An iterative approach to barycentric rational Hermite interpolation

Emiliano Cirillo · Kai Hormann

Abstract

In this paper we study an iterative approach to the Hermite interpolation problem, which first constructs an interpolant of the function values at $n+1$ nodes and then successively adds m correction terms to fit the data up to the m-th derivatives. In the case of polynomial interpolation, this simply reproduces the classical Hermite interpolant, but the approach is general enough to be used in other settings. In particular, we focus on the family of rational Floater–Hormann interpolants, which are based on blending local polynomial interpolants of degree d with rational blending functions. For this family, the proposed method results in rational Hermite interpolants, which depend linearly on the data, with numerator and denominator of degree at most $(m+1)(n+1)-1$ and $(m+1)(n-d)$, respectively. They converge at the rate of $O\big(h^{(m+1)(d+1)}\big)$ as the mesh size h converges to zero. After deriving the barycentric form of these interpolants, we prove the convergence rate for $m=1$ and $m=2$, and show that the approximation results compare favourably with other constructions.

Citation Info

Journal
Numerische Mathematik
Volume
140(4), December 2018
Pages
939–962

1 Introduction

For $m \in \mathbb{N}$, given an m times differentiable function $f\colon [a,b] \to \mathbb{R}$, and the $n+1$ interpolation nodes

$$a = x_0 < x_1 < \cdots < x_n = b,$$

the Hermite interpolation problem consists in finding a function $r_m\colon \mathbb{R} \to \mathbb{R}$, such that

$$r_m^{(k)}(x_i) = f_i^{(k)} = f^{(k)}(x_i), \qquad i = 0,\dots,n, \quad k = 0,\dots,m. \tag{1}$$

In the case of polynomial interpolation, this problem has a unique solution [13] $r_m \in \Pi_{(m+1)(n+1)-1}$, where Π_d denotes the space of polynomials of degree at most d. This polynomial Hermite interpolant can either be expressed in terms of the given data $f_i^{(k)}$ and the explicit formulas of the Hermite basis polynomials or in the Newton form [13, 6], but it can also be obtained *iteratively* in the following way.

Using the Lagrange basis polynomials

$$\ell_i(x) = \prod_{j=0, j\neq i}^{n} \frac{x-x_j}{x_i-x_j}, \qquad i = 0,\dots,n, \tag{2}$$

the polynomial interpolant $r_0 \in \Pi_n$ to the function values $f_0^{(0)},\dots,f_n^{(0)}$ at $x_0,\dots,x_n$ can be written as

$$r_0(x) = \sum_{i=0}^{n} \ell_i(x) f_i^{(0)}.$$

The polynomial $r_1 \in \Pi_{2n+1}$ that additionally interpolates the first derivatives $f_0^{(1)},\dots,f_n^{(1)}$ at $x_0,\dots,x_n$, can then be obtained by adding the correction term

$$q_1(x) = \sum_{i=0}^{n} \ell_{i,1}(x)\big(f_i^{(1)} - r_0'(x_i)\big),$$

where

$$\ell_{i,1}(x) = (x-x_i)\ell_i(x)^2, \qquad i = 0,\dots,n. \tag{3}$$

Indeed, since $\ell_{i,1}(x_j) = 0$ and $\ell_{i,1}'(x_j) = \delta_{i,j}$ for $i,j = 0,\dots,n$, it is clear that

$$r_1(x) = r_0(x) + q_1(x) = \sum_{i=0}^{n} \Big(\ell_i(x) f_i^{(0)} + \ell_{i,1}(x)\big(f_i^{(1)} - r_0'(x_i)\big)\Big),$$

satisfies the conditions in (1) for $m=1$, and by the uniqueness of the polynomial Hermite interpolant, r_1 coincides with the interpolant obtained in the classical way as

$$r_1(x)=\sum_{i=0}^{n}\Big(\ell_{i,0}(x)f_i^{(0)}+\ell_{i,1}(x)f_i^{(1)}\Big),$$

where

$$\ell_{i,0}(x)=\big(1-2(x-x_i)\ell_i'(x_i)\big)\ell_i(x)^2, \qquad i=0,\dots,n,$$

and $\ell_{i,1}(x)$ as in (3). A similar approach can be used to construct the polynomial $r_m\in\Pi_{(m+1)(n+1)-1}$ that fits the data up to the m-th derivatives by iteratively adding appropriate correction terms.

We discuss the details of this iterative Hermite interpolation approach in Section 2. Our key observation is that this construction works for any set of sufficiently smooth initial basis functions that satisfy the Lagrange property, and the main purpose of this paper is to discuss the combination of this approach with the rational basis functions of the Floater–Hormann interpolation scheme.

For any d with $0\le d\le n$, the Floater–Hormann interpolant is a special kind of rational interpolant with the guaranteed absence of real poles [3]. It may be written as

$$r(x)=\sum_{i=0}^{n-d}\lambda_i(x)p_i(x)\Bigg/\sum_{i=0}^{n-d}\lambda_i(x), \tag{4}$$

where $p_i\in\Pi_d$ denotes the unique polynomial that interpolates the function values $f_i^{(0)},\dots,f_{i+d}^{(0)}$ at the nodes $x_i,\dots,x_{i+d}$, and the blending functions λ_i are defined as

$$\lambda_i(x)=\frac{(-1)^i}{(x-x_i)\cdots(x-x_{i+d})}, \qquad i=0,\dots,n-d. \tag{5}$$

It has been proven [3] that for $d>0$ this rational interpolant converges to f at the rate of $O(h^{d+1})$ as $h\to 0$, where

$$h=\max_{0\le i\le n-1}(x_{i+1}-x_i), \tag{6}$$

and that its k-th derivative behaves like

$$|f^{(k)}(x)-r^{(k)}(x)|\le Ch^{d+1-k}, \qquad 0<k\le d,$$

for $k=1,2$ [1] and for general k under certain assumptions on the distribution of the interpolation nodes [9, 2].

Using the iterative construction in Section 2, we can extend the Floater–Hormann interpolation scheme to the Hermite case (Section 3), resulting in rational Hermite interpolants with numerator and denominator of degree at most $(m+1)(n+1)-1$ and $(m+1)(n-d)$ and no poles in $\mathbb{R}$. After deriving the barycentric form of these interpolants (Section 4), we analyse the approximation error for the cases $m=1$ and $m=2$ (Section 5) and show that it is of the order $O\big(h^{(m+1)(d+1)}\big)$. Our numerical experiments (Section 6) confirm these results and indicate that this is also the correct approximation order for $m>2$.

The use of rational functions for solving the Hermite interpolation problem is not new, and the most relevant reference in our context is the work by Schneider and Werner [11], who were the first to investigate rational Hermite interpolants with the barycentric approach. They provide an algorithm for computing the weights of the barycentric form of such interpolants and derive formulas for determining their derivatives. Despite the advantages of their approach, the main difficulty remains to find sets of weights that guarantee the absence of poles. One possibility is to prescribe the denominator q of the rational Hermite interpolant $r=p/q$, such that $q(x)>0$ for $x\in[a,b]$, and to use the algorithm in [11] to get the barycentric form of r, but the particular choice of q suggested in [11] can give huge approximation errors near the centre of the interpolation interval. Zhao et al. [14] propose to find the "optimal" weights by minimizing the square of the approximation error subject to certain constraints, including the positivity of q, but this requires to solve a nonlinear optimization problem and the resulting weights depend on f.

