| + |\hat{u}_i|) + O(\boldsymbol{u}^2), & 1 \le i \le n-1, \\
|\Delta G_i| &\le (D+2)\boldsymbol{u}|G_i| + O(\boldsymbol{u}^2), & 1 \le i \le n-1, \\
|\Delta\hat{b}_i| &\le \boldsymbol{u}|\hat{b}_i|, & 1 \le i \le n, \\
|\Delta\hat{g}_i| &\le 2\boldsymbol{u}|\hat{g}_i| + O(\boldsymbol{u}^2), & 1 \le i \le n-1.
\end{aligned}
$$

Proof.

First observe that

$$\hat{t}_i = \left(B_i\left(1 + \epsilon_{B_i}\right) - \hat{u}_i\right)\left(1 + \epsilon_{t_i}\right), \; |\epsilon_{t_i}| \le \mathbf{u}$$

and we get

$$|\Delta B_i| = \left|\hat{t}_i + \hat{u}_i - B_i\right| \le \left((D+1)\mathbf{u} + D\mathbf{u}^2\right)(|B_i| + |\hat{u}_i|).$$

Assume that the floating point number closer to $\mathscr{L}_0$ is $\mathscr{L}_0(1 + \epsilon_L)$. Then,

$$\hat{u}_1 = \frac{\mathscr{L}_0(1 + \epsilon_L)(1 + \epsilon_{u_1})(1 + \delta_{u_1})}{\hat{C} + \hat{\mu}_0}, \quad |\epsilon_L|, |\epsilon_{u_1}|, |\delta_{u_1}| \le \mathbf{u}.$$

Therefore,

$$|\Delta\mathscr{L}_0| = \left|\mathscr{L}_0 - \hat{u}_1(\hat{C} + \hat{\mu}_0)\right| \le (3\mathbf{u} + 3\mathbf{u}^2 + \mathbf{u}^3)|\mathscr{L}_0|.$$

$$\hat{u}_{i+1} = \frac{G_i\left(1 + \epsilon_{G_i}\right)}{\hat{t}_i - \hat{\alpha}}\left(1 + \delta_{u_{i+1}}\right)\left(1 + \epsilon_{u_{i+1}}\right), \; \left|\delta_{u_{i+1}}\right|, \left|\epsilon_{u_{i+1}}\right| \le \mathbf{u}$$

which implies

$$|\Delta G_i| = \left|\left(\hat{t}_i - \hat{\alpha}\right)\hat{u}_{i+1} - G_i\right| \le \left((D+2)\mathbf{u} + (2D+1)\mathbf{u}^2 + D\mathbf{u}^3\right)|G_i|.$$

Finally,

$$\hat{b}_i\left(1 + \epsilon_{b_i}\right) = \hat{u}_{i+1} + \hat{t}_i, \; \left|\epsilon_{b_i}\right| \le \mathbf{u}.$$

$$\hat{g}_i\left(1 + \epsilon_{g_i}\right)\left(1 + \delta_{g_i}\right) = \left(\hat{t}_{i+1} - \hat{\alpha}\right)\hat{u}_i, \; |\epsilon_{g_i}|, |\delta_{g_i}| \le \mathbf{u},$$

and the results follow in a straightforward way. $\square$

In plain words, Theorem 1 says that the computed Geronimus transform $\tilde{J}_n(\hat{b}, \hat{g})$ with shift α and parameters C, μ_0, and $\mathscr{L}_0$ is almost the exact Geronimus transform of $J_n(B + \Delta B, G + \Delta G)$ with shift $\hat{\alpha}$ and parameters $C + \Delta C$, $\hat{\mu}_0$, and $\hat{\mathscr{L}}_0$.

Definition 1 [19] A method for computing $y = f(x)$ is called mixed forward-backward stable (or numerically stable) if, for any x, it produces a computed $\hat{y}$ satisfying

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

provided that ϵ and η are sufficiently small. Informally, a mixed forward-backward stable algorithm produces almost the right answer for almost the right data.

We conclude that Algorithm 3 is componentwise stable in a mixed forward-backward sense [19] if $|\hat{u}_i| = O(|B_i|)$, for $1 \leq i \leq n$. However the following problem arises: $|\Delta B_i|/|B_i|$ can be much larger than **u** if $|\hat{u}_i|$ is much larger than $|B_i|$. Unfortunately, this can happen as the following numerical experiments show. Consider the sequence of Jacobi polynomials with parameters $-1/3$, $1/7$, and the shift $\alpha = -2$. Taking into account Theorem 1, we compute a bound for the backward error as $(\epsilon \cdot errback)$, where $errback = \max_{i=1:n-1} \{1 + |\hat{u}_i/B_i|\}$, and we get

	$n = 10$	$n = 100$	$n = 1000$
$errback$, $C = 0$	$7.23 \ 10^3$	$3.5 \ 10^5$	$5.9 \ 10^6$
$errback$, $C = 10$	418	$5.7 \ 10^4$	$5.9 \ 10^6$

The previous table shows that the upper bound of the backward error is not "small". Therefore, we cannot assure mixed forward-backward stability.

4.2 Condition number

The main goal of this section is to develop a bound that allows us to estimate the forward errors of Algorithm 3 in $O(n)$ operations. We also present some numerical experiments showing that the bound obtained gives a good prediction of the forward errors produced by this algorithm.

To bound the errors in Algorithm 3, we study the sensitivity of the Geronimus transformation with respect to perturbations of the initial data, i.e., the parameters of the monic Jacobi matrix $J_n(B, G)$, the shift α, and the parameters C, μ_0 and $\mathscr{L}_0$. We consider perturbations associated with the backward errors found in Theorem 1 and we measure the sensitivity of the problem by using the notion of componentwise relative condition number. This condition number, together with Theorem 1, allows us to get a tight upper bound on the forward errors obtained by the application of Algorithm 3 to a monic Jacobi matrix. This bound is presented in Theorem 2. In the following definition the variables $u_1, u_2, \ldots, u_n$ correspond to the diagonal entries of U in the UL factorization of $J_n(B, G) - \alpha I$.

Definition 2 Let $\tilde{J}_n(b, g)$ be the Geronimus transform of order n with shift α and parameters C, μ_0, and $\mathscr{L}_0$ of the $n \times n$ monic Jacobi matrix $J_n(B, G)$. Let $\tilde{J}_n(b + \Delta b, g + \Delta g)$ be the Geronimus transform of order n with shift $\alpha + \Delta \alpha$ and parameters $C + \Delta C$, $\mu_0 + \Delta \mu_0$, and $\mathscr{L}_0 + \Delta \mathscr{L}_0$ of the $n \times n$ monic Jacobi matrix $J_n(B + \Delta B, G + \Delta G)$. Let us define

$$DB := \max \left\{ \max_{1 \leq i \leq n-1} \left\{ \frac{|\Delta B_i|}{|B_i| + |u_i|} \right\}, \max_{1 \leq i \leq (n-1)} \left\{ \frac{|\Delta G_i|}{|G_i|} \right\}, \frac{|\Delta \alpha|}{|\alpha|}, \frac{|\Delta C|}{|C|}, \frac{|\Delta \mu_0|}{|\mu_0|}, \frac{|\Delta \mathscr{L}_0|}{|\mathscr{L}_0|}, \right\},$$

where the quotient $|\Delta \alpha|/|\alpha|$ has to be understood as zero if $\alpha = 0$. Then, the relative componentwise condition number of the Geronimus transformation with shift α and

parameters C, μ_0, and $\mathscr{L}_0$ with respect to perturbations associated with the backward errors in Theorem 1 is defined as

$$\kappa(B, G, \alpha, C, \mu_0, \mathscr{L}_0) := \lim_{\delta \to 0} \sup_{0 \leq DB \leq \delta} \frac{\max\left\{ \max\limits_{1 \leq i \leq n}\left\{ \frac{|\Delta b_i|}{|b_i|} \right\}, \max\limits_{1 \leq i \leq (n-1)}\left\{ \frac{|\Delta g_i|}{|g_i|} \right\} \right\}}{DB}.$$

The condition number $\kappa(B, G, \alpha, C, \mu_0, \mathscr{L}_0)$ is infinite if some of the denominators appearing in the relative changes of the outputs b_i, i.e. $|\Delta b_i|/|b_i|$, are zero. However, $b_i = 0$ will only happen for extremely particular values of the shift α and the rest of the parameters. In those cases, other condition numbers have to be considered. For instance, measuring absolute changes in the corresponding components of b, or measuring relative normwise changes of b. We do not consider these particular situations in this work. Notice that $g_i \neq 0$ for all i since $g_i = (t_i - \alpha)u_i$ and both factors $l_i = t_i - \alpha$ and u_i are nonzero.

The condition number $\kappa(B, G, \alpha, C, \mu_0, \mathscr{L}_0)$ allows us to give an upper bound on the forward errors produced by Algorithm 3, as the following theorem shows.

Theorem 2 *Let $\tilde{J}_n(b, g)$ and $\hat{\tilde{J}}_n(\hat{b}, \hat{g})$ be the exact and computed Geronimus transform with shift α and parameters C, μ_0, and $\mathscr{L}_0$ from Algorithm 3. Then,*

$$\max_k \left\{ \left| \frac{b_k - \hat{b}_k}{b_k} \right|, \left| \frac{g_k - \hat{g}_k}{g_k} \right| \right\} \leq \left(\frac{(D+2)\mathbf{u}}{1 - (D+2)\mathbf{u}} \right) (1 + \kappa(B, G, \alpha, C, \mu_0, \mathscr{L}_0)) + O(\mathbf{u}^2),$$

where the left hand side of the previous inequality is a shorthand expression for (11) and D is the constant used in Theorem 1.

