A quadrature formula based on Chebyshev rational functions

J. Van Deun and A. Bultheel

Department of Computer Science, K.U.Leuven, Belgium
E-mail: {joris.vandeun–adhemar.bultheel}@cs.kuleuven.be

Abstract

Several generalisations to the classical Gauss quadrature formulas have been made over the last few years. When the integrand has singularities near the interval of integration, formulas based on rational functions give more accurate results than the classical quadrature rules based on polynomials. In this paper we present one such generalisation which uses results from the theory of orthogonal rational functions. Compared to similar existing formulas, it has the advantage of improved stability and smaller quadrature weights.

1 Introduction

When faced with the problem of numerically integrating a given function over the interval $[-1, 1]$, the choice of a suitable quadrature formula will usually depend on at least two different considerations. First there is the question of accuracy: Which quadrature formula gives a more accurate result, for the same number of function evaluations of the integrand? Maximising the number of functions that is integrated exactly, we may hope to reduce the quadrature error for other functions as well. Gauss quadrature formulas then seem the most obvious choice; for a given number of nodes, they have a maximal domain of exactness. Classical Gauss rules like Gauss-Legendre or Gauss-Chebyshev choose nodes as the zeros of orthogonal polynomials (respectively Legendre and Chebyshev) and are often used when the integrand is smooth (polynomial-like), since they integrate exactly in polynomial spaces of largest possible dimension. When there are poles present in the integrand, Gauss formulas that integrate in rational spaces will give better results. Several such generalisations are discussed in [Gau99, VAV93, WL00].

But then we arrive at the second issue, which is equally important, i.e. that of the actual *construction* of the quadrature formula. If the effort involved in computing the nodes and weights is too large, the efficiency gained by superior accuracy may be totally lost. The nodes and weights in the (polynomial) Gauss-Chebyshev formula are explicitly known and have very simple expressions, but when there is no Chebyshev weight function in the integral, this formula behaves very poorly. For an integral without weight function, the Gauss-Legendre rule is more adequate, but its nodes and weights are not known explicitly and have to be computed, either by solving a tridiagonal eigenvalue problem [Gau99] or, more efficiently, using Newton iterations [Swa02]. In this case the nodes are also often precomputed (for several degrees) and hard-coded into the program.

However, for more general weights, or for the formulas based on rational functions, the computations for the Gauss rules are even more involved. First one has to compute the recurrence coefficients for the orthogonal functions. For an n-point rule, this comes down to computing approximately $2n$ inner products (integrals!), as explained in [Gau68, Gau99], or performing a two-dimensional recurrence if modified moments are used [Gau70]. Then these coefficients are gathered in a tridiagonal matrix whose eigenvalues are the quadrature nodes (the weights are obtained from the corresponding eigenvectors).

To compute the integral of a smooth function without a weight function, a good alternative is Fejér's first rule [DR84, p. 84-85]. The computational effort is minimal — both nodes and weights are known explicitly — and, using n nodes, it integrates polynomials of degree $n-1$ exactly. We recall that this formula uses the same nodes as the Gauss-Chebyshev rule (which are the zeros of the Chebyshev polynomials), wh ile the weights are determined to give maximal accuracy for polynomials. Since it does not use the nodes of the Legendre rule (corresponding to an integral with out weight function), its domain of exactness is the space of polynomials of degree $n-1$, instead of $2n-1$ for the true Gauss-Legendre rule.

In the presence of singularities, quadrature formulas which integrate rational functions with fixed poles are more accurate. In this paper we present a new quadrature rule, a rational variant of Fejér's rule, which requires less computational effort than the formulas from [Gau99] and in some cases of interest is more stable and gives more accurate results than the rule presented in [WL00].

In [VDBGV06] we constructed rational generalizations of the well-known classical Gauss-Chebyshev quadrature formula. These rational quadrature rules integrate functions with arbitrary real poles outside the interval $[-1,1]$, with respect to the different Chebyshev weight functions $(1-x)^\alpha(1+x)^\beta$, with α and β belonging to $\{\pm 1/2\}$. The poles are fixed in advance and there can be poles at infinity, in which case the formulas also integrate polynomials of a certain degree. If there are only m *distinct* poles (possibly repeated), then these formulas can be constructed in order $O(mn)$ operations, for arbitrarily high degree n. The main reason for the efficient computation of these rules is that we have explicit representations for the so-called Chebyshev orthogonal rational functions. These were also introduced in the same article. Putting all poles at infinity just gives the Chebyshev polynomials and corresponding Gauss-Chebyshev quadrature formula, which shows that it is indeed a true rational generalisation.

Unlike the formulas from [VDBGV06], the quadrature rule constructed in the present paper is not a Gaussian rule, but instead a rational generalization of Fejér's rule. We use the nodes of the rational Gauss-Chebysh ev formula from [VDBGV06], but there is no weight function in the integral. The quadrature weights are determined to give maximal accuracy in certain r ational function spaces. The reason for studying this rule instead of a rational Gauss-Legendre rule, is that we do not have explicit representations for the Legendre orthogonal rational functions, which makes the computational cost of the corresponding Gauss rule much higher, as explained above.

Before we conclude this section, let us explain the main difference between our rule and the rational Fejér rule from [WL00], which has the same domain of exactness. In [WL00], the authors use the zeros of Chebyshev polynomials as quadrature nodes, and then determine the weights to integrate rational functions exactly. The computational effort is moderate, requiring a two-dimensional recurrence. For poles close to the interval, this gives weights which are very large and of mixed sign. We, however, use the zeros of Chebyshev rational functions, as explained in the next section. For poles close to the interval this makes a big difference, because these zeros tend to cluster near the poles and the quadrature weights turn out to be much smaller and not of mixed sign. Numerical examples in section 5 will illustrate this. We further mention that the authors in [WL00] also discuss a rational Gauss-Legendre rule. The computational effort of this rule (being a Gaussian rule) is very high, as explained above.

