Algorithm XXX: Near best fixed pole rational interpolation with applications in spectral methods

JORIS VAN DEUN
Department of Mathematics and Computer Science, Universiteit Antwerpen
and
KARL DECKERS and ADHEMAR BULTHEEL
Department of Computer Science, Katholieke Universiteit Leuven
and
J.A.C. WEIDEMAN
Deparment of Applied Mathematics, University of Stellenbosch

We present a numerical procedure to compute the nodes and weights in rational Gauss-Chebyshev quadrature formulas. Under certain conditions on the poles, these nodes are near best for rational interpolation with prescribed poles (in the same sense that Chebyshev points are near best for polynomial interpolation). As an illustration, we use these interpolation points to solve a differential equation with an interior boundary layer using a rational spectral method.

The algorithm to compute the interpolation points (and, if required, the quadrature weights) is implemented as a MATLAB program.

Categories and Subject Descriptors: G.1.4 [**Numerical Analysis**]: Quadrature and Numerical Differentiation; G.4 [**Mathematical Software**]:

General Terms: Algorithms

Additional Key Words and Phrases: Rational interpolation, quadrature

The first author is a postdoctoral researcher of FWO-Flanders. Author's address: Joris Van Deun, Department of Mathematics and Computer Science, Universiteit Antwerpen, Middelheimlaan 1, B-2020 Antwerpen, Belgium; email: Joris.VanDeun@ua.ac.be.
The second and third authors' research is partially supported by the Fund for Scientific Research (FWO) project "RAM: Rational modelling: optimal conditioning and stable algorithms", grant #G.0423.05 and the Belgian Programme on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture, and by the K.U.Leuven research project OPTEC: Optimization and engineering. The scientific responsibility rests with the authors. Authors' addresses: Karl Deckers and Adhemar Bultheel, Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, B-3001 Heverlee, Belgium; email: {Karl.Deckers,Adhemar.Bultheel}@cs.kuleuven.be.
The fourth author's research was supported by the National Research Foundation in South Africa under grant FA2005032300018. Author's address: J.A.C. Weideman, Department of Applied Mathematics, University of Stellenbosch, Stellenbosch 7600, South Africa; email: weideman@dip.sun.ac.za.

1. INTRODUCTION

Rational approximation remains an active field of research. Over the years, several numerical integration routines based on rational functions have been proposed in the literature, see e.g. [Van Assche and Vanherwegen 1993; Gautschi 2000; Van Deun and Bultheel 2003] for general theoretical considerations and [Weideman and Laurie 2000; Van Deun et al. 2006; Van Deun and Bultheel 2006; Deckers et al. 2006; 2007] for some explicit cases.

Recently, focus seems to have shifted a little towards the solution of differential equations using rational spectral methods, an idea which looks very promising for functions that have singularities close to the interval. Some different strategies are discussed in [Weideman 1999; Berrut and Baltensperger 2001; Berrut and Mittelmann 2001; Baltensperger et al. 2003; Berrut and Mittelmann 2005; Tee and Trefethen 2006].

The rational interpolant that forms the basis for both the quadrature formulas and the spectral methods can be represented in several ways. A popular approach that has gained much interest uses barycentric formulas. Methods with both fixed and free poles have been studied, see e.g. [Baltensperger et al. 1999; Schneider and Werner 1991; Berrut and Mittelmann 1997; Berrut 1997; Berrut and Mittelmann 2004; Tee and Trefethen 2006; Polezzi and Sri Ranga 2007].

In the present article, however, we take a similar approach as in [Weideman 1999], where the poles are fixed and the interpolation points (the nodes in the quadrature formulas) are somehow determined by these poles. Specifically, for arbitrary complex poles outside $[-1, 1]$, the interpolation points are the zeros of a Chebyshev (quasi-orthogonal) rational function and they are the nodes in rational Gauss-Chebyshev quadrature formulas, as described in [Deckers et al. 2007]. Furthermore, for the case of real or complex conjugate poles (which means the denominator of the rational function is real), the corresponding rational function is of *minimal* Chebyshev norm (among all rational functions with the same poles and the same degree in the numerator). This extremal property of the Chebyshev rational functions was made explicit in [Van Deun 2007], which discusses an alternative method to compute the interpolation points. These points are near best for rational interpolation with prescribed poles in the same sense that the zeros of the Chebyshev polynomial of the first kind are near best for polynomial interpolation; the equi-oscillation property (a consequence of Chebyshev extremality) gives rise to a very uniform error. This is discussed in more detail in the case study of the next section.

2. APPLICATION IN SPECTRAL METHODS

Before we start the theoretical exposition about near best interpolation points, we first present a practical application in spectral methods as an extra motivation. Relevant definitions (specifically for the functions $F(\theta)$, $\mathcal{T}_n(x)$ and the points x_k) are given in the next section. Instead of giving a complete theoretical exposition about the use of rational functions in spectral collocation methods, we prefer to illustrate this by means of an example. The necessary background can be found in the references mentioned in the introduction. More general information about spectral methods is given in [Gottlieb and Orszag 1977; Fornberg 1996; Trefethen

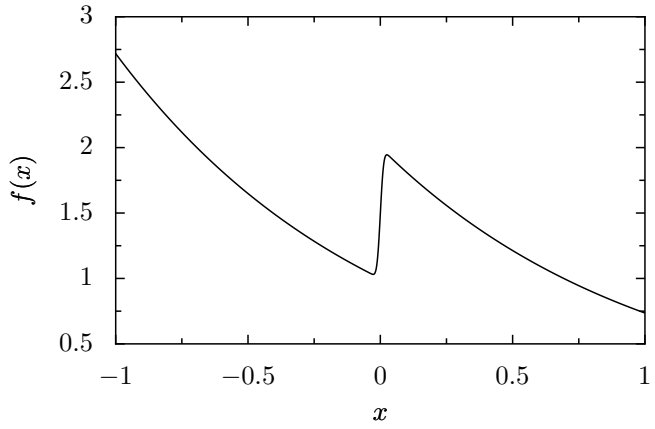

Fig. 1. The solution of (1) for $\varepsilon = 0.0001$.

2000].

We take an example from [Bush 1992, p. 161]. Assume we are given the differential equation

$$\varepsilon \frac{d^2 f}{dx^2} + x\frac{df}{dx} + xf = 0, \quad -1 < x < 1 \tag{1}$$

with boundary conditions $f(-1) = e$ and $f(1) = 2e^{-1}$. It can be verified that the exact solution to this equation is given by

$$f(x) = e^{x - \frac{x^2}{2\varepsilon}} \left[c_1 (x - 2\varepsilon) M\left(1 - \frac{\varepsilon}{2}, \frac{3}{2}; \frac{(x-2\varepsilon)^2}{2\varepsilon}\right) + c_2 M\left(\frac{1-\varepsilon}{2}, \frac{1}{2}; \frac{(x-2\varepsilon)^2}{2\varepsilon}\right)\right],$$

where $M(a, b; x)$ is Kummer's confluent hypergeometric function [Abramowitz and Stegun 1964, p. 504] and the constants c_1 and c_2 are easily solved from the boundary conditions. We mention that this explicit formula for $f(x)$ is not given in [Bush 1992]. When ε is very small, there is an interior boundary layer close to $x = 0$ (for $\varepsilon = 0$ the equation changes from second to first order). Figure 1 shows the solution for $\varepsilon = 0.0001$.

