ON THE COMPUTATION OF GAUSS QUADRATURE RULES FOR MEASURES WITH A MONOMIAL DENOMINATOR

CARL JAGELS* AND LOTHAR REICHEL†

Abstract. Let $d\mu$ be a nonnegative measure with support on the real axis and let $\alpha \in \mathbb{R}$ be outside the convex hull of the support. This paper describes a new approach to determining recursion coefficients for Gauss quadrature rules associated with measures of the form $d\breve{\mu}(x) := d\mu(x)/(x-\alpha)^{2\ell}$. The proposed method is based on determining recursion coefficients for a suitable family of orthonormal Laurent polynomials. Numerical examples show this approach to yield higher accuracy than available methods.

1. Introduction. Let $d\mu$ be a nonnegative measure on the real axis with an infinite number of points of support and such that all required moments

$$\mu_j := \int_{\mathbb{R}} x^j d\mu(x), \qquad j = 0, \pm 1, \pm 2, \dots \ , \tag{1.1}$$

exist. We assume for notational simplicity that $\mu_0 = 1$. Introduce the inner product and associated norm

$$(f,g)_\mu := \int_{\mathbb{R}} f(x)g(x)d\mu(x), \qquad \|f\|_\mu := (f,f)_\mu^{1/2}. \tag{1.2}$$

One can then determine orthonormal bases with respect to (1.2) for the spaces of Laurent polynomials

$$\mathbb{L}_{j,k} := \text{span}\{x^{-j}, x^{-j+1}, \dots, 1, \dots, x^{k-1}, x^k\}, \qquad j,k = 0,1,2,\dots \ .$$

Consider the ordering

$$\{1, x, x^2, \dots, x^i, x^{-1}, x^{i+1}, \dots, x^{2i}, x^{-2}, \dots, x^{-m+1}, \dots, x^{im}\} \tag{1.3}$$

of a basis for the space $\mathbb{L}_{-m+1,im}$ of dimension

$$\tau := m(i+1). \tag{1.4}$$

Orthonormalization of the basis (1.3) by the Gram–Schmidt process yields the orthonormal Laurent polynomials

$$\{\phi_0, \phi_1, \phi_2, \dots, \phi_i, \phi_{-1}, \phi_{i+1}, \dots, \phi_{-2}, \dots, \phi_{-m+1}, \dots, \phi_{im}\}. \tag{1.5}$$

Thus, the first j orthonormal Laurent polynomials form a basis for the space spanned by the first j powers of x ordered according to (1.3) for $1 \le j \le \tau$. It is shown in [13] that orthonormal Laurent polynomials (1.5) satisfy recursion formulas with few terms. They therefore can be computed fairly inexpensively by a Stieltjes-type procedure [13, Algorithm 2.1].

[0]Version March 11, 2015

*Department of Mathematics and Computer Science, Hanover College, Hanover, IN 47243, USA. E-mail: jagels@hanover.edu. Research supported in part by a grant from the Hanover College Faculty Development Committee.

†Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA. E-mail: reichel@math.kent.edu. Research supported in part by NSF grant DMS-1115385.

Let $m > 1$ and introduce, for $0 \leq j \leq i$, the vectors of orthonormal Laurent polynomials

$$\Phi_{\tau-j}(x) := [\phi_0(x), \ldots, \phi_i(x), \phi_{-1}(x), \ldots, \phi_{-m+1}(x), \ldots, \phi_{im-j}(x)]^T. \tag{1.6}$$

For $j = i$, the last entry is $\phi_{-m+1}(x)$. Throughout this paper the superscript ^{T} denotes transposition. The recursion coefficients for the ϕ_j determine the entries of symmetric pentadiagonal matrices $H_{\tau-j} \in \mathbb{R}^{(\tau-j)\times(\tau-j)}$ such that

$$x\Phi_{\tau-j}(x) = H_{\tau-j}\Phi_{\tau-j}(x) + \psi_{\tau-j}(x)\boldsymbol{e}_{\tau-j}, \qquad 0 \leq j < i. \tag{1.7}$$

For $j = i$ and $i > 1$, we have instead

$$x\Phi_{\tau-i}(x) = H_{\tau-i}\Phi_{\tau-i}(x) + \psi_{\tau-i}(x)(\boldsymbol{e}_{\tau-i} + \boldsymbol{e}_{\tau-i-1});$$

see [13] for details. The $\psi_{\tau-j}(x)$ are Laurent polynomials that, for $0 \leq j \leq i$, have numerator degree $im+1-j$ and denominator degree $m-1$. Moreover, for $0 < j \leq i$, $\psi_{\tau-j}(x)$ only differs from ϕ_{im+1-j} by a multiplicative constant. We remark that $\psi_\tau(x)$ does not correspond to a polynomial in the sequence of orthogonal Laurent polynomials, rather it is a linear combination of ϕ_{-m} and ϕ_{im+1}; see [13].

Let $\{x_k\}_{k=1}^{\tau-j}$ and $\{w_k^2\}_{k=1}^{\tau-j}$ denote the eigenvalues and the squares of the first components of normalized eigenvectors, respectively, of the matrix $H_{\tau-j}$. It is shown in [14, Section 5] that the x_k are the zeros of $\psi_{\tau-j}$ and

$$\mathcal{R}_\kappa(f) := \sum_{k=1}^{\kappa} w_k^2 f(x_k), \qquad \kappa = \tau - j, \tag{1.8}$$

is a rational Gauss quadrature rule such that

$$\mathcal{R}_\kappa(f) = \int_{\mathbb{R}} f(x)d\mu(x) \qquad \forall f \in \mathbb{L}_{2m-2,2(mi-j)+1}. \tag{1.9}$$

This rule can be interpreted as a (standard) Gauss quadrature rule with respect to the measure

$$d\breve{\mu}(x) = \breve{c}\frac{d\mu(x)}{x^{2m-2}}, \tag{1.10}$$

where the coefficient $\breve{c} > 0$ is chosen so that $d\breve{\mu}$ has total mass one. This paper is concerned with the computation of (standard) Gauss quadrature rules associated with measures of the form (1.10). The numerical method we propose is based on this relation between standard and rational Gauss quadrature rules. We remark that the determination of Gauss quadrature rules associated with a measure of the form $d\mu(x)/(x-\alpha)^{2\ell}$, with α outside the convex hull of the support of $d\mu$ and ℓ a positive integer, can be transformed by a change of variables to the problem of determining a Gauss quadrature rule with respect to the measure (1.10) with $\ell = m-1$ and the support of $d\mu$ on the positive real axis. We henceforth assume that this transformation has been carried out.

We briefly outline available methods for computing (standard) Gauss quadrature rules for a measure of the form (1.10) with $m > 1$ an integer and the support of $d\mu$ on the positive real axis before outlining our new scheme. A simple approach is to first determine orthonormal polynomials with respect to the inner product and norm

$$(f,g)_{\breve{\mu}} := \int_{\mathbb{R}} f(x)g(x)d\breve{\mu}(x), \qquad \|f\|_{\breve{\mu}} := (f,f)_{\breve{\mu}}^{1/2}. \tag{1.11}$$

The recursion coefficients of the orthonormal polynomials define a symmetric tridiagonal matrix, whose eigenvalues and squares of the first components of associated normalized eigenvectors yield the nodes and weights, respectively, of the desired quadrature rule. The nodes and weights can be computed efficiently by the Golub–Welsch algorithm [10], which is a modification of the standard QR algorithm for the spectral factorization of a symmetric tridiagonal matrix. However, computed results reported by López Lagomasino et al. [17] indicate that the computed nodes and weights can be sensitive to round-off errors introduced during the computations when m in (1.10) is not small. This approach is discussed further in Section 2.