Proof. By definition of $\kappa(B, G, \alpha, C, \mu_0, \mathscr{L}_0)$,

$$\left| \frac{\Delta b_i}{b_i} \right| \leq \kappa(B, G, \alpha, C, \mu_0, \mathscr{L}_0) DB,$$

where

$$\left| \frac{\Delta b_i}{b_i} \right| = \left| \frac{\hat{b}_i + \Delta \hat{b}_i - b_i}{b_i} \right|$$

by Theorem 1. Because of Theorem 1 again,

$$DB \leq (D+2)\mathbf{u} + O(\mathbf{u}^2).$$

Therefore, to first order,

$$\left| \frac{\Delta b_i}{b_i} \right| = \left| \frac{\hat{b}_i + \Delta \hat{b}_i - b_i}{b_i} \right| \leq \kappa(B, G, \alpha, C, \mu_0, \mathscr{L}_0)(D+2)\mathbf{u}.$$

Since

$$\left| \frac{\hat{b}_i - b_i}{b_i} \right| - \left| \frac{\Delta \hat{b}_i}{b_i} \right| \leq \left| \frac{\hat{b}_i + \Delta \hat{b}_i - b_i}{b_i} \right|,$$

we get

$$\left| \frac{\hat{b}_i - b_i}{b_i} \right| \leq \kappa(B, G, \alpha, C, \mu_0, \mathscr{L}_0)(D+2)\mathbf{u} + \left| \frac{\Delta \hat{b}_i}{b_i} \right|.$$

Notice that by Theorem 1,

$$\left|\frac{\Delta \hat{b_i}}{b_i}\right| = \left|\frac{\Delta \hat{b_i}}{\hat{b_i}}\right|\left|\frac{\hat{b_i}}{b_i}\right| \leq \mathbf{u}\left|\frac{\hat{b_i} - b_i + b_i}{b_i}\right| \leq \mathbf{u}\left(1 + \left|\frac{\hat{b_i} - b_i}{b_i}\right|\right).$$

Therefore,

$$\left|\frac{\hat{b_i} - b_i}{b_i}\right| \leq (D+2)\mathbf{u}\left(1 + \kappa(B, G, \alpha, C, \mu_0, \mathscr{L}_0) + \left|\frac{\hat{b_i} - b_i}{b_i}\right|\right)$$

and the result follows for the $b_i's$. The result for g_i can be proven similarly.
□

We will provide a way to compute $\kappa(B, G, \alpha, C, \mu_0, \mathscr{L}_0)$, and therefore a bound on the forward errors, with $\mathcal{O}(n)$ cost. It is essential to remark that we have checked on the reliability of the bound on the forward errors running many numerical experiments, where we have observed that the bound does not overestimate significantly the actual errors. For an example, check Tables 5 and 6.

The entries b and g of the Geronimus transform $\tilde{J}_n(b, g)$ of $J_n(B, G)$ are rational functions of the inputs B, G, α, C, μ_0, and $\mathscr{L}_0$, and, as a consequence, b and g are differentiable functions of these parameters whenever the denominators are different from zero. Therefore, $\kappa(B, G, \alpha, C, \mu_0, \mathscr{L}_0)$ can be expressed in terms of partial derivatives [3]. More precisely:

$$\kappa(B, G, \alpha, C, \mu_0, \mathscr{L}_0) = \max\{\max_{1 \leq k \leq n}\{\kappa(b_k)\}, \max_{1 \leq k \leq n-1}\{\kappa(g_k)\}\}, \tag{12}$$

where

$$\kappa(b_k) := \sum_{i=1}^{k-1} \kappa_{B_i}(b_k) + \sum_{i=1}^{k-1} \kappa_{G_i}(b_k) + \kappa_\alpha(b_k) + \kappa_C(b_k) + \kappa_{\mathscr{L}_0}(b_k) + \kappa_{\mu_0}(b_k), \tag{13}$$

$$\kappa(g_k) := \sum_{i=1}^{k} \kappa_{B_i}(g_k) + \sum_{i=1}^{k-1} \kappa_{G_i}(g_k) + \kappa_\alpha(g_k) + \kappa_C(g_k) + \kappa_{\mathscr{L}_0}(g_k) + \kappa_{\mu_0}(g_k), \tag{14}$$

where, for $k = 1$, the sums $\sum_{i=1}^{0}$ are understood to be zero and

$$\kappa_{B_i}(b_k) := \left|\frac{|B_i| + |u_i|}{b_k}\frac{\partial b_k}{\partial B_i}\right|, \quad \kappa_C(b_k) := \left|\frac{C}{b_k}\frac{\partial b_k}{\partial C}\right|, \tag{15}$$

$$\kappa_\alpha(b_k) := \left|\frac{\alpha}{b_k}\frac{\partial b_k}{\partial \alpha}\right|, \quad \kappa_{G_i}(b_k) := \left|\frac{G_i}{b_k}\frac{\partial b_k}{\partial G_i}\right|, \tag{16}$$

$$\kappa_{\mathscr{L}_0}(b_k) := \left|\frac{\mathscr{L}_0}{b_k}\frac{\partial u_k}{\partial \mathscr{L}_0}\right|, \quad \kappa_{\mu_0}(b_k) := \left|\frac{\mu_0}{b_k}\frac{\partial b_k}{\partial \mu_0}\right|, \tag{17}$$

and analogously for $\kappa(g_k)$.

In Theorem 3, we give recurrence relations for computing $\kappa(b_k)$ and $\kappa(g_k)$ that lead to an explicit expression for $\kappa(B, G, \alpha, C, \mu_0, \mathscr{L}_0)$. Our first step to prove Theorem 3 is to express the intermediate variables u_k in Algorithm 3, and the outputs b_k and g_k as functions of the data B, G, α, C, μ_0, and $\mathscr{L}_0$. Then, we obtain expressions for the partial derivatives of each of these functions with respect to their arguments. A detailed proof of this theorem can be found in Appendix 1.

Theorem 3 *Let $J_n(B,G)$ be any $n \times n$ Jacobi matrix, and let α, C, μ_0, and $\mathscr{L}_0$ be real numbers such that $J_n(B,G) - \alpha I$ has an UL factorization, where $u_1 = \mathscr{L}_0/(C + \mu_0)$. Let U be the upper bidiagonal factor in the UL factorization of $J_n(B,G) - \alpha I$. If $u_1, u_2, ..., u_n$ are the entries of U in positions (1,1),(2,2),...,(n,n), then*

$$\kappa(b_1) = \left| \frac{\alpha}{b_1} \right| \left| 1 + \frac{\partial u_1}{\partial \alpha} \right| + \left| \frac{u_1}{b_1} \right| |\kappa^*(u_1)|,$$

$$\kappa(b_k) = \frac{|u_k|}{|b_k|} + \frac{|\gamma_{k-1} u_k - 1|}{|b_k|} \left[|B_{k-1}| + |u_{k-1}| \left(1 + \kappa^*(u_{k-1}) \right) \right]$$

$$+ \frac{|\alpha|}{|b_k|} \left| (\gamma_{k-1} u_k - 1) \frac{\partial u_{k-1}}{\partial \alpha} + \gamma_{k-1} u_k \right|, \quad k \geq 2,$$

$$\kappa(g_k) = |\gamma_k| \left[|B_k| + |u_k| + |\delta_k| \kappa^*(u_k) \right] + \left| \frac{\alpha}{g_k} \right| \left| \delta_k \frac{\partial u_k}{\partial \alpha} - u_k \right|, \quad k \geq 1,$$

where

$$\kappa^*(u_1) = 1 + \frac{|C| + |\mu_0|}{|C + \mu_0|},$$

$$\kappa^*(u_k) = 1 + |\gamma_{k-1}| \left[|B_{k-1}| + |u_{k-1}|(1 + \kappa^*(u_{k-1})) \right], \quad k \geq 2,$$

$$\frac{\partial u_k}{\partial \alpha} = \begin{cases} -\dfrac{u_1}{C + \mu_0} \dfrac{\partial \mu_0}{\partial \alpha}, & k = 1 \\[2mm] \gamma_{k-1} u_k \left(1 + \dfrac{\partial u_{k-1}}{\partial \alpha} \right), & k > 1 \end{cases}$$

and

$$\gamma_k := \frac{1}{B_k - u_k - \alpha}, \qquad \delta_k := B_k - 2u_k - \alpha, \qquad k \geq 1.$$

Remark 1 It is possible to develop a roundoff error analysis of Algorithm 1 similar to the analysis done for Algorithm 3. To begin with, backward error bounds for Algorithm 1 can be found. Then, it is also possible to deduce recurrence relations for a relative componentwise condition number, $\kappa_A(B, G, \alpha, C, \mu_0, \mathscr{L}_0)$, for the Geronimus transformation with respect to perturbations in the input data associated with the backward errors of Algorithm 1. Finally, the condition number $\kappa_A(B, G, \alpha, C, \mu_0, \mathscr{L}_0)$ can be used in a counterpart version of Theorem 2 for Algorithm 1 to bound the forward errors. We do not include the details of these results to keep the paper concise. However, we would like to remark that it is easy to prove that

$$\kappa(B, G, \alpha, C, \mu_0, \mathscr{L}_0) \leq \kappa_A(B, G, \alpha, C, \mu_0, \mathscr{L}_0)$$

for all monic Jacobi matrices $J_n(B,G)$, all shifts α, and all the possible values of the parameters C, μ_0 and $\mathscr{L}_0$. This fact, together with the numerical experiments in Subsection 3.1, show that Algorithm 3 is more accurate than Algorithm 1.

Similar remarks can be made regarding Algorithm 2.