The classical spectral method is based on polynomial interpolation in Chebyshev points. However, the presence of the interior boundary layer makes a numerical solution using this standard method inadvisable, since a polynomial of very high degree is needed to cope with the abrupt transition. A rational spectral method with poles close to the boundary layer is expected to work much better. The problem is of course: how do we choose the poles? The approach we take is the same as in Section 2.2 of [Weideman 1999]. Without having to solve equation (1), boundary layer analysis [Bush 1992, p. 163] shows that an inner solution, valid for x in the boundary layer and $\varepsilon \to 0$, is given by

$$f_{\text{in}}(x) = 0.5\,\text{erf}(x/\sqrt{2\varepsilon}) + 1.5, \quad x = O(\sqrt{\varepsilon}).$$

It is obviously the error function that causes the abrupt transition in the boundary layer. We can model this behaviour with a rational function if we construct a Padé

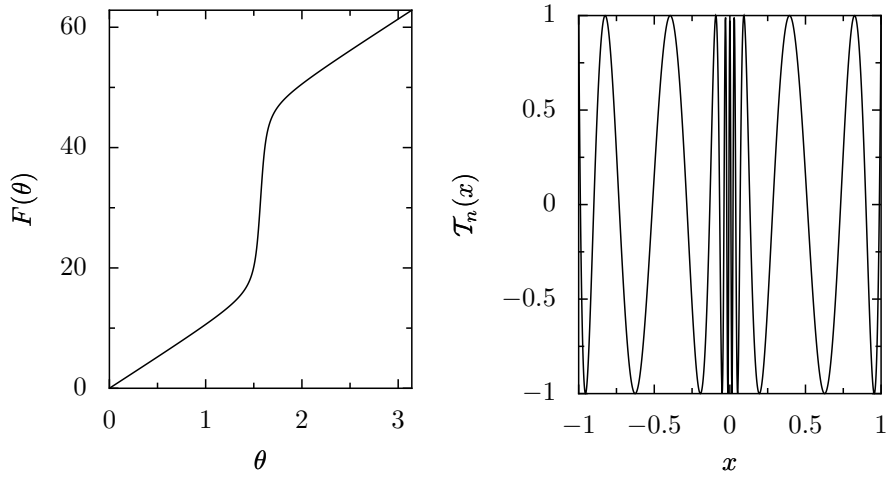

Fig. 2. The functions $F(\theta)$ and $\mathcal{T}_n(x)$ (scaled) for the poles (2) and (3) with $n = 20$.

approximation to the error function and compute its poles. A series expansion for $\mathrm{erf}(x)$ is explicitly known [Abramowitz and Stegun 1964, p. 297] and given by

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k+1}}{k!\,(2k+1)} \ .$$

From this expansion the coefficients of the denominator polynomial of an (m, m) Padé approximant are readily obtained by solving a Toeplitz system, see e.g. [Cuyt and Wuytack 1987, p. 63]. The poles we need are then of course just the zeros of this polynomial, scaled by $\sqrt{2\varepsilon}$. For $m = 10$ and $\varepsilon = 0.0001$ this gives the set of poles

$$\alpha_{1,2} \approx \pm 0.0403\mathbf{i},\ \alpha_{3,4,5,6} \approx \pm 0.0094 \pm 0.0398\mathbf{i},\ \alpha_{7,8,9,10} \approx \pm 0.0200 \pm 0.0384\mathbf{i}. \quad (2)$$

If we want n interpolation points, where $n > m = 10$, then we need to add the additional poles

$$\alpha_{11} = \alpha_{12} = \ldots = \alpha_n = \infty. \quad (3)$$

The left side of Figure 2 shows the function $F(\theta)$ for these poles when $n = 20$. Note the steep slope near $\theta = \pi/2$, which corresponds to $x = 0$.

Next, we compare three spectral methods to solve the differential equation (1). The number of interpolation points for each method is $n = 20$.

The first one (CHEB) is the classical (polynomial) method, where the interpolation points are the extrema of the Chebyshev polynomial of the first kind and degree n (they include the endpoints $x = -1$ and $x = 1$).

The second one (NCOP) is the rational method proposed in [Weideman 1999], where the poles are given by (2) and (3). The interpolation points are the zeros of a nonclassical orthogonal polynomial with respect to a rational weight function with these poles. The points $x = -1$ and $x = 1$ have to be included as additional nodes to facilitate the incorporation of boundary conditions.

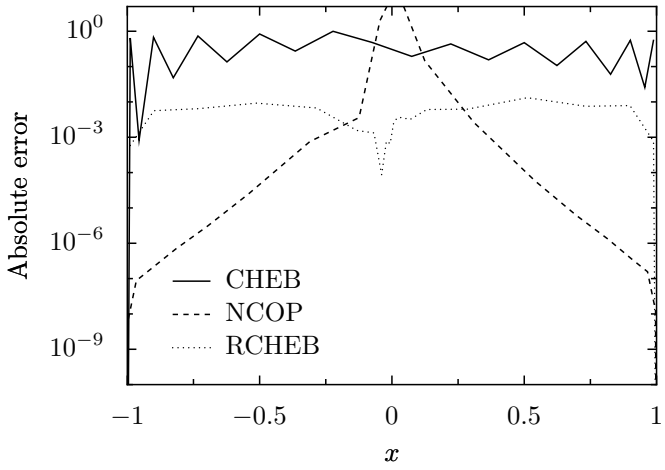

Fig. 3. Pointwise errors in the numerical solution of (1). 'CHEB' refers to the classical (polynomial) case, 'NCOP' is the rational method based on nonclassical orthogonal polynomials and 'RCHEB' is the new method proposed here. In this example, $n = 20$ and $m = 10$.

The third method (RCHEB), finally, is the new one we propose in this paper. It is very similar to the second, but now the interpolation points are our x_k, together with the endpoints $x = -1$ and $x = 1$. The function $\mathcal{T}_n(x)$ discussed in Theorem 3.2 is shown on the right of Figure 2, scaled such that its range is $[-1, 1]$. Note the equi-oscillatory behaviour and the zeros close to the middle of the interval (attracted by the poles). This is indeed the rational analogue of a Chebyshev polynomial. It is expected that the interpolation error will be rather uniform on the interval $[-1, 1]$. This statement can be made more exact using the formula for the interpolation error, but we will not do this here.

The pointwise errors in the numerical solution of (1) for each of the three methods is shown in Figure 3. Note that the polynomial method gives a uniform error but is not accurate at all. The second method gives very good results for points far from the boundary layer, but fails to approximate the function in the boundary layer. Our method is more accurate than the polynomial one and at the same time gives a rather uniform error, as was expected.

Our method becomes especially interesting if we increase the number of poles, as is shown in Figure 4. Here we take $n = 50$ and compare the solution for $m = 10$ to the one for $m = 24$. The difference between the two rational methods is striking (of course the results for the polynomial method are the same for both values of m since it does not use any information about the poles). It was pointed out in [Weideman 1999] by the author that the accuracy of his method improves with an increasing number of poles, but that the condition number of the differentiation matrices also grows, eventually leading to a complete loss of signifant digits. It appears that this is what occurred in the right side of Figure 4. Our method, on the contrary, does not suffer from this instability; the condition numbers of the matrices involved do not grow with increasing number of poles. We think this is another consequence of the equi-oscillation property. Extremely large values simply do not occur.

