

Fast and Stable Rational Interpolation in Roots of Unity and Chebyshev points

Ricardo Pachón Pedro Gonnet
Oxford University Mathematical Institute
24–29 St Giles', Oxford, OX1 3LB, UK.

Joris van Deun
Universiteit Antwerpen
Department of Mathematics and Computer Science
Middelheimlaan 1, B-2020 Antwerpen, Belgium

A new method for interpolation by rational functions of prescribed numerator and denominator degrees is presented. When the interpolation nodes are roots of unity or Chebyshev points, the algorithm is particularly simple and relies on discrete Fourier transform matrices, which results in a fast implementation using the Fast Fourier Transform. The method is generalised for arbitrary grids, which requires the construction of polynomials orthogonal on the set of interpolation nodes. The appearance of common factors in the numerator and denominator due to finite precision arithmetic is explained by the behaviour of the singular values of the linear system associated to rational interpolation problem. The new algorithm has connections with other methods, particularly the work of Jacobi and Kronecker, Berrut and Mittelman, and Egecioğlu and Koç. Short codes in Matlab and numerical experiments are included.

Oxford University Mathematical Institute
Numerical Analysis Group
24-29 St Giles'
Oxford, England OX1 3LB
E-mail: pachon@maths.ox.ac.uk

June, 2010

Report no. 10/02

Oxford University Mathematical Institute
Numerical Analysis Group
24-29 St Giles'
Oxford, England OX1 3LB
E-mail: pachon@maths.ox.ac.uk

June, 2010

1 Introduction

Let $\mathcal{P}(N)$ be the space of polynomials of degree less than or equal to N and $\mathcal{R}(m, n)$ the space of rational functions of the form p/q , where p and q are functions in $\mathcal{P}(m)$ and $\mathcal{P}(n)$ respectively, and m, n and N are nonnegative integers. In this paper we consider the numerical solution of the following rational interpolation problem, also known as the *Cauchy interpolation problem* [7]: Given $\mathbf{x} = [x_0, \dots, x_N]^\top$ and $\mathbf{f} = [f_0, \dots, f_N]^\top$, both in \mathbb{C}^{N+1} , with $x_i \neq x_j$ if $i \neq j$, and nonnegative integers m and n such that $N = m + n$, determine a function $r \in \mathcal{R}(m, n)$ such that

$$(1.1) \quad r(x_j) = \frac{p(x_j)}{q(x_j)} = f_j, \quad \text{for } j = 0, \dots, N.$$

We refer to \mathbf{x} as the *set of nodes* or the *grid*, and a rational function in $\mathcal{R}(m, n)$ as being of *type* $[m/n]$.

Polynomial interpolation, the particular case of (1.1) in which $n = 0$, provides a convenient framework to introduce a discussion of rational interpolation. In this case the interpolant exists and is unique, and its representation in the Lagrange or Newton basis, for example, allows one to explicitly write the polynomial in terms of \mathbf{f} , a useful property for theoretical and practical purposes (for an introduction to polynomial interpolation see, for example, [9, Chap. 2]). The barycentric formula [5] is an efficient and numerically stable efficient Lagrange method for its evaluation.

Two particular grids for polynomial interpolation have been extensively studied in connection with the problem of approximating functions: roots of unity and Chebyshev points, i.e. extrema of Chebyshev polynomials [31]. Some of the properties of polynomial interpolation in these nodes are the following: simple barycentric weights [22, 44]; computation of Taylor/Chebyshev coefficients with the Fast Fourier Transform (FFT) [17, 28]; connection with least squares approximation [52, Sec. 7.10]; “near-best” approximation [6, 18]; algebraic convergence of smooth functions [32]; and exponential convergence of analytic functions [52, Sec. 7.1].

The distribution of the nodes is decisive in polynomial interpolation. For example, the asymptotic distribution of Chebyshev points is $\mu(x) = \pi^{-1}(1 - x^2)^{-1/2}$ as $N \rightarrow \infty$, which determines a constant logarithmic potential $U(z) = \int \log \frac{1}{|z-x|} d\mu(x)$ (for the connection of potential theory with polynomial approximation see [27, 30, 41]). A similar property holds for roots of unity [41]. From this follows, for polynomial interpolation in these nodes, the uniform convergence of analytic functions and the well-conditioning of the problem [48], i.e. small perturbations of the data only cause small perturbations of the interpolant.

However, if the asymptotic distribution of the grid determines a nonconstant logarithmic potential, the interpolant may not converge regardless of the smoothness of the function, and even for functions for which the interpolant converges in theory, the problem is ill-conditioned, affecting its numerical solution [48]. The failure of interpolating polynomials to converge on arbitrary grids is called the *Runge phenomenon*.

Moreover, even if interpolating in “good points”, the rate of convergence may be slower than what might be desired. For example, if f is an analytic function on $[-1, 1]$

the convergence rate is ρ^{-N} , where ρ is the sum of the semimajor and semiminor axes of the maximum ellipse with foci ± 1 where f is analytic. Thus, if f is a function with singularities too close to $[-1, 1]$, the parameter ρ will be $1 + \epsilon$ with $\epsilon > 0$ very small.

The motivation for using rational functions is that by using a larger class of interpolatory functions it should be possible to overcome some of the deficiencies of polynomial interpolation. Theoretical and practical examples offer some evidence for this. For instance, when approximating meromorphic functions, the rate of convergence can be greatly improved when interpolating with rational functions of the appropriate fixed denominator degree [40, 42]. Furthermore, in recent years, spectral collocation methods for differential equations, in which an approximation of the solution is constructed by an interpolant that satisfies the equation exactly on a grid of points, have been shown to be improved by considering rational instead of polynomial interpolation [3, 47].

The advantages of rational over polynomial approximation come at the cost of increasing the complexity of the problem, since the construction of the former, in contrast with that of the latter, is nonlinear. Particular issues such as the existence of a solution, poles that may arise in the interpolation interval, or degeneracy of the rational function, are well known to affect the Cauchy problem [54]. All these difficulties must be overcome by an algorithm designed to compute rational interpolants, and there is a substantial literature on this subject. We will not attempt to present a survey of the methods which have been proposed in the past, but refer the interested reader to the expositions in [1, Sec. 7.1] and [33] and the references therein.

In this paper we present a novel solution of Cauchy's interpolation problem. We are driven by the simplicity and robustness that can be achieved for the polynomial case and we consider that a similar reliability would be desired when using rational functions. The algorithm is simple and useful for practical experimentation. Our motivation to study rational interpolation comes from our work in *chebfun*, a system in Matlab for numerical computations based on polynomial interpolation on Chebyshev points [50]. The algorithms we present in this paper are particularly easy to implement in this system and have been included since version 3 in the command `ratinterp`.

A fundamental aspect of our method is that the basis of $\mathcal{P}(N)$ used to represent p and q depends on the grid, and the main idea is the following. First, construct the matrix $\hat{Z} \in \mathbb{C}^{(N+1) \times (N+1)}$ that transforms the coefficients $\hat{\beta} \in \mathbb{C}^{N+1}$ of an arbitrary polynomial $\hat{q} \in \mathcal{P}(N)$ (in a basis to be specified) into the coefficients $\hat{\alpha} \in \mathbb{C}^{N+1}$ of a polynomial $\hat{p} \in \mathcal{P}(N)$ such that $\hat{p}(x_i) = \hat{q}(x_j)f_j$. Second, constrain \hat{Z} to obtain a linear system Z such that the associated numerator and denominator polynomials, p and q , are in $\mathcal{P}(m)$ and $\mathcal{P}(n)$ respectively. Notice that the resulting polynomials satisfy the conditions

$$(1.2) \quad p(x_j) = f_j q(x_j), \quad \text{for } j = 0, \dots, N.$$

If q is such that $q(x_j) \neq 0$, $j = 0, \dots, N$, the function $r = p/q \in \mathcal{R}(m, m)$ satisfies the original conditions (1.1).

In Section 2 we introduce our method for rational interpolation in roots of unity. In this case the method is particularly simple and can be efficiently implemented using the FFT algorithm. In Section 3 we present our solution for rational interpolation on a general grid, taking particular interest in the cases of Chebyshev and equidistant points.

The connection between conditions (1.1) and (1.2) is presented in Section 4, where we discuss aspects of rational interpolation such as unattainable points and common factors. The arise of artificial common factors in p and q is explained by the computation in finite precision of the singular values of Z . Additionally we present a strategy to produce rational interpolants with denominators of reduced degree.

The Cauchy interpolation problem has been studied for almost two centuries and it has a rich history. Our method has a connection with the classic algorithm by Jacobi [24] and in Section 5 we present some of the many relations it has with other works, such as the barycentric rational interpolation method proposed by Schneider and Werner [46] and continued by Berrut and Mittelmann [4], and the extension of Jacobi's algorithm developed by Egecioğlu and Koç [10] based on orthogonal polynomials. In Section 6 we present a discussion of our method and suggestions for further work.

A word of notation: If A is a matrix of size $(N+1) \times (N+1)$, we denote by $A_{j:k}$, $j < k$, the submatrix of size $(N+1) \times (k-j+1)$ formed with the columns j to k of A . A^* denotes the conjugate transpose of A and \bar{z} the conjugate of $z \in \mathbb{C}$. When using the notation $A_{j:k}^\top$ (or $A_{j:k}^*$), we refer to the transpose (or its conjugate) of $A_{j:k}$. A'' denotes the matrix that is the same as A except that the top-left and bottom-right entries of A are halved. A sum with the first and last terms halved is denoted by Σ'' . Given a polynomial ϕ , we write $\phi(\mathbf{x}) = [\phi(x_0), \dots, \phi(x_N)]^\top$, and in particular $\mathbf{x}^s = [x_0^s, \dots, x_N^s]^\top$. The entrywise product of two vectors \mathbf{v} and \mathbf{w} is denoted by $\mathbf{v} \cdot \mathbf{w}$. I denotes the identity matrix, the size of which is implied by the context, and $\text{diag}(\mathbf{v})$ is a diagonal matrix with entries given by the vector \mathbf{v} . Finally, we denote by Φ the diagonal matrix of size $(N+1) \times (N+1)$ with (j, j) entry $\Phi_j = f_j$.