The outline of this paper is as follows. In the next section we present the necessary theoretical preliminaries. The main results are given in section 3, but since the computations are rather technical, all the proofs have been moved to the end of the paper, in section 7. The numerical issues of constructing this quadrature formula are discussed in section 4. In section 5, finally, we give several numerical examples and compare our formulas to some of the abovementioned alternatives.

2 Preliminaries

The main concern in this paper is the computation of the integral

$$I(f) = \int_{-1}^{1} f(x)dx \approx I_n(f) = \sum_{k=1}^{n} A_{nk} f(x_{nk}) \tag{2.1}$$

for a function $f(x)$ which has (real) singularities close to the interval $[-1, 1]$. We will approximate $I(f)$ by a quadrature formula $I_n(f)$ which is exact for a certain class of rational functions with prescribed poles. Therefore, let us fix in advance a sequence of real numbers (poles) $A = \{\alpha_1, \alpha_2, \ldots\}$ outside the interval $[-1, 1]$. Some or all of these poles may be equal to infinity. The actual choice of the poles of course depends on the location of the singularities of $f(x)$, as illustrated by the examples in section 5. It is important to keep in mind throughout the entire exposition, that we can always return to the polynomial case if we fix all poles at infinity, as will be clear from the definitions that follow.

With this sequence of poles we now associate spaces of rational functions $\mathcal{L}_n$. First define the polynomial

$$\pi_n(x) = \prod_{k=1}^{n} \left(1 - \frac{x}{\alpha_k}\right)$$

whose zeros are the first n poles, and the basis functions

$$b_0 \equiv 1, \quad b_k(x) = \frac{x^k}{\pi_k(x)}, \quad k = 1, 2, \ldots$$

The rational function space $\mathcal{L}_n$ is the linear span of the first $n+1$ basis functions,

$$\mathcal{L}_n = \operatorname{span}\{b_0, \ldots, b_n\}.$$

Note that, if all poles are at infinity, the basis functions $b_k(x)$ just become the monomials x^k and $\mathcal{L}_n = \mathcal{P}_n$, the space of polynomials of degree n.

The Chebyshev polynomials $T_n(x)$ are well-known and we only briefly recall some of their properties. They satisfy the orthogonality conditions

$$\int_{-1}^{1} T_n(x) x^k \frac{dx}{\sqrt{1-x^2}} = 0, \quad k = 0, 1, 2, \ldots, n-1$$

and are given explicitly by $T_n(x) = \cos(n \arccos x)$. From this equation we readily obtain expressions for the zeros x_{nk} of $T_n(x)$,

$$x_{nk} = \cos\left(\pi \frac{2k-1}{2n}\right), \quad k = 1, 2, \ldots, n.$$

Now consider the quadrature formula $I_n(f)$ whose nodes are exactly these zeros x_{nk} and whose weights are determined such that the approximation (2.1) is exact if $f(x)$ is a polynomial of degree $n-1$ or less. This is Fejér's first quadrature rule and the weights are known explicitly [DR84, p. 84-85].

A logical rational generalisation of Fejér's first rule is obtained in the following way. Consider a rational function $\varphi_n \in \mathcal{L}_n$ which satisfies the orthogonality conditions

$$\int_{-1}^{1} \varphi_n(x) b_k(x) \frac{dx}{\sqrt{1-x^2}} = 0, \quad k = 0, 1, 2, \ldots, n-1.$$

These functions are uniquely defined, up to a normalisation factor which is irrelevant for our discussion, and we shall call them Chebyshev rational functions. They were discussed in great detail in [VDBGV06], where explicit expressions were given (which we repeat below) and a method presented to compute their zeros. Now let x_{nk} denote these zeros of $\varphi_n(x)$ and take them as nodes in a quadrature formula whose weights are determined such that (2.1) is exact for any $f \in \mathcal{L}_{n-1}$. This quadrature formula is the rational generalisation of Fejér's first rule which we will study in the rest of this paper.

Expressions for the weights A_{nk} are given in the following theorem, which is given in a more general setting in [VDB04]. We rephrase it according to our discussion.

Theorem 2.1. *Let $\{x_{nk}\}_{k=1}^{n}$ denote the zeros of the Chebyshev rational function $\varphi_n(x)$. Define the function $\mathrm{k}_n(x)$ as*

$$\mathrm{k}_n(x) = \sum_{k=0}^{n} \nu_k \varphi_k(x)$$

where

$$\nu_k = \int_{-1}^{1} \varphi_k(x)dx$$

and define the numbers λ_{nk} *as*

$$\lambda_{nk} = \left[\sum_{k=0}^{n-1} \varphi_k^2(x_{nk})\right]^{-1}, \quad k = 1, 2, \ldots, n.$$

Now consider the quadrature formula

$$I_n(f) = \sum_{k=1}^{n} A_{nk} f(x_{nk})$$

which approximates the integral

$$I(f) = \int_{-1}^{1} f(x)dx$$

such that this approximation is exact for any function $f \in \mathcal{L}_{n-1}$. *Then the weights* A_{nk} *are given by*

$$A_{nk} = \mathrm{k}_{n-1}(x_{nk})\lambda_{nk}$$

for $k = 1, 2, \ldots, n$.

As explained in [VDB04], the numbers ν_k are in fact *modified moments* and the λ_{nk} are the quadrature weights for the rational Gauss-Chebyshev formula studied in [VDBGV06]. For more information we refer to these articles.

Before we present the explicit expressions for the Chebyshev rational functions, we briefly discuss the Joukowski transformation $x = J(z)$, which is of great importance in the rest of the paper. This transformation is defined as

$$x = J(z) = \frac{1}{2}\left(z + \frac{1}{z}\right)$$

and it maps any point z inside the complex unit disc to a point x in the complex plane outside the interval $[-1, 1]$. The complex unit circle is mapped to the interval $[-1, 1]$. The inverse mapping is denoted by $z = J^{-1}(x)$ and is chosen so that $|z| < 1$ for any x outside the interval $[-1, 1]$. Note that the point $x = \infty$ corresponds to $z = 0$ and that the set $\{x : x \in \mathbb{R} \setminus [-1, 1]\}$ corresponds to the interval $\{z : z \in (-1, 1)\}$.