4.3 Stability and accuracy of the new algorithm for large shifts

There are some interesting results that we can prove related to the stability and accuracy of Algorithm 3 beyond the fact of being more accurate than Algorithm 1 or 2. It can be proven that, for large enough values of the shift α and under some small constraints, for $C \neq 0$, Algorithm 3 is *accurate*, i.e., it produces outputs with componentwise forward errors of order $O(\mathbf{u})$. To prove this, we will show that

$$\lim_{|\alpha| \to \infty} \kappa(B, G, \alpha, C, \mu_0, \mathscr{L}_0) = \max \left\{ 3, \frac{|B_1| + 3|\mathscr{L}_0/C|}{|B_1 - \mathscr{L}_0/C|} \right\}$$

Therefore, Theorem 2 guarantees accuracy if the quantity on the right is "small". The numerical experiments in Subsection 3.1 show that this is not the case for Algorithms 1, 2, or the backward algorithm. In fact, it can be proven that the accuracy of those algorithms decreases as $|\alpha|$ grows.

Let us recall that, if $C \neq 0$, according to Theorem 1, if $|\hat{u}_i| = O(|B_i|)$ for $1 \leq i \leq n$, then Algorithm 3 is mixed forward-backward stable, which is the usual requirement for a numerical algorithm to be considered stable [19, p. 7]. More precisely, in this case, it can be said that the computed Geronimus transform $\tilde{J}_n(\hat{b}, \hat{g})$ with shift α and parameters C, μ_0, and $\mathscr{L}_0$ of $J_n(B, G)$ is an $O(\mathbf{u})$ relative componentwise perturbation of the *exact* Geronimus transform with shift $\hat{\alpha}$ and parameters $\hat{C}$, $\hat{\mu}_0$, and $\Delta\mathscr{L}_0$ of $J_n(B + \Delta B, G + \Delta G)$, where $\Delta\mathscr{L}_0$, ΔB and ΔG are $O(\mathbf{u})$ relative componentwise perturbations of the *exact* inputs $\mathscr{L}_0$, B and G. In this context, another goal of this subsection is to prove that for large enough values of the shift, $|u_i| \ll |B_i|$ and then Algorithm 3 is stable. We have to admit that this will be proven for the exact values of u_i and not for the computed values $\hat{u}_i$, thus we are only proving stability up to $O(\mathbf{u}^2)$ terms.

Here we will also show that the condition number $\kappa(B, G, \alpha, C, \mu_0, \mathscr{L}_0)$ becomes very large as $|\alpha|$ grows when $C = 0$. In Appendix 2 we show that this condition number has the same magnitude as the standard condition number of the problem which implies that no accuracy can be expected from any algorithm to compute the Geronimus transformation when $C = 0$ and the shift moves away from the support of the measure. Moreover, Lemma 5 shows that $|B_1| + |u_1| \gg |B_1|$ when $|\alpha|$ grows which implies that no stability can either be expected from Algorithm 3.

We start with some technical lemmas. Firstly, the different numerical behavior of the Geronimus transformation when $C = 0$ and when $C \neq 0$ can be partially explained by using the following result. Notice that the parameter u_1 can be seen as a function of α.

Lemma 5 *Let $J_n(B, G)$ be the leading principal submatrix of a monic Jacobi matrix J. Let α, C, μ_0, and $\mathscr{L}_0$ be real numbers such that there is a unique UL factorization of $J_n(B, G) - \alpha I$. Let u_k, $1 \leq k \leq n$, be the main diagonal elements in the U factor. Then,*

− *if $C \neq 0$,*

$$\lim_{|\alpha| \to \infty} u_1 = \frac{\mathscr{L}_0}{C}, \quad \lim_{|\alpha| \to \infty} u_k = 0, \quad k \geq 2.$$

As a consequence, when $C \neq 0$, Algorithm 3 is stable for $|\alpha|$ large enough if $|\mathscr{L}_0/C| = O(|B_1|)$.

− *if $C = 0$,*

$$\lim_{|\alpha| \to \infty} |u_1| = \infty, \quad (u_1 \sim -\alpha),$$

$$\lim_{|\alpha| \to \infty} u_2 = \frac{G_1}{B_1}, \quad \lim_{|\alpha| \to \infty} u_k = 0, \quad k \geq 3.$$

Proof. First, assume $C \neq 0$. The proof follows directly from the expressions:

$$u_1 = \frac{\mathscr{L}_0}{C + \mu_0}, \qquad u_k = \frac{G_{k-1}}{B_{k-1} - u_{k-1} - \alpha}, \qquad k \geq 2,$$

using induction and the fact that $\mu_0 \to 0$, since

$$\lim_{|\alpha| \to \infty} \alpha \mu_0 = \lim_{|\alpha| \to \infty} \alpha \int \frac{d\mu}{x - \alpha} = -\mathscr{L}_0.$$

The limit and the integral can be interchanged if $\alpha \notin \text{supp } \mu$, because $\alpha/(x - \alpha)$ is a continuous function. As a consequence, $u_1 \to \mathscr{L}_0/C$ when $|\alpha| \to \infty$ if $C \neq 0$.

When $C = 0$, the previous result gives $u_1 \sim -\alpha$, so $u_2 \to G_1/B_1$ when $|\alpha| \to \infty$. This implies that $u_3 \to 0$ and the second claim follows by induction.
□

Lemma 6 *When $C \neq 0$ it is true that $\gamma_k \to 0$ and $\gamma_k \delta_k \to 1$ when $|\alpha| \to \infty$, $k \geq 1$.*

Proof. It follows from the definition of γ_k and δ_k and the asymptotic properties of u_k in Lemma 5. □

Lemma 7 *Let $J_n(B, G)$ be the leading principal submatrix of a monic Jacobi matrix J of size n. Let α, C, μ_0, and $\mathscr{L}_0$ be real numbers such that there is a unique UL factorization of $J_n(B, G) - \alpha I$. Let u_1 be the element in position $(1, 1)$ in the U factor. Then,*

$$\lim_{|\alpha| \to \infty} \frac{\partial u_k}{\partial \alpha} = \begin{cases} -1, & C = 0 \text{ and } k = 1, \\ 0, & \text{otherwise.} \end{cases}$$

Proof. Taking into account the definition of u_1, when $\alpha \notin \text{supp } \mu$ then

$$\frac{\partial u_1}{\partial \alpha} = \frac{-\mathscr{L}_0}{(C + \mu_0)^2} \frac{\partial \mu_0}{\partial \alpha} = -\frac{\mathscr{L}_0}{(C + \mu_0)^2} \int_a^b \frac{d\mu}{(x - \alpha)^2}.$$

The result follows from the observation that

$$\mu_0 = -\frac{\mathscr{L}_0}{\alpha} + \mathcal{O}(\alpha^{-2}), \qquad \frac{\partial \mu_0}{\partial \alpha} = \frac{\mathscr{L}_0}{\alpha^2} + \mathcal{O}(\alpha^{-3}), \qquad |\alpha| \to \infty.$$

For $k \geq 2$ we can use induction on k, noting that

$$\frac{\partial u_k}{\partial \alpha} = \frac{u_k}{B_{k-1} - u_{k-1} - \alpha} \left(1 + \frac{\partial u_{k-1}}{\partial \alpha} \right).$$

and considering Lemma 5. Note that $\partial u_k/\partial \alpha = \mathcal{O}(\alpha^{-1})$ when $|\alpha| \to \infty$.
□

4.4 Asymptotic analysis of the condition number when $C \neq 0$.

In this subsection we present an analysis of the condition number of Algorithm 3 when $C \neq 0$.

Lemma 8 *If $C \neq 0$, then*

$$\lim_{|\alpha| \to \infty} \frac{B_1 - 2u_1 - \alpha}{u_1} \frac{\partial u_1}{\partial \alpha} = 0, \qquad \lim_{|\alpha| \to \infty} \alpha \frac{\partial u_k}{\partial \alpha} = 0, \qquad k \geq 1$$

Proof. From the previous estimations it follows that when $C \neq 0$ then

$$\frac{\partial u_1}{\partial \alpha} = -\frac{\mathscr{L}_0^2}{C\alpha^2} + \mathcal{O}(\alpha^{-3}),$$

so the second part of the lemma is true for $k = 1$. Assume that the result holds for $k - 1$. Then, notice that

$$\alpha \frac{\partial u_k}{\partial \alpha} = u_k \frac{\alpha}{B_{k-1} - u_{k-1} - \alpha} + \frac{u_k}{B_{k-1} - u_{k-1} - \alpha} \left(\alpha \frac{\partial u_{k-1}}{\partial \alpha} \right).$$

Taking limits the second result follows. The first part of the lemma is obtained directly from the asymptotic estimations of $\partial u_1 / \partial \alpha$ and $\alpha \partial u_1 / \partial \alpha$ given above and the fact that $u_1 \to \mathscr{L}_0 / C$ when $\alpha \to \infty$ and $C \neq 0$. $\square$

Theorem 4 *If $C \neq 0$, then*

$$\lim_{|\alpha| \to \infty} \kappa^*(u_1) = 2, \qquad \lim_{|\alpha| \to \infty} \kappa^*(u_k) = 1, \quad k \geq 2.$$

$$\lim_{|\alpha| \to \infty} \kappa(b_k) = 1, \quad for \ k \neq 2, \qquad \lim_{|\alpha| \to \infty} \kappa(b_2) = \frac{|B_1| + 3|\mathscr{L}_0 / C|}{|B_1 - \mathscr{L}_0 / C|}.$$

$$\lim_{|\alpha| \to \infty} \kappa(g_1) = 3, \qquad \lim_{|\alpha| \to \infty} \kappa(g_k) = 1, \quad k \geq 2.$$

Proof. We prove the result by induction on k. Since $\lim_{|\alpha| \to \infty} \mu_0 = 0$,

$$\lim_{|\alpha| \to \infty} \kappa^*(u_1) = 2.$$

It is easy to show that $\kappa^*(u_2) = 1$. Assume that $\lim_{|\alpha| \to \infty} \kappa^*(u_{k-1}) = 1$ for some $k \geq 3$. Then, taking into account Lemma 5, we get

$$\lim_{|\alpha| \to \infty} |\gamma_{k-1} B_{k-1}| = 0, \qquad \lim_{|\alpha| \to \infty} |\gamma_{k-1} u_{k-1}| = 0,$$

which implies the result for u_k. Recall that $b_1 = u_1 + \alpha$. Then, taking into account Theorem 4, Lemmas 5 and 7 the result follows for $k = 1$. For $k = 2$, we apply Theorem 4, Lemmas 5–8, bearing in mind that $b_k = B_{k-1} + u_k - u_{k-1}$, $k \geq 2$.