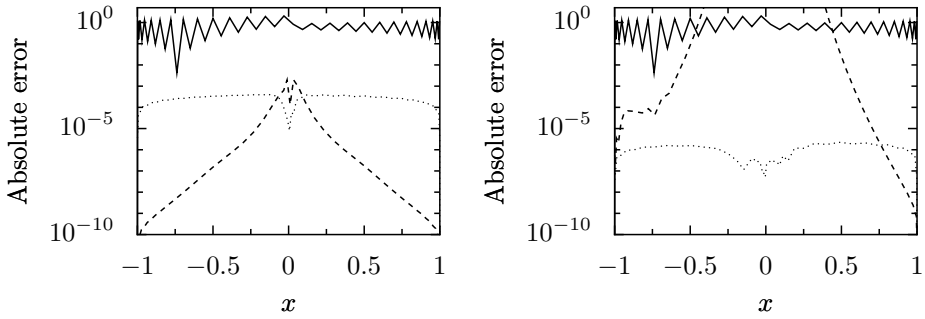

Fig. 4. Same as Figure 3, but with $n = 50$ and $m = 10$ (left) or $m = 24$ (right).

Of course there is much more to rational spectral methods than the example we have given and the methods we have discussed. The only purpose of this section, however, is to illustrate that the interpolation points returned by our algorithm are certainly worth considering for practical applications.

REMARK 2.1. *Several authors working on rational spectral methods have discussed a* mapping *function $g(x)$ from the interval $[-1, 1]$ onto itself to transform the polynomial Chebyshev points to a new set of interpolation points and thus obtain better results, see e.g. [Berrut and Mittelmann 2005; Tee and Trefethen 2006]. The method we propose can be interpreted in this sense and then our mapping function is given by*

$$g(x) = \cos(F^{-1}(n \cos^{-1} x)).$$

With all poles at infinity, this indeed becomes $g(x) = x$.

3. PRELIMINARIES

Let there be fixed a finite sequence of complex numbers (poles) $A = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ that are outside the interval $[-1, 1]$. Some or all of them may be at infinity. The rational functions we deal with in this article are of the form

$$f_i(x) = \frac{c_i x^i + c_{i-1} x^{i-1} + \cdots + c_0}{(1 - x/\alpha_1)(1 - x/\alpha_2) \ldots (1 - x/\alpha_i)} \,. \tag{4}$$

Thus the poles at infinity are automatically accounted for.

The Joukowski transformation, which maps the complex unit circle to the interval $[-1, 1]$ and the unit disc to the exterior of the interval, is denoted by

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

and in everything that follows, x and z will always be related by this transformation. The inverse transformation is denoted by $z = J^{-1}(x)$. We define the complex numbers β_k by

$$\beta_k = J^{-1}(\alpha_k), \quad k = 1, 2, \ldots, n.$$

They are, by definition, inside the unit disc. Note that $\beta_k = 0$ if $\alpha_k = \infty$.

Given the poles α_k (and thus also the numbers β_k), the near best interpolation points for rational interpolation with these prescribed poles are defined as follows.

DEFINITION 3.1. *For $k = 1, \ldots, n$ let θ_k denote the (unique) solution to the equation*

$$F(\theta_k) = \pi \left(k - \frac{1}{2} \right), \tag{5}$$

where the function $F(\theta)$ is given by

$$F(\theta) = \sum_{j=1}^{n-1} [\arg(z - \beta_j) + \arg(z - \overline{\beta}_j)] + \arg(z - \Re(\beta_n)) - (n-1)\theta$$

for $\theta \in [0, \pi]$, where $\Re(\cdot)$ refers to the real part, $z = e^{\mathbf{i}\theta}$ and $\arg(\cdot)$ is the complex argument, taken in the interval $[-\pi/2, 3\pi/2)$. Then the interpolation points x_k are defined as

$$x_k = \cos \theta_k$$

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

Note that in the case where all poles are at infinity (which corresponds to all β's equal to zero), we have

$$x_k^\infty = \cos \theta_k^\infty, \quad \text{where} \quad \theta_k^\infty = \frac{\pi}{n} \left(k - \frac{1}{2} \right), \tag{6}$$

which are exactly the zeros of the Chebyshev polynomial of the first kind and degree n.

The following theorem gives some properties of the interpolation points x_k. It provides the connection with Gauss quadrature and minimax approximations. For the proof we refer to [Deckers et al. 2007] and [Van Deun 2007].

THEOREM 3.2. *Let the points x_k be as in Definition 3.1. Define the weights λ_k as*

$$\lambda_k = 2\pi \left[1 + \sum_{j=1}^{n-1} \left(\frac{1 - |\beta_j|^2}{|z_k - \beta_j|^2} + \frac{1 - |\beta_j|^2}{|z_k - \overline{\beta}_j|^2} \right) + \frac{1 - \Re(\beta_n)^2}{|z_k - \Re(\beta_n)|^2} \right]^{-1},$$

where $z_k = J^{-1}(x_k)$. Then the quadrature formula

$$\int_{-1}^{1} f(x) \frac{dx}{\sqrt{1 - x^2}} \approx \sum_{k=1}^{n} \lambda_k f(x_k)$$

is exact for any $f(x) = f_{n-1}(x)\overline{g_{n-1}(\overline{x})}$ where f_{n-1} and g_{n-1} are of the form (4). If α_n is real, then the formula is exact for any $f(x) = f_n(x)\overline{g_{n-1}(\overline{x})}$.

Furthermore, if the poles are real or appear in complex conjugate pairs and if $\alpha_n = \infty$, let $\mathcal{T}_n(x)$ denote the rational function of the form (4) of degree n whose zeros are the points x_k and for which $c_n = 1$ (i.e. the numerator is monic). Then it holds that

$$\max_{x \in [-1,1]} |\mathcal{T}_n(x)| \leq \max_{x \in [-1,1]} |f_n(x)|$$

for any f_n of the form (4) with $c_n = 1$.

If all poles are at infinity, this is a classical theorem about polynomial Gauss-Chebyshev quadrature and the extremal property of Chebyshev polynomials. In the more general (rational) case, it is precisely this extremal property which makes the points x_k so interesting, as was shown in Section 2.

REMARK 3.3. *Although we only present the quadrature formula for the weight function $1/\sqrt{1-x^2}$, the above exposition could be made more general to include other Chebyshev weight functions, see [Deckers et al. 2007]. The extremal property, however, no longer holds for the other weight functions.*

For reasons of notational simplicity and clarity, we restrict our exposition to the first Chebyshev weight function. The software that comes with this article also deals with the other weight functions. This is briefly discussed in Section 5.

4. NUMERICAL SOLUTION

In order to compute the interpolation points x_k, we need to solve equation (5) for $k = 1, \ldots, n$. Since an analytical solution is in general impossible, we have to use numerical methods. In essence, the procedure we propose here is the following:

> Approximate the inverse function $F^{-1}(\theta)$ using a piecewise cubic Hermite interpolating polynomial (PCHIP), evaluate this approximation in the points $\pi(k-1/2)$ and refine the results using Newton-Raphson iteration.