The following theorem can be found in [VDBGV06].

Theorem 2.2. *The Chebyshev orthogonal rational functions* φ_n *are given by*

$$\varphi_n(x) = \sqrt{\frac{1-\beta_n^2}{2}}\left(\frac{zB_{n-1}(z)}{1-\beta_n z} + \frac{1}{(z-\beta_n)B_{n-1}(z)}\right), \quad n = 1, 2, \ldots$$

where $x = J(z)$ *and* $\alpha_k = J(\beta_k)$, *with* J *the Joukowski transform defined above. The functions* $B_n(z)$ *are defined by,*

$$B_0 \equiv 1, \quad B_n(z) = \frac{z-\beta_1}{1-\beta_1 z} \cdot \frac{z-\beta_2}{1-\beta_2 z} \cdot \cdots \cdot \frac{z-\beta_n}{1-\beta_n z}, \quad n = 1, 2, \ldots$$

Furthermore, the Chebyshev rational functions $\varphi_n(x)$ are normalised such that

$$\int_{-1}^{1} \varphi_n^2(x) \frac{dx}{\sqrt{1-x^2}} = \pi.$$

If all poles are at infinity, then all β_k are equal to zero and we get $\varphi_n(x) = (z^n + z^{-n})/\sqrt{2}$. Apart from a constant factor which is due to our different normalisation, this is exactly the expression for the Chebyshev polynomial $T_n(x)$ as given in [Sze75, p. 296].

It was shown in [VDBGV06] that the Gaussian nodes and weights $\{x_{nk}\}$ and $\{\lambda_{nk}\}$ can be computed very efficiently for arbitrary poles A. In fact, the effort is perfectly comparable to that of computing the classical (polynomial) Gauss-Legendre rule using the method from [Sze75]. The main cost in both algorithms comes from n Newton iterations to determine the nodes, starting from accurate initial values. This should be compared to the effort of constructing the rational Gauss-Legendre quadrature rule from [WL00] or [Gau99]. As we explained in the introduction, this requires performing a two-dimensional recurrence starting from modified moments or evaluating $2n$ integrals, and then solving an n-dimensional tridiagonal eigenvalue problem. Just to give the reader an idea of what we are talking about, computing the rational Gauss-Chebyshev nodes in Matlab 7 on a Pentium III with a CPU speed of 733 MHz for $n = 1600$ in the worst-case scenario where all poles are different, takes little more than 5 seconds. Solving the eigenvalue problem takes more than 4 minutes.

This said, the problem of computing the rational Fejér rule thus reduces to the computation of the modified moments ν_k. The following section describes two different approaches to compute these ν_k. Which approach is preferable, depends on the location of the poles, as illustrated in sections 4 and 5.

3 Computing the modified moments

As we mentioned in the introduction, this section only contains the results. All proofs can be found in section 7. Both methods are based on decomposing φ_k into functions which can be integrated 'exactly'.

3.1 Chebyshev polynomials

This method is based on an explicit representation of the numerator of φ_k in terms of Chebyshev polynomials, as given in the following theorem.

Theorem 3.1. *The Chebyshev rational function $\varphi_k(x)$ can be written as*

$$\varphi_k(x) = \sqrt{2} \cdot \frac{\sqrt{1-\beta_k^2}}{\prod_{i=1}^{k}(1+\beta_i^2)} \cdot \frac{\sum_{i=0}^{\prime k} \left[c_i^{(k)} + c_{-i}^{(k)}\right] T_i(x)}{\pi_k(x)}$$

where $\sum_i' a_i = a_0/2 + a_1 + a_2 + \ldots$ and the coefficients $c_i^{(k)}$ can be computed recursively from

$$c_i^{(k+1)} = c_{i-1}^{(k)} - (\beta_k + \beta_{k+1}) c_i^{(k)} + \beta_k \beta_{k+1} c_{i+1}^{(k)} \tag{3.1}$$

with the initial conditions $c_0^{(1)} = -\beta_1$ *and* $c_1^{(1)} = 1$*, and the convention that* $c_i^{(k)} = 0$ *for* $i \leq -k$ *or* $i > k$ *(from which it also follows that* $c_k^{(k)} = 1$ *for all* k*). The function* $T_i(x)$ *is a Chebyshev polynomial of degree* i*.*

Furthermore, if all poles are equal to $\alpha = J(\beta)$*, then explicit expressions for the coefficients* $c_i^{(k)}$ *are given by*

$$c_i^{(k)} = (-\beta)^{k-i}\binom{2k-1}{k-i}. \tag{3.2}$$

Using this theorem, the computation of the modified moments ν_k comes down to computing the 'moments' $\int_{-1}^{1} T_i(x)/\pi_k(x)dx$. This can be done recursively, as described in [WL00] for the case where all poles are different from each other. If there are repeated poles, some (technical) modifications are needed, but we do not go into details.

3.2 Rational non-orthogonal basis functions

Now we decompose $\varphi_k(x)$ as follows,

$$\varphi_k(x) = \sum_{i=0}^{k} a_i^{(k)} f_i(x) \tag{3.3}$$

where $f_0 \equiv 1$ and

$$f_i(x) = \left(\frac{1-\alpha_i x}{x-\alpha_i}\right)^j, \quad i = 1, 2, \ldots, \quad j = 1, 2, \ldots, \#\alpha_i$$

and $\#\alpha_i$ denotes the multiplicity of the pole α_i. Equation (3.3) can be written in matrix form for $k = 0, \ldots, n-1$ as $\mathbf{\Phi} = \mathbf{AF}$, where $\mathbf{\Phi} = [\varphi_0 \cdots \varphi_{n-1}]^T$, the lower triangular matrix $\mathbf{A}$ contains the coefficients $a_i^{(k)}$ and $\mathbf{F} = [f_0 \cdots f_{n-1}]^T$.

The following theorem is needed for the computation of the matrix $\mathbf{A}$.