2 Rational interpolation in roots of unity

In this section we present our solution of the rational interpolation problem when the grid is $\mathbf{z} = [z_0, \dots, z_N]^\top$ of $(N+1)$ -st roots of unity, that is,

$$z_j = \exp(2\pi i j / (N+1)), \quad j = 0, \dots, N.$$

The derivation for this particular case in Section 2.1 is useful to highlight the ideas of the method for arbitrary grids that will be presented in Section 3. An implementation using FFTs and the barycentric interpolation formula is presented in Section 2.2. See [25] for an application, where a method based on rational interpolation in roots of unity for the location of zeros of analytic functions is presented.

2.1 Derivation of the method

We begin by presenting the construction of a matrix \hat{Z} that maps the coefficients $\hat{\beta} = [\hat{\beta}_0, \dots, \hat{\beta}_N]^\top$ of $\hat{q} \in \mathcal{P}(N)$, expressed as

$$\hat{q}(z) = \sum_{j=0}^N \hat{\beta}_j z^j,$$

into the coefficients $\hat{\alpha} = [\hat{\alpha}_0, \dots, \hat{\alpha}_N]^\top$ of $\hat{p} \in \mathcal{P}(N)$, expressed as

$$\hat{p}(z) = \sum_{j=0}^N \hat{\alpha}_j z^j,$$

the values of which at \mathbf{z} are given by the entrywise product of \mathbf{f} and $\hat{\mathbf{q}}$, that is

$$[\hat{p}(z_0), \dots, \hat{p}(z_N)]^\top = [f_0 \hat{q}(z_0), \dots, f_N \hat{q}(z_N)]^\top,$$

or $\hat{\mathbf{p}} = \mathbf{f} \cdot \hat{\mathbf{q}}$, where $\hat{\mathbf{p}} = \hat{p}(\mathbf{z})$ and $\hat{\mathbf{q}} = \hat{q}(\mathbf{z})$ in this section.

Consider the Vandermonde matrix $A = [\mathbf{z}^0 | \dots | \mathbf{z}^N] \in \mathbb{C}^{(N+1) \times (N+1)}$ that transforms monomial coefficients of a polynomial into its values at roots of unity, e.g.

$$A\hat{\alpha} = \hat{\mathbf{p}}, \quad \text{and} \quad A\hat{\beta} = \hat{\mathbf{q}}.$$

Since monomials satisfy the discrete orthogonality property on roots of unity given by

$$(2.1) \quad \sum_{s=0}^N z_s^j \bar{z}_s^k = \sum_{s=0}^N \exp(2\pi i s(j-k)) = \begin{cases} N+1, & \text{if } j=k, \\ 0, & \text{if } j \neq k, \end{cases}$$

it follows that $A^*A = (N+1)I$, and A^* maps values at roots of unity of a polynomial into its (scaled) monomial coefficients. It follows that

$$(A^* \Phi A) \hat{\beta} = A^* \Phi \hat{\mathbf{q}} = A^* (\mathbf{f} \cdot \hat{\mathbf{q}}) = A^* \hat{\mathbf{p}} = A^* A \hat{\alpha} = (N+1) \hat{\alpha}.$$

Thus, the matrix

$$(2.2) \quad \hat{Z}_{\mathbf{f}} = \hat{Z} := A^* \Phi A,$$

transforms the coefficients $\hat{\beta}$ into the values $\hat{\mathbf{q}}$, computes $\hat{\mathbf{q}} \cdot \mathbf{f}$, and transforms these values back to the coefficients $\hat{\alpha}$ times a constant factor. Notice that \hat{Z} depends on \mathbf{f} but we drop the subscript to simplify the notation. A schematic representation of \hat{Z} is

$$(2.3) \quad \hat{Z} = \begin{bmatrix} \frac{(\mathbf{z}^0)^*}{(\mathbf{z}^N)^*} \\ \vdots \\ \frac{(\mathbf{z}^0)^*}{(\mathbf{z}^N)^*} \end{bmatrix} \begin{bmatrix} f_0 & & \\ & \ddots & \\ & & f_N \end{bmatrix} \begin{bmatrix} \left| \begin{array}{c} \mathbf{z}^0 \\ \vdots \\ \mathbf{z}^N \end{array} \right| \end{bmatrix}.$$

If $\hat{q}(x_j) \neq 0$, $j = 0, \dots, N$, then the rational function $\hat{r} = \hat{p}/\hat{q}$ satisfies the interpolation condition (1.1) for \mathbf{z} and \mathbf{f} . However, \hat{r} is in general a function of type $[N/N]$ and it is necessary to constrain the numerator and denominator to be polynomials in $\mathcal{P}(m)$ and $\mathcal{P}(n)$ respectively.

To constrain the degree of the denominator, we restrict the row space of \hat{Z} to \mathbb{C}^{n+1} by considering coefficient vectors of the form

$$(2.4) \quad \hat{\beta} = [\beta_0, \dots, \beta_n, \underbrace{0, \dots, 0}_{m \text{ zeros}}]^\top,$$

in which case it follows from (2.3) that

$$(2.5) \quad (N+1)\hat{\boldsymbol{\alpha}} = \hat{Z}\hat{\boldsymbol{\beta}} = A^*\Phi \left[\begin{array}{c|c|c} \mathbf{z}^0 & \cdots & \mathbf{z}^n \end{array} \right] \begin{bmatrix} \beta_0 \\ \vdots \\ \beta_n \end{bmatrix}.$$

Hence, the last m columns of the rightmost matrix A in (2.2) can be removed when computing the first $n+1$ coefficients of q . To constrain the degree of the numerator, we need to obtain conditions that guarantee that its coefficients are of the form

$$\hat{\boldsymbol{\alpha}} = [\alpha_0, \dots, \alpha_m, \underbrace{0, \dots, 0}_{n \text{ zeros}}]^\top.$$

For such $\hat{\boldsymbol{\alpha}}$, an arbitrary vector $\hat{\boldsymbol{\beta}}$ satisfies $\hat{Z}\hat{\boldsymbol{\beta}} = (N+1)\hat{\boldsymbol{\alpha}}$ if and only if it also satisfies the reduced system

$$(2.6) \quad \left[\begin{array}{c} (\mathbf{z}^{m+1})^* \\ \vdots \\ (\mathbf{z}^N)^* \end{array} \right] \Phi A \hat{\boldsymbol{\beta}} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Thus, to restrict the degree of the numerator, we need to keep only the last n rows of the leftmost matrix A^* in the right-hand side product of (2.2).

Combining (2.5) and (2.6) we obtain the matrix

$$(2.7) \quad Z = A_{m+1:N}^* \Phi A_{0:n},$$

and it follows that a vector $\boldsymbol{\beta} = [\beta_0, \dots, \beta_n]^\top$ such that

$$Z\boldsymbol{\beta} = 0,$$

i.e. $\boldsymbol{\beta}$ is in the kernel of Z , and a vector $\boldsymbol{\alpha} = [\alpha_0, \dots, \alpha_m]^\top$ given by $A_{0:m}^* \Phi A_{0:n} \boldsymbol{\beta} = (N+1)\boldsymbol{\alpha}$, determine the polynomials $p = \sum_{j=0}^m \alpha_j z^j$ and $q = \sum_{j=0}^n \beta_j z^j$ that satisfy (1.2) for a grid of roots of unity.

2.2 Implementation with FFTs and barycentric formula

The matrix-vector multiplication $A^*\mathbf{x}$ computes $\tilde{\mathbf{x}} = [\tilde{x}_0, \dots, \tilde{x}_N]^\top$, the Discrete Fourier Transform (DFT) of \mathbf{x} , that is

$$\tilde{x}_k = \sum_{j=0}^N \zeta^{kj} x_j,$$

where $\zeta = \exp(2\pi i/(N+1))$. In particular, A^* is known as the $(N+1)$ -point DFT matrix [51]. This matrix-vector product can be quickly computed by the fast Fourier transform (FFT) in $\mathcal{O}(N \log N)$ operations instead of the usual $\mathcal{O}(N^2)$. Similarly, the

matrix-vector product $A\tilde{\mathbf{x}}$ produces \mathbf{x} , the inverse DFT of the vector $\tilde{\mathbf{x}}$, and it can be efficiently computed by the inverse fast Fourier transform.

Since

$$\hat{Z} = A^* \Phi A = A^* (A^* (\Phi^*))^*,$$

we can obtain the matrix \hat{Z} by forming the matrix D the columns of which are the FFT of each column of Φ^* , and then computing the FFT of D^* . The matrix Z is computed in the same way but requires the deletion of the last m rows of D and of the first $m+1$ rows of the FFT of D^* . In Matlab, if \mathbf{fk} is a vector of length $N+1$ with the values to be interpolated at roots of unity, the following three lines compute the matrix Z :

```
>> D = fft( diag(conj(fk)) );
>> DD = fft( D(1:n+1,:)') ;
>> Z = DD(m+2:N+1,:);
```

From an element β in the kernel of Z , obtained for example from the Singular Value Decomposition [49], we obtain the $m+1$ coefficients α of the numerator p by computing $A_{0:m}^* \Phi A_{0:n} \beta = \alpha$. The matrix-vector product $A_{0:n} \beta = \mathbf{q}$ computes the values of the denominator polynomial q in roots of unity, i.e. $\mathbf{q} = q(\mathbf{z})$. Since

$$A_{0:n} \beta = A \hat{\beta},$$

where $\hat{\beta}$ is given by (2.4), we can compute \mathbf{q} more efficiently by taking the $(N+1)$ -point inverse FFT of β . The vector α corresponds to the first $m+1$ entries of the FFT of $\mathbf{f} \cdot \mathbf{q}$ (the rest of the entries, by construction, are zero).

However, for the evaluation of a rational interpolant \hat{r} at a single point x , a method other than explicitly evaluating the quotient $\hat{p}(x)/\hat{q}(x)$ is preferable: The *rational barycentric formula*

$$(2.8) \quad \hat{r}(x) = \frac{\sum_{j=0}^N \frac{u_j}{x - x_j} f_j}{\sum_{j=0}^N \frac{u_j}{x - x_j}},$$

where $u_j := w_j \hat{q}(x_j)$, with barycentric weights w_j defined as $w_j = 1/\ell'(x_j)$ with $\ell(x) = \prod_{j=0}^N (x - x_j)$, or equivalently,

$$(2.9) \quad w_j = \left[\prod_{i \neq j}^N (x_i - x_j) \right]^{-1}, \quad j = 0, \dots, N.$$

For rational interpolation in \mathbf{z} the barycentric weights are known explicitly [22]:

$$(2.10) \quad w_j = z_j.$$

If $\hat{q}(\mathbf{x})$ is a constant vector then (2.8) collapses to the barycentric formula for polynomial interpolation (see [5] for a recent survey). The barycentric formula is numerically

stable, as shown by Rack and Reimer [38] and Higham [23] for the polynomial case and by Salazar Celis for the rational case [43].