This procedure is similar to what we described in [Deckers et al. 2006], which is also based on Newton iteration, but which uses a variety of techniques to obtain the initial values. These techniques include linear extrapolation from two previous zeros to the next, estimating the internal inflection points of $F(\theta)$, and using the asymptotic distribution of the x_k. The current procedure is more efficient (requiring less iterations in general). It should be mentioned that the methods described in [Van Deun 2007] are completely different. They are based on eigenvalue problems and do not use the function $F(\theta)$. Although that implementation is much shorter and more straightforward than the current one, it is less efficient, especially for large values of n. Solving the eigenvalue problem is too expensive if no use is made of the special matrix structure, which is something we have not attempted yet.

4.1 The function $F(\theta)$

Before going into details, let us recall some properties of the function $F(\theta)$. For the proof (and some additional properties) we refer to [Deckers et al. 2006].

THEOREM 4.1. *The function $F(\theta)$ as given in Definition 3.1 is continuous and strictly increasing on $[0, \pi]$. It takes the values $F(0) = 0$ and $F(\pi) = n\pi$.*

Furthermore, define

$$f_\beta(\theta) = \arg(z - \beta) + \arg(z - \overline{\beta}), \quad z = e^{\mathbf{i}\theta}$$

(with the same convention for the argument as in Definition 3.1, this way the function is continuous and positive), so that

$$F(\theta) = \sum_{j=1}^{n-1} f_{\beta_j}(\theta) + \frac{1}{2} f_{\Re(\beta_n)}(\theta) - (n-1)\theta.$$

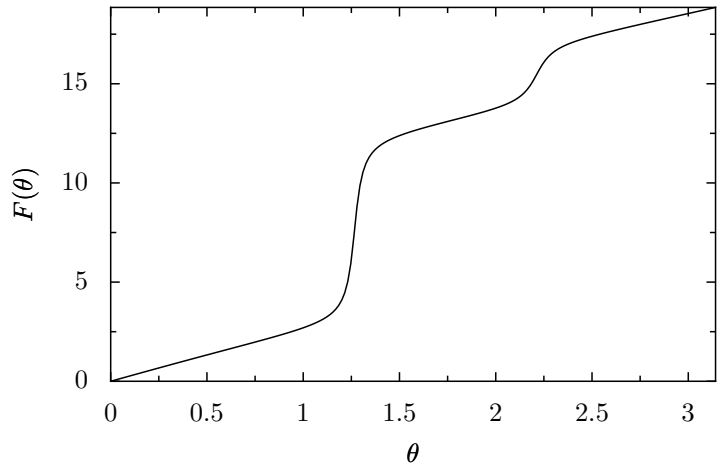

Fig. 5. The function $F(\theta)$ for $n = 6$ and the sequence of poles (8).

If $\beta \neq 0$ then $f'_\beta(\theta)$ has a local maximum in

$$\theta_* = \cos^{-1}\left[\frac{1+|\beta|^2}{2\Re(\beta)} - \frac{|\Im(\beta)|\sqrt{(1+|\beta|^2)^2 - 4\Re(\beta)^2}}{2\Re(\beta)|\beta|}\right], \tag{7}$$

where $\Im(\cdot)$ is the imaginary part and $\cos^{-1}(\cdot)$ is the arccosine. If the argument of the arccosine is greater than 1 (less than -1), the local maximum occurs in 0 (respectively π). In the limit case where m poles coalesce in the point $\alpha \in [-1, 1]$, the function $F(\theta)$ is discontinuous in $\theta_* = \cos^{-1}(\alpha)$ with a jump equal to

$$F(\theta_*+) - F(\theta_*-) = m\pi.$$

The last two statements indicate that poles very close to the interval lead to steep gradients (discontinuities in the limit case) in $F(\theta)$, whose location can be estimated by (7)[1]. If $\alpha \in [-1, 1]$ then $|\beta| = 1$ and the expression between the square brackets in (7) is equal to $\Re(\beta) = \alpha$. Figure 5 illustrates the previous theorem for $n = 6$ and the sequence of poles

$$\alpha_1 = 2,\ \alpha_2 = \alpha_3 = \alpha_4 = 0.3 + 0.03\mathbf{i},\ \alpha_5 = -0.6 + 0.05\mathbf{i},\ \alpha_6 = -2. \tag{8}$$

Note the steep gradient close to $\theta = 1.25$, caused by the triple pole at $\alpha = 0.3+0.03\mathbf{i}$. The size of the jump is approximately 3π and estimating its location through (7) gives $\theta_* \approx 1.266$. The smaller jump caused by α_5 can be estimated in the same way.

[1]If $\Re(\beta) = 0$ then this equation should be interpreted in a limiting sense. In Section 5 we give an equivalent formula which is numerically stable for all β.

4.2 PCHIP

Since $F(\theta)$ is strictly increasing, so is its inverse, and we would like to approximate it by a function with the same property, hence the idea of using a PCHIP. This is essentially an interpolating spline that preserves monotonicity. An ordinary spline is not appropriate, since steep gradients in F tend to produce overshoot and 'wiggles' in the approximant. For details about properties and the construction of the PCHIP we refer to [Fritsch and Carlson 1980]. Compared to the (ordinary) cubic interpolating spline, the PCHIP is less expensive to set up, but its second derivative is not necessarily continuous in the interpolation points. The algorithm is available as the standard MATLAB function `pchip`.

As interpolation points for the PCHIP we take $(F(\theta_k^\infty), \theta_k^\infty)$, together with the endpoints $(0, 0)$ and $(n\pi, \pi)$. Here θ_k^∞ refers to the solutions for the polynomial case as defined in (6). Specifically, this means that we do not take into account the position of the poles when constructing the PCHIP. If the approximation is not good enough and the Newton iterations diverge for certain θ_k, this is quickly detected and new initial values are obtained as described in Sections 4.3 and 4.4.

So let $P(\theta)$ denote the PCHIP approximation to $F^{-1}(\theta)$ such that

$$\begin{aligned} &P(0) = F^{-1}(0) = 0, \\ &P(F(\theta_k^\infty)) = F^{-1}(F(\theta_k^\infty)) = \theta_k^\infty, \quad k = 1, \ldots, n, \\ &P(n\pi) = F^{-1}(n\pi) = \pi. \end{aligned} \tag{9}$$

Then the approximations $\theta_k^{(0)}$ to the true zeros θ_k defined by (5) are given by

$$\theta_k^{(0)} = P(\pi(k - 1/2)), \quad k = 1, \ldots, n. \tag{10}$$

The superscript $^{(0)}$ indicates that these numbers are used as initial values for the Newton iterations to obtain a better approximation to θ_k. Regarding the accuracy of these initial values, we prove the following theorem.