$$\lim_{|\alpha| \to \infty} \frac{|u_2|}{|b_2|} = 0, \qquad \lim_{|\alpha| \to \infty} \frac{|B_1| + |u_1|}{|b_2|} = \frac{|B_1| + |\mathscr{L}_0 / C|}{|B_1 - \mathscr{L}_0 / C|},$$

$$\lim_{|\alpha| \to \infty} |\gamma_1 u_2 - 1| \left| \frac{u_1}{b_2} \right| \kappa^*(u_1) = 2 \left| \frac{\mathscr{L}_0 / C}{B_1 - \mathscr{L}_0 / C} \right|,$$

$$\lim_{|\alpha|\to\infty} \left|\frac{\alpha}{b_2}\right| \left|\frac{\partial b_2}{\partial \alpha}\right| = \lim_{|\alpha|\to\infty} \left|\frac{\alpha}{b_2}\right| \left|\frac{\partial u_2}{\partial \alpha} - \frac{\partial u_1}{\partial \alpha}\right| = 0.$$

Let $k \geq 3$, then

$$\lim_{|\alpha|\to\infty} \frac{|u_k|}{|b_k|} = 0, \qquad \lim_{|\alpha|\to\infty} |\gamma_{k-1} u_k - 1| \frac{|B_{k-1}| + |u_{k-1}|}{|b_k|} = 1,$$

$$\lim_{|\alpha|\to\infty} |\gamma_{k-1} u_k - 1| \left|\frac{u_{k-1}}{b_k}\right| \kappa^*(u_{k-1}) = 0,$$

$$\lim_{|\alpha|\to\infty} \left|\frac{\alpha}{b_k}\right| \left|\frac{\partial b_k}{\partial \alpha}\right| = \lim_{|\alpha|\to\infty} \left|\frac{\alpha}{b_k}\right| \left|\frac{\partial u_k}{\partial \alpha} - \frac{\partial u_{k-1}}{\partial \alpha}\right| = 0,$$

by Lemma 8 and the result for b_k follows. Finally, for $k = 1$,

$$\lim_{|\alpha|\to\infty} |\gamma_1| [|B_1| + |u_1|] = 0, \qquad \lim_{|\alpha|\to\infty} |\gamma_1 \delta_1| \kappa^*(u_1) = 2,$$

$$\lim_{|\alpha|\to\infty} \left|\frac{\alpha}{g_1}\right| \left|(B_1 - 2u_1 - \alpha)\frac{\partial u_1}{\partial \alpha} - u_1\right| =$$

$$\lim_{|\alpha|\to\infty} \left|\frac{\alpha}{B_1 - u_1 - \alpha}\right| \left|\frac{B_1 - 2u_1 - \alpha}{C + \mu_0}\frac{\partial \mu_0}{\partial \alpha} - 1\right| = 1.$$

The last equality follows from Lemma 8.

For $k \geq 2$, notice that

$$\lim_{|\alpha|\to\infty} |\gamma_k| [|B_k| + |u_k|] = 0, \qquad \lim_{|\alpha|\to\infty} |\gamma_k \delta_k| \kappa^*(u_1) = 1,$$

$$\lim_{|\alpha|\to\infty} \left|\frac{\alpha}{g_k}\right| \left|(B_k - 2u_k - \alpha)\frac{\partial u_k}{\partial \alpha} - u_k\right| =$$

$$\left|\frac{\alpha}{B_k - u_k - \alpha}\right| \left|\frac{B_k - 2u_k - \alpha}{u_k}\frac{u_k}{B_{k-1} - u_{k-1} - \alpha}\left(1 + \frac{\partial u_{k-1}}{\partial \alpha}\right) - 1\right| = 0.$$

taking into account Lemma 7.

∎

Theorem 5 *Let $\kappa(B, G, \alpha, C, \mu_0, \mathscr{L}_0)$ be the condition number for the Geronimus transformation with shift α and parameters $C \neq 0$, μ_0, and $\mathscr{L}_0$ introduced in Definition 12. Then*

$$\lim_{|\alpha|\to\infty} \kappa(B, G, \alpha, C, \mu_0, \mathscr{L}_0) = \max\left\{3, \frac{|B_1| + 3|\mathscr{L}_0/C|}{|B_1 - \mathscr{L}_0/C|}\right\}.$$

This implies that Algorithm 3 is accurate for $|\alpha|$ large enough as long as $\frac{|B_1| + 3|\mathscr{L}_0/C|}{|B_1 - \mathscr{L}_0/C|}$ is small.

Proof. It is a direct consequence of Theorems 3 and 4. ∎

4.5 Asymptotic analysis of the condition number when $C = 0$

Next we present a similar analysis of the condition number for the case $C = 0$. Note the different behavior of $\kappa(b_k)$ and $\kappa(g_k)$ with respect to the previous subsection.

Theorem 6 *If $C = 0$, then*

$$\lim_{|\alpha| \to \infty} \kappa^*(u_1) = 2, \quad \lim_{|\alpha| \to \infty} \kappa^*(u_2) = \infty.$$

$$\lim_{|\alpha| \to \infty} \kappa^*(u_3) = 1 + 3 \left| \frac{G_1}{B_1^2} \right|, \quad \lim_{|\alpha| \to \infty} \kappa^*(u_k) = 1, \quad \text{for all } k \geq 4,$$

$$\lim_{|\alpha| \to \infty} \kappa(b_1) = \infty, \quad \lim_{|\alpha| \to \infty} \kappa(b_2) = 3 \left| 1 - \frac{G_1}{B_1^2} \right|,$$

$$\lim_{|\alpha| \to \infty} \kappa(b_3) = \infty, \quad \lim_{|\alpha| \to \infty} \kappa(b_k) = 1, \quad \text{for } k \geq 4$$

$$\lim_{|\alpha| \to \infty} \kappa(g_1) = \infty, \quad \lim_{|\alpha| \to \infty} \kappa(g_2) = \infty,$$

$$\lim_{|\alpha| \to \infty} \kappa(g_3) = 1 + 3 \left| \frac{G_1}{B_1^2} \right|, \quad \lim_{|\alpha| \to \infty} \kappa(g_k) = 1, \quad \text{for } k \geq 4$$

Proof. The result for u_1 follows in a straightforward way. In the expression for $\kappa^*(u_2)$ notice that

$$\lim_{|\alpha| \to \infty} \frac{|u_1|}{|B_1 - u_1 - \alpha|} (1 + \kappa^*(u_1)) = \infty$$

taking into account Lemma 5.

Notice that $\kappa^*(u_3)$ can also be expressed as

$$1 + \frac{|B_2|}{|B_2 - u_2 - \alpha|} + \frac{|u_2|}{|B_2 - u_2 - \alpha|} \left(2 + \frac{|B_1|}{|B_1 - u_1 - \alpha|} \right) +$$

$$\frac{|u_2|}{|B_1 - u_1 - \alpha|} \frac{|u_1|}{|B_2 - u_2 - \alpha|} (1 + \kappa^*(u_1)).$$

Notice that the limit when $|\alpha|$ grows of the first three terms in the previous expression is 1, while the limit of the last term is $3|G_1/B_1^2|$. Now it is easy to show the result for $k = 4$. The rest of the cases follow by induction.

In order to compute the condition numbers of the b's, note first that

$$\kappa(b_1) = \left| \frac{\alpha}{b_1} \right| \left| 1 + \frac{\partial u_1}{\partial \alpha} \right| + 2 \left| \frac{u_1}{b_1} \right|.$$

Taking into account Lemmas 7 and 5, the result follows. In a similar way it is possible to prove the result for the other condition numbers $\kappa(b_k)$ and $\kappa(g_k)$. $\square$

The previous results suggest that better accuracy can be obtained when computing the Geronimus transformation with $C = 0$ using the new algorithm (Algorithm 3) if at least the following outputs are computed with extended accuracy: $u_1, u_2, u_3, b_1, b_2, b_3, b_4, g_1, g_2, g_3$ and then use these values as inputs of the same algorithm. Check Table 7 for new numerical results. The computations of the 4-by-4 principal leading submatrix of the Geronimus transform $\tilde{J}$ as well as the the first three main diagonal entries of the factor U were done with 64 decimal digits of precision.

α	Error **b**	Error **g**
-1.0001	$1.31\ 10^{-11}$	$2.22\ 10^{-16}$
-1.1	91.26	1.74
-2	$9.3\ 10^{-3}$	$1.67\ 10^{-2}$
-10	$1.41\ 10^{-5}$	$5.73\ 10^{-7}$
-100	$5.29\ 10^{-10}$	$5.28\ 10^{-10}$
-1000	$1.59\ 10^{-12}$	$1.59\ 10^{-12}$
-10^6	$2.21\ 10^{-16}$	$2.22\ 10^{-16}$

α	Error **b**	Error **g**
-0.0001	$2.1\ 10^{-16}$	$3.64\ 10^{-16}$
-0.1	$1.83\ 10^{-16}$	$2.31\ 10^{-16}$
-1	$1.41\ 10^{-7}$	$2.34\ 10^{-7}$
-10	$4.5\ 10^{-3}$	$9.3\ 10^{-3}$
-100	$2.38\ 10^{-8}$	$4\ 10^{-8}$
-1000	$3.65\ 10^{-12}$	$3.59\ 10^{-12}$
-10^6	$2.2\ 10^{-16}$	$2.89\ 10^{-16}$

Table 7 Algorithm with extended accuracy. Forward Errors for $n = 60$ and $C = 0$. On the left, Jacobi Polynomials with $a = -1/3$, $b = 1/7$. On the right, Laguerre Polynomials with $a = -1/3$.