For an arbitrary vector $\hat{q}(\mathbf{x}) \in \mathbb{C}^{N+1}$ with entries all different from zero, (2.8) defines a function \hat{r} in $\mathcal{R}(N, N)$ that interpolates \mathbf{f} in \mathbf{x} . The underlying polynomials \hat{p} and \hat{q} in $\mathcal{P}(N)$ for the numerator and denominator are made evident when writing them as Lagrange interpolants of the values $\hat{q}(\mathbf{x}) \cdot \mathbf{f}$ and $\hat{q}(\mathbf{x})$ respectively:

$$(2.11) \quad \hat{p}(x) = \ell(x) \sum_{j=0}^N \frac{w_j \hat{q}(x_j)}{x - x_j} f_j, \quad \hat{q}(x) = \ell(x) \sum_{j=0}^N \frac{w_j \hat{q}(x_j)}{x - x_j}.$$

It is evident from (2.11) that \hat{r} always interpolates \mathbf{f} exactly in \mathbf{x} . When $x = x_i$ the evaluation of r should be f_i instead of attempting to compute the value from formula (2.8), which is not defined in that case. Conversely, using the same argument it follows that any interpolant of type $[N/N]$ can be written in the barycentric form (2.8).

The arbitrary choice of $\hat{\mathbf{q}}$ gives $N + 1$ degrees of freedom that can be used to impose additional characteristics on the rational interpolant \hat{r} . For example, for the Cauchy interpolation problem we have seen that these values are chosen in such a way that the rational function is of type $[m/n]$, as noticed for the first time by Schneider and Werner [46]. Another example is the interpolant of Floater and Hormann [13] for which they impose conditions that guarantee that $\hat{r} \in \mathcal{R}(N, N)$ is free of poles while keeping quadratic or higher rates of convergence as $N \rightarrow \infty$. For a survey of rational barycentric interpolation, see [3].

The code in Figure 1 presents a Matlab implementation of the ideas described so far. To obtain an element of the kernel of Z we compute its last right singular vector obtained from the Matlab command `svd`, which takes no more than a couple of seconds in our workstations even when n is in the hundreds. The singular values of Z provide information of the numerical aspects of the rational interpolation problem, and we present these observations in Section 4.

The only command that is not part of standard Matlab is `bary`, which is included in the `chebfun` system to evaluate the barycentric formula with arbitrary weights. In general, `r_x = bary(x,fvals,xk,uk)` interpolates the values `fvals` at nodes `xk` in the point `x` using the weights `uk`.

The following lines compute the interpolant in $\mathcal{R}(45, 4)$ of the function $f(z) = \log(2 - z)\sqrt{z + 2}/(1 - 16z^4)$ in 50 roots of unity and evaluate it on a grid of 200 points. As expected, the rational interpolant is a good approximation of the function in the unit disk [42]:

```
>> fh = @(z) log(2-z).*sqrt(z+2)./(1-16*z.^4);
>> tic, rh = ratinterproots(fh,50,4); toc
Elapsed time is 0.000590 seconds.
>> zz = exp(1i*linspace(0,2*pi,200));
>> error_zz = max(abs(fh(zz)-rh(zz)))
error_zz =
    1.792609524364659e-16
```

```

function rh = ratinterproots(fh,m,n)

N = m+n;                                % N+1 = # of nodes
z = exp(2i*pi*(0:N)/(N+1)).';           % roots of unity
f = fh(z);                               % function values
D = fft(diag(conj(f)));                  % compute A'(Phi')
Z = fft(D(1:n+1,:))';                   % compute A'(A'(Phi'))'
[u,s,v] = svd(Z(m+2:N+1,:));             % svd of syst w size nx(n+1)
q = ifft(v(:,end),N+1);                  % values of q at roots of unity
rh = @(x) bary(x,f,z,q.*z);              % rat. bary. interpolation

```

Figure 1: Matlab code for the rational interpolation algorithm in roots of unity. The input arguments are a function handle **fh** that specifies the values to be interpolated at **z**, and the degrees **m** and **n**. The output argument is a function handle **rh** for the evaluation of the rational interpolant of type $[m/n]$. If the vector **q** does not have zeros, the rational interpolant satisfies (1.1).

The total computation cost of our method for rational interpolation in roots of unity is $\mathcal{O}(N^2 \log N)$ operations to compute the matrix Z in (2.7), $\mathcal{O}(n^3)$ operations to obtain its kernel vector, $\mathcal{O}(N \log N)$ operations to compute the values in **q**, and $\mathcal{O}(N)$ operations to evaluate the barycentric formula (2.8).

3 Rational interpolation in other grids

In this section we present a solution to Cauchy's interpolation problem for an arbitrary grid $\mathbf{x} \in \mathbb{C}^{N+1}$. The basis that we use to represent the numerator and denominator is a family $\{\phi_j\}$, $j = 0, \dots, N$, of $N + 1$ polynomials such that the degree of ϕ_j is exactly j and the ϕ_j are orthogonal with respect to the inner product

$$(3.1) \quad \langle f, g \rangle_{\mathbf{x}} := \sum_{k=0}^N f(x_k) \overline{g(x_k)},$$

that is,

$$(3.2) \quad \langle \phi_j, \phi_k \rangle_{\mathbf{x}} = \begin{cases} \kappa_j > 0 & \text{if } j = k, \\ 0 & \text{if } j \neq k. \end{cases}$$

Defining the Vandermonde-type matrix $C = [\phi_0(\mathbf{x}) | \dots | \phi_N(\mathbf{x})] \in \mathbb{C}^{(N+1) \times (N+1)}$ and $\boldsymbol{\kappa} = [\kappa_0, \dots, \kappa_N]^\top$, (3.2) can be stated as $C^* C = \text{diag}(\boldsymbol{\kappa})$. The following theorem summarises the method presented in this paper.

Theorem 1 *If $\boldsymbol{\alpha} = [\alpha_0, \dots, \alpha_m]^\top$ and $\boldsymbol{\beta} = [\beta_0, \dots, \beta_n]^\top$ are such that*

$$\boldsymbol{\beta} \in \ker(C_{m+1:N}^* \Phi C_{0:n}) \quad \text{and} \quad C_{0:m}^* \Phi C_{0:n} \boldsymbol{\beta} = \boldsymbol{\kappa} \cdot \boldsymbol{\alpha},$$

where then the polynomials

$$p(x) = \sum_{j=0}^m \alpha_j \phi_j(x), \quad \text{and} \quad q(x) = \sum_{j=0}^n \beta_j \phi_j(x),$$

satisfy the conditions (1.2), and if $q(x_j) \neq 0$, the function $r = p/q \in \mathcal{R}(m, n)$ satisfies the conditions (1.1).

Proof Let $\hat{p}(x) = \sum_{j=0}^N \hat{\alpha}_j \phi_j(x)$ and $\hat{q}(x) = \sum_{j=0}^N \hat{\beta}_j \phi_j(x)$, such that $\hat{\mathbf{p}} = \mathbf{f} \cdot \hat{\mathbf{q}}$, where $\hat{\mathbf{p}} = \hat{p}(\mathbf{x})$ and $\hat{\mathbf{q}} = \hat{q}(\mathbf{x})$. Then it follows that

$$(C^* \Phi C) \hat{\beta} = C^* \Phi \hat{\mathbf{q}} = C^* (\mathbf{f} \cdot \hat{\mathbf{q}}) = C^* \hat{\mathbf{p}} = C^* C \hat{\alpha} = \kappa \cdot \hat{\alpha}.$$

Equations (2.4)–(2.7) in Section 2.1 apply for the matrix $\hat{Z} = C^* \Phi C$, with A replaced by C , from which the result of the theorem follows. ■

In the remaining of the paper we denote $C_{m+1:N}^* \Phi C_{0:n} \in \mathbb{C}^{n \times (n+1)}$ in Theorem 1 by Z .

We use Theorem 1 to obtain a solution of the rational interpolation problem on various specific grids. For each of them we establish the family of polynomials $\{\phi_j\}$ orthogonal with respect to (3.1). These polynomials, in most cases, can be defined, from certain initial conditions, by a three-term recurrence relation of the form

$$(3.3) \quad \phi_{k+1}(x) = a_k(x - b_k)\phi_k(x) - c_k\phi_{k-1}(x).$$

3.1 Chebyshev points

Consider the family of *Chebyshev polynomials of the first kind* $\{T_j\}$, simply referred in this paper as *Chebyshev polynomials*, defined in $[-1, 1]$ by

$$T_0(x) = 1, \quad T_1(x) = x,$$

and for higher degrees by (3.3) with

$$a_k = 2, \quad b_k = 0, \quad \text{and} \quad c_k = 1, \quad k = 1, \dots, N-1.$$

The $N+1$ *Chebyshev points of the first kind* $\mathbf{y}^{(1)} = [y_0^{(1)}, \dots, y_N^{(1)}]^\top$ are the zeros of $T_{N+1}(x)$, given by

$$y_j^{(1)} = \cos \frac{(2j+1)\pi}{2N+2}.$$

The following discrete orthogonality property is satisfied [31, Sec. 4.6.1]:

$$(3.4) \quad \langle T_j, T_k \rangle_{\mathbf{y}^{(1)}} = \begin{cases} \kappa_j & \text{if } j = k, \\ 0 & \text{if } j \neq k, \end{cases}$$

with $\kappa_0 = N+1$ and $\kappa_j = \frac{1}{2}(N+1)$, $j = 1, \dots, N$. Thus, the matrix C above can be constructed with Chebyshev polynomials and the rational interpolant in $\mathbf{y}^{(1)}$ is established from Theorem 1.