Theorem 3.2. *Put by definition* $\mathbf{B} = [b_i^{(k)}] = \mathbf{A}^{-1}$*. The entries* $b_i^{(k)}$ *are given by*

$$b_i^{(k)} = \frac{1}{\pi}\int_{-1}^{1} f_k(x)\varphi_i(x)\frac{dx}{\sqrt{1-x^2}}. \tag{3.4}$$

It is clear that $b_0^{(0)} = 1$*. Explicit expressions for the other coefficients are available in the following two cases.*

1. *If all poles are equal to* $\alpha \neq \infty$*, we have*

$$b_i^{(k)} = \sqrt{\frac{1-\beta^2}{2}}\frac{1}{2^{k-1}}\sum_{j=0}^{\lfloor (k-i)/2 \rfloor}\binom{k}{j}(-\beta)^{k-2j-i}, \quad i \geq 1, \tag{3.5}$$

$$b_0^{(k)} = \frac{1}{2^{k-1}}\sum_{j=0}^{\lfloor k/2 \rfloor}\binom{k}{j}(-\beta)^{k-2j} - \frac{1}{2^k}\binom{k}{k/2}(1 - k \bmod 2),$$

where $\lfloor x \rfloor$ *denotes the largest integer less than or equal to* x*. Note that the last term in the second formula only appears when* k *is even.*

2. If all poles are different from each other, we have

$$\begin{aligned} b_i^{(k)} &= \sqrt{\frac{1-\beta_i^2}{2}}\frac{1-\beta_k^2}{1-\beta_i\beta_k}B_{i-1}(\beta_k), \quad i \geq 1, \\ b_0^{(k)} &= -\beta_k. \end{aligned}$$

In all formulas $\alpha_i = J(\beta_i)$ and $\alpha = J(\beta)$.

For the general case where there are both repeated and different poles, the explicit formulas are too complicated to be of use. We come back to this in section 4.

To be able to compute the modified moments ν_k, we also need the integrals $\int_{-1}^{1} f_k(x)dx$. The next theorem shows how they can be computed.

Theorem 3.3. *For $|\alpha| > 1$, define the integrals*

$$I_m^{(\alpha)} = \int_{-1}^{1} \left(\frac{1-\alpha x}{x-\alpha}\right)^m dx.$$

They satisfy the (backward) recurrence relation

$$I_{m-1}^{(\alpha)} = \left(1-\frac{1}{\alpha^2}\right)\frac{1-(-1)^m}{m} - \frac{2}{\alpha}I_m^{(\alpha)} - \frac{1}{\alpha^2}I_{m+1}^{(\alpha)},$$

and the series expansions

$$\begin{aligned} I_m^{(\alpha)} &= \frac{2}{m+1} + \frac{4m}{\alpha^2}\sum_{j=0}^{\infty}\frac{1}{(2j+m+3)(2j+m+1)}\frac{1}{\alpha^{2j}}, \quad m \text{ even}, \\ I_m^{(\alpha)} &= -\frac{4m}{\alpha}\sum_{j=0}^{\infty}\frac{1}{(2j+m+2)(2j+m)}\frac{1}{\alpha^{2j}}, \quad m \text{ odd}. \end{aligned}$$

As will be explained in section 4, we use the backward recurrence relation for stability reasons. The series expansions are of course needed to start this recurrence. If all poles are different from each other, we only need integrals of type $I_1^{(\alpha)}$, which can be computed explicitly,

$$I_1^{(\alpha)} = (\alpha^2-1)\log\frac{\alpha+1}{\alpha-1} - 2\alpha.$$

The previous two theorems contain all the information we need to solve the lower triangular system

$$\int_{-1}^{1}\mathbf{F}dx = \mathbf{A}^{-1}\int_{-1}^{1}\mathbf{\Phi}dx,$$

thus providing the modified moments ν_k.

4 Numerical issues

While the previous section only presented the theoretical formulas to compute the modified moments, in this section we discuss some numerical considerations to determine which method is more suitable, depending on the location of the poles. Some examples are given as illustration. All computations from this and the following section were done in Matlab 7 on a Pentium III (Coppermine) with a CPU speed of 733 MHz.

4.1 Chebyshev polynomials

The applicability of this method depends on the size of the coefficients $d_i^{(k)} = c_i^{(k)} + c_{-i}^{(k)}$ and the integrals $I_i^{(k)} = \int_{-1}^{1} T_i(x)/\pi_k(x)dx$. We first consider two limit cases.

In the case where all poles are at infinity (corresponding to the classical Fejér rule), we obviously have $\varphi_k(x) = \sqrt{2}\, T_k(x)$, which was already mentioned before. The coefficients $d_i^{(k)}$ satisfy $d_i^{(k)} = 0$ for $i < k$ and $d_k^{(k)} = 1$, and the values of $I_i^{(k)}$ remain bounded. From a numerical point of view, computing the modified moments ν_k in this case is a perfectly well-conditioned problem. It is safe to say that this will also be true for poles far from the interval $[-1, 1]$ (e.g. poles tending to infinity), since in that case the rational function φ_k is still very much 'polynomial-like'.

The other limit case corresponds to all poles equal to $\alpha = 1$. This is of course not allowed in practice, but it is interesting to study the growth of the coefficients $d_i^{(k)}$. It follows from equation (3.1) that in this case

$$d_i^{(k+1)} = d_{i-1}^{(k)} - 2d_i^{(k)} + d_{i+1}^{(k)}.$$

Together with the fact that $d_k^{(k)} = 1$, it can easily be proved by induction that we have the following explicit formula,

$$d_i^{(k)} = (-1)^{k-i}\binom{2k}{k-i},$$

which means that for large k and small i this coefficient becomes very large. The same will be true if all poles are equal to α very close to 1 or -1, as follows from equation (3.2). In general, since the coefficients are continuous functions of the poles, if all poles are close to the interval (and at the same side), these coefficients can become very large. In that case also the integrals $I_i^{(k)}$ can be very large, but the modified moments ν_k are typically of order $O(1)$. It is well-known that summing very large numbers to obtain a small number is an ill-conditioned problem.