In order to get barycentric rational Hermite interpolants without poles in $\mathbb{R}$ and good approximation rates, the Floater–Hormann interpolant has been generalized in two ways. Jing, Kang, and Zhu [8] focus on the special case $m=1$ and propose to define r as in (4), but with p_i denoting the unique polynomial of degree $2d+1$ that interpolates the function values $f_j^{(0)}$ and the first derivatives $f_j^{(1)}$ at x_j for $j=i,\dots,i+d$, and with the blending functions λ_i replaced by λ_i^2. They show that these interpolants converge to f at the rate of $O(h^{2d+1})$ as $h \to 0$, which is one order less than the convergence rate of our interpolant for $m = 1$. Floater and Schulz [4] instead derive a Hermite version of the Floater–Hormann interpolant by considering multiple interpolation nodes. The resulting interpolants have the same degree and convergence rate as our interpolants, but they turn out to have a larger maximum approximation error in all our numerical tests (Section 6).

2 Iterative Hermite interpolation

Let $m \in \mathbb{N}$ and $b_0, \dots, b_n$ be some basis functions that satisfy the Lagrange property $b_i(x_l) = \delta_{i,l}$ for $i, l = 0, \dots, n$ and are m times differentiable at x_l for $l = 0, \dots, n$. We then define the functions

$$b_{i,j}(x) = \frac{(x - x_i)^j}{j!} b_i(x)^{j+1}, \qquad i = 0, \dots, n, \quad j = 0, \dots, m. \tag{7}$$

Lemma 1. *The functions $b_{i,j}$ in* (7) *satisfy*

$$b_{i,j}^{(k)}(x_l) = \begin{cases} 0, & \text{if } k < j, \\ \delta_{i,l}, & \text{if } k = j, \end{cases} \qquad l = 0, \dots, n.$$

Proof. For $j = 0$, the statement follows directly from the Lagrange property of the functions b_i, and for $j > 0$, we prove it by induction over j. To this end, let

$$c_i(x) = (x - x_i) b_i(x),$$

so that we can write $b_{i,j}$ as

$$b_{i,j}(x) = \frac{1}{j} c_i(x) b_{i,j-1}(x).$$

By the Leibniz rule,

$$b_{i,j}^{(k)}(x) = \frac{1}{j} \sum_{p=0}^{k} \binom{k}{p} c_i^{(k-p)}(x) b_{i,j-1}^{(p)}(x),$$

and since $c_i(x_l) = 0$ and $b_{i,j-1}^{(p)}(x_l) = 0$ for $p < j - 1$ by the induction hypothesis, we get

$$b_{i,j}^{(k)}(x_l) = \frac{1}{j} \sum_{p=j-1}^{k-1} \binom{k}{p} c_i^{(k-p)}(x_l) b_{i,j-1}^{(p)}(x_l).$$

The statement then follows by noticing that the sum is empty if $k < j$, and that if $k = j$, then

$$b_{i,j}^{(k)}(x_l) = \frac{1}{j} \binom{j}{j-1} c_i'(x_l) b_{i,j-1}^{(j-1)}(x_l) = \delta_{i,l},$$

again by the induction hypothesis and the fact that $c_i'(x_i) = 1$. □

Starting from the Lagrange interpolant

$$r_0(x) = \sum_{i=0}^{n} b_{i,0}(x) f_i^{(0)},$$

we can now use the functions $b_{i,j}$ to construct

$$r_j(x) = r_{j-1}(x) + q_j(x), \qquad j = 1, \dots, m, \tag{8}$$

by iteratively adding the correction terms

$$q_j(x) = \sum_{i=0}^{n} b_{i,j}(x) \big(f_i^{(j)} - r_{j-1}^{(j)}(x_i) \big), \qquad j = 1, \dots, m,$$

resulting in the Hermite interpolant r_m.

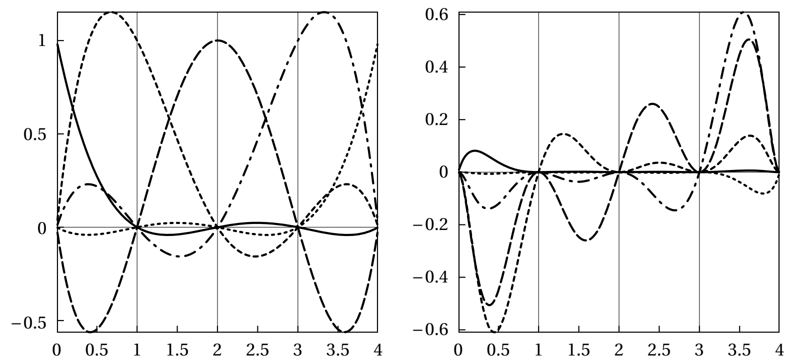

Figure 1: Polynomial basis functions $b_i = b_{i,0} = \ell_i$ (left) for the interpolation nodes $x_i = i$, $i = 0,\dots,4$ and corresponding functions $b_{i,1}$ (right).

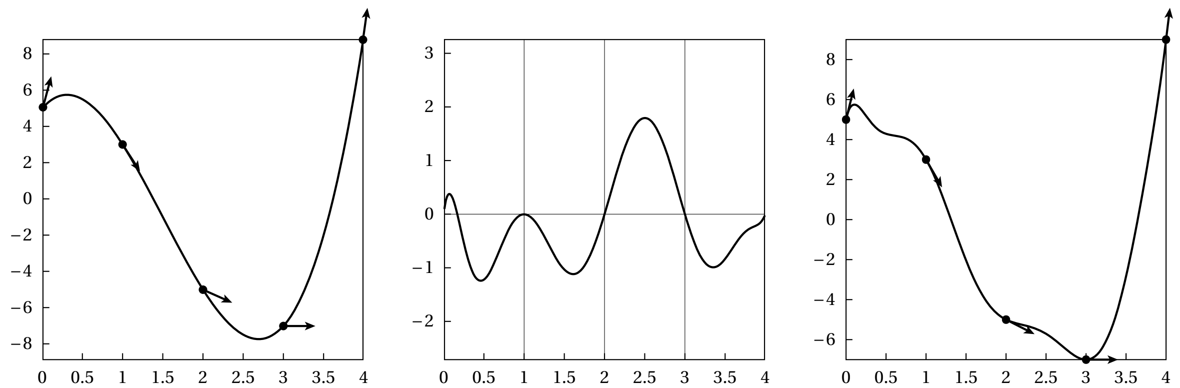

Figure 2: Lagrange interpolant r_0 (left), correction term q_1 (middle), and Hermite interpolant r_1 (right) for the data in Example 1 and the basis functions in Figure 1.

Theorem 1. *The function r_m in* (8) *satisfies the interpolation conditions in* (1).

Proof. By Lemma 1 we have

$$q_j^{(k)}(x_l) = \sum_{i=0}^{n} b_{i,j}^{(k)}(x_l)\big(f_i^{(j)} - r_{j-1}^{(j)}(x_i)\big) = \begin{cases} 0, & \text{if } k < j, \\ f_l^{(k)} - r_{j-1}^{(k)}(x_l), & \text{if } k = j, \end{cases}$$

hence

$$r_j^{(k)}(x_l) = \begin{cases} r_{j-1}^{(k)}(x_l), & \text{if } k < j, \\ f_l^{(k)}, & \text{if } k = j, \end{cases}$$

and the statement then follows by induction over j. □

By construction, it is clear that the Hermite interpolant r_m depends linearly on the given data. In the special case of polynomial interpolation, where the b_i are the Lagrange basis functions in (2), we notice that $b_{i,j} \in \Pi_{(j+1)(n+1)-1}$ for $i = 0,\dots,n$ and $j = 0,\dots,m$, hence $r_m \in \Pi_{(m+1)(n+1)-1}$. Therefore, by the uniqueness of the polynomial Hermite interpolant, the iteratively constructed and the classical Hermite interpolant are the same.