The efficient implementation of the interpolant presented in Section 2 for roots of unity, using the FFT algorithm and the rational barycentric formula, can be repeated for interpolation in Chebyshev points of the first kind. In this case, the appropriate tool to accelerate the computation of Z is the Discrete Cosine Transform (DCT), which computes the matrix-vector product $C^* \mathbf{x}$. And the barycentric weights used in formula (2.8) are given in [22] by

$$(3.5) \quad w_j = (-1)^j \sin \frac{(2j+1)\pi}{2N+2}.$$

We present in Figure 2 an implementation of our method for rational interpolation in Chebyshev points of the first kind. This code is essentially the same as the one in Figure 1 for roots of unity, except for the explicit computation of the barycentric weights for the grid of Chebyshev points and the use of the DCT¹ instead of the FFT.

```
function rh = ratinterpcheb1(fh,m,n)

N = m+n;                                % N+1 = # of nodes
y1 = cos((2*(0:N)+1)*pi/(2*N+2))';      % Cheb pts of 1st kind
f = fh(y1);                             % function values
D = dct(diag(f));                       % compute C'(Phi')
Z = dct(D(1:n+1,:))';                  % compute C'(C'(Phi'))'
[u,s,v] = svd(Z(m+2:N+1,:));           % svd of syst w size nx(n+1)
q = idct(v(:,end),N+1);                 % values of q at Cheb pts
w = (-1).^(0:N)' .* sin((2*(0:N)+1)*pi/(2*N+2))'; % barycentric weights
rh = @(x) bary(x,f,y1,q.*w);           % rat. bary. interpolation
```

Figure 2: Matlab code for the rational interpolation algorithm for Chebyshev points of the first kind. Input and output arguments are similar to those in Figure 1.

The following lines compute an interpolant in $\mathcal{R}(12,12)$ of the function $f(x) = (1.5 - \cos(5x))^{-1}$ in 25 Chebyshev points of the first kind. The error of the rational interpolant in 200 equispaced points is of order 10^{-15} :

¹The Matlab command `dct` is only available in the Signal Processing Toolbox but it can be replaced by the following four lines (where x and y are the input and output vectors respectively):

```
n = size(x,1); w = (exp(-1i*(0:n-1)*pi/(2*n))/sqrt(2*n)).'; w(1) = w(1)/sqrt(2);
if mod(n,2) == 1 || isreal(x), y = fft([x;x(n:-1:1,:)]); y = diag(w)*y(1:n,:);
else y = fft([x(1:2:n,:); x(n:-2:2,:)]); y = diag(2*w)*y; end;
if isreal(x), y = real(y); end;
```

Similarly, the `idct` command for the inverse DCT can be replaced by the following five lines (where y and x are the input and output vectors respectively):

```
n = size(y,1); w = (exp(1i*(0:n-1)*pi/(2*n)) * sqrt(2*n)).';
if mod(n,2) == 1 || isreal(y), w(1) = w(1)*sqrt(2);
x = ifft([diag(w)*y; zeros(1,size(y,2)); -1i*diag(w(2:n))*y(n:-1:2,:)]);
x = x(1:n,:); else w(1) = w(1)/sqrt(2); x([1:2:n,n:-2:2],:) = ifft(diag(w)*y);
end; if isreal(y), x = real(x); end;
```

```
>> fh = @(x) 1./(1.5-cos(5*x));
>> tic, rh = ratinterpcheb1(fh,12,12); toc
Elapsed time is 0.000853 seconds.
>> xx = linspace(-1,1,200);
>> error_xx = max(abs(fh(xx)-rh(xx)))
error_xx =
    1.332267629550188e-15
```

Also of importance are the $N + 1$ *Chebyshev points of the second kind* $\mathbf{y}^{(2)} = [y_0^{(2)}, \dots, y_N^{(2)}]^\top$, which are the local extrema in $[-1, 1]$ of $T_N(x)$, given by

$$y_j^{(2)} = \cos \frac{j\pi}{N}, \quad j = 0, \dots, N.$$

Orthogonal polynomials with respect to $\langle \cdot, \cdot \rangle_{\mathbf{y}^{(2)}}$ have been recently derived by Eisenberg and Fedele [12]. They are

$$\phi_0(x) = N - 1, \quad \phi_1(x) = Nx, \quad \text{and} \quad \phi_2(x) = (N + 1)x^2 - \frac{N + 2}{2},$$

and higher degree polynomials are defined by the three-term recurrence (3.3) with coefficients given by

$$a_k = \frac{N + k}{N + k - 1}, \quad b_k = 0, \quad \text{and} \quad c_k = \frac{N + k + 1}{4(N + k - 1)},$$

for $k = 2, \dots, N - 1$. For $\langle \cdot, \cdot \rangle_{\mathbf{y}^{(2)}}$, they satisfy a similar condition as (3.4), with the values $\kappa_j = \langle \phi_j, \phi_j \rangle_{\mathbf{y}^{(2)}}$ given exactly by

$$\kappa_j = \begin{cases} (N + 1)(N - 1)^2, & \text{if } j = 0, \\ \frac{N(N + j + 1)(N + j - 1)}{2^{2j-1}}, & \text{if } 0 < j < N, \\ \frac{N^2(2N - 1)}{2^{2N-3}}, & \text{if } j = N. \end{cases}$$

It follows from Theorem 1 that a rational interpolant for $\mathbf{y}^{(2)}$ can be constructed from these polynomials. The method, however, can be slightly modified in this case to make it simpler by using Chebyshev polynomials instead. In particular, Chebyshev polynomials are orthogonal with respect to the inner product [31, Sec. 4.6.1]

$$(3.6) \quad (f, g)_{\mathbf{y}^{(2)}} = \sum_{k=0}^N {}'' f(y_k^{(2)}) \overline{g(y_k^{(2)})},$$

that is, similarly to (3.4), they satisfy

$$(3.7) \quad (T_j, T_k)_{\mathbf{y}^{(2)}} = \begin{cases} \kappa_j & \text{if } j = k, \\ 0 & \text{if } j \neq k, \end{cases}$$

with $\kappa_0 = \kappa_N = N$ and $\kappa_j = \frac{1}{2}N$, $j = 1, \dots, N-1$, i.e. if C is defined by Chebyshev polynomials evaluated on $\mathbf{y}^{(2)}$, we have that $C^\top I'' C = \text{diag}(\boldsymbol{\kappa})$.

If $\hat{p}(x) = \sum_{j=0}^N \hat{\alpha}_j T_j(x)$ and $\hat{q}(x) = \sum_{j=0}^N \hat{\beta}_j T_j(x)$ such that $\hat{\mathbf{p}} = \hat{\mathbf{q}} \cdot \mathbf{f}$, where $\hat{\mathbf{p}} = \hat{p}(\mathbf{y}^{(2)})$ and $\hat{\mathbf{q}} = \hat{q}(\mathbf{y}^{(2)})$, then, similarly as in the proof of Theorem 1, we have that

$$(3.8) \quad (C^\top \Phi'' C) \hat{\boldsymbol{\beta}} = C^\top \Phi'' \hat{\mathbf{q}} = C^\top I'' (\mathbf{f} \cdot \hat{\mathbf{q}}) = C^\top I'' \hat{\mathbf{p}} = C^\top I'' C \hat{\boldsymbol{\alpha}} = \boldsymbol{\kappa} \cdot \hat{\boldsymbol{\alpha}}.$$

Notice that an alternative derivation of (3.8) can be obtained by applying the Lobatto–Markov quadrature rule to the integral formula $\hat{\alpha}_j = \int_{-1}^1 \hat{p}(x)(1-x^2)^{-1} dx$ of the coefficient $\hat{\alpha}_j$ of the polynomial \hat{p} [39, p. 151].

Following the same reasoning as in Theorem 1, the coefficients $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ of p and q are given by

$$\boldsymbol{\beta} \in \ker(C_{m+1:N}^\top \Phi'' C_{0:n}) \quad \text{and} \quad C_{0:m}^\top \Phi'' C_{0:n} \boldsymbol{\beta} = \boldsymbol{\kappa} \cdot \boldsymbol{\alpha},$$

For the implementation with the barycentric formula (2.8), the weights for $\mathbf{y}^{(2)}$ are given by [44]

$$(3.9) \quad w_j = (-1)^j \delta_j, \quad \delta_j = \begin{cases} 1/2, & j = 0 \text{ or } j = N, \\ 1, & \text{otherwise.} \end{cases}$$

In Figure 3 we present a code for rational interpolation in Chebyshev points of the second kind using the modified method. In this case we explicitly form the Vandermonde-type matrix C by the three-term recurrence of Chebyshev polynomials.

```
function rh = ratinterpcheb2(fh,m,n)

N = m+n;                                % N+1 = # of nodes
y2 = cos((0:N)*pi/N)';                  % Cheb pts of 2nd kind
f = fh(y2);                              % function values
C(:,1) = ones(N+1,1); C(:,2) = y2;      % Vandermonde-type matrix
for k = 2:N,
    C(:,k+1) = 2*y2.*C(:,k) - C(:,k-1); % 3-term recurrence
end
half = [1/2; ones(N-1,1); 1/2];
Z = C(:,m+2:N+1) .* diag(half.*f) * C(:,1:n+1); % modified matrix Z
[u,s,v] = svd(Z);                        % svd of syst w size nx(n+1)
q = C(:,1:n+1) * v(:,end);              % values of q at Cheb pts
w = half.*(-1).^((0:N)');               % barycentric weights
rh = @(x) bary(x,f,y2,q.*w);             % rat. bary. interpolation
```

Figure 3: Matlab code of the rational interpolation algorithm for Chebyshev points of the second kind. Input and output arguments are similar to the ones in Figure 1.