The previous considerations are illustrated in table 1, which shows the maximum absolute values of $d_i^{(n)}$, $I_i^{(n)}$ and the relative error on ν_n for different locations of the poles when $n = 15$. Note that the accuracy seems to be higher for poles which are more or less symmetric with respect to the origin. This is due to the computation of the integrals $I_i^{(k)}$ using the algorithm from [WL00].

Table 1: Size of coefficients and auxiliary integrals in the method based on Chebyshev polynomials and relative error of modified moment ν_n when $n = 15$.

α_k	$\max_i \lvert d_i^{(n)} \rvert$	$\max_i \lvert I_i^{(n)} \rvert$	rel. err. ν_n
$2k$	$1.68e+00$	$7.96e-17$	$1.95e-09$
$(-1)^k 2k$	$1.00e+00$	$5.15e-17$	$2.89e-12$
1.001	$7.47e+07$	$7.14e+40$	$2.87e+34$
$1+2^{-k}$	$6.44e+06$	$2.81e+31$	$7.74e+18$
$(-1)^k(1+2^{-k})$	$3.58e+02$	$5.22e+12$	$2.51e-02$

As mentioned by the authors, their algorithm seems to be unstable for poles which are all at the same side of the interval. However, some digits are also lost in the case of $\alpha_k = (-1)^k 2k$. It is not immediately clear what causes this loss. Obviously, when there are many poles close to the interval (at one or both sides), all digits are lost because the integrals $I_i^{(k)}$ become too large.

4.2 Rational non-orthogonal basis functions

If there are both repeated and different poles, it is very difficult to derive explicit formulas for the inner products (3.4). However, the functions f_k and φ_i all belong to $\mathcal{L}_{n-1}$, so we can compute the inner products *exactly* using the n-point rational Gauss-Chebyshev quadrature formula which we already computed (see section 2), since it is exact in the space $\mathcal{L}_n \cdot \mathcal{L}_{n-1}$, as explained in [VDBGV06].

In the case where all poles are equal, we can simplify the computations as follows. If, for fixed k, we only consider the sum in (3.5) and denote it by

$$\tilde{b}_i = \sqrt{\frac{2}{1-\beta^2}}2^{k-1}b_{k-2i}^{(k)}$$

then it is easily checked that we have the recurrence formula

$$\tilde{b}_i = \beta^2\tilde{b}_{i-1} + \binom{k}{i}$$

and $\tilde{b}_0 = 1$. For the other coefficients it holds that $b_{k-2i-1}^{(k)} = -\beta b_{k-2i}^{(k)}$.

Similarly, if all poles are different, there is no need to recompute the $B_i(\beta_k)$ for each i. Obviously,

$$B_{i-1}(\beta_k) = B_{i-2}(\beta_k)\frac{\beta_k - \beta_{i-1}}{1-\beta_k\beta_{i-1}}.$$

Taking into account the above considerations seriously reduces the computational cost of this approach. In fact, this makes the difference between a complexity of order $O(n^2)$ and $O(n^3)$.

As for the integrals $I_m^{(\alpha)}$, we use the backward recurrence relation because it is stable. The homogeneous solution of this recurrence consists of the two

Table 2: Condition number for the matrix $\mathbf{B}$ in the method based on rational non-orthogonal basis functions when $n = 15$.

α_k	$\operatorname{cond} \mathbf{B}$
$2k$	$4.32e+24$
$(-1)^k 2k$	$4.11e+19$
1.001	$1.10e+06$
$1+2^{-k}$	$4.19e+05$
$(-1)^k(1+2^{-k})$	$1.06e+03$

modes $(-1/\alpha)^k$ and $k(-1/\alpha)^k$ which both remain bounded for $k \to \infty$ because $|\alpha| > 1$. For the forward recurrence the homogeneous solution consists of $(-\alpha)^k$ and $k(-\alpha)^k$ which both become unbounded. Summing the series expansion is a well-conditioned problem, since all terms are of the same sign. The convergence rate of course depends on both m and α. As an example, for $m = 30$ and $\alpha = 2$, we needed 23 terms to reach full precision. However, for poles very close to the interval, or for small values of m, it is preferable to use the forward recurrence since it does not need the series expansions (which converge very slowly if $|\alpha| \approx 1$).

Finally, the applicability of this method depends on the condition number of the matrix $\mathbf{B}$, which corresponds roughly to the number of digits lost in solving the lower triangular system. In table 2 we show the 2-norm condition number for different locations of the poles when $n = 15$. Contrary to the method based on Chebyshev polynomials, this approach fails completely when the poles are far from the interval, and seems to work rather well for poles *extremely* close to the interval. This is confirmed by figure 1, which shows the error growth (the relative error divided by the machine precision) for the modified moments ν_k and the same locations of the poles as in table 2. For poles tending to infinity, all digits are lost very quickly. However, when the poles converge to the interval exponentially, only a few digits are lost. Also, the integrals $I_m^{(\alpha)}$ remain bounded as is obvious from their definition. This kind of behaviour is exactly what we want and shows that this method is perfectly complementary to the method based on Chebyshev polynomials. It should be mention ed, however, that the condition number grows rather quickly with increasing n, so this method will only work for moderate values of n, say $n < 50$ or so. But then again, integrals with a high number of poles extremely close to the interval do not seem likely to arise very often in practice.

5 Examples

In this section we apply our quadrature formula based on orthogonal rational functions (ORF) to some test integrals and compare the accuracy and stability with the results obtained using the formulas from [WL00]. In all examples, the 'exact' result was obtained from a multiprecision computation in Maple.

Figure 1: Error growth of modified moment ν_k for different locations of the poles in the method based on rational non-orthogonal basis functions.