When the recursion coefficients for orthonormal polynomials associated with the measure $d\mu$ are explicitly known or are simple to compute, one may first determine a symmetric tridiagonal matrix associated with a Gauss quadrature rule for this measure. This matrix then can be modified to obtain a symmetric tridiagonal matrix associated with a Gauss quadrature rule for the measure (1.10). Algorithms for modification of the original symmetric tridiagonal matrix are discussed by Gautschi [5, 6], Golub and Meurant [9], and Verlinden [18]. These algorithms also can be applied to more general rational modifications of a given measure $d\mu$ than (1.10). However, these modification methods are sensitive to round-off errors introduced during the computations. Therefore, the positive integer m in (1.10) has to be quite small to achieve accurate results. Indeed, Gautschi [7] recently proposed the use of high-precision arithmetic to reduce the loss of accuracy.

We propose to compute Gauss quadrature rules for measures of the form (1.10) by determining the eigenvalues and the squares of the first components of normalized eigenvectors of a symmetric pentadiagonal matrix determined by the recursion coefficients of suitable orthonormal Laurent polynomials. The computations can be organized similarly as in the Golub–Welsch algorithm. This allows the computation of the nodes and weights for an n-point Gauss rule from the recursion coefficients in only $\mathcal{O}(n^2)$ arithmetic floating point operations (flops). Numerical examples illustrate that this approach is less sensitive to round-off errors introduced during the computations than the other available approaches discussed above.

This paper is organized as follows. Section 2 describes the three approaches outlined above for computing Gauss quadrature rules associated with measures of the form (1.10). We are interested in the application of these Gauss rules to the approximation of functionals of the form

$$F(A) := \boldsymbol{v}^T f(A)\boldsymbol{v}, \tag{1.12}$$

where $A \in \mathbb{R}^{n\times n}$ is a large symmetric positive definite matrix, $\boldsymbol{v} \in \mathbb{R}^n$ is a unit vector, and f is a function defined on the convex hull of the spectrum of A. The need to evaluate expressions of the form (1.12) arises in many applications, including in the solution of linear discrete ill-posed problems, network analysis, and the solution of differential equations; see, e.g., [1, 2, 3, 15]. A nice overview of methods based on Gauss quadrature for approximating (1.12) is provided by Golub and Meurant [9]. The application of the Gauss quadrature rules of interest in this paper to the approximation of expressions (1.12) is discussed in Section 3. A Golub–Welsch-type algorithm for computing the nodes and weights of desired rational Gauss quadrature rules is described in Section 4. A few computed examples are presented in Section 5. The examples consider the computation of approximations of expressions of the form (1.12) by rational Gauss quadrature, as well as the computation of Gauss quadrature rules for a measure of the type (1.10) with $d\mu$ a measure with infinitely many points of support. Concluding remarks can be found in Section 6.

2. Computation of Gauss quadrature rules. We describe in some detail the three approaches to compute Gauss quadrature rules associated with a measure of the form (1.10) outlined at the end of the previous section. Let p and q be polynomials. Then

$$(p.q)_{\breve{\mu}} = \left(\frac{p(x)}{x^{m-1}}, \frac{q(x)}{x^{m-1}} \right)_{\mu}.$$

This relation shows that standard Gauss rules with respect to the inner product (1.11) may be considered rational Gauss rules with respect to the inner product (1.2). Rational Gauss rules were first discussed by Gonchar and López Lagomasino [11, 16] and have more recently received considerable attention by Gautschi and his collaborators; see [6] and references therein.

We first consider the construction of Gauss quadrature rules for the measure (1.10) by modification of the recursion coefficients for the orthonormal polynomials $p_0, p_1, p_2, \ldots$ associated with the inner product (1.2). These polynomials satisfy a recursion relation of the form

$$\begin{aligned} \beta_1 p_1(x) &= (x-\alpha_1)p_0(x), \qquad p_0(x) = 1, \\ \beta_j p_j(x) &= (x-\alpha_j)p_{j-1}(x) - \beta_{j-1}p_{j-2}(x), \qquad j = 2, 3, \ldots . \end{aligned}$$

The recursion coefficients α_j and $\beta_j > 0$ may be explicitly known or can be computed with a Stieltjes procedure; see, e.g., Gautschi [4, 6] for the latter.

Assume that the entries of the symmetric tridiagonal matrix

$$T_\tau = \begin{bmatrix} \alpha_1 & \beta_1 & & & & 0 \\ \beta_1 & \alpha_2 & \beta_2 & & & \\ & \beta_2 & \alpha_3 & & & \\ & & & \ddots & & \\ & & & \ddots & \ddots & \beta_{\tau-1} \\ 0 & & & & \beta_{\tau-1} & \alpha_\tau \end{bmatrix} \tag{2.1}$$

of order τ, as well as the next subdiagonal entry $\beta_{\tau-2m-1}$, are available and compute the moments $\mu_{-1}, \mu_{-2}, \ldots, \mu_{2-2m}$; cf. (1.1). Here and elsewhere in this paper τ is defined by (1.4). Application of Algorithm 2.8 in Gautschi [6, p. 129] $2m-2$ times, first to the matrix (2.1) and then to the matrices determined by the algorithm, yields the symmetric tridiagonal matrix

$$\breve{T}_\tau = \begin{bmatrix} \breve{\alpha}_1 & \breve{\beta}_1 & & & & 0 \\ \breve{\beta}_1 & \breve{\alpha}_2 & \breve{\beta}_2 & & & \\ & \breve{\beta}_2 & \breve{\alpha}_3 & & & \\ & & & \ddots & & \\ & & & \ddots & \ddots & \breve{\beta}_{\tau-1} \\ 0 & & & & \breve{\beta}_{\tau-1} & \breve{\alpha}_\tau \end{bmatrix} \in \mathbb{R}^{\tau\times\tau}. \tag{2.2}$$

Its eigenvalues $\breve{x}_1, \breve{x}_2, \ldots, \breve{x}_\tau$ are the nodes and the squares of the first components of normalized eigenvectors are the weights $\breve{w}_1^2, \breve{w}_2^2, \ldots, \breve{w}_\tau^2$ of the τ-point (standard)

Gauss quadrature rule

$$\breve{\mathcal{G}}_\tau(f) = \sum_{k=1}^{\tau} \breve{w}_k^2 f(\breve{x}_k) \tag{2.3}$$

associated with the measure (1.10). Thus,

$$\breve{\mathcal{G}}_\tau(f) = \int_{\mathbb{R}} f(x) d\breve{\mu}(x) \qquad \forall f \in \mathbb{L}_{0,2\tau-1}.$$

Assume that m is independent of τ. Then the computational effort to determine the matrix (2.2) from (2.1) with Algorithm 2.8 in Gautschi [6, p. 129] requires $\mathcal{O}(\tau)$ flops. The computation of the nodes and weights of the Gauss rule (2.3) by the Golub–Welsch algorithm can be carried out in $\mathcal{O}(\tau^2)$ flops.