Example 1. For $n = 4$, let us consider the interpolation nodes

$$x_0 = 0, \qquad x_1 = 1, \qquad x_2 = 2, \qquad x_3 = 3, \qquad x_4 = 4,$$

the function values

$$f_0^{(0)} = 5, \qquad f_1^{(0)} = 3, \qquad f_2^{(0)} = -5, \qquad f_3^{(0)} = -7, \qquad f_4^{(0)} = 9,$$

and the derivative data

$$f_0^{(1)} = 17, \qquad f_1^{(1)} = -7, \qquad f_2^{(1)} = -2, \qquad f_3^{(1)} = 0, \qquad f_4^{(1)} = 33.$$

Taking the Lagrange basis functions (see Figure 1, left) in (2) as b_i in (7) gives the polynomial Lagrange interpolant (see Figure 2, left)

$$r_0(x) = 2x^3 - 9x^2 + 5x + 5,$$

whose first order derivatives

$$r_0'(0) = 5, \qquad r_0'(1) = -7, \qquad r_0'(2) = -7, \qquad r_0'(3) = 5, \qquad r_0'(4) = 29$$

do not match the given derivative data, except at $x_1 = 1$. This can be fixed by adding the correction term (see Figure 2, middle)

$$q_1(x) = 12b_{0,1}(x) + 5b_{2,1}(x) - 5b_{3,1}(x) + 4b_{4,1}(x),$$

because the basis functions (see Figure 1, right)

$$b_{i,1}(x) = (x - x_i)b_i(x)^2 = (x - x_i)\ell_i(x)^2$$

of this correction term modify only the first derivatives at the interpolation nodes, but not the function values, resulting in the polynomial Hermite interpolant (see Figure 2, right)

$$r_1(x) = r_0(x) + q_1(x) = \tfrac{29}{144}x^9 - \tfrac{91}{24}x^8 + \tfrac{237}{8}x^7 - 124x^6 + \tfrac{14371}{48}x^5 - \tfrac{3343}{8}x^4 + \tfrac{2887}{9}x^3 - \tfrac{370}{3}x^2 + 17x + 5.$$

3 Iterative rational Hermite interpolation

In order to combine the iterative construction in Section 2 with the Floater–Hormann interpolation scheme, we recall from [3] that r in (4) can be expressed in *barycentric form* as

$$r(x) = \sum_{i=0}^{n} b_i(x) f_i^{(0)},$$

with the basis functions

$$b_i(x) = \frac{w_i}{x - x_i} \Bigg/ \sum_{j=0}^{n} \frac{w_j}{x - x_j}, \qquad i = 0, \dots, n \tag{9}$$

that satisfy the Lagrange property and the *barycentric weights*

$$w_i = (-1)^{i+d} \sum_{j=\max(0,i-d)}^{\min(i,n-d)} \prod_{k=j,k\neq i}^{j+d} \frac{1}{|x_i - x_k|}, \qquad i = 0, \dots, n, \tag{10}$$

which depend only on the interpolation nodes x_i, but not on the data $f_i^{(0)}$.

Following the construction in (8), we now define the *iterative rational Hermite interpolant* as

$$r_m(x) = \sum_{i=0}^{n} \sum_{j=0}^{m} (x - x_i)^j b_i(x)^{j+1} g_{i,j}, \tag{11}$$

where

$$g_{i,0} = f_i^{(0)}, \qquad g_{i,j} = \big(f_i^{(j)} - r_{j-1}^{(j)}(x_i)\big)/j!, \qquad j = 1, \dots, m. \tag{12}$$

It follows from Theorem 1 that r_m satisfies the conditions in (1), and since Floater–Hormann interpolants and in particular the basis functions in (9) do not have any poles in $\mathbb{R}$, it is clear by construction that the same holds for r_m. Let us now investigate the degree of r_m.

Proposition 1. *The numerator and denominator of the iterative rational Hermite interpolant r_m in (11) have degree at most $(m+1)(n+1)-1$ and $(m+1)(n-d)$, respectively.*

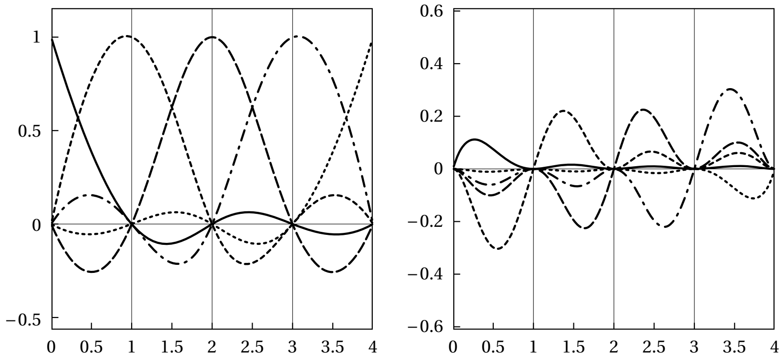

Figure 3: Rational Floater–Hormann basis functions $b_i = b_{i,0}$ (left) for $d = 1$ and the interpolation nodes $x_i = i$, $i = 0, \dots, 4$ and corresponding functions $b_{i,1}$ (right).

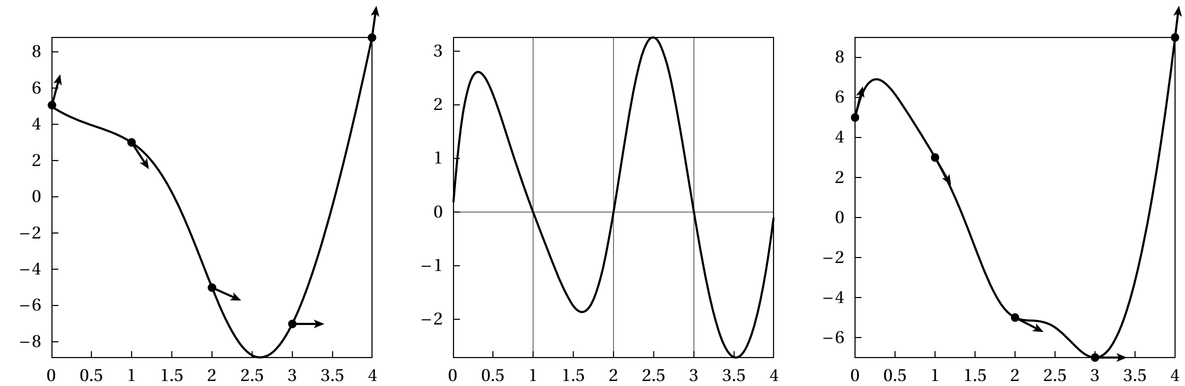

Figure 4: Lagrange interpolant r_0 (left), correction term q_1 (middle), and Hermite interpolant r_1 (right) for the data in Example 1 and the basis functions in Figure 3.

Proof. We first recall from [3] that the degrees of the numerator and the denominator of the Floater–Hormann interpolant r are at most n and $n-d$, respectively, which can be seen after multiplying the numerator and denominator in (4) by

$$\mu(x) = \prod_{k=0}^{n} (x - x_k).$$

Therefore, the basis functions b_i in (9) can be written in rational form as

$$b_i(x) = \frac{P_i(x)}{Q(x)}, \qquad i = 0, \dots, n,$$

with certain numerators $P_i \in \Pi_n$ and a common denominator $Q \in \Pi_{n-d}$, so that

$$r_m(x) = \sum_{i=0}^{n} \sum_{j=0}^{m} (x - x_i)^j \left(\frac{P_i(x)}{Q(x)} \right)^{j+1} g_{i,j} = \frac{\sum_{i=0}^{n} \sum_{j=0}^{m} (x - x_i)^j P_i(x)^{j+1} Q(x)^{m-j} g_{i,j}}{Q(x)^{m+1}}. \tag{13}$$