THEOREM 4.2. *Let θ_k and $\theta_k^{(0)}$ be as defined above. Then it holds that*

$$|\theta_k - \theta_k^{(0)}| \leq \frac{\pi}{n}, \quad k = 1, \ldots, n.$$

PROOF. This is a simple consequence of the interpolation and monotonicity properties of $P(\theta)$. Fix k and determine j so that

$$F(\theta_j^\infty) \leq \pi \left(k - \frac{1}{2} \right) \leq F(\theta_{j+1}^\infty)$$

assuming that this can be done. Since both $P(\theta)$ and $F^{-1}(\theta)$ are strictly increasing, applying P on both sides it follows from these inequalities, the interpolation property (9) and equation (10) that

$$\begin{aligned} \theta_j^\infty &\leq \theta_k^{(0)} \leq \theta_{j+1}^\infty, \\ \theta_j^\infty &\leq \theta_k \leq \theta_{j+1}^\infty, \end{aligned}$$

which leads to

$$|\theta_k - \theta_k^{(0)}| \leq \theta_{j+1}^\infty - \theta_j^\infty = \frac{\pi}{n}.$$

In the case where no such j can be found, we either have $0 \le \pi(k-1/2) \le F(\theta_1^\infty)$ or $F(\theta_n^\infty) \le \pi(k-1/2) \le n\pi$, which both lead to

$$|\theta_k - \theta_k^{(0)}| \le \frac{\pi}{2n}.$$

This completes the proof. □

As a consequence of this theorem, the initial values $\theta_k^{(0)}$ will be better if n is large. Also, because of the choice of the interpolation points for $P(\theta)$, the approximation will deteriorate if the distribution of the poles deviates much from the polynomial case (which corresponds to all poles at infinity). As explained above, the presence of poles very close to the interval causes abrupt transitions in $F(\theta)$ that are difficult to follow for $P(\theta)$. In that case, the initial values may be too far from the exact solutions for the Newton iterations to converge. We then need additional starting values, which can be obtained with the aid of formula (7). However, first we need to find out which poles are close to the interval and cause difficulties.

4.3 Poles close to the interval

Poles attract zeros. If there are many poles close to a point of the interval, then the x_k will also gather around this point. This vague statement can be made exact in several ways.

One way is to study the asymptotic distribution of the x_k's as a function of the asymptotic distribution of the poles. This was done using logarithmic potential theory for the case of real poles in [Van Assche and Vanherwegen 1993] and extended to the case of arbitrary complex poles in [Deckers et al. 2007]. It is also possible to give an interpretation for a finite number of poles in terms of electrostatic equilibrium, where negatively charged poles attract positively charged zeros, but that is outside the scope of this article.

Here we look at this statement in terms of the function $F(\theta)$. It follows from Theorem 4.1 that in the limit case where m poles coalesce in $x \in [-1,1]$, there will also be m (or at most $m+1$) of the x_k equal to x. We can also be more quantitative if the poles are not *on* but *close to* the interval, but first we need some simple lemmas.

LEMMA 4.3. *Let $f(z)$ be a function analytic in a small annulus containing the complex unit circle and put $z = e^{\mathbf{i}\theta}$. Then*

$$\frac{d}{d\theta} \arg f(z) = \Re\left(z\frac{f'(z)}{f(z)}\right),$$

regardless of which branch of the argument is taken. If z is on a branch cut of $\arg f(z)$, the derivative on the left should be interpreted in a limiting sense.

PROOF. First note that, with $z = e^{\mathbf{i}\theta}$, we have

$$\frac{df}{d\theta} = \frac{df}{dz} \cdot \frac{dz}{d\theta} = \mathbf{i}zf'(z), \tag{11}$$

and also, since $\overline{f(z)}$ is an analytic function of $\overline{z}$,

$$\frac{d\overline{f}}{d\theta} = \frac{d\overline{f}}{d\overline{z}} \cdot \frac{d\overline{z}}{d\theta} = -\mathbf{i}\overline{z}\overline{f'(z)}. \tag{12}$$

Next write

$$\arg f(z) = \tan^{-1} \frac{\Im(f(z))}{\Re(f(z))} , \tag{13}$$

where $\tan^{-1}$ refers to the arctangent with the same branch structure as the argument.

Then write $\Re(f(z)) = (f(z) + \overline{f(z)})/2$ and $\Im(f(z)) = -\mathbf{i}(f(z) - \overline{f(z)})/2$ and use formulas (11) and (12) to differentiate (13). If z is on a branch cut of $\arg f(z)$, then strictly speaking the derivative does not exist because of the discontinuity, but the slopes on both sides of the jump are the same. The derivative should be interpreted in this sense (to be more exact: the value of the derivative at points where it is undefined should be replaced by its limiting value).

Some computations now yield

$$\frac{d}{d\theta} \arg f(z) = \frac{z f'(z)\overline{f(z)} + \overline{z}\,\overline{f'(z)} f(z)}{2 f(z)\overline{f(z)}} ,$$

which completes the proof. □

We use this lemma to prove the following result.

LEMMA 4.4. *Let $f_\beta(\theta)$ be as defined in Theorem 4.1. Then*

$$\frac{2}{1+|\beta|} \leq f'_\beta(\theta) \leq \frac{2}{1-|\beta|} .$$

PROOF. Use the previous lemma to obtain

$$f'_\beta(\theta) = \Re\left(\frac{z}{z-\beta}\right) + \Re\left(\frac{z}{z-\overline{\beta}}\right),$$

then use $\Re(\cdot) \leq |\cdot|$ and the fact that $|z| = 1$ and $|z - \beta| \geq 1 - |\beta|$ to find the upper bound.

For the lower bound we have

$$\Re\left(\frac{z}{z-\beta}\right) = 1 + \Re\left(\frac{\beta}{z-\beta}\right) = 1 + \frac{\Re(\beta\overline{z}) - |\beta|^2}{|z-\beta|^2} .$$

Now use $\Re(\cdot) \geq -|\cdot|$ and the fact that $|z| = 1$ and $|z - \beta|^2 \leq (1 + |\beta|)^2$ to get

$$\Re\left(\frac{z}{z-\beta}\right) \geq \frac{1}{1+|\beta|} .$$

The same applies of course for the term with $\overline{\beta}$. □

With the aid of this lemma we obtain information about the minimum and maximum distance between the θ_k, as shown in the next theorem.

THEOREM 4.5. *Define*

$$r = \max_{1 \leq k \leq n} |\beta_k|$$

and

$$\Delta = \frac{1+r}{1-r} .$$

Then it holds that

$$\min_{1\le k\le n-1} |\theta_k - \theta_{k+1}| \ge \frac{1}{\Delta}\frac{\pi}{n} ,$$
$$\max_{1\le k\le n-1} |\theta_k - \theta_{k+1}| \le \Delta\frac{\pi}{n} .$$

This means that there can be no more than Δ of the x_j between any two consecutive Chebyshev points x_k^∞ and x_{k+1}^∞ and vice versa. Note that $0 \le r < 1$. This number gives an indication of how close the poles are to the interval: the closer r is to 1, the closer some poles are to the interval.