An alternative way to determining the entries of the matrix (2.2) is to apply the Stieltjes procedure to compute the recursion coefficients for orthonormal polynomials $\breve{p}_0, \breve{p}_1, \breve{p}_2, \ldots$ associated with the inner product (1.11). These polynomials satisfy a three-term recursion relation

$$\begin{aligned} \breve{\beta}_1 \breve{p}_1(x) &= (x - \breve{\alpha}_1)\breve{p}_0(x), \qquad \breve{p}_0(x) = 1, \\ \breve{\beta}_j \breve{p}_j(x) &= (x - \breve{\alpha}_j)\breve{p}_{j-1}(x) - \breve{\beta}_{j-1}\breve{p}_{j-2}(x), \qquad j = 2, 3, \ldots . \end{aligned}$$

Finally, we can compute the rational Gauss rule (1.8) with $j = 0$ by first determining the entries of the pentadiagonal matrix H_τ, defined by (1.7) with $j = 0$, and then evaluate its eigenvalues and the squares of the first components of normalized eigenvectors. As pointed out in Section 1, the rule (1.8) may be considered a τ-point (standard) Gauss rule associated with the measure (1.10). The entries of H_τ can be determined with a Stieltjes-type procedure with recursion formulas with few terms. The eigenvalues and the squares of the first components of normalized eigenvectors of H_τ can be computed in a variety of ways, for instance by first bringing the matrix to symmetric tridiagonal form using a suitably chosen sequence of Givens reflectors. This requires $\mathcal{O}(\tau^2)$ flops and is described in Section 4. The Golub–Welsch algorithm can then be applied to the symmetric tridiagonal matrix obtained to determine the nodes and weights of the quadrature rule (1.8) (with $j = 0$). The performance of the numerical methods described is illustrated in Section 5.

We conclude this section by pointing out a relation between the matrices (1.7) and (2.2) for $0 \leq j \leq i$.

PROPOSITION 2.1. *The matrices $H_{\tau-j}$ and $\breve{T}_{\tau-j}$ have the same spectra for each $0 \leq j \leq i$. This is not true when $j > i$.*

Proof. The result follows from the way the denominator is increased in (1.3) and, therefore, in (1.5). The denominator power does not change when $0 \leq j \leq i$, and agrees with the denominator in (1.10). The matrices $H_{\tau-j}$ and $\breve{T}_{\tau-j}$ therefore determine the same quadrature rules for $0 \leq j \leq i$. The denominators for the rational Gauss quadrature rule associated with $H_{\tau-j}$ for $j > i$ differs from that for the rational Gauss rule associated with $T_{\tau-i}$. Therefore, the associated quadrature rules are not the same in this case. □

COROLLARY 2.2. *The eigenvalues of H_τ are the nodes of the Gauss rule (2.3). Let $w_1^2, w_2^2, \ldots, w_\tau^2$ be the squared first components of the corresponding eigenvectors of H_τ. Then the weights in (2.3) are given by $\breve{w}_k^2 = c w_k^2 / x_k^{2\ell-2}$ for $1 \leq k \leq \tau$, where the scaling factor c is chosen so that the weights $\breve{w}_k^2$ sum to one.*

Proof. The fact that the eigenvalues of H_τ are the nodes of the rule (2.3) follows from Proposition 2.1. It is shown in [14] that the w_j^2 are weights of the rational Gauss rule (1.8). The relation between these weights and the weights of (2.3) follows from the relation between Gauss and rational Gauss rules; see, e.g., Gautschi [6, p. 181] for details. □

3. Approximation of matrix functionals. Let the matrix $A \in \mathbb{R}^{n\times n}$ be symmetric positive definite with spectral factorization

$$A = Q\Lambda Q^T, \qquad \Lambda = \mathrm{diag}[\lambda_1, \lambda_2, \ldots, \lambda_n], \quad 0 < \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n, \tag{3.1}$$

where the λ_j are eigenvalues and the columns of $Q = [\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_n] \in \mathbb{R}^{n\times n}$ are orthonormal eigenvectors. Substituting (3.1) into (1.12) yields

$$F(A) = \sum_{j=1}^{n} f(\lambda_j) w_j^2, \qquad w_j := \boldsymbol{v}^T \boldsymbol{q}_j.$$

This shows that (1.12) is a Stieltjes integral, which we write as

$$F(A) = \int f(x) d\mu(x), \tag{3.2}$$

where μ is an increasing piecewise constant distribution function with jumps at the eigenvalues λ_j and $d\mu$ denotes the associated measure. The approach to determine approximations of matrix functionals (1.12) by viewing them as Stieltjes integrals and approximating the latter by Gauss-type quadrature rules is discussed in detail by Golub and Meurant [8, 9]. An application of rational Gauss rules to the approximation of matrix functionals is described in [17]. These references focus on the determination of upper and lower bounds for (3.2) for certain integrands.

We use the measure $d\mu$ in (3.2) to define the inner product

$$(f, g)_\mu := \int_{\mathbb{R}} f(x) g(x) d\mu(x) \tag{3.3}$$

for functions f and g belonging to a suitably restricted set. This inner product also can be written as

$$(f, g)_\mu = (f(A)\boldsymbol{v})^T g(A)\boldsymbol{v}.$$

Introduce the rational Krylov subspace

$$\mathbb{K}^{-m+1,im}(A, \boldsymbol{v}) = \mathrm{span}\{A^{-m+1}\boldsymbol{v}, A^{-m+2}\boldsymbol{v}, \ldots, A^{-1}\boldsymbol{v}, \boldsymbol{v}, A\boldsymbol{v}, \ldots, A^{im}\boldsymbol{v}\},$$

which we assume to be of dimension (1.4). Let

$$\boldsymbol{v}_j = \phi_j(A)\boldsymbol{v}, \qquad j = -m+1, -m+2, \ldots, -1, 0, 1, \ldots im,$$

denote an orthonormal basis. The functions ϕ_j are Laurent polynomials. Thus,

$$\boldsymbol{v}_j^T \boldsymbol{v}_k = (\phi_j, \phi_k)_\mu = \begin{cases} 1, & j = k, \\ 0, & j \neq k. \end{cases}$$

Let $\|\cdot\|$ denote the Euclidean vector norm and define the norm $\|\cdot\|_\mu$ induced by the inner product (3.3). Then

$$\|\boldsymbol{v}_j\| = (\boldsymbol{v}_j^T \boldsymbol{v}_j)^{1/2} = (\phi_j, \phi_j)_\mu^{1/2} = \|\phi_j\|_\mu.$$

Thus, the orthonormality of the basis $\{\boldsymbol{v}_j\}_{j=-m+1}^{im}$ with respect to the standard inner product and Euclidean norm is equivalent to the orthonormality of the Laurent polynomials ϕ_j with respect to the inner product (3.3) and associated norm $\|\cdot\|_\mu$.