3.2 Equidistant points

For the set of $N + 1$ equidistant nodes in $[-1, 1]$, i.e. $\mathbf{t} = [t_0, \dots, t_N]^\top$ with

$$t_j = -1 + \frac{2j}{N}, \quad j = 0, \dots, N,$$

the family $\{G_j\}$ of orthogonal polynomials with respect to $\langle \cdot, \cdot \rangle_{\mathbf{t}}$, required in Theorem 1, are the *Gram polynomials*

$$G_{-1} = 0, \quad G_0(x) = (N + 1)^{-1/2},$$

and higher degrees polynomials defined by the three-term recurrence relation (3.3) with coefficients

$$a_k = \frac{N}{k + 1} \left(\frac{4(k + 1)^2 - 1}{(N + 1)^2 - (k + 1)^2} \right)^{1/2}, \quad b_k = 0,$$

for $k = 0, \dots, N - 1$, and $c_k = a_k/a_{k-1}$ for $k = 1, \dots, N - 1$ (for more on Gram polynomials see [8, Sec. 4.4.4] and [11]). Specifically,

$$(G_j, G_k)_{\mathbf{t}} = \begin{cases} (N + 1)^2 & \text{if } j = k, \\ 0 & \text{if } j \neq k. \end{cases}$$

For the implementation with the formula (2.8), the barycentric weights are given by

$$(3.10) \quad w_j = (-1)^j \binom{N}{j}.$$

Two difficulties arise for interpolation in equidistant points using the method presented in this paper. First when N is large, Gram polynomials of degrees close to N have exponentially large oscillations near the endpoints, making them unsuitable for approximation of functions [2]. This introduces large numerical errors in the computation of the product $C_{m+1:N}^* \Phi C_{0:n}$ in Theorem 1, affecting the accuracy of the coefficients vector β . Second, the weights (3.10) near the origin and near the endpoints vary by exponentially large factors. Thus, even if the computation of the values $q(t_j)$ were accurate, the interpolation with formula (2.8) would be ill-conditioned.

These difficulties are not surprising though, since it is known that the equispaced grid is a poor choice for the interpolation problem. A recent result by Platte, Trefethen and Kuijlaars, for example, shows that a stable algorithm for (linear or nonlinear) approximation from equispaced data cannot converge geometrically [35].

3.3 Arbitrary Grids

For an arbitrary grid \mathbf{x} for which a family of orthogonal polynomials with respect to $\langle \cdot, \cdot \rangle_{\mathbf{x}}$ is not known explicitly, it is possible to obtain the associated matrix C by a numerical method [16, Chp. 2].

If $\langle \cdot, \cdot \rangle_{\mathbf{x}}$ satisfies the shift property $\langle xf, g \rangle_{\mathbf{x}} = \langle f, xg \rangle_{\mathbf{x}}$, then it follows that a three-term recurrence as (3.3) can be established [16, Sec. 1.3], and the coefficients can be computed, for example, by the Stieltjes method [16, Sec. 2.2.3.1], where

$$a_k = 1, \quad b_k = \frac{\langle x\phi_k, \phi_k \rangle_{\mathbf{x}}}{\langle \phi_k, \phi_k \rangle_{\mathbf{x}}} \quad \text{and} \quad c_k = \frac{\langle \phi_k, \phi_k \rangle_{\mathbf{x}}}{\langle \phi_{k-1}, \phi_{k-1} \rangle_{\mathbf{x}}},$$

with initial condition $\phi_{-1}(x) = 0$ and $\phi_0(x) = 1$, or by the orthogonal reduction method based on Lanczos algorithm [16, Sec. 2.2.3.2]. See [14, Sec. 6] for a discussion on both methods, and the Matlab suite OPQ, written by Gautschi, for their implementation in the commands `stieltjes.m` and `lanczos.m` [15].

If the shift property is not satisfied, for example for grids in the complex plane that do not include all the conjugates of the grid points, the matrix C in Theorem 1 can be obtained by computing the QR decomposition of a $(N+1) \times (N+1)$ Vandermonde-type matrix, the columns of which are polynomials of an arbitrary basis evaluated on \mathbf{x} [49]. Notice, however, that the number of operations of the QR decomposition is of order $\mathcal{O}(N^3)$.

4 Common factors of the rational interpolant

Theorem 1 establishes the conditions required to obtain $p \in \mathcal{P}(m)$ and $q \in \mathcal{P}(n)$ that satisfy (1.2). The following theorem is the main result that connects these polynomials with the solution of (1.1), the original rational interpolation problem. This result is due to Maehly and Witzgall [29] and can also be found, for example, in [54] and [21].

Theorem 2 *Let $p \in \mathcal{P}(m)$ and $q \in \mathcal{P}(n)$ satisfy equations (1.2). Then, p and q are of the form*

$$\begin{aligned} p(x) &= \bar{p}(x)s(x)v(x), \\ q(x) &= \bar{q}(x)s(x)v(x), \end{aligned}$$

where \bar{p} and \bar{q} are relatively prime polynomials, unique up to normalisation, and of exact degrees \bar{m} and \bar{n} , $0 \leq \bar{m} \leq m$ and $0 \leq \bar{n} \leq n$; $s(x)$ is a monic polynomial of degree δs , where $0 \leq \delta s \leq \delta := \min(m - \bar{m}, n - \bar{n})$, and its zeros are the interpolation points x_j such that $q(x_j) = 0$; and $v(x)$ is an arbitrary polynomial of degree $\delta v = \delta - \delta s$.

Moreover, we have that

$$\text{rank}(Z) = n - \delta + \delta s.$$

Proof The first part of the Theorem follows immediately from the definitions of \bar{p} , \bar{q} and s . The rank of Z follows from the observation that v is arbitrary, hence the nullity of Z is $\delta v + 1 = \delta - \delta s + 1$. ■

The value δ is known as the *defect* and the rational function p/q is *nondegenerate* if $\delta = 0$. Suppose that p and q satisfy (1.2). If none of the values $q(x_j)$ is zero, then we can define the rational function $r = p/q$ that interpolates \mathbf{f} in the grid \mathbf{x} . However, if $q(x_j) = 0$ for some j , i.e. x_j is a zero of s , then $p(x_j) = 0$ and both p and q have

the common term $(x - x_j)$ that can be cancelled out. Thus, in this case the rational function r in its reduced form may or may not satisfy the interpolation condition, i.e. $\bar{p}(x_j)/\bar{q}(x_j) = f_j$ may not hold for a certain x_j . Grid points where the interpolation conditions (1.1) cannot be satisfied are called *unattainable points* and their presence indicates the nonexistence of a solution to Cauchy's interpolation problem. If there exists a solution to Cauchy's interpolation problem, then it is unique [54].

Notice that $q(x_j) = 0$ is a necessary but not sufficient condition for a point x_j to be unattainable. Unattainable points can of course be detected *a posteriori* simply by evaluating the rational function at interpolation points where q is zero.

Although it is trivial to construct artificial examples of rational interpolants with unattainable points, they seem to appear rarely in practical examples. A more frequent source of difficulties is the polynomial v which collects the common factors of p and q other than those due to s . From Theorem 2 it follows that δv can be detected by looking at the rank (or the nullity) of this matrix. In practice, however, it may happen that Z is numerically not of full rank while there are nevertheless no common factors in p and q . In infinite precision Z would be of full rank but linear independence is lost due to finite precision effects.

Consider for example the function [4]

$$(4.1) \quad f = \frac{\exp((x + 1.2)^{-1})}{1 + 25x^2}.$$

Its interpolant of type [18/18] in 37 Chebyshev points of the first kind in $[-1, 1]$ does not have common factors, as can be verified with a symbolic computation software. In the top left panel of Figure 4 we plot 16 pairs of zeros of p and q computed numerically with Maple using 100 decimal digits of precision (the others are two zeros of p at infinity and two zeros of q at $\pm i/5$).

On the other hand, if we use ordinary IEEE double-precision arithmetic to compute the same rational interpolant, the Matlab command `rank` determines that the rank of Z is 9. Since $q(x_j) \neq 0$, $j = 0, \dots, 37$, then we should have that $\delta s = 0$ and $\delta = \delta v = 9$. In the top right panel of Figure 4 we plot 16 pairs of zeros of p and q computed in this way (the others are two zeros of p at approximately $\pm 1.535 \times 10^5$ and two zeros of q at $\pm 0.5i$). Clearly the zeros and poles of the rational interpolant differ completely from the true ones. Notice that there are 9 pairs of common factors, in agreement with the computed rank of Z .

The lower panels of Figure 4 can give us more insight into this phenomenon. We compute the interpolants of f in $\mathcal{R}(k, k)$, $k = 1, \dots, 18$, that is, the first 18 elements in the diagonal of the rational interpolation table [54]. Each line corresponds to the singular values of the Z for each problem, normalized with respect to the largest value (the singular values when $k = 18$ are marked with a white circle, and for $k = 1, \dots, 17$, only the smallest one is marked with a black dot).

Similarly as before, the left panel shows the singular values obtained with Maple using 100 decimal digits, and the right panel shows the singular values computed in double-precision. A striking feature of these plots is the fast decay of the singular values as we increase k . When $k = 18$, the smallest singular value is approximately 1.4105×10^{-60} .

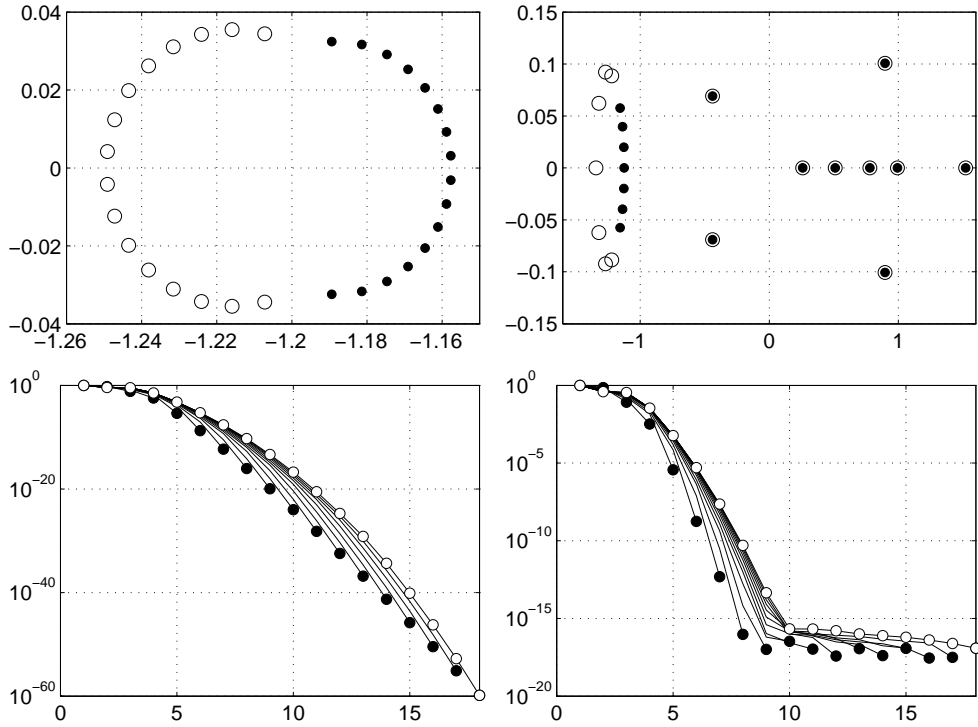


Figure 4: Rational interpolation of (4.1) in Chebyshev points of the first kind. Left panels show computations using Maple with 100 decimal digits and right panels show computations in ordinary IEEE double-precision arithmetic. Upper: Sixteen pairs of zeros of the numerator (larger white circles) and denominator (smaller black dots) of the $[18/18]$ interpolant. Two other pairs are outside the plotted region. Lower: Singular values of Z , normalised with respect the largest one, for the $[k/k]$ interpolant $k = 1, \dots, 18$. For $k = 18$ the values are marked with a white circle, and for the rest only the smallest ones are marked with black dots. The nine pairs of overlapping zeros of p and q in the upper right panel arise due to the nine singular values below machine precision in the lower right panel.