4.6 Forward stability of Algorithm 3

The purpose of this section is to prove that the forward error bound we have found for Algorithm 3 is the best one can expect, because it reflects the sensitivity of the transformation to componentwise relative perturbations in the data. We have seen that Algorithm 3 is neither backward stable nor stable in the mixed backward-forward sense, and therefore we consider a weaker notion of stability. An algorithm is said to be *forward stable* if it produces forward errors of similar magnitude to those produced by a backward stable algorithm [19, p. 9]. In this section we show that Algorithm 3 is componentwise forward stable. In order to prove that, we define the relative componentwise condition number of the Geronimus transformation with shift α and parameters C, μ_0, and $\mathscr{L}_0$ with respect to small componentwise relative perturbations of B, G, α, C, μ_0, and $\mathscr{L}_0$.

$$\kappa_S(B,G,\alpha,C,\mu_0,\mathscr{L}_0) = \lim_{\delta\to 0}\sup_{0\le DC\le\delta}\frac{\max\left\{\max\limits_{1\le i\le(n)}\left\{\frac{|\Delta b_i|}{|b_i|}\right\},\max\limits_{1\le i\le(n-2)}\left\{\frac{|\Delta g_i|}{|g_i|}\right\}\right\}}{DC},$$

(18)

where

$$DC = \max\left\{\max_{1\le i\le(n)}\left\{\frac{|\Delta B_i|}{|B_i|}\right\},\max_{1\le i\le(n-1)}\left\{\frac{|\Delta G_i|}{|G_i|}\right\},\frac{|\Delta\alpha|}{|\alpha|},\frac{|\Delta C|}{|C|},\frac{|\Delta\mu_0|}{|\mu_0|},\frac{|\Delta\mathscr{L}_0|}{|\mathscr{L}_0|}\right\}.$$

Recurrent expressions for $\kappa_S(B,G,\alpha,C,\mu_0,\mathscr{L}_0)$ can be obtained in a similar way as we got recurrent expressions for $\kappa(B,G,\alpha,C,\mu_0,\mathscr{L}_0)$.

Theorem 7 *Let $J_n(B,G)$ be any $n\times n$ monic Jacobi matrix, and let α, C, μ_0, and $\mathscr{L}_0$ be real numbers such that $J_n(B,G) - \alpha I$ has a unique UL factorization with $u_1 = \mathscr{L}_0/(C+\mu_0)$. Let U be the upper bidiagonal factor in the UL factorization of $J_n(B,G) - \alpha I$. If $u_1, u_2, ..., u_n$ are the entries of U in positions $(1,1),(2,2),...,(n,n)$, then*

$$\kappa_S(b_1) = \left|\frac{\alpha}{b_1}\right|\left|1+\frac{\partial u_1}{\partial\alpha}\right| + \left|\frac{u_1}{b_1}\right||\kappa_S^*(u_1)|,$$

$$\kappa_S(b_k) = \frac{|u_k|}{|b_k|} + \frac{|\gamma_{k-1}u_k - 1|}{|b_k|}\left[|B_{k-1}| + |u_{k-1}|\kappa_S^*(u_{k-1})\right],$$

$$+ \left|\frac{\alpha}{b_k}\right|\left|(\gamma_{k-1}u_k - 1)\frac{\partial u_{k-1}}{\partial\alpha} + \gamma_{k-1}u_k\right|, \quad k\ge 2,$$

$$\kappa_S(g_k) = |\gamma_k| \left[|B_k| + |\delta_k|\kappa_S^*(u_k)\right] + \left|\frac{\alpha}{g_k}\right| \left|\delta_k \frac{\partial u_k}{\partial \alpha} - u_k\right|, \quad k \geq 1,$$

where

$$\kappa_S^*(u_1) = 1 + \frac{|C| + |\mu_0|}{|C + \mu_0|},$$
$$\kappa_S^*(u_k) = 1 + |\gamma_{k-1}| \left[|B_{k-1}| + |u_{k-1}|\kappa_S^*(u_{k-1})\right], \quad k \geq 2,$$

and

$$\frac{\partial u_k}{\partial \alpha} = \begin{cases} -\dfrac{u_1}{C + \mu_0} \dfrac{\partial \mu_0}{\partial \alpha}, & k = 1, \\[12pt] \gamma_{k-1} u_k \left(1 + \dfrac{\partial u_{k-1}}{\partial \alpha}\right), & k > 1. \end{cases}$$

Proof. Analogous to the proof of Theorem 3. □

To prove that Algorithm 3 is componentwise forward stable is equivalent to prove that $\kappa_S(B, G, \alpha, C, \mu_0, \mathscr{L}_0)$ and $\kappa(B, G, \alpha, C, \mu_0, \mathscr{L}_0)$ have the same order of magnitude, by taking into account Theorem 2.

By using Theorem 7, we can prove Theorem 8, after considerably long and delicate algebraic manipulations are performed. The complete proof can be found in Appendix 2. This theorem states that the condition numbers, $\kappa(B, G, \alpha, C, \mu_0, \mathscr{L}_0)$ and $\kappa_S(B, G, \alpha, C, \mu_0, \mathscr{L}_0)$ that we have defined for the Geronimus transformation are of the same order of magnitude, which implies that Algorithm 3 is forward stable.

Theorem 8 *Let $\kappa(B, G, \alpha, C, \mu_0, \mathscr{L}_0)$ and $\kappa_S(B, G, \alpha, C, \mu_0, \mathscr{L}_0)$ be the condition numbers introduced, respectively, in Definition 2 and (18) for the Geronimus transformation with shift α and parameters C, μ_0 and $\mathscr{L}_0$, then*

$$\kappa_S(B, G, \alpha, C, \mu_0, \mathscr{L}_0) \leq \kappa(B, G, \alpha, C, \mu_0, \mathscr{L}_0) \leq 8\,\kappa_S(B, G, \alpha, C, \mu_0, \mathscr{L}_0). \qquad (19)$$

This result together with the fact that $\kappa(B, G, \alpha, C, \mu_0, \mathscr{L}_0) \geq 1$ implies that Algorithm 3 is componentwise forward stable.

References

1. M.I. Bueno and F. Marcellán *Darboux transformation and perturbation of linear functionals*, Linear Algebra and Applications, **384** 215–242(2004).
2. M. Buhmann and A. Iserles, *On orthogonal polynomials transformed by the QR algorithm*, Journal Comp. Appl. Math. **43** 117–134(1992).
3. F. Chaitin-Chatelin and V. Fraysse, *Lectures on finite precision computations*, SIAM, Philadelphia, (1996)
4. T.S. Chihara, *An Introduction to Orthogonal Polynomials*, Gordon and Breach, New York, (1957)
5. S. Elhay and J. Kautsky, *Jacobi matrices for measures modified by a rational factor*, Numerical Algorithms **6**, pp. 205-227(1994)
6. D.K. Faddeev and V.N. Faddeeva, *Computational methods of linear algebra*, Transl. from Russian, W.H. Freeman, 1963.
7. D. Galant, *An implementation of Christoffel's Theorem in the Theory of Orthogonal Polynomials*, Math. Comput., **25**, 111–113(1971).
8. D. Galant, *Algebraic Methods for modified orthogonal polynomials*, Math. Comput., **59**, 541–546(1992).

9. W. GAUTSCHI, *Minimal solutions of three-term recurrence relations and orthogonal polynomials*, Mathematics of Computation, **36**, No.154, pp. 547-554(1981)

10. W. GAUTSCHI, *Orthogonal polynomials: Computations and approximation*, Oxford University Press, New York, (2004)

11. W. GAUTSCHI, *Computational aspects of three-tem recurrence relations*, SIAM Review 9, no. 1 24-82(1967). .

12. W. GAUTSCHI, *The interplay between classical analysis and (numerical) linear algebra- A tribute to Gene H. Golub*, Electron. Trans. Numer. Anal., **13**, 119-147(2002).

13. YA. L. GERONIMUS, *On the polynomials orthogonal with respect to a given number sequence and a theorem* ed W. Hahn, Izv. Akad. Nauk, **4**, 215-228(1940). (in Russian)

14. YA. L. GERONIMUS, *On the polynomials orthogonal with respect to a given number sequence*, Zap. Mat. Otdel. Khar'kov. Univers. i NII Mat. i Mehan., **17**, 3-18(1940).

15. G.H. GOLUB AND J. KAUTSKY, *Calculation of Gauss Quadratures with Multiple Free and Fixed Knots*, Numer. Math., **41**, 147–163(1983).

16. F.A. GRÜNBAUM AND L. HAINE, *Orthogonal Polynomials satisfying Differential Equations: the Role of the Darboux Transformation*, CRM Proceedings and Lecture Notes, American Mathematical Society. Providence, Rhode Island. **9**, 143–154(1996).

17. F.A. GRÜNBAUM AND L. HAINE, *Bispectral Darboux Transformations:An Extension of the Krall Polynomials*, Internat. Math. Res. Notices, **8**, 359–392(1997).

18. F.A. GRÜNBAUM, J. HAINE, AND E. HOROZOV, *Some functions that generalize the Krall-Laguerre polynomials*, J. of Comp. Appl. Math., **106**, 271–297 (1999).

19. N.J. HIGHAM, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, (2002)

20. V.B. MATVEEV AND M.A. SALLE, *Differential-Difference Evolution Equations. II (Darboux Transformation for the Toda Lattice)* , Letters in Math. Physics, **3**, 425–429(1979).

21. H. RUTISHAUSER, *Der Quotienten-differenzen-algorithmus*, Birkhäuser, 1957.

22. V. SPIRIDONOV, L. VINET, AND A. ZHEDANOV, *Spectral transformations, self-similar reductions and orthogonal polynomials*, J. Phys. A: Math. & Gen., **30**, 7621–7637(1997).