Independently of i, the degrees of the terms in the numerator of r_m in (13) are

$$j + (j+1)n + (m-j)(n-d) \le (m+1)(n+1) - 1, \qquad j = 0, \dots, m,$$

and the degree of the denominator of r_m is at most $(m+1)(n-d)$. □

Example 2. For the interpolation nodes, functions values, and derivative data from Example 1, the rational basis functions b_i in (9) of the Floater–Hormann interpolant for $d = 1$ (see Figure 3, left) give rise to the rational Lagrange interpolant (see Figure 4, left)

$$r_0(x) = \frac{3x^4 - 17x^3 + 31x^2 - 38x + 30}{x^2 - 4x + 6},$$

whose first order derivatives

$$r_0'(0) = -3, \qquad r_0'(1) = -3, \qquad r_0'(2) = -11, \qquad r_0'(3) = 9, \qquad r_0'(4) = 21$$

do not match the given derivative data. This can be fixed by adding the correction term (see Figure 4, middle)

$$q_1(x) = 20b_{0,1}(x) - 4b_{1,1} + 9b_{2,1}(x) - 9b_{3,1}(x) + 12b_{4,1}(x),$$

resulting in the rational Hermite interpolant (see Figure 4, right)

$$r_1(x) = \frac{4x^9 - 81x^8 + 699x^7 - 3321x^6 + 9445x^5 - 16446x^4 + 17120x^3 - 9520x^2 + 1488x + 720}{4(x^2 - 4x + 6)^2}.$$

4 The barycentric form

Neither of the formulas in (11) and (13) are suitable for an efficient construction and evaluation of the rational Hermite interpolant r_m, because the data values $g_{i,j}$ in (12) are defined recursively in terms of the derivatives of the interpolants r_j, $j = 0, \ldots, m-1$ and depend on the data $f_i^{(k)}$. A better choice is to write r_m in *barycentric form* [11],

$$r_m(x) = \frac{\displaystyle\sum_{i=0}^{n}\sum_{j=0}^{m} \frac{w_{i,j}^{[m]}}{(x-x_i)^{j+1}} \sum_{k=0}^{j} \frac{f_i^{(k)}}{k!}(x-x_i)^k}{\displaystyle\sum_{i=0}^{n}\sum_{j=0}^{m} \frac{w_{i,j}^{[m]}}{(x-x_i)^{j+1}}}, \tag{14}$$

with *barycentric weights* $w_{i,j}^{[m]}$, which depend only on the interpolation nodes x_i, but not on the data $f_i^{(k)}$. Schneider and Werner [11] show that any rational Hermite interpolant can be written in this form, and they provide tools for determining the weights $w_{i,j}^{[m]}$, which we can use in our case.

Theorem 2. *The iterative rational Hermite interpolant r_m in* (11) *can be written in barycentric form* (14) *with barycentric weights*

$$w_{i,j}^{[m]} = (-1)^{j+1} \sum_{|\gamma|=m-j} \prod_{k=1}^{m+1} \vartheta_{i,\gamma_k}, \qquad i = 0, \ldots, n, \quad j = 0, \ldots, m, \tag{15}$$

where the sum ranges over all $(m+1)$-dimensional multi-indices $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_{m+1})$ whose non-negative integer components sum up to $m-j$ and

$$\vartheta_{i,0} = -w_i, \qquad \vartheta_{i,j} = \sum_{k=0,k\neq i}^{n} \frac{w_k}{(x_i - x_k)^j}, \qquad j = 1, \ldots, m, \tag{16}$$

with w_i as defined in (10).

Proof. Letting

$$\omega_i(x) = w_i + (x - x_i) \sum_{k=0,k\neq i}^{n} \frac{w_k}{x - x_k}$$

and

$$\Omega_i(x) = \omega_i(x)^{m+1},$$

it follows from [11, Lemma 2.6] that

$$w_{i,m-j}^{[m]} = \frac{1}{j!}\Omega_i^{(j)}(x_i).$$

By the general Leibniz rule for higher order derivatives of a product of several functions,

$$\Omega_i^{(j)}(x) = \sum_{|\gamma|=j} \binom{j}{\gamma_1, \ldots, \gamma_{m+1}} \prod_{k=1}^{m+1} \omega_i^{(\gamma_k)}(x),$$

	$w_{i,0}^{[m]}$	$w_{i,1}^{[m]}$	$w_{i,2}^{[m]}$	$w_{i,3}^{[m]}$
$m=0$	w_i			
$m=1$	$2w_i\vartheta_{i,1}$	w_i^2		
$m=2$	$-3w_i^2\vartheta_{i,2}+3w_i\vartheta_{i,1}^2$	$3w_i^2\vartheta_{i,1}$	w_i^3	
$m=3$	$4w_i^3\vartheta_{i,3}-12w_i^2\vartheta_{i,2}\vartheta_{i,1}+4w_i\vartheta_{i,1}^3$	$-4w_i^3\vartheta_{i,2}+6w_i^2\vartheta_{i,1}^2$	$4w_i^3\vartheta_{i,1}$	w_i^4

Table 1: Barycentric weights of the iterative rational Hermite interpolant r_m for $m \le 3$.

where the sum ranges over all $(m+1)$-dimensional multi-indices $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_{m+1})$ whose non-negative integer components sum up to j. Since

$$\omega_i^{(j)}(x_i) = (-1)^{j+1} j! \vartheta_{i,j},$$

with $\vartheta_{i,j}$ as defined in (16), we then have

$$w_{i,m-j}^{[m]} = (-1)^{m-j+1} \sum_{|\boldsymbol{\gamma}|=j} \prod_{k=1}^{m+1} \vartheta_{i,\gamma_k},$$

and the statement follows after substituting $m-j$ with j. □

Table 1 lists the weights in (15) for $m \le 3$. Since these weights do not depend on the data $f_i^{(k)}$, they can be precomputed once for a given set of interpolation nodes x_i, and the iterative rational Hermite interpolant r_m can then be evaluated for arbitrary data $f_i^{(k)}$ and at any x with $O(nm^2)$ operations, using the barycentric form (14).

Remark 1. In the special case of equidistant interpolation nodes, the weights w_i of the Floater–Hormann interpolant are known to be very simple [3], and the same is true for the weights $w_{i,m}^{[m]} = w_i^{m+1}$. For $j < m$, however, the weights $w_{i,j}^{[m]}$ do not seem to have a simple form, and unlike the weights of the interpolants in [4] and [8], they depend on n.

Remark 2. Although we did not notice any numerical problems in our experiments, it remains future work to study the stability of computing the weights $w_{i,j}^{[m]}$ as in (15). Indeed, if two nodes x_i and x_{i+1} are very close and m is large, then the evaluation of $\vartheta_{i,j}$ in (16) may suffer from cancellation. However, the barycentric form (14) comes with the advantage of maintaining the interpolation property even if rounding errors occur during the computation of the weights [11]. And since the weights are determined in a preprocessing step, it is also possible to carry out these computations in high precision arithmetic, despite the additional cost.

5 Approximation error

Let us now analyse the approximation power of our iterative rational Hermite interpolant. We start with the case $m=1$. Denoting the denominator of (9) by

$$W(x) = \sum_{j=0}^{n} \frac{w_j}{x-x_j},$$

it follows from (11) that

$$\begin{aligned} f(x) - r_1(x) &= f(x) - \frac{1}{W(x)} \sum_{i=0}^{n} \frac{w_i}{x-x_i} g_{i,0} - \frac{1}{W(x)^2} \sum_{i=0}^{n} \frac{w_i^2}{x-x_i} g_{i,1} \\ &= \frac{1}{W(x)^2} \sum_{i=0}^{n} \frac{w_i}{x-x_i} \left(\sum_{j=0}^{n} \frac{w_j}{x-x_j} (f(x) - f(x_j)) - w_i g_{i,1} \right) \\ &= \frac{A(x)}{W(x)^2}, \end{aligned} \tag{17}$$

with

$$A(x) = \sum_{i=0}^{n} \frac{w_i}{x-x_i} \left(\sum_{j=0}^{n} w_j f[x,x_j] - w_i g_{i,1} \right).$$

Recalling from [10, Proposition 11] that

$$-w_i r_0'(x_i) = \sum_{j=0, j\neq i}^{n} w_j f[x_i, x_j],$$

hence

$$w_i g_{i,1} = w_i f[x_i, x_i] - w_i r_0'(x_i) = \sum_{j=0}^{n} w_j f[x_i, x_j], \tag{18}$$

we observe that $A(x)$ simplifies to

$$A(x) = \sum_{i=0}^{n} w_i \sum_{j=0}^{n} w_j f[x, x_i, x_j]. \tag{19}$$

But before we proceed to bound the error, we need an auxiliary result.