We compute the basis $\{\boldsymbol{v}_j\}_{j=-m+1}^{im}$ with the aid of Algorithm 2.1 in [13]. The vectors $\boldsymbol{v}_j$ are determined in the order indicated by (1.5). The algorithm yields the recursion coefficients for the ϕ_j. As outlined in Section 1, the recursion relations for the vectors $\boldsymbol{v}_j$ can be expressed with a pentadiagonal matrix $H_{\tau-j}$, cf. (1.7). We obtain analogously to (1.6) the matrix

$$V_{\tau-j} := [\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_i, \boldsymbol{v}_{-1}, \boldsymbol{v}_{i+1}, \ldots, \boldsymbol{v}_{2i}, \ldots, \boldsymbol{v}_{-m+1}, \ldots, \boldsymbol{v}_{im-j}] \in \mathbb{R}^{n\times(\tau-j)}.$$

Here $0 \le j \le i$. The matrix $H_{\tau-j}$ satisfies

$$H_{\tau-j} = V_{\tau-j}^T A V_{\tau-j} \tag{3.4}$$

and defines the quadrature rule

$$\mathcal{R}_{\tau-j}(f) := \boldsymbol{e}_1^T f(H_{\tau-j})\boldsymbol{e}_1. \tag{3.5}$$

Analogously to (1.9), we have

$$\mathcal{R}_{\tau-j}(f) = \boldsymbol{v}^T f(A)\boldsymbol{v} \qquad \forall f \in \mathbb{L}_{2m-2,2(mi-j)+1}. \tag{3.6}$$

This result, as well as (3.4), are shown in [14]. Substituting the spectral factorization of $H_{\tau-j}$ into the right-hand side of (3.5) shows that $\mathcal{R}_{\tau-j}(f)$ is an $(\tau-j)$-point quadrature rule, and by (3.6) it is a rational Gauss rule analogous to (1.8).

We provide some details of the second approach described in Section 2 for determining orthonormal polynomials with respect to the inner product and associated norm

$$(f,g)_{\breve{\mu}} := (A^{-m+1}f(A)\boldsymbol{v})^T A^{-m+1} g(A)\boldsymbol{v}, \qquad \|f\|_{\breve{\mu}} := \|A^{-m+1}f(A)\boldsymbol{v}\|. \tag{3.7}$$

The inner product is analogous to (1.11). This approach to generating orthonormal polynomials has previously been applied in [17].

The following algorithm is the Lanczos method equipped with the inner product (3.7). We apply it to computing an orthonormal basis $\{\boldsymbol{u}_j\}_{j=0}^{\tau-1}$ for the (standard) Krylov subspace $\mathbb{K}^\tau(A,\boldsymbol{v}) := \mathrm{span}\{\boldsymbol{v}, A\boldsymbol{v}, \ldots, A^{\tau-1}\boldsymbol{v}\}$, which we assume to be of dimension τ. This is the generic situation.

ALGORITHM 3.1 (Orthonormalization process with respect to (3.7)).
Input: *m, i, τ, $\boldsymbol{v}$, functions for evaluating matrix-vector products and solving linear systems of equations with A;*
Output: *orthonormal columns of $\breve{U}_\tau = [\breve{\boldsymbol{u}}_0, \breve{\boldsymbol{u}}_1, \ldots, \breve{\boldsymbol{u}}_{\tau-1}] \in \mathbb{R}^{n\times\tau}$, non-trivial entries of symmetric tridiagonal matrix $\breve{T}_\tau \in \mathbb{R}^{\tau\times\tau}$.*
$\breve{\beta}_0 := \|A^{-m+1}\boldsymbol{v}\|$; $\breve{\boldsymbol{u}}_0 := \boldsymbol{v}/\breve{\beta}_0$;
$\boldsymbol{w} := A\breve{\boldsymbol{u}}_0$;
$\breve{\alpha}_1 := (A^{-m+1}\boldsymbol{w})^T(A^{-m+1}\breve{\boldsymbol{u}}_0)$;
$\boldsymbol{w} := \boldsymbol{w} - \breve{\alpha}_1\breve{\boldsymbol{u}}_0$;
for $k = 1, 2, \ldots, \tau-1$ **do**
  $\breve{\beta}_k := \|A^{-m+1}\boldsymbol{w}\|$; $\breve{\boldsymbol{u}}_k := \boldsymbol{w}/\breve{\beta}_k$;
  $\boldsymbol{w} := A\breve{\boldsymbol{u}}_k$;
  $\breve{\alpha}_{k+1} := (A^{-m+1}\boldsymbol{w})^T(A^{-m+1}\breve{\boldsymbol{u}}_k)$;

$\boldsymbol{w} := \boldsymbol{w} - \breve{\alpha}_{k+1}\breve{\boldsymbol{u}}_k - \breve{\beta}_k\breve{\boldsymbol{u}}_{k-1}$;
end

The matrix $\breve{U}_\tau$ computed by the algorithm satisfies

$$(A^{-m+1}\breve{U}_\tau)^T(A^{-m+1}\breve{U}_\tau) = I,$$

where I denotes the identity. The scalars $\breve{\alpha}_j$ and $\breve{\beta}_j$ of the the algorithm comprise the nontrivial entries of the symmetric tridiagonal matrix $\breve{T}_\tau$. This matrix is analogous to (2.2) and is such that

$$\breve{T}_\tau = (A^{-m+1}\breve{U}_\tau)^T A(A^{-m+1}\breve{U}_\tau).$$

The (standard) Gauss rule determined by $\breve{T}_\tau$ is given by

$$\breve{\mathcal{G}}_\tau(f) := (\breve{T}_\tau^{-m+1}\boldsymbol{e}_1)^T f(\breve{T}_\tau)(\breve{T}_\tau^{-m+1}\boldsymbol{e}_1)$$

and satisfies

$$\breve{\mathcal{G}}_\tau(f) = (A^{-m+1}\boldsymbol{v})^T f(A)(A^{-m+1}\boldsymbol{v}) \qquad \forall f \in \mathbb{L}_{2m-2,2mi+1}.$$

We conclude this section with some remarks concerning the computational aspects of $\breve{\mathcal{G}}_\tau(f)$ and $\mathcal{R}_\tau(f)$.

1. The denominator degree exactly integrated by $\mathcal{R}_\tau(f)$ can be increased incrementally, whereas the denominator for $\breve{\mathcal{G}}_\tau(f)$ is determined *a priori* by Algorithm 3.1. The computed examples in Section 5 illustrate the advantage of increasing the denominator degree incrementally.
2. The entries of $\breve{T}_\tau$ determined by Algorithm 3.1 require the computation of an inner product that involves the quantities $A^{-m+1}\boldsymbol{w}$. The numerical stability of the computations then depends on

   $$||A^{-m+1}|| = \lambda_1^{-m+1}.$$

   This norm is large when A has an eigenvalue close to the origin. In this case, one can expect $\breve{\mathcal{G}}_\tau(f)$ to be severely contaminated by propagated round-off errors introduced during the computations. The reason for this is that we can expect $\|A^{-m+1}\boldsymbol{v}\|$ to be large in this situation. If $|\breve{\mathcal{G}}_\tau(f)|$ is much smaller than $\|A^{-m+1}\boldsymbol{v}\|$, then severe cancellation of significant digits may occur during the evaluation of $\breve{\mathcal{G}}_\tau(f)$. We note that the computation of the entries of H_τ only depends on $A^{-1}\boldsymbol{w}$. Cancellation of significant digits during the evaluation of the quadrature rule therefore is less of an issue. These observations are supported by the computed examples presented in Section 5.
3. The tridiagonal structure of $\breve{T}_\tau$ is on the surface more attractive than the pentadiagonal structure of H_τ. However, Section 4 describes an efficient algorithm to compute an orthogonal matrix G such that

   $$H_\tau = GT_{H_\tau}G,$$

   where T_{H_τ} is tridiagonal and $G\boldsymbol{e}_1 = \boldsymbol{e}_1$. The quadrature rule (3.5) with $j = 0$ reduces to

   $$\mathcal{R}_\tau(f) = \boldsymbol{e}_1^T f(T_{H_\tau})\boldsymbol{e}_1.$$

   The nodes and weights of the quadrature rule (1.8) can then be determined directly from the spectral decomposition of T_{H_τ}. This is not true of $\breve{\mathcal{G}}_\tau(f)$. As indicated by Corollary 2.2, the weights in of the latter rule must be normalized by the constant $\breve{c} = \boldsymbol{v}^T A^{-(2m-2)}\boldsymbol{v}$, which may be quite large in magnitude.