23. V. SPIRIDONOV AND A. ZHEDANOV, *Discrete Darboux transformations, discrete time Toda lattice and the Askey-Wilson polynomials*, Methods and Appl. Anal., **2**, 369–398(1995).

24. V.B. UVAROV, *The connection between systems of polynomials that are orthogonal with respect to different distribution functions*, Ah. Vychisl. Matem. i Mat. Fiz., **9**, 1253–62(1969).

25. G.J. YOON, *Darboux transformation and orthogonal polynomials*, Bull. Korean Math. Soc. **39**, No.3, pp. 359-376(2002)

APPENDIX 1: Proof of Theorem 3

In this section we include the proof of Theorem 3. First, we express the intermediate variables u_k of Algorithm 1, and the outputs b_k and g_k as functions of the data B, G, α, C, μ_0, and $\mathscr{L}_0$. Then we obtain expressions of the partial derivatives of each of the functions with respect to their arguments. From Algorithm 1, we get

$$u_1 = \frac{\mathscr{L}_0}{C + \mu_0}, \quad u_k = \frac{G_{k-1}}{B_{k-1} - u_{k-1} - \alpha}, \quad k \geq 2, \tag{A-1}$$

and hence, for $k \geq 2$, u_k can be seen as a function of $B_1, ..., B_{k-1}, G_1, ..., G_{k-1}, \alpha, C, \mu_0$, and $\mathscr{L}_0$. Notice that u_1 is a function of α, C, μ_0, and $\mathscr{L}_0$ only.

The proofs of the next three lemmas have been omitted because the results can easily be obtained through direct computations.

Lemma 9 *If α, C, μ_0 and $\mathscr{L}_0$ are real numbers such that $J_n(B, G) - \alpha I$ has a unique UL factorization, then u_k has the following partial derivatives with respect to $B_1, ..., B_{k-1}, G_1, ..., G_{k-1}, \alpha$, C, μ_0 and $\mathscr{L}_0$.*

$$\frac{\partial u_k}{\partial B_i} = \begin{cases} 0, & k = 1 \\ -\gamma_{k-1} u_k, & i = k-1, \ k > 1 \\ \gamma_{k-1} u_k \ \dfrac{\partial u_{k-1}}{\partial B_i}, & i < k-1, \ k > 1 \end{cases} \qquad \frac{\partial u_k}{\partial C} = \begin{cases} \dfrac{-u_1}{C + \mu_0}, & k = 1 \\ \gamma_{k-1} u_k \ \dfrac{\partial u_{k-1}}{\partial C}, & k > 1 \end{cases}$$

$$\frac{\partial u_k}{\partial G_i} = \begin{cases} 0, & k = 1, \\ \gamma_{k-1}, & i = k-1, \ k > 1 \\ \gamma_{k-1} u_k \ \dfrac{\partial u_{k-1}}{\partial G_i}, & i < k-1, \ k > 1 \end{cases} \qquad \frac{\partial u_k}{\partial \mathscr{L}_0} = \begin{cases} \dfrac{1}{C + \mu_0}, & k = 1 \\ \gamma_{k-1} u_k \ \dfrac{\partial u_{k-1}}{\partial \mathscr{L}_0}, & k > 1 \end{cases}$$

$$\frac{\partial u_k}{\partial \alpha} = \begin{cases} -\dfrac{u_1}{C + \mu_0} \dfrac{\partial \mu_0}{\partial \alpha}, & k = 1 \\ \gamma_{k-1} u_k \ \left(1 + \dfrac{\partial u_{k-1}}{\partial \alpha}\right), & k > 1 \end{cases} \qquad \frac{\partial u_k}{\partial \mu_0} = \begin{cases} \dfrac{-u_1}{C + \mu_0}, & k = 1 \\ \gamma_{k-1} u_k \ \dfrac{\partial u_{k-1}}{\partial \mu_0}, & k > 1 \end{cases}$$

Here

$$\gamma_{k-1} := \frac{1}{B_{k-1} - u_{k-1} - \alpha}, \qquad k \geq 2. \tag{A-2}$$

From Algorithm 1, we also get

$$b_1 = u_1 + \alpha, \quad b_k = B_{k-1} + u_k - u_{k-1}, \quad k \geq 2 \tag{A-3}$$

and, therefore, for $k \geq 2$, the variable b_k can be seen as a function of $B_1, ..., B_{k-1}$, $G_1, ..., G_{k-1}, \alpha, C, \mu_0, \mathscr{L}_0$. Notice that b_1 is only a function of α, C, μ_0, and $\mathscr{L}_0$.

Lemma 10 *If α, C, μ_0 and $\mathscr{L}_0$ are real numbers such that $J_n(B, G) - \alpha I$ has a unique UL factorization, then the partial derivatives of b_k with respect to $B_1, ..., B_{k-1}, G_1, ..., G_{k-1}$, α, C, μ_0, and $\mathscr{L}_0$ are*

$$\frac{\partial b_k}{\partial B_i} = \begin{cases} 0, & i=1,\ k=1 \\ 1 + \dfrac{\partial u_k}{\partial B_{k-1}}, & i=k-1,\ k>1 \\ \dfrac{\partial u_k}{\partial B_i} - \dfrac{\partial u_{k-1}}{\partial B_i}, & i<k-1,\ k>1 \end{cases} \qquad \frac{\partial b_k}{\partial C} = \begin{cases} \dfrac{\partial u_1}{\partial C}, & k=1 \\ \dfrac{\partial u_k}{\partial C} - \dfrac{\partial u_{k-1}}{\partial C}, & k>1 \end{cases}$$

$$\frac{\partial b_k}{\partial G_i} = \begin{cases} 0, & k=1 \\ \dfrac{\partial u_k}{\partial G_{k-1}}, & i=k-1,\ k>1 \\ \dfrac{\partial u_k}{\partial G_i} - \dfrac{\partial u_{k-1}}{\partial G_i}, & i<k-1,\ k>1 \end{cases} \qquad \frac{\partial b_k}{\partial \mathscr{L}_0} = \begin{cases} \dfrac{\partial u_1}{\partial \mathscr{L}_0}, & k=1 \\ \dfrac{\partial u_k}{\partial \mathscr{L}_0} - \dfrac{\partial u_{k-1}}{\partial \mathscr{L}_0}, & k>1 \end{cases}$$

$$\frac{\partial b_k}{\partial \alpha} = \begin{cases} \dfrac{\partial u_1}{\partial \alpha} + 1, & k=1 \\ \dfrac{\partial u_k}{\partial \alpha} - \dfrac{\partial u_{k-1}}{\partial \alpha}, & k>1 \end{cases} \qquad \frac{\partial b_k}{\partial \mu_0} = \begin{cases} \dfrac{\partial u_1}{\partial \mu_0}, & k=1 \\ \dfrac{\partial u_k}{\partial \mu_0} - \dfrac{\partial u_{k-1}}{\partial \mu_0}, & k>1 \end{cases}$$

It also happens that $g_k = (B_k - u_k - \alpha)u_k$, $k \geq 1$, so g_k is a function of $B_1, ..., B_k, G_1, ..., G_{k-1}, \alpha, C, \mu_0, \mathscr{L}_0$.

Lemma 11 *If α, C, μ_0 and $\mathscr{L}_0$ are real numbers such that $J_n(B,G) - \alpha I$ has a unique UL factorization, then the partial derivatives of g_k with respect to $B_1, ..., B_k, G_1, ..., G_{k-1}$, α and C are*

$$\frac{\partial g_k}{\partial B_i} = \begin{cases} u_k, & i=k \\ \delta_k \dfrac{\partial u_k}{\partial B_i}, & i<k \end{cases} \qquad\qquad \frac{\partial g_k}{\partial C} = \delta_k \frac{\partial u_k}{\partial C}, \quad k \geq 1$$

$$\frac{\partial g_k}{\partial G_i} = \delta_k \frac{\partial u_k}{\partial G_i}, \quad i<k \qquad\qquad \frac{\partial g_k}{\partial \mathscr{L}_0} = \delta_k \frac{\partial u_k}{\partial \mathscr{L}_0}, \quad k \geq 1$$

$$\frac{\partial g_k}{\partial \alpha} = \delta_k \frac{\partial u_k}{\partial \alpha} - u_k, \quad k \geq 1 \qquad\qquad \frac{\partial g_k}{\partial \mu_0} = \delta_k \frac{\partial u_k}{\partial \mu_0}, \quad k \geq 1$$

Here $\delta_k := B_k - 2u_k - \alpha$ for $k \geq 1$.

Next, we define some quantities that will be useful in order to compute the condition number $\kappa(B, G, \alpha, C, \mu_0, \mathscr{L}_0)$ introduced in (12). Let us call

$$\kappa^*(u_k) := \sum_{i=1}^{k-1} \kappa_{B_i}(u_k) + \sum_{i=1}^{k-1} \kappa_{G_i}(u_k) + \kappa_C(u_k) + \kappa_{\mathscr{L}_0}(u_k) + \kappa_{\mu_0}(u_k), \qquad \text{(A-4)}$$

where

$$\kappa_{B_i}(u_k) := \left| \frac{|B_i| + |u_i|}{u_k} \frac{\partial u_k}{\partial B_i} \right|, \quad \kappa_C(u_k) := \left| \frac{C}{u_k} \frac{\partial u_k}{\partial C} \right|, \qquad \text{(A-5)}$$

$$\kappa_{G_i}(u_k) := \left| \frac{G_i}{u_k} \frac{\partial u_k}{\partial G_i} \right|, \quad \kappa_{\mathscr{L}_0}(u_k) := \left| \frac{\mathscr{L}_0}{u_k} \frac{\partial u_k}{\partial \mathscr{L}_0} \right|, \quad \kappa_{\mu_0}(u_k) := \left| \frac{\mu_0}{u_k} \frac{\partial u_k}{\partial \mu_0} \right|. \qquad \text{(A-6)}$$

Note that the subscript of these auxiliary "condition numbers" indicates with respect to which input variable the specific condition number is computed.