Lemma 2. *The barycentric weights in* (10) *satisfy*

$$\sum_{i=0}^{n} w_i f[x, x_i] = \sum_{i=0}^{n-d} (-1)^i f[x, x_i, \dots, x_{i+d}]$$

for any $x \in \mathbb{R}$.

Proof. Following Hormann and Schaefer [7], we let

$$V_i^d = 1, \qquad i = 0, \dots, n-d$$

and

$$V_i^{j-1} = \frac{V_{i-1}^j}{x_{i+j-1} - x_{i-1}} + \frac{V_i^j}{x_{i+j} - x_i}, \qquad i = 0, \dots, n-j+1,$$

for $j = d, d-1, \dots, 1$, tacitly assuming that $V_i^j = 0$ for $i < 0$ and $i > n-j$ to keep the notation simple. Then,

$$\begin{aligned}
\sum_{i=0}^{n-d} (-1)^i f[x, x_i, \dots, x_{i+d}] &= \sum_{i=0}^{n-d} (-1)^i V_i^d \frac{f[x, x_{i+1}, \dots, x_{i+d}] - f[x, x_i, \dots, x_{i+d-1}]}{x_{i+d} - x_i} \\
&= \sum_{i=1}^{n-d+1} (-1)^{i-1} \frac{V_{i-1}^d}{x_{i+d-1} - x_{i-1}} f[x, x_i, \dots, x_{i+d-1}] \\
&\quad - \sum_{i=0}^{n-d} (-1)^i \frac{V_i^d}{x_{i+d} - x_i} f[x, x_i, \dots, x_{i+d-1}] \\
&= \sum_{i=0}^{n-d+1} (-1)^{i-1} V_i^{d-1} f[x, x_i, \dots, x_{i+d-1}] = \cdots = \sum_{i=0}^{n} (-1)^{i-d} V_i^0 f[x, x_i],
\end{aligned}$$

and the statement follows by recalling from [7, Section 3] that $V_i^0 = (-1)^{i+d} w_i$. □

Note that Lemma 2 is also true if x is replaced by two or more variables. Now we are ready to get an error bound in the maximum norm. To this end, let $\|g\| = \max_{a \le x \le b} |g(x)|$ for any $g \in C^0[a, b]$.

Theorem 3. *Suppose* $d \ge 0$ *and* $f \in C^{2(d+2)}[a, b]$, *and let* h *be as in* (6). *Then,*

$$\|f - r_1\| \le C h^{2(d+1)},$$

where the constant C *depends only on* d, *the derivatives of* f, *the interval length* $b-a$, *and, only in the case* $d = 0$, *on the local mesh ratio*

$$\beta = \max_{1 \le i \le n-2} \min\left\{ \frac{x_{i+1} - x_i}{x_i - x_{i-1}}, \frac{x_{i+1} - x_i}{x_{i+2} - x_{i+1}} \right\}. \tag{20}$$

Proof. Since r_1 interpolates f at the interpolation nodes, it suffices to consider $x \in [a,b] \setminus \{x_0,\dots,x_n\}$. Our main idea is to derive an upper and a lower bound on the numerator and the denominator of the quotient in (17), respectively, and we proceed as in the proof of [3, Theorem 2].

For the numerator, we first apply Lemma 2 twice to (19) and thus get

$$A(x) = \sum_{i=0}^{n-d} (-1)^i \sum_{j=0}^{n-d} (-1)^j f[x, x_i, \dots, x_{i+d}, x_j, \dots, x_{j+d}].$$

Let us now assume that $n-d$ is odd, so that the number of terms in both sums is even. Combining the first and second terms of the second sum, the third and forth, and so on, we then have

$$A(x) = -\sum_{i=0}^{n-d} (-1)^i \sum_{j=0,\ j \text{ even}}^{n-d} (x_{j+d+1} - x_j) f[x, x_i, \dots, x_{i+d}, x_j, \dots, x_{j+d+1}],$$

and after applying the same strategy with respect to the first sum, we arrive at

$$A(x) = \sum_{i=0,\ i \text{ even}}^{n-d} (x_{i+d+1} - x_i) \sum_{j=0,\ j \text{ even}}^{n-d} (x_{j+d+1} - x_j) f[x, x_i, \dots, x_{i+d+1}, x_j, \dots, x_{j+d+1}].$$

Since

$$\sum_{i=0}^{n-d-1} (x_{i+d+1} - x_i) \le (d+1)(b-a),$$

as shown in the proof of [3, Theorem 2], it follows that

$$|A(x)| \le (d+1)^2 (b-a)^2 \frac{\|f^{(2d+4)}\|}{(2d+4)!}. \tag{21}$$

If $n-d$ is even, then a similar reasoning reveals that

$$|A(x)| \le (d+1)^2 (b-a)^2 \frac{\|f^{(2d+4)}\|}{(2d+4)!} + 2(d+1)(b-a) \frac{\|f^{(2d+3)}\|}{(2d+3)!} + \frac{\|f^{(2d+2)}\|}{(2d+2)!}. \tag{22}$$

For the denominator, we remember from [3, Section 4] that

$$W(x) = \sum_{i=0}^{n-d} \lambda_i(x),$$

with λ_i as defined in (5), and from the proofs of [3, Theorem 2] and [3, Theorem 3] that

$$|W(x)| \ge \frac{1}{d!\, h^{d+1}}$$

if $d \ge 1$ and

$$|W(x)| \ge \frac{1}{(1+\beta)h}$$

if $d = 0$. The statement then follows by combining these bounds. □

Equations (21) and (22) allow us to deduce the degree of polynomial reproduction of r_1.

Corollary 1. *The iterative rational Hermite interpolant r_1 reproduces polynomials of degree $2d+1$ and even of degree $2d+3$, if $n-d$ is odd.*

Let us now turn to the case $m = 2$. By (11) and (17), we have

$$f(x) - r_2(x) = f(x) - r_1(x) - \frac{1}{W(x)^3} \sum_{i=0}^{n} \frac{w_i^3}{x - x_i} g_{i,2} = \frac{B(x)}{W(x)^3}, \tag{23}$$

with

$$B(x) = \sum_{i=0}^{n} \frac{w_i}{x - x_i} \big(A(x) - w_i^2 g_{i,2} \big).$$

To simplify $B(x)$, we first note that

$$\begin{aligned}\sum_{j=0}^{n} w_{j,0}^{[1]} f[x_i,x_j] &= 2\sum_{j=0}^{n} w_j \sum_{k=0,k\neq j}^{n} \frac{w_k}{x_j-x_k} f[x_i,x_j]\\ &= \sum_{j=0}^{n} w_j \sum_{k=0,k\neq j}^{n} w_k \frac{f[x_i,x_j]}{x_j-x_k} - \sum_{j=0}^{n} w_j \sum_{k=0,k\neq j}^{n} w_k \frac{f[x_i,x_k]}{x_j-x_k}\\ &= \sum_{j=0}^{n} w_j \sum_{k=0,k\neq j}^{n} w_k f[x_i,x_j,x_k].\end{aligned}$$