The right panel shows that when $k \geq 9$ the magnitude of the smallest singular values is in the order of machine precision $\epsilon_M \approx 2.22 \times 10^{-16}$. In particular, when $k = 18$, only the first 9 are larger than ϵ_M , which corresponds to the numerically computed rank of Z . Moreover, the remaining values introduce noise into the polynomials p and q , and since their singular values are numerically negligible, they correspond to common terms in the polynomial v .

In our experience we have seen this behaviour quite often, specifically when computing elements in the diagonal of the rational interpolation table: the singular values decrease very quickly reaching at some point machine precision, which in turn introduce artificial common terms in p and q that make the zeros and poles of the rational interpolant to deviate from the expected ones. To our knowledge, these observations have not been discussed before in the literature.

When the artificial common factors are introduced in the interpolant we can attempt to combine the (numerically) linearly independent vectors in the kernel of Z to produce a reduced denominator q , i.e. a coefficient vector β whose last entries are zero. This modifies the problem of computing an interpolant of type $[m/n]$ into one of type $[m/\nu]$, where $\nu < n$ is the largest degree of the denominator such that no artificial common factors are present.

For example, let σ_k , $k = 0, \dots, n-1$ be the singular values of Z and V the matrix of the corresponding right singular vectors. If $\sigma_k < \epsilon_M$ for $k = \nu, \dots, n-1$, the columns of $V_{\nu:n}$ correspond to a basis for the kernel of Z . Let R be the triangular matrix of the QR factorisation of $(PV_{\nu:n})^\top$, where P is the anti-diagonal matrix that flips $V_{\nu:n}$ upside-down. Then, the row spaces of $(PV_{\nu:n})^\top$ and R are the same, but the first $n - \nu$ entries of the last row of R are zero. Hence, we can use the last row of R as a vector of coefficients for a reduced q . If the null space \mathbf{V} of Z has already been computed, the following two lines compute the vector of coefficients β of the reduced q :

```
>> [Q,R] = qr(flipud( V ).' );
>> beta = flipud( R(end,:).' );
```

We use this strategy to obtain reduced denominators of rational interpolants of type $[k/k]$, $k = 1, \dots, 250$, in Chebyshev points of the first kind, for the function

$$(4.2) \quad f(x) = \frac{xg(x)}{\sinh(g(x))}, \quad x \in [-1, 1],$$

where

$$g(x) = \frac{\pi}{\omega}(x^2 - c^2), \quad \text{with } c = 0.6, \text{ and } \omega = 0.02.$$

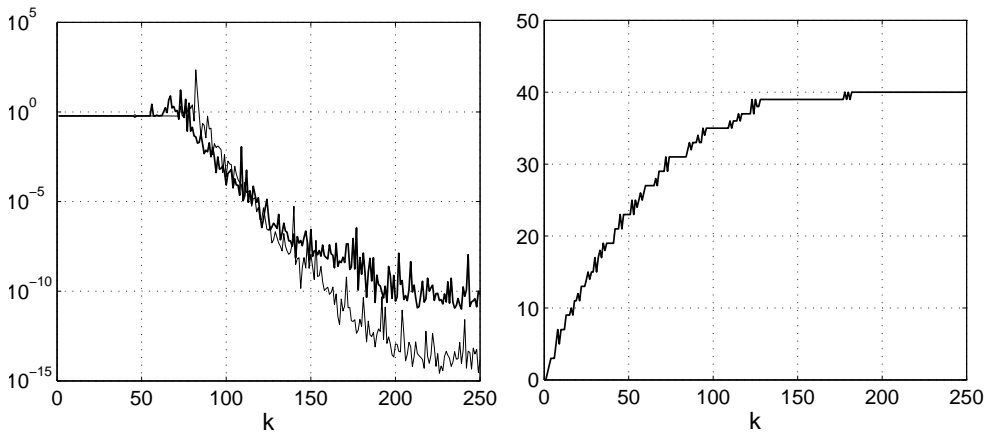


Figure 5: Rational interpolation of the function (4.2). Left: Error measured in a grid of 300 equispaced points using rational functions with full degree (thin line) and reduced degree (bold line). Right: Degree of the reduced denominator.

In the left panel of Figure 5 we plot the error of the rational interpolant as k increases. The thin line shows the error of the rational function with a denominator of full degree. The bold line shows the error when modifying the code to reduce the degree of the denominator, which is presented in the right panel of the Figure. The degree of the denominator slowly increases reaching 40 when $k = 182$, after which it remains almost unchanged.

Finally we should mention that zeros of q may appear anywhere in the complex plane. In particular, when interpolating values defined on $[-1, 1]$, it may happen that the zeros of q are located inside the interval. See Figure 6. This may prevent the use of rational interpolants in certain applications.

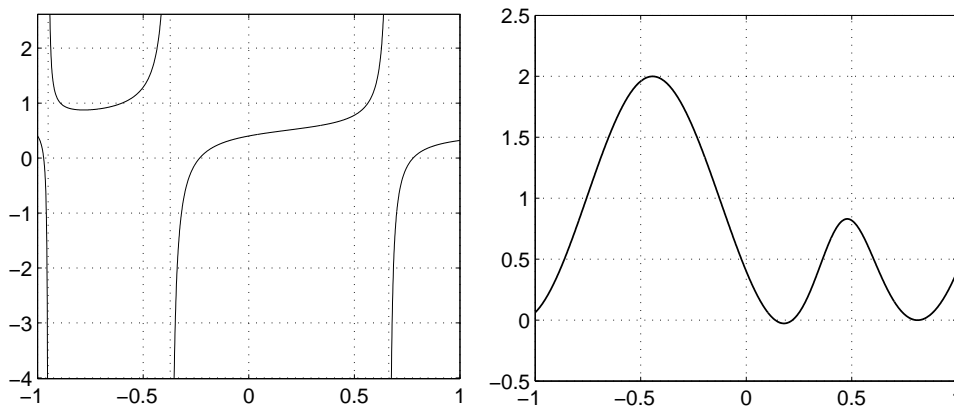


Figure 6: Rational interpolation of $f(x) = 1 - \sin(5|x - 0.5|)$. Left: Interpolant in $\mathcal{R}(3,3)$. Notice that q has zeros at -0.949409857044933 , -0.371655244598090 , and 0.663444249729421 . Right: Interpolant in $\mathcal{R}(6,6)$. In this case q does not have any zeros in the interval.

5 History and Related Work

The first reference on the rational interpolation problem studied in this paper was Cauchy's 1821 treatise on analysis for the École Royale Polytechnique [7]. In one of the appendices (*Note V. Sur la Formule de Lagrange relative à l'Interpolation*), he studies the generalisation of the Lagrange formula for interpolation to rational functions of a prescribed type. After showing that the condition $m + n = N$ guarantees the uniqueness of a solution, Cauchy derives an explicit though cumbersome formula for the rational interpolant². See [33] for more on the history of this problem.

A major contribution to the understanding of the rational interpolation problem was published by Jacobi in 1846 [24]. His solution is one of the main approaches for rational interpolation, another being the one based on continued fractions, of which

²Salzer vividly describes the obscurity of this solution [45]: “Cauchy’s formula is so rarely given, or even cited, that perhaps less than one person out of every thousand who can write down the Lagrange polynomial interpolation formula, can do the same for the former”.

we say nothing in this paper. Jacobi's approach is as follows. Consider the Newton representation of a polynomial interpolant $\varphi \in \mathcal{P}(N)$ in an arbitrary grid \mathbf{x} ,

$$\varphi(x) = \varphi[x_0] + \varphi[x_0, x_1](x - x_0) + \varphi[x_0, x_1, x_2](x - x_0)(x - x_1) + \cdots + \varphi[x_0, x_1, \dots, x_N](x - x_0)(x - x_1) \cdots (x - x_{N-1}),$$

where the coefficients are given by the divided differences

$$\varphi[x_j, x_{j+1}, \dots, x_{k-1}, x_k] = \frac{\varphi[x_{j+1}, \dots, x_k] - \varphi[x_j, \dots, x_{k-1}]}{x_k - x_j},$$

with initial condition $\varphi[x_j] = \varphi(x_j)$. In particular, it follows by induction that the coefficient of degree N has the representation

$$\varphi[x_0, x_1, \dots, x_N] = \sum_{i=0}^N w_i \varphi(x_i),$$

where w_i are the barycentric weights given by (2.9). Hence, if φ is a polynomial of degree strictly less than N , then

$$(5.1) \quad \sum_{i=0}^N w_i \varphi(x_i) = 0.$$

Suppose now that $p \in \mathcal{P}(m)$, $q \in \mathcal{P}(n)$, and $p(\mathbf{x}) = q(\mathbf{x}) \cdot \mathbf{f}$. If we replace $\varphi(x)$ in (5.1) by $p(x)\phi_j(x)$, where $\phi_j \in \mathcal{P}(j)$, $j = 0, \dots, n-1$, and express q as

$$q(x) = \sum_{k=0}^n \beta_k \psi_k(x),$$

then it follows from (5.1) that

$$\begin{aligned} 0 &= \sum_{i=0}^N w_i p(x_i) \phi_j(x_i) = \sum_{i=0}^N w_i f_i q(x_i) \phi_j(x_i) \\ &= \sum_{i=0}^N w_i f_i \phi_j(x_i) \sum_{k=0}^n \beta_k \psi_k(x_i). \end{aligned}$$

In matrix form, solving $\boldsymbol{\beta}$ in the previous set of equations is equivalent to computing the kernel of the matrix

$$(5.2) \quad Z = \begin{bmatrix} \phi_0(\mathbf{x}) \\ \vdots \\ \phi_{n-1}(\mathbf{x}) \end{bmatrix} \begin{bmatrix} f_0 w_0 & & \\ & \ddots & \\ & & f_N w_N \end{bmatrix} \begin{bmatrix} \psi_0(\mathbf{x}) & \cdots & \psi_n(\mathbf{x}) \end{bmatrix}.$$

If the same basis $\{\phi_j\}$ is used to expand q , then

$$(5.3) \quad Z = C_{0:n-1}^\top \Phi \Omega C_{0:n},$$

where Ω is the diagonal matrix of size $(N+1) \times (N+1)$ with (j, j) entry $\Omega_j = w_j$. Notice the similarity of (5.3) with the matrix Z of Theorem 1.