The quantities $\kappa^*(u_k)$ can be computed recursively as the following lemma shows:

Lemma 12 *Let α, C, $\mathscr{L}_0$, and μ_0 be real numbers such that $J_n(B,G) - \alpha I$ has a unique UL factorization. Then,*

$$\kappa^*(u_1) = 1 + \frac{|C| + |\mu_0|}{|C + \mu_0|}$$

$$\kappa^*(u_k) = 1 + |\gamma_{k-1} B_{k-1}| + |\gamma_{k-1} u_{k-1}|(1 + \kappa^*(u_{k-1})), \quad k \geq 2,$$

$$\kappa(b_1) = \left|\frac{\alpha}{u_1 + \alpha}\right| \left|1 + \frac{\partial u_1}{\partial \alpha}\right| + \left|\frac{u_1}{u_1 + \alpha}\right| \kappa^*(u_1),$$

$$\kappa(b_k) = |\gamma_{k-1} u_k| + \frac{|\gamma_{k-1} u_k - 1|}{|b_k|} \left[|B_{k-1}| + |u_{k-1}| \left(1 + \kappa^*(u_{k-1})\right)\right] +$$

$$+ \left|\frac{\alpha}{b_k}\right| \left|(\gamma_{k-1} u_k - 1) \frac{\partial u_{k-1}}{\partial \alpha} + \gamma_{k-1} u_k\right|, \quad k \geq 2.$$

where γ_{k-1} is defined in (A-2).

Proof.
If $k = 1$ then

$$\kappa^*(u_1) = \kappa_C(u_1) + \kappa_{\mathscr{L}_0}(u_1) + \kappa_{\mu_0}(u_1) = 1 + \frac{|C| + |\mu_0|}{|C + \mu_0|}.$$

Assume now that $k > 1$. Then, if $i = k - 1$,

$$\kappa_{B_{k-1}}(u_k) = \frac{|B_{k-1}| + |u_{k-1}|}{|u_k|} \left|\frac{u_k}{B_{k-1} - u_{k-1} - \alpha}\right| = \frac{|B_{k-1}| + |u_{k-1}|}{|B_{k-1} - u_{k-1} - \alpha|}.$$

$$\kappa_{G_{k-1}}(u_k) = \left|\frac{G_{k-1}}{u_k}\right| \left|\frac{1}{B_{k-1} - u_{k-1} - \alpha}\right| = 1.$$

Similarly, if $i < k - 1$,

$$\kappa_{B_i}(u_k) = \frac{|B_i| + |u_i|}{|u_k|} \left|\frac{u_k}{B_{k-1} - u_{k-1} - \alpha}\right| \left|\frac{\partial u_{k-1}}{\partial B_i}\right| = \left|\frac{u_{k-1}}{B_{k-1} - u_{k-1} - \alpha}\right| \kappa_{B_i}(u_{k-1}).$$

$$\kappa_{G_i}(u_k) = \left|\frac{G_i}{u_k}\right| \left|\frac{u_k}{B_{k-1} - u_{k-1} - \alpha}\right| \left|\frac{\partial u_{k-1}}{\partial G_i}\right| = \left|\frac{u_{k-1}}{B_{k-1} - u_{k-1} - \alpha}\right| \kappa_{G_i}(u_{k-1}).$$

Finally,

$$\kappa_C(u_k) = \left|\frac{C}{u_k}\right| \left|\frac{u_k}{B_{k-1} - u_{k-1} - \alpha}\right| \left|\frac{\partial u_{k-1}}{\partial C}\right| = \left|\frac{u_{k-1}}{B_{k-1} - u_{k-1} - \alpha}\right| \kappa_C(u_{k-1}).$$

The remaining two condition numbers are computed in a similar way.

These expressions lead us to the recurrence relation for $\kappa^*(u_k)$ in a straightforward way from (A-4).

For $k = 1$,

$$\kappa_C(b_1) = \left|\frac{C}{b_1}\right| \left|\frac{u_1}{C + \mu_0}\right|, \quad \kappa_\alpha(b_1) = \left|\frac{\alpha}{b_1}\right| \left|1 + \frac{\partial u_1}{\partial \alpha}\right|.$$

$$\kappa_{\mathscr{L}_0}(b_1) = \left|\frac{u_1}{b_1}\right|, \quad \kappa_{\mu_0}(b_1) = \left|\frac{u_1}{b_1}\right| \left|\frac{\mu_0}{C + \mu_0}\right|.$$

Assume now that $k > 1$. For $i = k - 1$,

$$\kappa_{B_{k-1}}(b_k) = \frac{|B_{k-1}| + |u_{k-1}|}{|b_k|} \left| \frac{u_k}{B_{k-1} - u_{k-1} - \alpha} - 1 \right| = \frac{|B_{k-1}| + |u_{k-1}|}{|b_k|} |\gamma_k u_k - 1|,$$

$$\kappa_{G_{k-1}}(b_k) = \left| \frac{G_{k-1}}{b_k} \right| \left| \frac{1}{B_{k-1} - u_{k-1} - \alpha} \right| = \left| \frac{u_k}{b_k} \right|.$$

For $i < k - 1$

$$\kappa_{B_i}(b_k) = \frac{|B_i| + |u_i|}{|b_k|} |\gamma_{k-1} u_k - 1| \left| \frac{\partial u_{k-1}}{\partial B_i} \right| = |\gamma_{k-1} u_k - 1| \left| \frac{u_{k-1}}{b_k} \right| \kappa_{B_i}(u_{k-1}),$$

$$\kappa_{G_i}(b_k) = \left| \frac{G_i}{b_k} \right| |\gamma_{k-1} u_k - 1| \left| \frac{\partial u_{k-1}}{\partial G_i} \right| = |\gamma_{k-1} u_k - 1| \left| \frac{u_{k-1}}{b_k} \right| \kappa_{G_i}(u_{k-1}).$$

Finally,

$$\kappa_\alpha(b_k) = \left| \frac{\alpha}{b_k} \right| \left| (\gamma_{k-1} u_k - 1) \frac{\partial u_{k-1}}{\partial \alpha} + \gamma_{k-1} u_k \right|,$$

$$\kappa_C(b_k) = \left| \frac{C}{b_k} \right| \left| (\gamma_{k-1} u_k - 1) \frac{\partial u_{k-1}}{\partial C} \right| = |\gamma_{k-1} u_k - 1| \left| \frac{u_{k-1}}{b_k} \right| \kappa_C(u_{k-1}).$$

The rest of the condition numbers can be obtained in a similar way. The result follows by (13) and (A-4). □

The expression for $\kappa(g_k)$ can be found following a similar procedure.

APPENDIX 2: Proof of Theorem 8

It can be seen from their explicit expressions that both numbers $\kappa^*(u_k)$ and $\kappa_S^*(u_k)$ are larger than one. Moreover they are of the same order of magnitude as the following lemma shows.

Theorem 9

$$\kappa_S^*(u_k) \leq \kappa^*(u_k) \leq 2\kappa_S^*(u_k) \quad \text{for all } k \geq 1.$$

Proof. The first inequality is clear. Notice that the second inequality is true for $k = 1$. In order to prove the second inequality for $k > 1$, note that

$$\kappa^*(u_k) = 1 + |\gamma_{k-1} B_{k-1}| + \sum_{i=1}^{k-2} (2 + |\gamma_i B_i|) \prod_{j=i+1}^{k-1} |\gamma_j u_j| + \prod_{j=1}^{k-1} |\gamma_j u_j| \left(2 + \frac{|C| + |\mu_0|}{|C + \mu_0|} \right),$$

$$\kappa_S^*(u_k) = 1 + |\gamma_{k-1} B_{k-1}| + \sum_{i=1}^{k-2} (1 + |\gamma_i B_i|) \prod_{j=i+1}^{k-1} |\gamma_j u_j| + \prod_{j=1}^{k-1} |\gamma_j u_j| \left(1 + \frac{|C| + |\mu_0|}{|C + \mu_0|} \right),$$

where $\sum_{i=1}^0 \equiv 0$ and $\sum_{i=1}^{-1} \equiv 0$, i.e., for $k = 1$ the summations are not present. The result follows from the previous expressions. □

It is also easy to prove that $\kappa(b_k)$ and $\kappa_S(b_k)$ are of the same order of magnitude for all $k \geq 1$.

Theorem 10 *For* $1 \le k \le n$,

$$\kappa_S(b_k) \le \kappa(b_k) \le 3\kappa_S(b_k).$$

Proof. Again, the first inequality is obvious. In order to prove the second one take into account Theorem 9 and the fact that $1 \le \kappa_S^*(u_k)$ for all k to get

$$\kappa(b_k) \le \left|\frac{u_k}{b_k}\right| + \left|\frac{\gamma_{k-1}u_k - 1}{b_k}\right| [|B_{k-1}| + |u_{k-1}|3\kappa_S^*(u_{k-1})]$$
$$+ \left|\frac{\alpha}{b_k}\right| \left|(\gamma_{k-1}u_k - 1)\frac{\partial u_{k-1}}{\partial\alpha} + \gamma_{k-1}u_k\right|,$$

and the result follows. $\square$

Proving that $\kappa(g_k)$ and $\kappa_S(g_k)$ are of the same magnitude is not always possible. It is not true in general that $\kappa(g_k)$ is upper bounded by a multiple of $\kappa_S(g_k)$. However, the lemma below shows that whenever $\kappa(g_k)$ and $\kappa_S(g_k)$ have different orders of magnitude, then $\kappa(g_k)$ is bounded by $8\kappa_S(b_{k+1})$. The technical Lemma 13 will be needed to prove our claim.