4. A Golub–Welsch-type algorithm for rational Gauss rules. We describe an efficient approach to determine the nodes and weights of rational Gauss quadrature rules determined by the pentadiagonal matrices $H_{\tau-j}$ in (1.7) or (3.4) of recursion coefficients for orthonormal Laurent polynomials. We let $j = 0$ for notational simplicity. The Laurent polynomials are assumed to be ordered according to (1.5). Then the recursion matrix H_τ may have sub-subdiagonal elements in the columns $i+1, 2(i+1), \ldots, (m-1)(i+1)$ and is otherwise tridiagonal. The standard procedure for reducing a symmetric matrix to symmetric tridiagonal form proceeds by applying symmetric Givens reflectors from the right and from the left to H_τ, from top to bottom, to determine a matrix that is orthogonally similar. However, for the matrices H_τ under consideration, this approach results in unnecessarily much fill-in as the process unfolds. It is more advantageous, in the sense of creating less fill-in, to first tridiagonalize the bottom $(i+1)\times(i+1)$ block of H_τ, then retrocede to the bottom $2(i+1)\times 2(i+1)$ block, and to continue in this fashion until the matrix has been tridiagonalized. The following example illustrates this procedure.

Example 4.1. Consider the matrix H_τ for $i = 2$ and $m = 4$. Then $\tau = 12$ and H_{12} is of the form

$$H_{12} = \begin{bmatrix}
* & * & & & & & & & & & & \\
* & * & * & & & & & & & & & \\
 & * & * & * & * & & & & & & & \\
 & & * & * & * & & & & & & & \\
 & & * & * & * & * & & & & & & \\
 & & & & * & * & * & * & & & & \\
 & & & & & * & * & * & & & & \\
 & & & & & * & * & * & * & & & \\
 & & & & & & & * & * & * & * & \\
 & & & & & & & & * & * & * & \\
 & & & & & & & & \circledast & * & * & * \\
 & & & & & & & & & & * & *
\end{bmatrix}.$$

There are non-zero sub-subdiagonal entries in the 3^{rd}, 6^{th} and 9^{th} columns. The Givens reflector G_1 is applied to H_{12} to zero the sub-subdiagonal entry (circled) in the 9^{th} column. This creates a sub-subdiagonal entry in the 10^{th} column, or, more generally, in column $\tau - 2$. The matrix obtained is depicted below.

$$G_1 H_{12} G_1 = \begin{bmatrix}
* & * & & & & & & & & & & \\
* & * & * & & & & & & & & & \\
 & * & * & * & * & & & & & & & \\
 & & * & * & * & & & & & & & \\
 & & * & * & * & * & & & & & & \\
 & & & & * & * & * & * & & & & \\
 & & & & & * & * & * & & & & \\
 & & & & & * & * & * & * & & & \\
 & & & & & & & * & * & * & & \\
 & & & & & & & & * & * & * & * \\
 & & & & & & & & & * & * & * \\
 & & & & & & & & & \circledast & * & *
\end{bmatrix}$$

We now apply the Givens reflector G_2 to zero the sub-subdiagonal entry in column 10 (circled). This yields the matrix below. The lower 4×4 principal submatrix is now tridiagonal and the first pass is complete. It required the application of $i = 2$ Givens

reflectors from the right and from the left.

$$G_2G_1H_{12}G_1G_2 = \begin{bmatrix} * & * & & & & & & & & & \\ * & * & * & & & & & & & & \\ & * & * & * & * & & & & & & \\ & & * & * & * & & & & & & \\ & & * & * & * & * & & & & & \\ & & & & * & * & * & * & & & \\ & & & & & * & * & * & & & \\ & & & & & \boxed{*} & * & * & * & & \\ & & & & & & & * & * & * & \\ & & & & & & & & * & * & * \\ & & & & & & & & & * & * \end{bmatrix}.$$

The tridiagonalization process now is continued with a second pass that begins with the sub-subdiagonal entry in the 6^{th} column (marked with a square). The second pass requires the application on five Givens reflectors from the right and from the left. To tridiagonalize the matrix a third pass is required, and it demands the application of eight Givens reflectors from the right and from the left. □

Each application of a Givens reflector from the left introduces a non-zero sub-subdiagonal entry in the next column. The second pass requires the application of $i + i + 1$ reflectors from the right and from the left. There are $m - 1$ columns with a non-zero sub-subdiagonal entry in H_τ. Therefore the procedure requires $m-1$ passes. Each new pass demands the application of $i+1$ more Givens reflectors from the right and from the left than the previous one. The last pass begins by zeroing the sub-subdiagonal entry in column $i+1$ of the original matrix H_τ. Zeroing the entries as we "chase the bulge" during the last pass requires the application of $i(m-1)+(m-2)$ Givens reflectors from the right and from the left. The procedure thus requires the application of

$$\frac{(m-1)(im+m-2)}{2}$$

reflectors from the right and from the left. The same number of reflectors are required when using the standard procedure of tridiagonalizing H_τ from top to bottom. Our approach has the advantage of preserving the structure of H_τ in the sense that all intermediate transformed matrices consist of 2×2 and 3×3 diagonal blocks.

We now consider in some detail the effect of applying a Givens reflector

$$G = \begin{bmatrix} c & s \\ s & -c \end{bmatrix}$$

to the pentadiagonal matrix $H_\tau = [h_{jk}]$ with a non-zero sub-subdiagonal entry in column k. Assume for the moment that $i > 1$. We would like to zero this sub-subdiagonal entry by applying the Givens reflector G from the right and from the left. Define

$$c := \frac{h_{k+1,k}}{\sqrt{h_{k+1,k}^2 + h_{k+2,k}^2}}, \qquad s := \frac{h_{k+2,k}}{\sqrt{h_{k+1,k}^2 + h_{k+2,k}^2}}.$$

Premultiplying H_τ by G effects rows $k+1$ and $k+2$ of H_τ. This effect is described by

$$\begin{bmatrix} c & s \\ s & -c \end{bmatrix} \begin{bmatrix} h_{k+1,k} & h_{k+1,k+1} & h_{k+1,k+2} & 0 \\ h_{k+2,k} & h_{k+2,k+1} & h_{k+2,k+2} & h_{k+2,k+3} \end{bmatrix}$$

$$= \begin{bmatrix} \tilde{h}_{k+1,k} & \tilde{h}_{k+1,k+1} & \tilde{h}_{k+1,k+2} & \tilde{h}_{k+1,k+3} \\ 0 & 0 & \tilde{h}_{k+2,k+2} & \tilde{h}_{k+2,k+3} \end{bmatrix}.$$