PROOF. From the definition of $F(\theta)$ and the previous lemma it follows that

$$F'(\theta) \le (n-1)\frac{2}{1-r} + \frac{1}{1-r} - (n-1) = \frac{n(1+r)-r}{1-r} \le n\,\frac{1+r}{1-r}$$

and similarly

$$F'(\theta) \ge n\,\frac{1-r}{1+r} .$$

Hence we have

$$\frac{1}{\Delta}\frac{1}{n} \le (F^{-1})'(\theta) \le \Delta\frac{1}{n} , \tag{14}$$

where we use the symbol $(F^{-1})'(\theta)$ to denote the derivative of $F^{-1}(\theta)$. Now apply the mean value theorem to the function $F^{-1}(\theta)$ on the interval $[\pi(k-1/2), \pi(k+1/2)]$ to find that

$$\frac{\theta_{k+1}-\theta_k}{\pi} = (F^{-1})'(\xi)$$

for some ξ inside this interval. Inserting the preceding result in formula (14) and taking the minimum (respectively maximum) over all k proves the first part of the theorem.

The distance between two consecutive θ_k^∞ and θ_{k+1}^∞ is exactly π/n so there can be no more than Δ of the θ_j between them and vice versa. Since the cosine is monotonous on $[0,\pi]$, this also holds for the x_j and x_k^∞. □

In particular, if $r = 1/3$ then $\Delta = 2$ so there can be at most two rational interpolation points between any pair of successive Chebyshev points and there can be at most two Chebyshev points between any pair of successive rational interpolation points. Thus, the distribution of the x_k will not differ much from the polynomial case. Consequently, we consider a pole to be *close to the interval* when it satisfies $|\beta| > 1/3$. Through the Joukowski transformation this translates to

$$\left(\frac{\Re(\alpha)}{5/3}\right)^2 + \left(\frac{\Im(\alpha)}{4/3}\right)^2 < 1, \tag{15}$$

which means the pole is inside an ellipse with semimajor axis 5/3 and semiminor axis 4/3, centered around the interval. Taking $\Delta = 2$ may seem rather arbitrary, but the main reason we use it, is that it seems to work very well in practice. In all our experiments, the Newton iterations discussed in the previous section have never diverged when all poles were outside this ellipse.

So when the Newton iterations for one or more of the θ_k diverge, we need only consider formula (7) for poles close to the interval (i.e. satisfying the above criterion) to find out where $F(\theta)$ is steep. However, it may happen that a pole α_k is close to the interval and nevertheless $F(\theta)$ is not particularly steep near the point θ_* obtained from (7) with $\beta = \beta_k$, e.g. when n is large and all other poles are far away. Therefore, we need an additional criterion to decide which points θ_* to use to obtain new starting values.

Suppose that $\theta_* \in [\theta_k^\infty, \theta_{k+1}^\infty]$, then we compare the slope of F at θ_* to the slope of the straight line connecting the points $(\theta_k^\infty, F(\theta_k^\infty))$ and $(\theta_{k+1}^\infty, F(\theta_{k+1}^\infty))$. In particular, if

$$F'(\theta_*) > 2\frac{F(\theta_{k+1}^\infty) - F(\theta_k^\infty)}{\theta_{k+1}^\infty - \theta_k^\infty}, \tag{16}$$

then we retain the point θ_* to obtain new initial values for the Newton iterations. Again, this criterion is mostly heuristic and gives satisfactory results in all the experiments we have done.

4.4 Obtaining new initial values

When the Newton iterations for a certain θ_k diverge, proceed as follows. First determine the poles that satisfy (15). From these poles, compute the corresponding θ_* values by (7) and only retain those for which (16) holds. Denote this set by Θ.

If Θ has only one element, then take this as initial value for θ_k.

If Θ has more elements, we need to determine the position of θ_k with respect to every value in Θ. Since $F(\theta)$ is monotonic, it suffices to evaluate F in every element of Θ. If θ_k is greater (less) than the largest (smallest) element of Θ, then take this element as initial value. Else, there exist $\theta_{*,1} \in \Theta$ and $\theta_{*,2} \in \Theta$ so that $\theta_{*,1} \leq \theta_k \leq \theta_{*,2}$ (assume that $[\theta_{*,1}, \theta_{*,2}]$ is the smallest such interval). Now define

$$\tilde{\theta} = \frac{\theta_{*,1} + \theta_{*,2}}{2} + \frac{\pi}{4}\left(\frac{n_1}{F'(\theta_{*,1})} - \frac{n_2}{F'(\theta_{*,2})}\right),$$

where n_i is the 'multiplicity' of $\theta_{*,i}$, i.e. the number of (not necessarily distinct) poles that lead to $\theta_{*,i}$. Then if $\theta_k \geq \tilde{\theta}$, take $\theta_{*,2}$ as initial value, else take $\theta_{*,1}$. Note that $\tilde{\theta}$ is the point halfway between $\theta_{*,1}$ and $\theta_{*,2}$ plus some small 'correction'. This formula is arrived at as follows: since $\theta_{*,1}$ attracts approximately n_1 points θ_k, and since the values $F(\theta_k)$ are all at a distance π from each other, we may estimate that the largest θ_k which is attracted by $\theta_{*,1}$, is given by

$$\theta_k \approx \theta_{*,1} + \frac{n_1\pi/2}{F'(\theta_{*,1})},$$

assuming that half of the θ_k attracted by $\theta_{*,1}$ are to the left of $\theta_{*,1}$ and the other half are to the right. Similarly we estimate the smallest θ_k attracted by $\theta_{*,2}$. The point $\tilde{\theta}$ is then simply the point halfway between these two estimates. This works remarkably well in practice.

However, if the Newton iterations still diverge with these new initial values, we finally resort to the method of bisection to obtain the exact values θ_k. This only happens if there are several distinct poles *very* close to the interval (e.g. at a

distance of 100 times machine epsilon) and is not likely to occur much in practical applications. An example is given in Section 6.

5. IMPLEMENTATION ISSUES

The algorithm is implemented as the MATLAB function `rcheb`, which consists of several subfunctions. The function `rcheb` itself takes two arguments. The first is a vector of poles, the second is optional and indicates the weight function for the corresponding quadrature formula. It defaults to the first Chebyshev weight, which is the one used in this article. For the other two Chebyshev weight functions, everything that has been said before about the algorithm remains valid, but the definition of $F(\theta)$ and equation (5) are slightly different. The general formulation (covering all weight functions) requires the introduction of two constants, as discussed in [Deckers et al. 2007].

The function `psort` is used to sort the sequence of poles and the sequence Θ mentioned in Section 4.4, and to compute the multiplicity of each pole α or each θ_*-value. It uses the MATLAB function `sort`, which sorts complex numbers first by absolute value and then by angle.

To evaluate the function $F(\theta)$ and its derivative, we use `ceval`, `ftheta` and `fdtheta`. If $F(\theta)$ is to be evaluated in several points at the same time, we can either loop over all evaluation points, or over the different poles, or even avoid a loop altogether by using matrix arguments. Which method is faster depends on the number of evaluation points and the number of distinct poles, and the decision is based on a test that was determined heuristically (by evaluating $F(\theta)$ in a substantial part of its parameter space, using each of the three methods, and for each set of arguments checking which method is faster). This test is done at the beginning of `ceval`.