We then recall from [11, Proposition 2.4] that

$$\begin{aligned}-\frac{1}{2} w_{i,1}^{[1]} r_1''(x_i) &= \sum_{j=0}^{n} w_{j,0}^{[1]} f[x_i,x_j] + \sum_{j=0,j\neq i}^{n} w_{j,1}^{[1]} f[x_i,x_j,x_j]\\ &= \sum_{j=0}^{n} w_j \sum_{k=0,k\neq j}^{n} w_k f[x_i,x_j,x_k] + \sum_{j=0,j\neq i}^{n} w_j^2 f[x_i,x_j,x_j],\end{aligned}$$

hence

$$\begin{aligned}w_i^2 g_{i,2} &= w_i^2 f[x_i,x_i,x_i] - \frac{1}{2} w_{i,1}^{[1]} r_1''(x_i)\\ &= \sum_{j=0}^{n} w_j \sum_{k=0,k\neq j}^{n} w_k f[x_i,x_j,x_k] + \sum_{j=0}^{n} w_j^2 f[x_i,x_j,x_j]\\ &= \sum_{j=0}^{n} w_j \sum_{k=0}^{n} w_k f[x_i,x_j,x_k].\end{aligned} \tag{24}$$

Using (19), we then get

$$B(x) = \sum_{i=0}^{n} w_i \sum_{j=0}^{n} w_j \sum_{k=0}^{n} w_k f[x,x_i,x_j,x_k].$$

The approximation order and degree of polynomial reproduction of r_2 can then be proven along the same lines as for r_1 above.

Theorem 4. *Suppose $d \geq 0$ and $f \in C^{3(d+2)}[a,b]$, and let h be as in* (6)*. Then,*

$$\|f - r_2\| \leq C h^{3(d+1)},$$

where the constant C depends only on d, the derivatives of f, the interval length $b-a$, and, only in the case $d=0$, on the local mesh ratio β in (20)*.*

Corollary 2. *The iterative rational Hermite interpolant r_2 reproduces polynomials of degree $3d+2$ and even of degree $3d+5$, if $n-d$ is odd.*

For $m > 2$, we conjecture that (18) and (24) generalize to

$$w_i^m g_{i,m} = \sum_{j_1=0}^{n} w_{j_1} \cdots \sum_{j_m=0}^{n} w_{j_m} f[x_i,x_{j_1},\ldots,x_{j_m}],$$

and the approximation order and degree of polynomial reproduction of r_m can then be proven along the same lines as for r_1 and r_2 above.

Conjecture 1. *Suppose $d \geq 0$ and $f \in C^{(m+1)(d+2)}[a,b]$, and let h be as in* (6)*. Then,*

$$\|f - r_m\| \leq C h^{(m+1)(d+1)},$$

where the constant C depends only on d, the derivatives of f, the interval length $b-a$, and, only in the case $d=0$, on the local mesh ratio β in (20)*.*

Conjecture 2. *The iterative rational Hermite interpolant r_m reproduces polynomials of degree $(m+1)(d+1)-1$ and even of degree $(m+1)(d+2)-1$, if $n-d$ is odd.*

rational Hermite interpolant	numerator degree		denominator degree		approximation order	
	$m=1$	$m>1$	$m=1$	$m>1$	$m=1$	$m>1$
r_m in (11)	$2n+1$	$(m+1)(n+1)-1$	$2(n-d)$	$(m+1)(n-d)$	$2d+2$	$(m+1)(d+1)$
Floater and Schulz [4]	$2n+1$	$(m+1)(n+1)-1$	$2(n-d)$	$(m+1)(n-d)$	$2d+2$	$(m+1)(d+1)$
Jing, Kang, and Zhu [8]	$2n+1$	—	$2(n-d)$	—	$2d+1$	—

Table 2: Properties of the rational Hermite interpolants that we compare in our numerical experiments.

Experiment	m	d	f	x_i	Figure	Table
1	1	0	$1/(1+25(2x-1)^2)$	$(1-\cos(i\pi/n))/2$	5	4
2	2	1	$(1+\tanh(-9x+1))/2$	i/n	6	5
3	4	1	$e^{-(x-1/2)^2/2}$	i/n	7	6
4	1	1	$101e^x/((100x-101)(100x+1))+1$	i/n	8	7
5	2	4	$\|3x-1\|+(3x-1)/2-(3x-1)^2$	i/n	9	8
6	1	3	$e^x/\cos(x)$	i/n	10	—
7	2	2	$\sin(10\pi x)x$	i/n	11	—

Table 3: Parameters m and d, functions f, and interpolation nodes x_i used in our numerical experiments.

6 Numerical experiments

We have tested and compared our rational Hermite interpolant r_m with the rational Hermite interpolants proposed by Floater and Schulz [4] and by Jing, Kang, and Zhu [8]. Table 2 lists the degrees of numerator and denominator, as well as the approximation orders of these three interpolants. Note that the interpolant of Jing, Kang, and Zhu is defined only for the case $m=1$ and that we use $(m+1)(d+1)-1$ as degree of the local polynomial interpolants in the construction of Floater and Schulz, so that both the denominator degree and the approximation order of their interpolant are the same as for our interpolant.

In our numerical experiments, we chose various values for the order m of interpolated derivatives and the degree d of the local polynomials used in the construction of the rational interpolants, and we tested different test functions f, with equidistant, Chebyshev, and other nodes, but the interpolation interval is

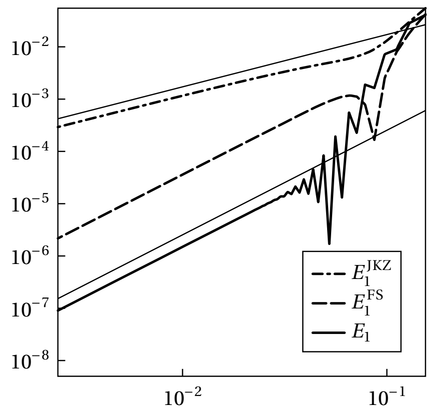

Figure 5: Log-log plot of the error with respect to h for Experiment 1.

n	E_1	order	E_1^{FS}	order	E_1^{JKZ}	order
10	4.07e–02		4.19e–02		5.49e–02	
20	1.89e–03	4.51	7.92e–04	5.83	7.43e–03	2.94
40	2.92e–05	6.05	5.40e–04	0.56	4.00e–03	0.90
80	5.72e–06	2.35	1.38e–04	1.97	2.20e–03	0.87
160	1.44e–06	1.99	3.47e–05	1.99	1.14e–03	0.95
320	3.61e–07	2.00	8.67e–06	2.00	5.80e–04	0.98
640	9.03e–08	2.00	2.17e–06	2.00	2.92e–04	0.99

Table 4: Error and approximation order for Experiment 1.

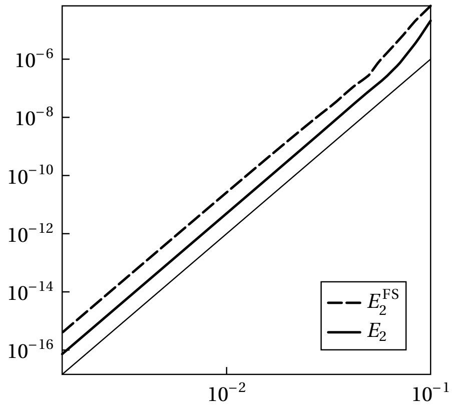

Figure 6: Log-log plot of the error with respect to h for Experiment 2.

n	E_2	order	E_2^{FS}	order
10	2.09e−05		6.79e−05	
20	8.11e−08	8.01	2.86e−07	7.89
40	1.23e−09	6.04	5.68e−09	5.65
80	1.90e−11	6.01	9.87e−11	5.85
160	2.98e−13	6.00	1.59e−12	5.95
320	4.66e−15	6.00	2.52e−14	5.98
640	7.28e−17	6.00	3.96e−16	5.99

Table 5: Error and approximation order for Experiment 2.

always $[a, b] = [0, 1]$. Table 3 summarizes the settings. Note that we decided to mainly focus on equidistant nodes, since polynomial Hermite interpolation behaves badly in this case. We observed similar results for other nodes.