The $(k+2, k+1)$ entry vanishes due to the fact that

$$\det \begin{vmatrix} h_{k+1,k} & h_{k+1,k+1} \\ h_{k+2,k} & h_{k+2,k+1} \end{vmatrix} = 0;$$

see [14]. Postmultiplication of GH_τ by G effects columns $k+1$ and $k+2$. We obtain

$$\begin{bmatrix} h_{k,k+1} & h_{k,k+2} \\ \tilde{h}_{k+1,k+1} & \tilde{h}_{k+1,k+2} \\ 0 & \tilde{h}_{k+2,k+2} \\ 0 & h_{k+3,k+2} \end{bmatrix} \begin{bmatrix} c & s \\ s & -c \end{bmatrix} = \begin{bmatrix} \tilde{h}_{k+1,k} & 0 \\ \tilde{\tilde{h}}_{k+1,k+1} & \tilde{h}_{k+2,k+1} \\ \tilde{h}_{k+2,k+1} & \tilde{\tilde{h}}_{k+2,k+2} \\ \tilde{h}_{k+3,k+1} & \tilde{h}_{k+3,k+2} \end{bmatrix}$$

where symmetry is used. Only the diagonal entries require two iterations to determine. For $k = \tau - 2$, a reflector is premultiplied to the block

$$\begin{bmatrix} h_{k+1,k} & h_{k+1,k+1} & h_{k+1,k+2} \\ h_{k+2,k} & h_{k+2,k+1} & h_{k+2,k+2} \end{bmatrix}$$

with an obvious extension for the postmultiplication of the reflector. These discussions lead to the following algorithm. It overwrites the entries of H_τ.

ALGORITHM 4.1 (Tridiagonalization of H_τ).
Input: *m, i, symmetric pentadiagonal matrix H_τ;*
Output: *non-zero entries of symmetric tridiagonal matrix H_τ.*
$\tau := m(i+1)$;
for $l = 1, 2, \ldots, m-1$ **do**
  $j := \tau - l(i+1)$;
  for $k = j \ldots, \tau - 2$ **do**
    $r := \sqrt{h_{k+1,k}^2 + h_{k+2,k}^2}$; $c := h_{k+1,k}/r$; $s := h_{k+2,k}/r$;
    $h_{k+1,k} := ch_{k+1,k} + sh_{k+2,k}$; $h_{k+2,k} := 0$;
    $h_{k+1,k+1} := ch_{k+1,k+1} + sh_{k+2,k+1}$; $h_{k+1,k+2} := ch_{k+1,k+2} + sh_{k+2,k+2}$;
    $h_{k+2,k+2} := sh_{k+1,k+2} - ch_{k+2,k+2}$; $h_{k+2,k+1} := sh_{k+2,k+2}$;
    if $k \neq \tau - 2$
      $h_{k+3,k+1} := sh_{k+3,k+2}$; $h_{k+3,k+2} := -ch_{k+3,k+2}$;
      $h_{k+1,k+3} := h_{k+3,k+1}$; $h_{k+2,k+3} := h_{k+3,k+2}$;
    end if
    $h_{k+1,k+1} := ch_{k+1,k+1} + sh_{k+2,k+1}$; $h_{k+2,k+2} := -ch_{k+2,k+2}$;
    $h_{k,k+1} := h_{k+1,k}$, $h_{k,k+2} = h_{k+2,k}$; $h_{k+1,k+2} := h_{k+2,k+1}$;
  end
end

The case $i = 1$ differs from $i > 1$ in that every other column of H_τ has five non-zero entries. The non-zero portions of rows $k+1$ and $k+2$ now have the form

$$\begin{bmatrix} h_{k+1,k} & h_{k+1,k+1} & h_{k+1,k+2} & 0 & 0 \\ h_{k+2,k} & h_{k+2,k+1} & h_{k+2,k+2} & h_{k+2,k+3} & h_{k+2,k+4} \end{bmatrix}.$$

It is fairly simple to modify Algorithm 4.1 so that it can be applied to tridiagonalize H_τ when $i = 1$. We omit the details.

5. Computed examples. The computations reported in this section are carried out using MATLAB with about 15 significant decimal digits. Examples 5.1–5.3 illustrate the approximation of integrals (1.12) with $f(x) = \ln(x)$ and symmetric positive definite matrices $A \in \mathbb{R}^{1000\times 1000}$. The vector $\boldsymbol{v} \in \mathbb{R}^{1000}$ in (1.12) has normally distributed random entries with mean zero and variance one. Example 5.4 computes the approximations of $f(x) = 1/\sqrt{x}$ when applied to a matrix used in the previous example that has been augmented in order to illustrate the sensitivity of the different methods to the occurrence of an eigenvalue of small magnitude. Example 5.5 compares the approximation of $(\ln(x), 1)_\mu$ determined by the quadrature rule (1.8), $j = 0$, to that of (2.3) when $d\mu(x)$ is the Hermite measure with its support modified to exclude an interval containing zero. Since f is singular at the origin, we use a rational Gauss rule that integrates a family of rational functions with a pole at the origin exactly, or equivalently, we use an inner product of the form (3.7). We compare the three methods to compute quadrature rules discussed in this paper:

1. Application of τ steps of the symmetric Lanczos process to the matrix A with initial vector $\boldsymbol{v}$ gives a symmetric tridiagonal matrix $T_\tau \in \mathbb{R}^{\tau\times\tau}$. It is made up of the recursion coefficients for orthonormal polynomials associated with the inner product (3.3). This matrix and the negative moments $\mu_{-j} = \boldsymbol{v}^T A^{-j}\boldsymbol{v}$ for $j = 1, 2, \ldots, 2m-2$ are used as input for Algorithm 2.8 in [6] to compute the non-zero entries of the tridiagonal matrix $\breve{T}'_\tau$. It contains the recursion coefficients for orthonormal polynomials associated with the inner product (3.7). The computation of the entries of $\breve{T}'_\tau$ requires $2m-2$ applications of Algorithm 2.8 in [6]. We use the MATLAB implementation `chri4.m` made available by Gautschi.[1]
2. Algorithm 3.1 of Section 3 directly computes the recursion coefficients for orthonormal polynomials associated with the inner product (3.7). Each evaluation of this inner product generally requires the solution of $2m-2$ linear systems of equations with the matrix A. We denote the computed symmetric tridiagonal matrix by $\breve{T}_\tau$. In exact arithmetic this matrix is the same as the matrix $\breve{T}'_\tau$ determined by Method 1. However, due to round-off errors introduced during the computations, the matrices evaluated by Methods 1 and 2 may differ.
3. The non-zero entries of the pentadiagonal matrix H_τ are computed by using Algorithm 2.1 in [13]. This requires the solution of one linear system of equations with the matrix A every $i+1$ steps. The tridiagonalization method of Section 4 gives a symmetric tridiagonal matrix which we refer to as T_{H_τ}. We let $i = 3$ in the computed examples.