Care has to be taken when computing $\arg(z-\beta)$ when z is very close to β. Recall that

$$\arg(z-\beta) = \tan^{-1}\frac{\sin\theta - r\sin\phi}{\cos\theta - r\cos\phi},$$

where $z = e^{\mathbf{i}\theta}$ and $\beta = re^{\mathbf{i}\phi}$, and digits are lost when computing the differences in numerator and denominator when z and β are very close to each other. To avoid this, we use Simpson's formulas to obtain the equivalent expression

$$\frac{\sin\theta - r\sin\phi}{\cos\theta - r\cos\phi} = \frac{2\sin\frac{\theta-\phi}{2}\cos\frac{\theta+\phi}{2} + (1-r)\sin\phi}{2\sin\frac{\theta+\phi}{2}\sin\frac{\phi-\theta}{2} + (1-r)\cos\phi}$$

which yields much more accurate results.

The Newton-Raphson iterations are done for several nodes at the same time in `newton`. As soon as a node has converged, it is omitted from subsequent computations to speed up performance. Since $F(\theta)$ is monotonic, it is also easy to determine when an iteration starts diverging. If this is the case, the node is also omitted from subsequent computations and marked as diverging. An interval containing the exact solution is returned in case it is needed later for the method of bisection. We point out that the maximum number of iterations is set to 10. Once convergence sets in, it follows from the quadratic convergence that it will take at most 5 iterations to reach machine precision (working in double precision). If convergence

does not start after 5 iterations, the initial value is considered not good enough. The accuracy tolerance is set to 50 times machine precision in the variable `xtol` in the main programme. This is also used for bisection and can be changed if less accuracy is needed.

For the nodes that have not converged, new initial values are determined using the approach discussed in Section 4.4. This is done in `newinit`. The programme does not use formula (7), but instead uses the equivalent formula

$$\theta_* = \cos^{-1}\left[\frac{\Re(\beta)((1+|\beta|^2)^2+4\Im(\beta)^2)}{2|\beta|(|\beta|(1+|\beta|^2)+|\Im(\beta)|\sqrt{(1+|\beta|^2)^2-4\Re(\beta)^2})}\right],$$

which is numerically stable for all values of β. When choosing between $\theta_{*,1}$ and $\theta_{*,2}$ based on the estimate for $\tilde{\theta}$, we do not throw away the other possibility, but keep it as a third initial value just in case the Newton iterations still diverge.

If all possible initial values have been tried and there are still nodes that have not been found, the programme uses bisection as a last means of obtaining the exact solution. This is done in the function `bisect`. The maximum number of iterations is set to 52. This way, convergence to machine precision (double precision) is always guaranteed. The iterations are started from an enclosing interval that is returned by `newton` and improved upon in `bound`. This improvement is based on the fact that $F(\theta)$ is monotonic. We know that $\theta_i < \theta_j < \theta_k$ if $i < j < k$, so we can use the nodes that have already been found to obtain bounds for the remaining ones. Under normal circumstances, bisection will never be necessary, but if $F(\theta)$ is almost discontinuous in several points (which means there are several distinct poles very close to the interval), then bisection is inevitable for at least a few nodes. Examples are given in the next section.

If a second output argument is present, it is used to return the quadrature weights, which are computed in the function `weight`. The same remark regarding the computational efficiency applies as in the case of `ceval`. However, the function does not use the formula from Theorem 3.2, but an equivalent formula based on the equality

$$|z-\beta|^2 = (1-r)^2 + 4r\sin^2\frac{\theta-\phi}{2}$$

where θ, r and ϕ are as above. This formula is more accurate, but when z and β are very close to each other, digits are inevitably lost. This is illustrated in the next section.

Finally, a third output argument may be used to obtain an accuracy estimate for each of the computed nodes. This is a vector that contains the values $F(\hat{\theta}_k)/F'(\hat{\theta}_k)$, where $\hat{\theta}_k$ is our computed approximation to the exact zero θ_k. Again, only in exceptional cases where there are poles extremely close to the interval, some of the estimates can be greater than `xtol`. Examples are given in the next section. This output parameter is only useful for verification purposes.

6. EXPERIMENTS

To test the code and to give the users some examples to work with, we have included a file `experiments.m` that illustrates the use of this software. For several sets of poles and each of the three weight functions, we compute the nodes x_k and

the quadrature weights λ_k and look at the accuracy estimates mentioned above. Furthermore, we use the nodes and weights to compute the integral of a function that should be integrated exactly by this formula. This is based on the following theorem, which is an immediate consequence of Theorem 3.2 in [Deckers et al. 2007]. For the proof, we refer to that article.

THEOREM 6.1. *Define the factors* $B_k(z)$ *as*

$$B_0 \equiv 1, \quad B_k(z) = \frac{z-\beta_k}{1-\overline{\beta}_k z} \cdot B_{k-1}(z), \quad k = 1, 2, \ldots$$

and for each weight function $w(x)$ *and each* k *define*

$$\varphi_k(x) = \frac{z\overline{B_{k-1}(\overline{z})}}{1-\beta_k z} + \frac{1}{(z-\beta_k)B_{k-1}(z)}, \qquad w(x) = \frac{1}{\sqrt{1-x^2}},$$

$$\varphi_k(x) = \frac{\sqrt{2}}{z-1}\left(\frac{z^2\overline{B_{k-1}(\overline{z})}}{1-\beta_k z} - \frac{1}{(z-\beta_k)B_{k-1}(z)}\right), \qquad w(x) = \sqrt{\frac{1-x}{1+x}},$$

$$\varphi_k(x) = \frac{2}{z^2-1}\left(\frac{z^3\overline{B_{k-1}(\overline{z})}}{1-\beta_k z} - \frac{1}{(z-\beta_k)B_{k-1}(z)}\right), \qquad w(x) = \sqrt{1-x^2},$$

then it follows that

$$\int_{-1}^{1} |\varphi_k(x)|^2 w(x)dx = \frac{2\pi}{1-|\beta_k|^2}.$$

The functions $\varphi_k(x)$ are in fact orthogonal rational functions with respect to the weight function $w(x)$. It can be verified that they are of the form (4) and thus the integral can be computed exactly with an n-point quadrature formula as long as $k < n$.

We only discuss some special cases in this text, plenty of 'normal' examples are given in the file `experiments.m`. For comparison purposes, we also include a file `experiments.out`, which contains the output of `experiments.m`, executed on a computer with a 2.4 GHz AMD Opteron processor.

First let us take 7 distinct poles, each of multiplicity 10 and just above the interval (at a distance of 100 times machine epsilon). The MATLAB code for this example is reproduced below and shows the exact location of the poles.

```
>> a = [-0.6:0.2:0.6] + 1e2*eps*i;
>> a = repmat(a,1,10);
>> [x,lambda,err] = rcheb(a);
>> 71 - find(abs(err) > 50*eps)
ans =
    10
>> err(61)
ans =
     2.841055762200743e-14
>> abs(1 - sum(lambda)/pi)
ans =
```

```
    2.157035816630071e-08
```

This code fragment requires some explanation, as it illustrates a kind of worst case scenario that we analyse in some more detail. The test on the fourth line shows that the node x_{10} cannot be computed to full accuracy, but only to approximately 14 digits (nodes are stored in reverse order since $x_k < x_l$ if $k > l$). It also turns out that for this node (and for x_1) bisection is necessary (this can, of course, not be deduced from this code fragment). It is surprising that the method of bisection does not yield accurate results. In fact, the only reasonable explanation is that the function $F(\theta)$ itself cannot be computed accurately, and that is exactly what happened. In spite of the more accurate formulas from the previous section (based on Simpson's formulas), one digit is still lost when computing $\arg(z-\beta)$ for z close to β. Without these more accurate formulas however, up to 6 or 7 digits would be lost in this example. The last test in this code fragment shows that additional digits are also lost when computing the quadrature weights, which should sum up to π. Again, since the zeros are so close to the poles, this is inevitable. The quadrature test based on the above theorem is not shown here, but is included in the file `experiments.m` and exhibits the same loss of digits.