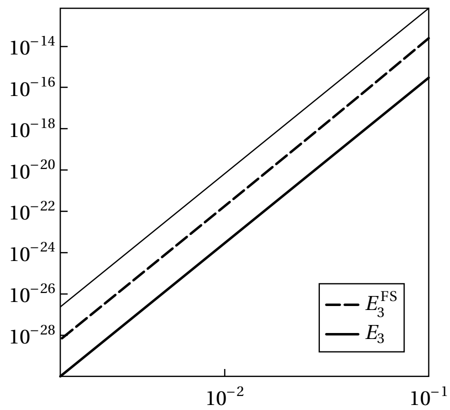

Figure 7: Log-log plot of the error with respect to h for Experiment 3.

n	E_3	order	E_3^{FS}	order
10	2.91e−16		2.38e−14	
20	1.14e−18	8.00	8.02e−17	8.21
40	4.44e−21	8.00	2.89e−19	8.12
80	1.73e−23	8.00	1.08e−21	8.06
160	6.77e−26	8.00	4.12e−24	8.03
320	2.64e−28	8.00	1.59e−26	8.02
640	1.03e−30	8.00	6.18e−29	8.01

Table 6: Error and approximation order for Experiment 3.

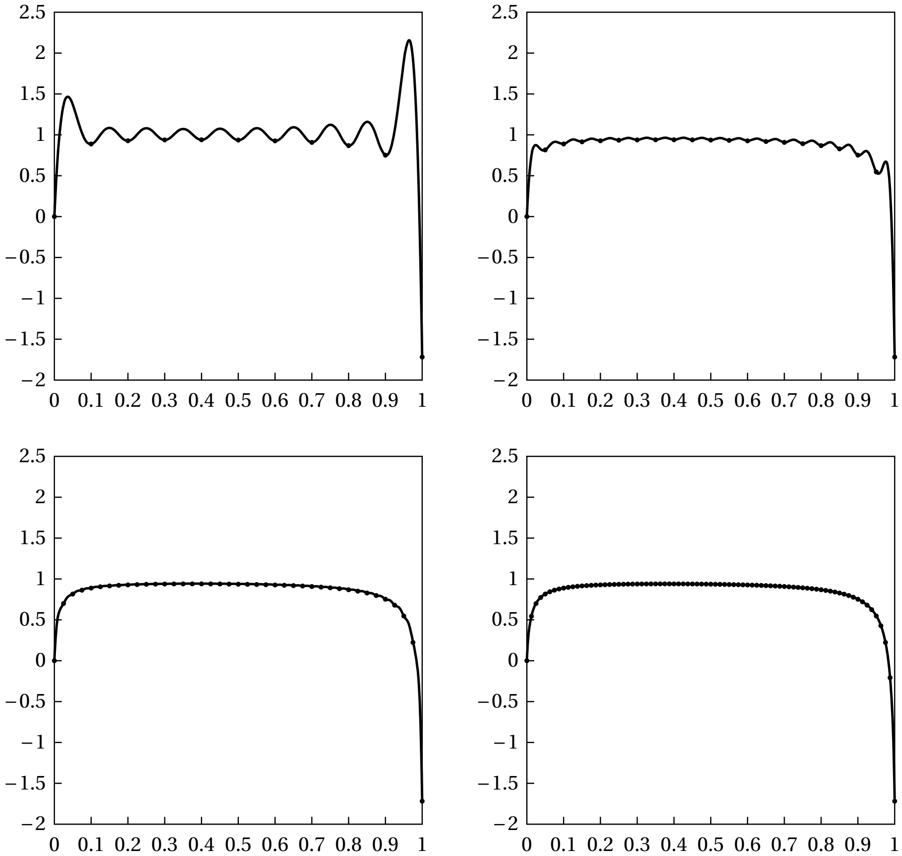

Figure 8: Iterative rational Hermite interpolant for $n = 10, 20, 40, 80$ for Experiment 4.

n	E_1	order	E_1^{FS}	order	E_1^{JKZ}	order	E_0^{FH}	order
10	1.78		2.01		2.51		7.82e–01	
20	5.64e–01	1.66	6.58e–01	1.61	8.71e–01	1.53	4.44e–01	0.82
40	1.35e–01	2.07	1.66e–01	1.99	2.44e–01	1.84	2.03e–01	1.13
80	2.23e–02	2.60	2.99e–02	2.47	5.31e–02	2.20	7.36e–02	1.46
160	2.51e–03	3.15	3.81e–03	2.97	9.14e–03	2.54	2.24e–02	1.72
320	2.10e–04	3.58	3.63e–04	3.39	1.34e–03	2.77	6.11e–03	1.87
640	1.48e–05	3.83	2.86e–05	3.67	1.82e–04	2.89	1.59e–03	1.94

Table 7: Error and approximation order for Experiment 4.

For each experiment we report the maximum error

$$E_m = \|f - r_m\|$$

and the approximation order, where E_m is computed by evaluating the pointwise error at 100 equidistant points in each of the n subintervals $[x_i, x_{i+1}]$, $i = 0, \dots, n-1$. We use the superscripts 'FS', 'JKZ', and 'FH' to refer to the Hermite interpolants proposed by Floater and Schulz [4] and by Jing, Kang, and Zhu [8], and to the classical Floater–Hormann interpolant [3], respectively.

The first three experiments support Theorems 3 and 4, as well as Conjecture 1, and more generally confirm the approximation orders listed in Table 2. In order to verify the approximation orders even for small h, all computations were performed in *C++* using the multiple-precision library *MPFR* [5]. Note that the plots in Figures 5–7 show the error only for the even values of n, from 10 to 640, because the errors for the odd values follow the same trend but with a lower constant and would thus have resulted in more confusing graphs. The thin straight reference lines represent and support the expected convergence rates, that is, $O(h^{(m+1)(d+1)})$ for E_m and E_m^{FS} and $O(h^{2d+1})$ for E_1^{JKZ}. Overall, these experiments show that for $m = 1$, our interpolant is better, in terms of approximation error and order, than the one proposed by Jing, Kang,

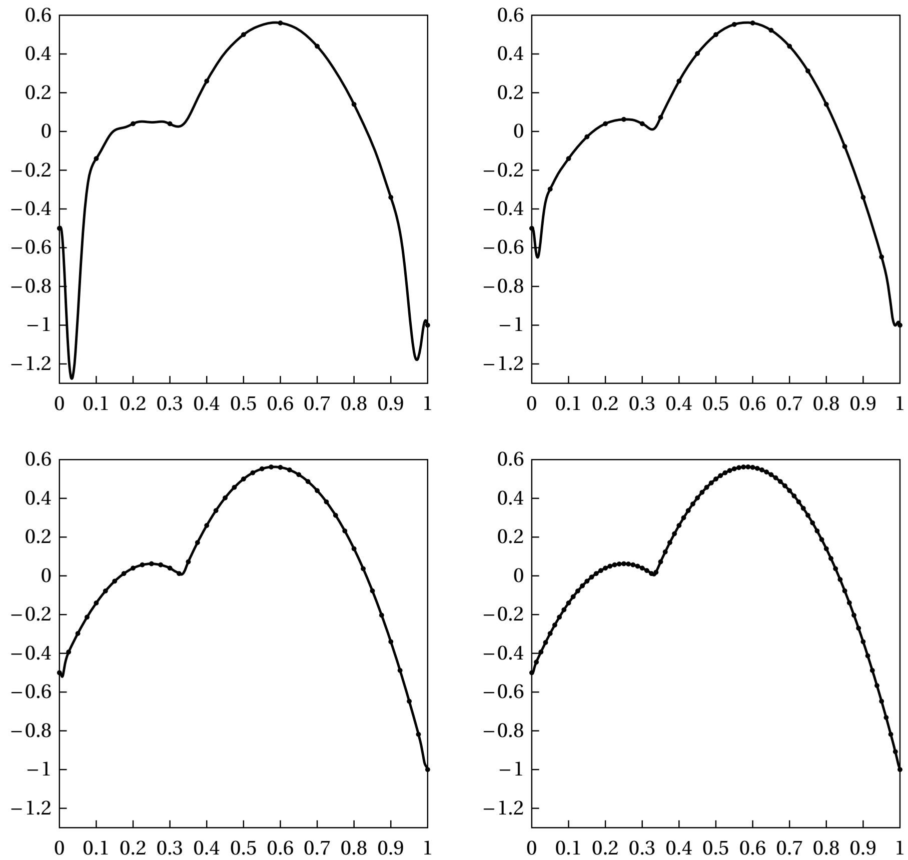

Figure 9: Iterative rational Hermite interpolant for $n = 10, 20, 40, 80$ for Experiment 5.