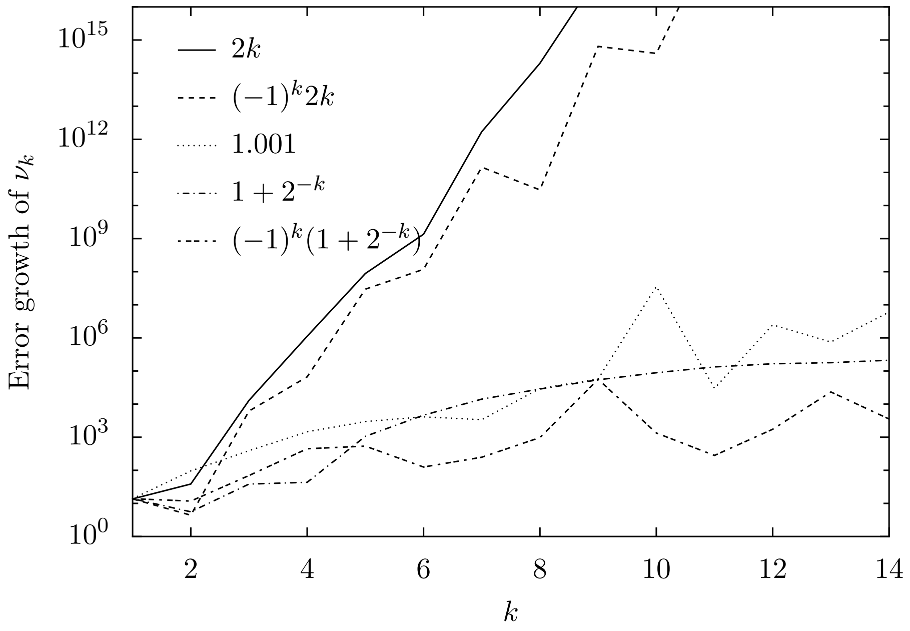

Table 3: Relative error for I_1 with $\omega = 1.1$.

n	ORF	WL
2	$4.15e-01$	$1.42e-02$
4	$1.76e-03$	$8.87e-04$
8	$1.36e-08$	$4.51e-08$
12	$9.41e-14$	$2.36e-13$
16	$< \epsilon_{\text{mach}}$	$4.44e-16$

For the first example we follow [WL00] and [Gau99] and compute

$$I_1 = \int_{-1}^{1} \frac{\pi x/\omega}{\sin(\pi x/\omega)} dx, \quad \omega > 1.$$

The integrand has poles at the integer multiples of ω, so we choose the sequence of poles $A = \{\omega, -\omega, 2\omega, -2\omega, \ldots\}$. Table 3 shows the relative error in the quadrature rule using our approach (ORF) based on Chebyshev polynomials and the method from Weideman and Laurie (WL) for the case $\omega = 1.1$. It seems that with increasing n, our quadrature formula is a little more accurate, but the difference is very small. For $n = 16$ Matlab returned a relative error of 0 for our method, which means less than machine precision, which is approximately $2.22e-16$. In all cases the weights were positive and less than 1.

In table 4 we repeat this experiment but now with $\omega = 1.001$. The results are comparable to the previous case, but it seems that we cannot get more than

Table 4: Relative error for I_1 with $\omega = 1.001$.

n	ORF	WL
2	$2.96e+00$	$1.79e-01$
4	$8.85e-03$	$3.89e-03$
8	$4.78e-08$	$2.07e-07$
12	$1.33e-13$	$1.56e-12$
16	$5.17e-14$	$1.75e-14$

Table 5: Relative error for I_2 with $\alpha = -2.5$.

n	ORF	WL
2	$2.52e-03$	$4.06e-04$
4	$2.26e-06$	$1.86e-06$
8	$6.20e-12$	$2.58e-11$
12	$5.55e-16$	$2.86e-12$
16	$2.22e-16$	$3.11e-11$

approximately 14 correct digits due to roundoff errors. Again all weights were positive and less than 1.

The next example is

$$I_2 = \int_{-1}^{1} \frac{1}{\sqrt{(x+3)(x+2)}} dx.$$

As noted by Van Assche in [VAV93], the integrand is in fact a Stieltjes function and can be well approximated by a rational function with poles on the branch cut $[-3, -2]$. First we take all poles equal to $\alpha = -2.5$. Table 5 gives the results. In our approach, we used the method based on rational non-orthogonal basis functions, but the other method (Chebyshev polynomials) gave similar results. It is worth mentioning that for $n = 16$, even though the condition number of the matrix $\mathbf{B}$ was of order $O(10^5)$, the quadrature sum is accurate up to machine precision. It is not uncommon in numerical integration to obtain an accurate approximation to the integral even with quadrature weights which are not very accurate. Also note that in this case the WL formula seems to stagnate at more or less 12 correct digits. We also noticed several negative weights in this formula, while the weights in our formula were all positive. Next we take all poles different from each other as in the second example of [VAV93]. They are the zeros of successive Chebyshev polynomials $T_{3^m}(x)$, transformed to $[-3, -2]$ and ordered in such a way that they are dense on this interval. For more information we refer to the article. The results are shown in table 6. This time we used the method based on Chebyshev polynomials for the ORF rule. Note that with increasing n, all digits are lost in the WL rule, while this does not occur in our method. This is very remarkable since both methods use the integrals $I_i^{(k)}$. The loss of accuracy must therefore occur not in the computation

Table 6: Relative error for I_2 with α_k distributed over $[-3, -2]$.

n	ORF	WL
2	$4.72e-03$	$2.51e-04$
4	$7.47e-08$	$6.88e-08$
8	$4.67e-14$	$1.20e-10$
12	$3.33e-16$	$2.99e-03$
16	$1.11e-16$	$3.42e+01$

Table 7: Relative error for I_3 with $\omega = 1.1$.

n	ORF	WL	PF
5	$4.56e-02$	$5.52e-01$	$1.86e-01$
10	$1.18e-04$	$4.34e+00$	$3.84e-02$
20	$3.14e-13$	$5.88e+14$	$7.38e-04$
30	$7.33e-15$	$2.04e+28$	$4.66e-06$

of these integrals, but when combining these integrals to obtain the quadrature weights. We mention that some of the weights in the WL rule were negative and of order $O(10^2)$. All weights in the ORF rule were positive and less than 1. Computing this integral with a classical (polynomial) Gauss-Legendre rule gives approximately 9 correct digits for $n = 8$ and 14 for $n = 12$.