The first two methods require in essence the computation of the quantity $\breve{\beta}_0 := \|A^{-m-1}\tilde{\boldsymbol{v}}\|$. This quantity may be large when A has an eigenvalue close to the origin, and this may cause poor accuracy in the computed recursion coefficients. The third method requires fewer solutions of linear systems with the matrix A. This can make the execution of Method 3 faster than the execution of the other methods when A is a large sparse symmetric matrix without exploitable structure.

The tables for the examples below show the absolute difference between the approximations provided by the quadrature rules and the desired value $\boldsymbol{v}^T f(A)\boldsymbol{v}$. All matrix functionals are evaluated by means of the spectral decomposition of the matrix. The table columns labeled m display the value of m in the inner product (3.7); thus, the values $m = 2, 3 \ldots, 6$ refer to the powers $A^{-2}, A^{-4}, \ldots, A^{-10}$. Since $i = 3$, we

[1] `http://www.cs.purdue.edu/archives/2002/wxg/codes`

have to increase τ by four for every increment of m in order for the quadrature rules to be equivalent in exact arithmetic. The designation 'NA' in the tables indicates that the algorithm returned a complex approximation or caused overflow. These difficulties depend on lack of accuracy in the entries of the computed symmetric tridiagonal matrices.

Example 5.1. Let A be the symmetric positive definite tridiagonal Toeplitz matrix $[-1, 2, -1]$ of order 1000 and let $f(x) = \ln(x)$. The minimal eigenvalue of A is about $1 \cdot 10^{-5}$. Table 5.1 displays the computed results. The computed approximations are essentially the same for denominator powers two and four. For larger denominator powers, the different methods do not yield the same results. The small eigenvalue of A is particularly problematic for Method 2 when the denominator power is larger than or equal to four. Method 1 yields lower accuracy for the denominator power of eight, but large errors do not occur until $m = 6$. Method 3 yields the smallest errors, but the method is also hampered by the presence of a small eigenvalue of A. We return to this example below. □

m	τ	Method 1	Method 2	Method 3
2	8	.0797	.0797	.0797
3	12	.0330	.0330	.0330
4	16	.0164	NA	.0164
5	20	.0142	NA	.0047
6	24	NA	NA	.0011

TABLE 5.1
Example 5.1: Errors in approximations obtained with different quadrature rules.

Example 5.2. Let $A = [a_{jk}]_{j,k=1}^{1000} \in \mathbb{R}^{1000\times 1000}$ be the symmetric positive definite Toeplitz matrix with entries $a_{jk} = 1/(1 + |j - k|)$ and let $f(x) = \ln(x)$. The smallest eigenvalue of A is about 0.386. Table 5.2 shows computed results. The quadrature error is seen to converge rapidly to zero for increasing values of m and τ for all methods. Only a small deviation between the approximations begins to occur for the denominator power eight. An imaginary component, though small, introduces itself into the approximation determined by Method 1 for $m = 6$. □

m	τ	Method 1	Method 2	Method 3
2	8	$5 \cdot 10^{-6}$	$5 \cdot 10^{-6}$	$5 \cdot 10^{-6}$
3	12	$3 \cdot 10^{-9}$	$3 \cdot 10^{-9}$	$3 \cdot 10^{-9}$
4	16	$2 \cdot 10^{-12}$	$2 \cdot 10^{-12}$	$2 \cdot 10^{-12}$
5	20	$1 \cdot 10^{-14}$	$3 \cdot 10^{-14}$	$3 \cdot 10^{-15}$
6	24	NA	$4 \cdot 10^{-13}$	$4 \cdot 10^{-15}$

TABLE 5.2
Example 5.2: Errors in approximations obtained with different quadrature rules.

Example 5.3. We return to the Toeplitz matrix $[-1, 2, -1]$ of Example 5.1 and compare the performance of Methods 1 and 3 for fixed values of m set to 2, 4, and 6. The rules determined by Method 1 for the different m-values are labeled by $\breve{T}_\tau^{(2)}$, $\breve{T}_\tau^{(4)}$, and $\breve{T}_\tau^{(6)}$, respectively, in Figure 5.1; the rules computed by Method 3 are denoted by T_{H_τ}. The dimension of the recursion matrices is increased from $\tau = 12$ to $\tau = 60$ in increments of 12. The numerator degree for Method 1 increases with τ, whereas both numerator and denominator degrees increase with τ for Method 3. Figure 5.1

illustrates the convergence for the different methods. Method 3 is seen to perform well also for large values of τ. $\square$

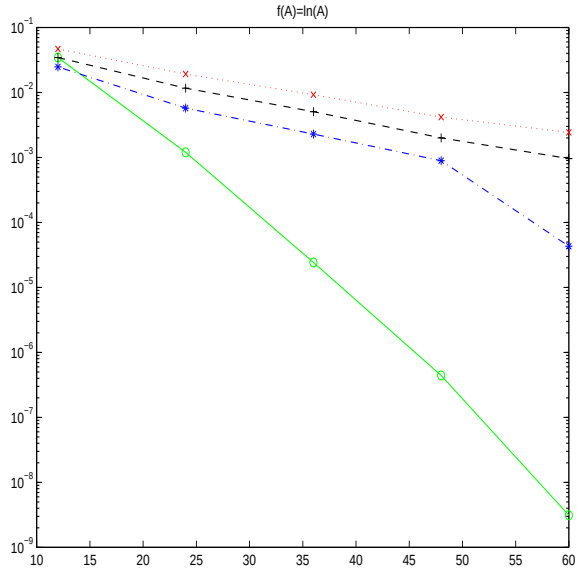

FIG. 5.1. *Example 5.3: Errors in approximations obtained with the quadrature rules* T_{H_τ} *(∘–∘),* $\breve{T}_\tau^{(2)}$ *(×···×),* $\breve{T}_\tau^{(4)}$ *(+ – –+),and* $\breve{T}_\tau^{(6)}$ *(∗–·–∗) for subspaces of dimensions* $\tau = 12, 24, 36, 48, 60$.

Example 5.4. This example is analogous to the previous example, but with the matrix A of Example 5.3 augmented to introduce an eigenvalue 10^{-7}. Specifically, the matrix is of order 1001 and of the form

$$A' := \begin{bmatrix} A & \mathbf{0} \\ \mathbf{0}^T & 10^{-7} \end{bmatrix}.$$

Figure 5.2 shows results for the function $f(x) = 1/\sqrt{x}$. The graphs illustrate the computational sensitivity of Method 1 when the matrix is close to a singular matrix. The results determined by Method 3 are seen to be less sensitive to round-off errors introduced during the computations and fast convergence can be observed.