For the next example, we take poles at the integer multiples of $\mathbf{i}\omega$ where $\omega = 0.001$. If all poles are at the imaginary axis, it can be shown that the nodes x_k are symmetric with respect to the origin, which provides us with an extra test for accuracy, as shown in the next code fragment.

```
>> a = [1:10]*i*0.001;
>> a = [a, -a];
>> [x,lambda,err] = rcheb(a);
>> find(abs(err) > 50*eps)
ans =
   Empty matrix: 1-by-0
>> max(abs(x(1:10) + x(20:-1:11)))
ans =
     4.996003610813204e-16
>> abs(1 - sum(lambda)/pi)
ans =
     1.110223024625157e-15
```

Bisection was not necessary to compute the nodes.

For the last example, we look at the case of a very large n. We take three distinct poles (one on either side of the interval and one just above the interval), repeated ten thousand times. The results are shown below.

```
>> a = [-1.1 0.1*i 1.1];
>> a = repmat(a,1,10000);
>> [x,lambda,err] = rcheb(a);
>> find(abs(err) > 50*eps)
ans =
   Empty matrix: 1-by-0
>> abs(1-sum(lambda)/pi)
ans =
```

```
6.106226635438361e-15
```

It takes approximately 6.1 seconds to execute this piece of code. Computing only the nodes, without the weights and accuracy estimates, takes 4.3 seconds. We are fairly confident that it is impossible to obtain the same efficiency using eigenvalue based methods. No bisection was used in this case either.

REFERENCES

Abramowitz, M. and Stegun, I. A. 1964. *Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Tables*. Applied Mathematics Series, vol. 55. National Bureau of Standards, Washington, D.C.

Baltensperger, R., Berrut, J.-P., and Dubey, Y. 2003. The linear rational pseudospectral method with preassigned poles. *Numer. Algorithms 33,* 1-4, 53–63. International Conference on Numerical Algorithms, Vol. I (Marrakesh, 2001).

Baltensperger, R., Berrut, J.-P., and Noël, B. 1999. Exponential convergence of a linear rational interpolant between transformed Chebyshev points. *Math. Comp. 68,* 227, 1109–1120.

Berrut, J.-P. 1997. The barycentric weights of rational interpolation with prescribed poles. *J. Comput. Appl. Math. 86,* 1, 45–52. Special issue dedicated to William B. Gragg (Monterey, CA, 1996).

Berrut, J.-P. and Baltensperger, R. 2001. The linear rational pseudospectral method for boundary value problems. *BIT 41,* 5, suppl., 868–879. BIT 40th Anniversary Meeting.

Berrut, J.-P. and Mittelmann, H. D. 1997. Matrices for the direct determination of the barycentric weights of rational interpolation. *J. Comput. Appl. Math. 78,* 2, 355–370.

Berrut, J.-P. and Mittelmann, H. D. 2001. The linear rational pseudospectral method with iteratively optimized poles for two-point boundary value problems. *SIAM J. Sci. Comput. 23,* 3, 961–975 (electronic).

Berrut, J.-P. and Mittelmann, H. D. 2004. Adaptive point shifts in rational approximation with optimized denominator. In Proceedings of the 10th International Congress on Computational and Applied Mathematics (ICCAM-2002). *J. Comput. Appl. Math. 164/165*, 81–92.

Berrut, J.-P. and Mittelmann, H. D. 2005. Optimized point shifts and poles in the linear rational pseudospectral method for boundary value problems. *J. Comput. Phys. 204,* 1, 292–301.

Bush, A. W. 1992. *Perturbation methods for engineers and scientists.* CRC Press Library of Engineering Mathematics. Boca Raton, FL: CRC Press. xii, 303 p.

Cuyt, A. and Wuytack, L. 1987. *Nonlinear methods in numerical analysis*. North-Holland Mathematics Studies, vol. 136. North-Holland Publishing Co., Amsterdam. Studies in Computational Mathematics, 1.

Deckers, K., Van Deun, J., and Bultheel, A. 2006. Computing rational Gauss-Chebyshev quadrature formulas with complex poles. In *Proceedings of the Fifth International Conference on Engineering Computational Technology*, B. H. V. Topping, G. Montero, and R. Montenegro, Eds. Civil-Comp Press, Stirlingshire, United Kingdom. Paper 30.

Deckers, K., Van Deun, J., and Bultheel, A. 2007. Rational Gauss-Chebyshev quadrature formulas for complex poles outside $[-1, 1]$. *Math. Comp.* To appear.

Fornberg, B. 1996. *A practical guide to pseudospectral methods*. Cambridge University Press, Cambridge.

Fritsch, F. N. and Carlson, R. E. 1980. Monotone piecewise cubic interpolation. *SIAM J. Numer. Anal. 17,* 2, 238–246.

Gautschi, W. 2000. Quadrature rules for rational functions. *Numer. Math. 86(4)*, 617–633.

Gottlieb, D. and Orszag, S. A. 1977. *Numerical analysis of spectral methods: theory and applications*. SIAM, Philadelphia.

Polezzi, M. and Sri Ranga, A. 2007. On the denominator values and barycentric weights of rational interpolants. *J. Comput. Appl. Math. 200(2)*, 576–590.

Schneider, C. and Werner, W. 1991. Hermite interpolation: the barycentric approach. *Computing 46,* 1, 35–51.

Tee, T. W. and Trefethen, L. N. 2006. A rational spectral collocation method with adaptively transformed chebyshev grid points. *SIAM Journal on Scientific Computing 28,* 5, 1798–1811.

Trefethen, L. N. 2000. *Spectral methods in MATLAB*. Software, Environments, and Tools, vol. 10. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA.

Van Assche, W. and Vanherwegen, I. 1993. Quadrature formulas based on rational interpolation. *Math. Comp. 61(204)*, 765–783.

Van Deun, J. 2007. Eigenvalue problems to compute almost optimal points for rational interpolation with prescribed poles. *Numer. Algorithms 45*, 89–99.

Van Deun, J. and Bultheel, A. 2003. Orthogonal rational functions and quadrature on an interval. *J. Comput. Appl. Math. 153(1–2)*, 487–495.

Van Deun, J. and Bultheel, A. 2006. A quadrature formula based on Chebyshev rational functions. *IMA J. Numer. Anal. 26,* 4, 641–656.

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

Weideman, J. A. C. 1999. Spectral methods based on nonclassical orthogonal polynomials. In *Applications and computation of orthogonal polynomials (Oberwolfach, 1998)*. Internat. Ser. Numer. Math., vol. 131. Birkhäuser, Basel, 239–251.

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