n	E_2	order	E_2^{FS}	order	E_0^{FH}	order
10	9.19e–01		2.35e–01		1.90e–02	
20	2.23e–01	2.05	5.30e–02	2.15	9.50e–03	1.00
40	5.58e–02	2.00	1.33e–02	2.00	4.75e–03	1.00
80	1.36e–02	2.04	3.74e–03	1.83	2.38e–03	1.00
160	3.40e–03	2.00	1.87e–03	1.00	1.19e–03	1.00
320	9.36e–04	1.86	9.36e–04	1.00	5.94e–04	1.00
640	4.68e–04	1.00	4.68e–04	1.00	2.97e–04	1.00

Table 8: Error and approximation order for Experiment 5.

and Zhu [8]. For general m, it matches the interpolant proposed by Floater and Schulz [4], but we observed that it typically gives an approximation error which is smaller by a factor of 2 to 5.

Experiments 4 and 5 show the interpolation quality of the proposed iterative rational Hermite interpolant at equidistant nodes for a C^∞ function with poles outside but near the endpoints of the interpolation interval in Figure 8 and for a C^0 function in Figure 9. Tables 7 and 8 report the corresponding numerical results for all rational Hermite interpolants and for the classical Floater–Hormann interpolant at $(m+1)(n+1)$ equidistant nodes, that is, for the same number of overall data values. All computations were carried out in *MATLAB* with standard precision. For the smooth function in Experiment 4, our interpolant has the smallest approximation error among the three Hermite interpolants. The non-Hermite Floater–Hormann interpolant is more accurate for small $n \leq 20$, but it is outperformed by the Hermite interpolants for larger n, because the latter have a higher approximation order. Experiment 5 shows that the smoothness condition on f in Theorems 3 and 4 is essential for the approximation order of our rational Hermite interpolant, which drops to $O(h)$, if f is only continuous. The same is true for the other interpolants, and we observe that the best approximation error is obtained by the Floater–Hormann interpolant in this experiment.

In Experiments 6 and 7, we compare the numerical stability of the rational Hermite interpolants in the case of equidistant interpolation nodes. All computations were performed in *C++* with double precision.

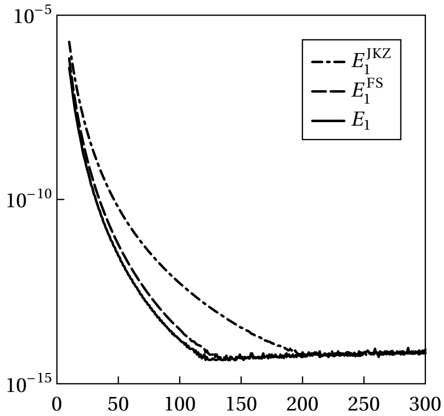

Figure 10: Semi-log plot of the error with respect to n for Experiment 6.

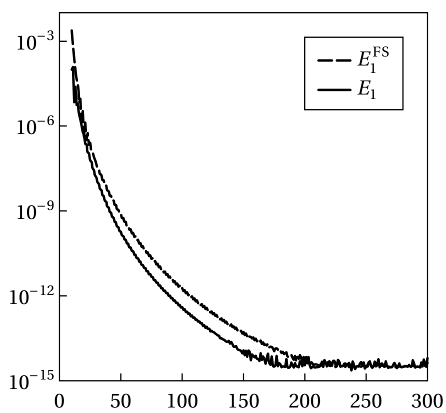

Figure 11: Semi-log plot of the error with respect to n for Experiment 7.

Figures 10 and 11 show that all interpolants reach the level of rounding errors for sufficiently large n and that our interpolant is the fastest to converge. However, we noticed that further increasing n may lead to a slight increase of the error for some test functions, as shown in Figure 10. Since this occurs for all three interpolants, it is probably not related to the computation of the barycentric weights, but it may indicate a numerical instability of the barycentric form (14). It remains future work to further investigate this phenomenon.

Acknowledgements

This work was supported by the Swiss National Science Foundation (SNSF) under project number 200021_150053. We thank the anonymous reviewers for their insightful comments, which helped to improve this paper.

References

[1] J.-P. Berrut, M. S. Floater, and G. Klein. Convergence rates of derivatives of a family of barycentric rational interpolants. *Applied Numerical Mathematics*, 61(9):989–1000, Sept. 2011.

[2] E. Cirillo, K. Hormann, and J. Sidon. Convergence rates of derivatives of Floater–Hormann interpolants for well-spaced nodes. *Applied Numerical Mathematics*, 116:108–118, June 2017.

[3] M. S. Floater and K. Hormann. Barycentric rational interpolation with no poles and high rates of approximation. *Numerische Mathematik*, 107(2):315–331, Aug. 2007.

[4] M. S. Floater and C. Schulz. *Rational Hermite interpolation without poles*, chapter 9, pages 117–139. In [12], 2009.

[5] L. Fousse, G. Hanrot, V. Lefèvre, P. Pélissier, and P. Zimmermann. MPFR: A multiple-precision binary floating-point library with correct rounding. *ACM Transactions on Mathematical Software*, 33(2):Article 13, 15 pages, June 2007.

[6] W. Gautschi. *Numerical Analysis: An Introduction*. Birkhäuser, Boston, 1997.

[7] K. Hormann and S. Schaefer. Pyramid algorithms for barycentric rational interpolation. *Computer Aided Geometric Design*, 42:1–6, Feb. 2016.

[8] K. Jing, N. Kang, and G. Zhu. Convergence rates of a family of barycentric osculatory rational interpolation. *Applied Mathematics and Computation*, 53(1):169–181, Dec. 2015.

[9] G. Klein and J.-P. Berrut. Linear rational finite differences from derivatives of barycentric rational interpolants. *SIAM Journal on Numerical Analysis*, 50(2):643–656, 2012.

[10] C. Schneider and W. Werner. Some new aspects of rational interpolation. *Mathematics of Computation*, 47(175):285–299, July 1986.

[11] C. Schneider and W. Werner. Hermite interpolation: The barycentric approach. *Computing*, 46(1):35–51, Mar. 1991.

[12] C. Schulz. *Topics in Curve Intersection and Barycentric Interpolation*. PhD thesis, University of Oslo, 2009.

[13] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*, volume 12 of *Texts in Applied Mathematics*. Springer, New York, 2nd edition, 1993.

[14] Q. Zhao, Y. Hao, Z. Yin, and Y. Zhang. Best barycentric rational Hermite interpolation. In *Proceedings of the 2010 International Conference on Intelligent System Design and Engineering Application (ISDEA 2010)*, volume 1, pages 417–419. IEEE Computer Society, Los Alamitos, CA, 2010.