Example 5.5. For the last example we consider a modified Hermite measure $d\mu(x) := c_a e^{-x^2}$ on $\mathbb{R}\backslash(-a, a)$, where the constant c_a is chosen to normalize the measure. Maple was used to evaluate the inner products defined by (1.2),

$$(f, g)_\mu = c_a \int_{-\infty}^{-a} f(x)g(x)e^{-x^2}\, dx + c_a \int_a^{\infty} f(x)g(x)e^{-x^2}\, dx$$

using 15-digit arithmetic. Analogous to the classical Hermite polynomials, the orthonormal Laurent polynomials (1.5) are even for even indices j and odd otherwise. The measure $d\mu$ is an example of the indefinite case discussed in [12], which requires some modification of the recursion formulas. Note that Gautschi's MATLAB function `chri4.m` cannot be used to evaluate the recursion coefficients for the measure $d\breve{\mu}(x)$ defined by (1.10). Instead, we apply Maple to compute the recursion coefficients for the matrices $\breve{T}_\tau^{(2k)}$, $k = 1, 2, 3$, and H_τ of Methods 2 and 3 for $\tau = 8, 16, 24, 32$. Figure 5.3 shows results for the function $f(x) = \ln(x)$ and $a = 1/4$. The intersection

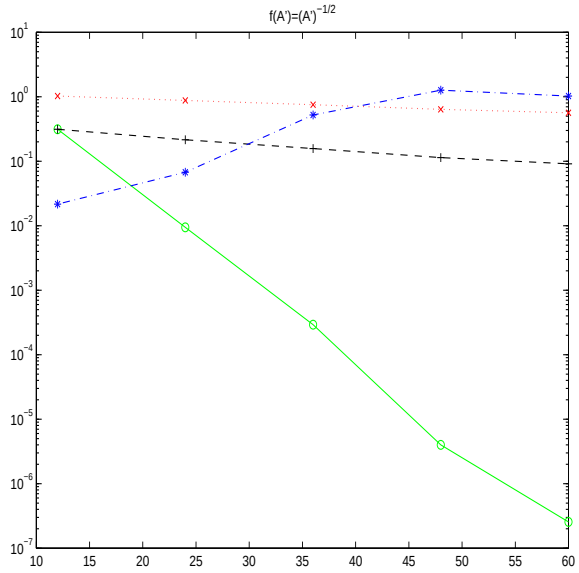

FIG. 5.2. *Example 5.4: Errors in approximations obtained with the quadrature rules* T_{H_τ} *(∘–∘),* $\breve{T}_\tau^{(2)}$ *(×···×),* $\breve{T}_\tau^{(4)}$ *(+ – –+),and* $\breve{T}_\tau^{(6)}$ *(∗–·–∗) for subspaces of dimensions* $\tau = 12, 24, 36, 48, 60$.

of the curve for Method 3 (green with circles) with those for Method 2 illustrates Proposition 2.1; the denominator powers integrated exactly for the two methods are equal at the points of intersection. We remark that the time required to compute the recursion coefficients for $\breve{T}_\tau^{(2k)}$ increases with k and is, already for $k = 2$, significantly longer than the computing time for determining H_τ. This example again illustrates that Method 3 is more stable than Method 2.

6. Conclusion. Three implementations of Gauss quadrature rules for measures with a monomial denominator are compared. A method based on the use of orthogonal Laurent polynomials is found to suffer the least from propagated errors caused by round-off errors introduced during the computations.

Acknowledgement. We would like to thank the referees for comments that lead to improvements of the presentation.

REFERENCES

[1] D. Calvetti, P. C. Hansen, and L. Reichel, L-curve curvature bounds via Lanczos bidiagonalization, Electron. Trans. Numer. Anal., 14 (2002), pp. 20–35.
[2] D. Calvetti, S. Morigi, L. Reichel, and F. Sgallari, Tikhonov regularization and the L-curve for large, discrete ill-posed problems, J. Comput. Appl. Math., 123 (2000), pp. 423–446.
[3] C. Fenu, D. Martin, L. Reichel, and G. Rodriguez, Network analysis via partial spectral factorization and Gauss quadrature, SIAM J. Sci. Comput., 35 (2013), pp. A2046–A2068.
[4] W. Gautschi, On generating orthogonal polynomials, SIAM J. Sci. Stat. Comput., 3 (1982), pp. 289–317.
[5] W. Gautschi, The interplay between classical analysis and (numerical) linear algebra - a tribute to Gene H. Golub, Electron. Trans. Numer. Anal., 13 (2002), pp. 119–147.
[6] W. Gautschi, Orthogonal Polynomials: Approximation and Computation, Oxford University Press, Oxford, 2004.
[7] W. Gautschi, Repeated modifications of orthogonal polynomials by linear divisors, Numer. Algorithms, 63 (2013), pp. 369–383.

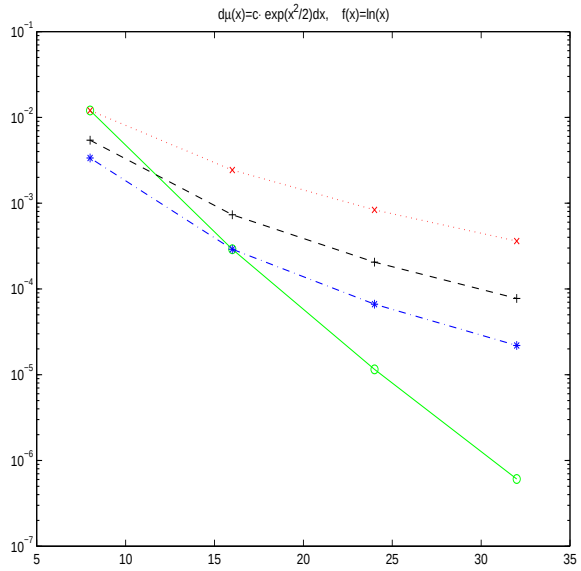

FIG. 5.3. *Example 5.5: Errors in approximations obtained with the quadrature rules* T_{H_τ} *(∘–∘),* $\check{T}_\tau^{(2)}$ *(× · · · ×),* $\check{T}_\tau^{(4)}$ *(+ − −+),and* $\check{T}_\tau^{(6)}$ *(∗–·–∗) for subspaces of dimensions* $\tau = 8, 16, 24, 32$.

[8] G. H. Golub and G. Meurant, Matrices, moments and quadrature, in Numerical Analysis 1993, eds. D. F. Griffiths and G. A. Watson, Longman, Essex, England, 1994, pp. 105–156.

[9] G. H. Golub and G. Meurant, Matrices, Moments and Quadrature with Applications, Princeton University Press, Princeton, 2010.

[10] G. H. Golub and J. Welsch, Calculation of Gauss quadrature rules, Math. Comp., 23 (1969), pp. 221–230.

[11] A. A. Gonchar and G. López Lagomasino, On Markov's theorem for multipoint Padé approximants, Math. USSR Sb., 34 (1978), pp. 449–459.

[12] C. Jagels and L. Reichel, The extended Krylov subspace method and orthogonal Laurent polynomials, Linear Algebra Appl., 431 (2009), pp. 441–458.

[13] C. Jagels and L. Reichel, Recursion relations for the extended Krylov subspace method, Linear Algebra Appl., 434 (2011), pp. 1716–1732.

[14] C. Jagels and L. Reichel, The structure of matrices in rational Gauss quadrature, Math. Comp., 82 (2013), pp. 2035–2060.

[15] J. V. Lambers, Enhancement of Krylov subspace spectral methods by block Lanczos iteration, Electron. Trans. Numer. Anal., 31 (2008), pp. 86–109.

[16] G. López Lagomasino, Conditions for the convergence of multipoint Padé approximants for Stieltjes type functions, Math. USSR Sb., 35 (1979), pp. 363–379.

[17] G. López Lagomasino, L. Reichel, and L. Wunderlich, Matrices, moments, and rational quadrature, Linear Algebra Appl., 429 (2008), pp. 2540–2554.

[18] P. Verlinden, Stable rational modification of a weight, Numer. Algorithms, 22 (1999), pp. 183–192.