Lemma 13 *Let us assume that* $\frac{3}{4} < \gamma_k u_k < \frac{3}{2}$ *for some* k. *If* $\gamma_k u_k > 4|\gamma_k B_k|$, *then*

- *if* $\gamma_k u_{k+1} > 15/8$ *or* $\gamma_k u_{k+1} < 3/8$, *then*

$$\frac{5}{12} < \left|\frac{u_{k+1} - 1/\gamma_k}{b_{k+1}}\right|.$$

- *if* $3/8 \le \gamma_k u_{k+1} \le 15/8$, *then*
$$\frac{1}{4} < \left|\frac{u_{k+1}}{b_{k+1}}\right|.$$

Proof. Since $\gamma_k u_k > 4|\gamma_k B_k|$,

$$-\frac{3}{8} < \gamma_k B_k < \frac{3}{8}. \tag{A-1}$$

We consider two possible situations: $\gamma_k > 0$ and $\gamma_k < 0$. Let us begin by assuming that $\gamma_k > 0$.

1. If $\gamma_k > 0$, then $u_k > 0$. From (A-1) we get

$$-\frac{3}{8}\frac{1}{\gamma_k} < l_k + u_k + \alpha < \frac{3}{8}\frac{1}{\gamma_k}.$$

Therefore,

$$-\frac{15}{8\gamma_k} - l_k < \alpha < -\frac{3}{8\gamma_k} - l_k. \tag{A-2}$$

Then, from (A-2), and taking into account that $b_{k+1} = u_{k+1} + l_k + \alpha$, we get the following bounds

$$u_{k+1} - \frac{15}{8\gamma_k} < b_{k+1} < u_{k+1} - \frac{3}{8\gamma_k}.$$

Notice that both bounds of b_{k+1} will be positive if $u_{k+1}\gamma_k > 15/8$, and both bounds will be negative if $u_{k+1}\gamma_k < 3/8$.

– Let us assume that $u_{k+1}\gamma_k > 15/8$, then $u_{k+1} - 1/\gamma_k > 0$ and

$$\frac{u_{k+1} - 1/\gamma_k}{u_{k+1} - \frac{3}{8\gamma_k}} < \left| \frac{u_{k+1} - 1/\gamma_k}{b_{k+1}} \right|.$$

Therefore,

$$\frac{7}{12} < \frac{1}{1 + \frac{5}{8}\frac{1}{u_{k+1}\gamma_k - 1}} < \left| \frac{u_{k+1} - 1/\gamma_k}{b_{k+1}} \right|.$$

– Let us assume now that $u_{k+1}\gamma_k < 3/8$. Then, $u_{k+1} - 1/\gamma_k < 0$ and

$$\frac{-u_{k+1} + 1/\gamma_k}{-u_{k+1} + \frac{15}{8\gamma_k}} < \left| \frac{u_{k+1} - 1/\gamma_k}{b_{k+1}} \right|.$$

As a consequence,

$$\frac{5}{12} < \frac{1}{1 + \frac{7}{8}\frac{1}{1 - u_{k+1}\gamma_k}} < \left| \frac{u_{k+1} - 1/\gamma_k}{b_{k+1}} \right|.$$

– Finally, suppose that $\frac{3}{8} \le u_{k+1}\gamma_k \le \frac{15}{8}$. Then, $u_{k+1} > 0$. If $b_{k+1} > 0$, we get

$$\frac{5}{4} < \frac{u_{k+1}\gamma_k}{u_{k+1}\gamma_k - \frac{3}{8}} < \left| \frac{u_{k+1}}{b_{k+1}} \right|.$$

If $b_{k+1} < 0$, then

$$\frac{1}{4} < \frac{u_{k+1}\gamma_k}{-u_{k+1}\gamma_k + \frac{15}{8}} < \left| \frac{u_{k+1}}{b_{k+1}} \right|.$$

2. When $\gamma_k < 0$, a similar proof gives the same bounds.

☐

Now we can prove Theorem 11. Let us remark that Theorem 8 is a trivial consequence of Theorems 10 and 11. Notice that, from the expressions for $\kappa(g_k)$ and $\kappa_S(g_k)$, and taking into account that $\kappa^*(u_k)$ and $\kappa_S^*(u_k)$ are of the same order of magnitude by Theorem 9, it can easily be deduced that $\kappa(g_k)$ and $\kappa_S(g_k)$ have similar orders of magnitude when $u_k\gamma_k$ is not close to one. This is covered in the first two items of Theorem 11. The most difficult situation, i.e., when $u_k\gamma_k$ is close to one, is presented in the last item. Let us recall that $u_k \ne 0$ for all k because $G_{k-1} \ne 0$ for monic Jacobi matrices corresponding to sequences of orthogonal polynomials.

Theorem 11 *For $1 \le k \le n - 1$,*

1 if $u_k\gamma_k < 0$, then

$$\kappa_S(g_k) \le \kappa(g_k) \le 3\kappa_S(g_k).$$

2 if $0 < u_k\gamma_k \le 3/4$ or $u_k\gamma_k \ge 3/2$, then

$$\kappa_S(g_k) \le \kappa(g_k) \le 8\kappa_S(g_k).$$

3 if $\frac{3}{4} < u_k\gamma_k < \frac{3}{2}$ for some k,
  3.1 if $u_k\gamma_k \le 4|B_k\gamma_k|$, then

$$\kappa_S(g_k) \le \kappa(g_k) \le 5\kappa_S(g_k).$$

3.2 if $u_k\gamma_k > 4|B_k\gamma_k|$, then

(a) if $\kappa(g_k) \geq \frac{4}{3}u_k\gamma_k$, then
$$\kappa_S(g_k) \leq \kappa(g_k) \leq 8\kappa_S(g_k).$$

(b) if $\kappa(g_k) < \frac{4}{3}u_k\gamma_k$, then
$$\kappa_S(g_k) \leq \kappa(g_k) \leq 8\kappa_S(b_{k+1}).$$

Proof.

Considering the definitions of $\kappa(g_k)$ and $\kappa_S(g_k)$, it is easy to see that
$$\kappa_S(g_k) \leq \kappa(g_k), \quad \text{for all } k.$$

In the rest of the proof, notice that
$$\gamma_k\delta_k = \frac{B_k - 2u_k - \alpha}{B_k - u_k - \alpha} = 1 - \frac{u_k}{B_k - u_k - \alpha} = 1 - \gamma_k u_k.$$

Denote $a = u_k\gamma_k$. We need to compare the quantities $|a| + 2|1 - a|\kappa_S^*(u_k)$ and $|1 - a|\kappa_S^*(u_k)$. Note also that $\kappa_S^*(u_k) \geq 1$.

1. If $a < 0$ then $|a| + 2|1 - a| = 2 - 3a \leq 3(1 - a)$, and hence
$$|a| + 2|1 - a|\kappa_S^*(u_k) \leq (|a| + 2|1 - a|)\kappa_S^*(u_k) \leq 3|1 - a|\kappa_S^*(u_k),$$
so $\kappa(g_k) \leq 3\kappa_S(g_k)$.

2. If $0 \leq a \leq 3/4$, then $|a| + 2|1 - a| = 2 - a$, so
$$|a| + 2|1 - a|\kappa_S^*(u_k) \leq (|a| + 2|1 - a|)\kappa_S^*(u_k) \leq 8|1 - a|\kappa_S^*(u_k),$$
and therefore $\kappa(g_k) \leq 8\kappa_S(g_k)$.

3. If $a \geq 3/2$ then $|a| + 2|1 - a| = 3a - 2 \leq 5(a - 1)$, so
$$|a| + 2|1 - a|\kappa_S^*(u_k) \leq (|a| + 2|1 - a|)\kappa_S^*(u_k) \leq 5|1 - a|\kappa_S^*(u_k),$$
and $\kappa(g_k) \leq 5\kappa_S(g_k)$.

3.1 If $\frac{3}{4} < \gamma_k u_k < \frac{3}{2}$ and $u_k\gamma_k \leq 4|B_k\gamma_k|$ then, taking into account the expressions for $\kappa(g_k)$ and $\kappa_S(g_k)$, the result follows.

3.2 If $\frac{3}{4} < \gamma_k u_k < \frac{3}{2}$ and $u_k\gamma_k > 4|B_k\gamma_k|$ then, the condition $\kappa(g_k) \geq \frac{4}{3}u_k\gamma_k$ implies
$$8\kappa_S(g_k) \geq 4\left\{|\gamma_k|[|B_k| + |\delta_k|\kappa(u_k)] + |\gamma_k\alpha|\left|1 - \frac{\delta_k}{u_k}\frac{\partial u_k}{\partial\alpha}\right|\right\} \geq 4\kappa(g_k) - 4|u_k\gamma_k| > \kappa(g_k).$$

On the other hand, if $\kappa(g_k) < \frac{4}{3}u_k\gamma_k$ and $u_{k+1}\gamma_k > \frac{15}{8}$ or $u_{k+1}\gamma_k < \frac{3}{8}$, then by Lemma 13
$$\kappa_S(b_{k+1}) \geq \left|\frac{u_{k+1} - 1/\gamma_k}{b_{k+1}}\right||u_k\gamma_k| > \frac{5}{12}|u_k\gamma_k|,$$
which implies
$$\kappa(g_k) < \frac{16}{5}\kappa_S(b_{k+1}).$$

When $\kappa(g_k) < \frac{4}{3}u_k\gamma_k$ and $\frac{3}{8} \leq u_k\gamma_k \leq \frac{15}{8}$, by Lemma 13
$$\kappa_S(b_{k+1}) \geq \frac{1}{4}.$$

Moreover, since $u_k\gamma_k < \frac{3}{2}$, $\kappa(g_k) < \frac{4}{3}u_k\gamma_k \leq 2$, which implies
$$\kappa(g_k) \leq 8\kappa_S(b_{k+1}).$$

$\square$