Consider the case in which

$$\begin{aligned} \phi_i(z) &= z^i \quad i = 0, \dots, n-1, \\ \psi_j(z) &= z^j \quad j = 0, \dots, n, \end{aligned}$$

and the grid is \mathbf{z} . From (2.10), and since $z_k^{(N-j+1)} = z_k^j$, it follows that (5.3) reduces to the same matrix Z obtained in Section 2 given by (2.7). Similarly, consider the case in which the basis in (5.3) is fixed as

$$\begin{aligned} \phi_i(x) &= T_i(x) \quad i = 0, \dots, n-1, \\ \psi_j(x) &= T_j(x) \quad j = 0, \dots, n, \end{aligned}$$

and the grid is $\mathbf{y}^{(2)}$. From (3.9), and since $T_j(y_k^{(2)}) = (-1)^k T_{N-j}(y_k^{(2)})$, it follows that (5.3) reduces to the system $C_{m+1:N}^* \Phi'' C_{0:n}$, where the columns of C are Chebyshev polynomials evaluated at $\mathbf{y}^{(2)}$. This is the same system (3.8) resulting from the “modified method” of Section 3.1 for Chebyshev points of the second kind.

Jacobi used monomials both for $\{\phi_i\}$ and $\{\psi_i\}$, however algorithms based on a system like (5.3) have been proposed and rediscovered several times in recent years. Authors that have used monomials include Kronecker [26], Werner [53], Meinguet [33], and Graves-Morris [20].

More recently, Berrut and Mittelmann [4] proposed to solve Cauchy’s interpolation problem by using monomials to define C , and directly computing $\mathbf{u} = \Omega C_{0:n} \boldsymbol{\beta} \in \mathbb{C}^{N+1}$, the weights of the rational barycentric formula (2.8), as an element in the kernel of

$$B = \begin{bmatrix} C_{0:m-1}^\top \\ C_{0:n-1}^\top \Phi \end{bmatrix} \in \mathbb{C}^{N \times (N+1)}.$$

The n conditions at the bottom of $B\mathbf{u} = 0$ are the same as those in (5.3). And the m conditions at the top are similarly derived from (5.1) by replacing $\varphi(x)$ for $q(x)\phi_j(x)$, where $\phi_j \in \mathcal{P}(j)$, $j = 0, \dots, m-1$. Reductions of the system to one of size $n \times (n+1)$ are studied in [4], [55] and [36].

Instead of monomials, Opitz [34] and Schneider and Werner [46] used the Newton basis, and Predonzan [37] and Salzer [45] used the Lagrange basis.

The choice of the basis is critical for the stability of the method. Figure 7 shows similar plots to the one in the right panel of Figure 4, with the singular values of the Z matrix (5.3) for the rational interpolation of type $[2k/2k]$, $k = 1, \dots, 25$, in Chebyshev points of the second kind for the function (4.1). For the plot in the left panel we use the monomial basis, and in the right panel we use the Chebyshev basis, i.e. the modified method of Section 3.1. Notice how for monomials, the tail of the singular values grows

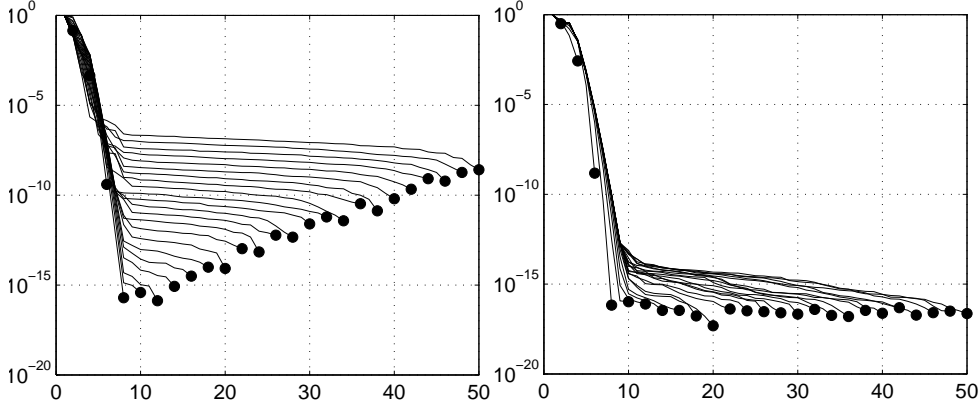


Figure 7: Normalised singular values of the Z matrix in (5.3) for the interpolants of (4.1) in $\mathbf{y}^{(2)}$ of type $[2k/2k]$, $k = 1, \dots, 25$. For each k , the smallest singular value is marked with a black dot. Left: Monomials basis. Right: Chebyshev basis (this is equivalent to the modified method for $\mathbf{y}^{(2)}$ presented in Section 3.1). If using the original method, i.e. with the polynomials of Eisinberg and Fedele, the plot obtained is indistinguishable from the one in the right panel.

as we increase k . For the Chebyshev basis, on the other hand, the first singular values seem to converge while the tail remains close to machine precision as we increase k .

Another method derived from (5.2), closer in spirit to ours, is the one proposed by Egecioğlu and Koç [10]. Instead of fixing the basis, they used a Stieltjes procedure to obtain polynomials $\phi_j \in \mathcal{P}(j)$, $j = 0, \dots, n$, orthogonal with respect to the bilinear form

$$(5.4) \quad \langle g, h \rangle_{\mathbf{f}, \mathbf{x}} = \sum_{j=0}^N f_j w_j g(x_j) h(x_j).$$

At the k th step of the Stieltjes algorithm the polynomial ϕ_k in the matrix C of (5.3) is obtained from ϕ_{k-1} and ϕ_{k-2} . Since these polynomials are orthogonal with respect to (5.4), at the end of the n th step all the entries off the diagonal of Z are zero, and it follows that $\phi_n(x)$ corresponds to the denominator polynomial $q(x)$. This method is very fast, since there is no need for explicitly assembling Z or using an SVD algorithm to compute an element in its kernel. However, since the weight of the bilinear form (5.4) depends on the values \mathbf{f} and the barycentric weights (2.8), the method is sensitive not only to the grid distribution but also to the data values. In particular a special procedure must be used if any value f_j is zero. This method has been developed further in [19].

6 Discussion

We have presented a new algorithm for the solution of Cauchy's rational interpolation problem. We established conditions that ensure that

- the interpolation condition (1.1) is satisfied, and
- the polynomials $p(x)$ and $q(x)$ are of the appropriate degrees.

The first set of conditions are represented in the linear transformation \hat{Z} that maps the coefficients of $q(x) \in \mathcal{R}(N, N)$ to the coefficients of $p(x) \in \mathcal{R}(N, N)$ such that $p(x_j) = q(x_j)f_j$, $j = 0, \dots, N$. The second are obtained by appropriately restricting the matrix \hat{Z} . Theorem 1 summarises the method presented in this paper.

Our method relies on the use of polynomials that satisfy the discrete orthogonal property (3.2). Some recent research has been focused on these polynomials, and for particular grids they are known explicitly. In Section 3.1 we modified our method for Chebyshev points of the second kind by requiring orthogonality with respect to (3.6) instead of (3.1). This suggests that the method can be modified and perhaps improved by introducing specific knowledge of polynomials orthogonal with respect to other inner products.

The barycentric formula (2.8) provides an efficient method to evaluate the rational interpolant. And for the specific cases of roots of unity and Chebyshev points of the first kind, the matrix Z can be constructed using the FFT or the DCT. Notice, however, that for certain pairs (m, n) of the degrees of p and q , the explicit computation of the product $C_{m+1:N}^* \Phi C_{0:n}$ may be faster. This could be exploited in a general code for rational interpolation, by having different ways of constructing Z depending on (m, n) .

Figures 4 and 7 hint at the relevance of the singular values of the matrix Z in understanding the rational interpolation problem. We have observed in our experiments that singular values decrease very rapidly (specifically when moving in the diagonal of the interpolation table), and when becoming negligible they introduce artificial factors in p and q that can be treated, at least numerically, as common. Another aspect that seems to be captured by the singular values of Z is the sensitivity of the solution which heavily depends on the basis used to represent p and q . Further work may look to obtain precise statements of the behaviour exhibited by the singular values discussed in this paper.

The identity (5.1) provides a family of methods to solve the rational interpolation problem by choosing different bases to represent p and q . This choice is important for numerical purposes, since it can affect the stability and speed of the method. We have found that researchers in this area are not always aware of the strong connections between their methods or of the importance of Jacobi's contribution. The algorithm that we present is not a member of this family but some connections between them were established.

An increasing number of applications of rational interpolation can be found nowadays. Some of the interests that have motivated this work are the computation of best rational approximants and the use of rational functions to approximate the singularity structure of functions in the complex plane. Additionally, we are currently exploring the generalisation of the method presented in this paper for least squares approximation with rational functions.

Acknowledgements

We wish to thank Nick Trefethen for many stimulating discussions, suggestions and comments during the preparation of this paper. We are also grateful to Jean-Paul Berrut and Rodrigo Platte. The second author was funded by the Swiss National Science Foundation Individual Support Fellowship PBEZP2-127959.