For the final example we look at the integral

$$I_3 = \int_{-1}^{1} \sin \frac{1}{\omega - x} dx, \quad \omega > 1$$

whose integrand has an essential singularity in ω. We take all poles equal to $\alpha = \omega$ and use the method based on rational non-orthogonal basis functions. In table 7 we compare our results with the WL rule and the classical (polynomial-based) Fejér rule (PF). This time the WL rule fails completely, not only because of loss of accuracy in the construction of the rule, but also because the weights are very large and of mixed sign. For $n = 20$ the largest weight was of order $O(10^{20})$. Note that our method performs very well (even though for $n = 30$ the condition number of the matrix $\mathbf{B}$ was equal to $1.59e + 11$). All the computed weights were positive and less than 1.

6 Conclusion

For the cases where the WL quadrature rule from [WL00] works well, there is not really a need to use our formulas. Although the results are a little better, this is at the expense of considerably more computations. However, for poles close to the interval, or at the same side of the interval, the WL rule fails, either because of rounding errors or because the weights become very large and of mixed sign. In this case the examples indicate that our rule performs much better, especially when the computations are done using the rational non-orthogonal basis functions. The weights remain small and the presence of poles attract the nodes to the endpoints (we refer to [VDBGV06] for a detailed explanation).

7 Proofs

In all proofs in this section, we use $x = J(z)$ and $\alpha_k = J(\beta_k)$ where J is the Joukowski transform.

Proof of theorem 3.1. From theorem 2.2 and the fact that

$$\pi_n(x) = \frac{\prod_{k=1}^n (1-\beta_k z)(z-\beta_k)}{z^n \prod_{k=1}^n (1+\beta_k^2)}$$

we get that

$$\varphi_n(x) = \sqrt{\frac{1-\beta_n^2}{2}} \frac{z^{-(n-1)}(z-\beta_n)\prod_{k=1}^{n-1}(z-\beta_k)^2 + z^{-n}(1-\beta_n z)\prod_{k=1}^{n-1}(1-\beta_k z)^2}{\pi_n(x)\prod_{k=1}^n (1+\beta_k^2)}.$$

If we put

$$z^{-(n-1)}(z-\beta_n)\prod_{k=1}^{n-1}(z-\beta_k)^2 = \sum_{k=-(n-1)}^{n} c_k^{(n)} z^k$$

then the numerator equals

$$z^{-(n-1)}(z-\beta_n)\prod_{k=1}^{n-1}(z-\beta_k)^2 + z^{-n}(1-\beta_n z)\prod_{k=1}^{n-1}(1-\beta_k z)^2 = \sum_{k=-(n-1)}^{n} c_k^{(n)}(z^k + z^{-k}) = 2\sum_{k=1}^{n-1}(c_k^{(n)} + c_{-k}^{(n)})T_k(x) + 2(T_n(x) + c_0^{(n)}).$$

To find the coefficients $c_k^{(n)}$ write

$$\begin{aligned} z^{-n}(z-\beta_{n+1})\prod_{k=1}^{n}(z-\beta_k)^2 \\ &= z^{-1}(z-\beta_{n+1})(z-\beta_n)z^{-(n-1)}(z-\beta_n)\prod_{k=1}^{n-1}(z-\beta_k)^2 \\ &= z^{-1}(z-\beta_{n+1})(z-\beta_n)\sum_{k=-(n-1)}^{n} c_k^{(n)} z^k \\ &= \sum_{k=-n}^{n+1}[c_{k-1}^{(n)} - (\beta_n+\beta_{n+1})c_k^{(n)} + \beta_n\beta_{n+1}c_{k+1}^{(n)}]z^k \end{aligned}$$

if we take the convention that $c_k^{(n)} = 0$ for $k \leq -n$ or $k > n$. This yields the recurrence relation (3.1).

If all poles are equal, formula (3.2) follows (by induction) from this recurrence relation using the well-known recurrence for binomial numbers. □

Proof of theorem 3.2. Formula (3.4) follows from the orthogonality and the normalisation of the φ_k. We only prove the explicit expressions corresponding to all poles equal. If all poles are different, the reasoning is analogous and the computations are easier.

If all poles are equal to α, the functions f_k have the form

$$f_k(x) = \left(\frac{1-\alpha x}{x-\alpha}\right)^k, \quad k = 1, 2, \ldots$$

From the definition of the Joukowski transformation it follows that

$$f_k(x) = \frac{1}{2^k}\left(\frac{1-\beta z}{z-\beta} + \frac{z-\beta}{1-\beta z}\right)^k$$

so the integral becomes

$$b_i^{(k)} = 2\Re\left\{\frac{1}{2\pi\mathbf{i}}\sqrt{\frac{1-\beta^2}{2}}\frac{1}{2^k}\oint\left(\frac{1-\beta z}{z-\beta} + \frac{z-\beta}{1-\beta z}\right)^k \frac{(z-\beta)^{i-1}}{(1-\beta z)^i}dz\right\}$$

where $\Re$ denotes the real part and the integral is over the complex unit circle. Using the residue theorem then gives

$$b_i^{(k)} = \sqrt{\frac{1-\beta^2}{2}}\frac{1}{2^{k-1}}\operatorname{Res}\left\{\left(\frac{1-\beta z}{z-\beta} + \frac{z-\beta}{1-\beta z}\right)^k \frac{(z-\beta)^{i-1}}{(1-\beta z)^i}, \beta\right\}$$

(since β is real, the residue at β is real as well, so the real part can be omitted from the formula). Using the binomial theorem, the integrand can be written as

$$\left(\frac{1-\beta z}{z-\beta} + \frac{z-\beta}{1-\beta z}\right)^k \frac{(z-\beta)^{i-1}}{(1-\beta z)^i} = \sum_{j=0}^{k}\binom{k}{j}\left(\frac{1-\beta z}{z-\beta}\right)^{k-2j-i}\frac{1}{z-\beta}$$

and the numerator in the summand equals

$$(1-\beta z)^{k-2j-i} = \sum_{m=0}^{k-2j-i}\binom{k-2j-i}{m}(1-\beta^2)^{k-2j-i-m}(-\beta)^m(z-\beta)^m.$$