References

- [1] G. A. BAKER AND P. R. GRAVES-MORRIS, *Padé Approximants*, vol. 59 of Encyclopedia of Mathematics and its Applications, Cambridge University Press, Cambridge, UK, 1996.
- [2] R. W. BARNARD, G. DAHLQUIST, K. PEARCE, L. REICHEL, AND K. C. RICHARDS, *Gram polynomials and the Kummer function*, J. Approx. Theory, 94 (1998), pp. 128–143.
- [3] J.-P. BERRUT, R. BALTENSPERGER, AND H. MITTELMANN, *Recent developments in barycentric rational interpolation*, Internat. Ser. Numer. Math., 151 (2005), pp. 27–51.
- [4] J.-P. BERRUT AND H. MITTELMANN, *Matrices for the direct determination of the barycentric weights of rational interpolation*, J. Comput. Appl. Math., 78 (1997), pp. 355–370.
- [5] J.-P. BERRUT AND L. N. TREFETHEN, *Barycentric Lagrange interpolation*, SIAM Review, 46 (2004), pp. 501–517.
- [6] L. BRUTMAN, *Lebesgue functions for polynomial interpolation — a survey*, Ann. Numer. Math., 4 (1997), pp. 111–128.
- [7] A. L. CAUCHY, *Cours d'Analyse de l'École Royale Polytechnique. 1re partie. Analyse Algébrique*, Imprimerie Royale, Paris, 1821.
- [8] G. DAHLQUIST AND Å. BJÖRCK, *Numerical Methods*, Prentice-Hall, Englewood Cliffs, N.J., 1974.
- [9] P. J. DAVIS, *Interpolation and Approximation*, Dover Publications, New York, 1975.
- [10] Ö. EĞECIOĞLU AND Ç. K. KOÇ, *A fast algorithm for rational interpolation via orthogonal polynomials*, Math. Comp., 53 (1989), pp. 249–264.
- [11] A. EISENBERG AND G. FEDELE, *Discrete orthogonal polynomials on equidistant nodes*, International Math. Forum, 21 (2007), pp. 1007–1020.
- [12] ———, *Discrete orthogonal polynomials on Gauss–Lobatto Chebyshev nodes*, J. Approx. Theory, 144 (2007), pp. 238–246.
- [13] M. FLOATER AND K. HORMANN, *Barycentric rational interpolation with no poles and high rates of approximation*, Numer. Math., 107 (2007), pp. 315–331.
- [14] W. GAUTSCHI, *Orthogonal polynomials: Applications and computations*, Acta Numerica, 5 (1996), pp. 45–119.

- [15] ———, *OPQ: A MATLAB suite of programs for generating orthogonal polynomials and related quadrature rules*. <http://www.cs.purdue.edu/archives/2002/wxg/codes/OPQ.html>, Purdue University, 2004.
- [16] ———, *Orthogonal Polynomials: Computation and Approximation*, Oxford University Press, Oxford, 2004.
- [17] K. O. GEDDES, *Near-minimax polynomial approximation in an elliptical region*, SIAM J. Numer. Anal., 15 (1978), pp. 1225–1233.
- [18] K. O. GEDDES AND J. C. MASON, *Polynomial approximation by projections on the unit circle*, SIAM J. Numer. Anal., 12 (1975), pp. 111–120.
- [19] L. GEMIGNANI, *Rational interpolation via orthogonal polynomials*, Comput. Math. Appl., 26 (1993), pp. 27–34.
- [20] P. R. GRAVES-MORRIS, *Symmetrical formulas for rational interpolants*, J. Comput. Appl. Math., 10 (1984), pp. 107–111.
- [21] M. H. GUTKNECHT, *In what sense is the rational interpolation problem well posed?*, Constr. Approx., 6 (1990), pp. 437–450.
- [22] P. HENRICI, *Essentials of Numerical Analysis*, Wiley, New York, 1982.
- [23] N. J. HIGHAM, *The numerical stability of barycentric Lagrange interpolation*, IMA J. Numer. Anal., 24 (2004), pp. 547–556.
- [24] C. G. J. JACOBI, *Über die Darstellung einer Reihe gegebner Werthe durch eine gebrochne rationale Function*, J. Reine Angew. Math., 30 (1846), pp. 127–156.
- [25] P. KRAVANJA, T. SAKURAI, AND M. VAN BAREL, *On locating clusters of zeros of analytic functions*, BIT, 39 (1999), pp. 646–682.
- [26] L. KRONECKER, *Zur Theorie der Elimination einer Variablen aus zwei algebraische Gleichungen*, Monatsber. Königl. Preussischen Akad. Wiss. (Berlin), (1881), pp. 535–600.
- [27] A. L. LEVIN AND E. B. SAFF, *Potential theoretic tools in polynomial and rational approximation*, in Harmonic Analysis and Rational Approximation: Their rôles in Signals, Control and Dynamical Systems, J.-D. Fournier, J. Grimm, J. Leblond, and J. R. Partington, eds., vol. 327 of Lecture Notes in Control and Information Sciences, Springer, 2006, pp. 71–94.
- [28] J. N. LYNESS AND G. SANDE, *Algorithm 413. ENTCAF and ENTCRE: Evaluation of normalized Taylor coefficients of an analytic function*, Comm. ACM, 14 (1971), pp. 669–675.
- [29] H. MAEHLY AND C. WITZGALL, *Tschebyscheff-Approximationen in kleinen Intervallen II, Stetigkeitssätze für gebrochen rationale Approximationen*, Numer. Math., 2 (1960), pp. 293–307.
- [30] A. MARTINEZ-FINKELSHTEIN, *Equilibrium problems of potential theory in the complex plane*, in Orthogonal Polynomials and Special Functions, W. Van Assche and F. Marcellán, eds., vol. 1883 of Lecture Notes Math., Springer, 2006, pp. 79–117.

- [31] J. MASON AND D. HANDSCOMB, *Chebyshev Polynomials*, Chapman and Hall/CRC, Boca Raton, FL, 2003.
- [32] G. MASTROIANNI AND J. SZABADOS, *Jackson order of approximation by Lagrange interpolation. II*, Acta Math. Hungar., 69 (1995), pp. 73–82.
- [33] J. MEINGUET, *On the solubility of the Cauchy interpolation problem*, in Approximation Theory, A. Talbot, ed., Academic Press, London, 1970, pp. 137–163.
- [34] G. OPITZ, *Steigungsmatrizen*, Z. Angew. Math. Mech., 44 (1964), pp. T52–T54.
- [35] R. B. PLATTE, L. N. TREFETHEN, AND A. B. J. KUIJLAARS, *Impossibility of fast stable approximation of analytic functions from equispaced samples*. SIAM Review (to appear).
- [36] M. POLEZZI AND A. SRI RANGA, *On the denominator values and bayreentric weights of rational interpolants*, J. Comput. Appl. Math., 200 (2007), pp. 576–590.
- [37] A. PREDONZAN, *Su una formula d'interpolazione per le funzioni razionali*, Rend. Sem. Mat. univ. Padova, 22 (1953), pp. 417–425.
- [38] H.-J. RACK AND M. REIMER, *The numerical stability of evaluation schemes for polynomials based on the Lagrange interpolation form*, BIT, 22 (1982), pp. 101–107.
- [39] T. J. RIVLIN, *The Chebyshev Polynomials*, John Wiley, New York, 1974.
- [40] E. B. SAFF, *An extension of Montessus de Ballore's theorem on the convergence of interpolating rational functions*, J. Approx. Theory, 6 (1972), pp. 63–67.
- [41] ———, *Polynomial and rational approximation in the complex domain*, in Approximation Theory, C. de Boor, ed., vol. 36 of Proc. Symp. Appl. Math., Providence, 1986, Amer. Math. Soc, pp. 21–49.
- [42] E. B. SAFF AND J. L. WALSH, *On the convergence of rational functions which interpolate in the roots of unity*, Pacific J. Math., 45 (1973), pp. 639–641.
- [43] O. SALAZAR CELIS, *Practical Rational Interpolation of Exact and Inexact Data. Theory and Algorithms*, PhD thesis, University of Antwerp, July 2008.
- [44] H. E. SALZER, *Lagrangian interpolation at the Chebyshev points $x_{n,\nu} = \cos(\nu\pi/n)$, $\nu = 0(1)n$; some unnoted advantages*, Computer J., 15 (1972), pp. 156–159.
- [45] ———, *Rational interpolation using incomplete barycentric forms*, Z. Angew. Math. Mech., 61 (1981), pp. 161–164.
- [46] C. SCHNEIDER AND W. WERNER, *Some new aspects of rational interpolation*, Math. Comp., 47 (1986), pp. 285 – 299.
- [47] T. W. TEE AND L. N. TREFETHEN, *A rational spectral collocation method with adaptively transformed Chebyshev grid points*, SIAM J. Sci. Comput., 28 (2006), pp. 1798–1811.
- [48] L. N. TREFETHEN, *Spectral Methods in MATLAB*, SIAM, Philadelphia, 2000.

- [49] L. N. TREFETHEN AND D. BAU III, *Numerical Linear Algebra*, SIAM, Philadelphia, PA., 1997.
- [50] L. N. TREFETHEN, N. HALE, R. B. PLATTE, T. A. DRISCOLL, AND R. PACHÓN, *Chebfun version 3*. <http://www.maths.ox.ac.uk/chebfun/>, Oxford University, 2009.
- [51] C. VAN LOAN, *Computational Frameworks for the Fast Fourier Transform*, SIAM, Philadelphia, 1992.
- [52] J. L. WALSH, *Interpolation and Approximation by Rational Functions in the Complex Domain*, vol. 20 of Colloquium Publications, American Mathematical Society, Providence, Rhode Island, 5th ed., 1969.
- [53] H. WERNER, *Rationale Tshcebyscheff-Approximation, Eigenwerttheorie und Differenzenrechnung*, Arch. Rat. Mech. Anal., 13 (1963), pp. 330–347.
- [54] L. WUYTACK, *On some aspects of the rational interpolation problem*, SIAM J. Numer. Anal., 11 (1974), pp. 52–60.
- [55] X. ZHU AND G. ZHU, *A method for directly finding the denominator values of rational interpolants*, J. Comput. Appl. Math., 148 (2002), pp. 341–348.