Since the residue is a linear function of its argument, we finally get

$$b_i^{(k)} = \sqrt{\frac{1-\beta^2}{2}}\frac{1}{2^{k-1}}\sum_{j=0}^{\lfloor (k-i)/2 \rfloor}\binom{k}{j}(-\beta)^{k-2j-i}.$$

The coefficient corresponding to $\varphi_0 \equiv 1$ can be found by similar reasoning, starting from

$$b_0^{(k)} = \frac{1}{2^k} \operatorname{Res}\left\{ \left(\frac{1-\beta z}{z-\beta} + \frac{z-\beta}{1-\beta z} \right)^k \frac{1}{z}, \beta \right\} + \frac{1}{2^k} \operatorname{Res}\left\{ \left(\frac{1-\beta z}{z-\beta} + \frac{z-\beta}{1-\beta z} \right)^k \frac{1}{z}, 0 \right\}.$$

Some computations yield that the first residue equals

$$\operatorname{Res}\left\{ \left(\frac{1-\beta z}{z-\beta} + \frac{z-\beta}{1-\beta z} \right)^k \frac{1}{z}, \beta \right\} = (-1)^k \sum_{j=0}^{\lfloor k/2 \rfloor} \binom{k}{j} \left(\beta^{k-2j} - \frac{1}{\beta^{k-2j}} \right)$$

while the second residue obviousely equals

$$\operatorname{Res}\left\{ \left(\frac{1-\beta z}{z-\beta} + \frac{z-\beta}{1-\beta z} \right)^k \frac{1}{z}, 0 \right\} = (-1)^k \left(\beta + \frac{1}{\beta} \right)^k .$$

Adding both terms and doing some algebra finishes the proof. □

Proof of theorem 3.3. To find the recurrence relation, note that we have

$$\frac{d}{dx} \left(\frac{1-\alpha x}{x-\alpha} \right)^m = \frac{m}{\alpha^2-1} \left[\left(\frac{1-\alpha x}{x-\alpha} \right)^{m+1} + 2\alpha \left(\frac{1-\alpha x}{x-\alpha} \right)^m + \alpha^2 \left(\frac{1-\alpha x}{x-\alpha} \right)^{m-1} \right].$$

Integrating both sides from -1 to 1 gives

$$1 - (-1)^m = \frac{m}{\alpha^2-1} \left(I_{m+1}^{(\alpha)} + 2\alpha I_m^{(\alpha)} + \alpha^2 I_{m-1}^{(\alpha)} \right).$$

Rearranging terms then yields the recurrence relation.

The series expansions are much more difficult to derive. If we define the generating function $I^{(\alpha)}(z)$ as the (formal) series

$$I^{(\alpha)}(z) = \sum_{m=0}^{\infty} I_m^{(\alpha)} z^m,$$

then using the recurrence relation and following the procedure outlined in [GKP89, Chap. 7], we find the explicit expression

$$I^{(\alpha)}(z) = \frac{(\alpha^2-1)z}{(1+\alpha z)^2} \log \frac{(1+z)(\alpha+1)}{(1-z)(\alpha-1)} + \frac{2}{1+\alpha z} .$$

A Taylor series expansion for this function now gives expressions for the integrals $I_m^{(\alpha)}$. We omit the computations, which are cumbersome but straightforward, and only give the result,

$$I_m^{(\alpha)} = (-\alpha)^{m-1} \left((\alpha^2-1) m \log \frac{\alpha+1}{\alpha-1} - 2\alpha \right) + 2(\alpha^2-1) c_m^{(\alpha)}, \quad m = 0, 1, \ldots \tag{7.1}$$

where the constants $c_m^{(\alpha)}$ are given by

$$c_m^{(\alpha)} = \sum_{i=0}^{\lfloor m/2 \rfloor - 1} \frac{m-2i-1}{2i+1}(-\alpha)^{m-2-2i}$$

and an empty sum is equal to zero. These expressions, however, are not very useful (especially for large m and α), because of heavy cancellation when summing both terms in (7.1). In fact, if we compute a Laurent series expansion for $I_m^{(\alpha)}$ as a function of α, all terms with positive powers of α cancel against each other (there is only a finite number of such terms) and we are left with only powers of $1/\alpha$. This Laurent series expansion corresponds to the last formulas in theorem 3.3. Again we omit the computations. □

References

[DR84] P.J. Davis and P. Rabinowitz. *Methods of numerical integration.* Academic Press, 2nd edition, 1984.

[Gau68] W. Gautschi. Construction of Gauss–Christoffel quadrature formulas. *Math. Comp.*, 22:251–270, 1968.

[Gau70] W. Gautschi. On the construction of Gaussian quadrature rules from modified moments. *Math. Comp.*, 24:245–260, 1970.

[Gau99] W. Gautschi. Algorithm 793: GQRAT - Gauss quadrature for rational functions. *ACM Trans. Math. Software*, 25:213–139, 1999.

[GKP89] R.L. Graham, D. Knuth, and O. Patashnic. *Concrete mathematics.* Addison Wesley, 1989.

[Swa02] P.N. Swarztrauber. On computing the points and weights for Gauss–Legendre quadrature. *SIAM J. Sci. Comput.*, 24(2):945–954, 2002.

[Sze75] G. Szegő. *Orthogonal polynomials*, volume 33 of *Amer. Math. Soc. Colloq. Publ.* Amer. Math. Soc., Providence, Rhode Island, 4th edition, 1975. First edition 1939.

[VAV93] W. Van Assche and I. Vanherwegen. Quadrature formulas based on rational interpolation. *Math. Comp.*, 16:765–783, 1993.

[VDB04] J. Van Deun and A. Bultheel. Modified moments and orthogonal rational functions. *Appl. Numer. Anal. Comput. Math.*, 1(3):455–468, 2004.

[VDBGV06] J. Van Deun, A. Bultheel, and P. González-Vera. On computing rational Gauss-Chebyshev quadrature formulas. *Math. Comp.*, 75:307–326, 2006.

[WL00] J.A.C. Weideman and D.P. Laurie. Quadrature rules based on partial fraction expansions. *Numer. Algorithms*, 24:159–178, 2000.