Proc. Computer Algebra in Scientific Computing (CASC) 2020
F. Boulier, M. England, T. Sadykov, E. Vorozhtsov (eds.)
Springer Lecture Notes in Computer Science

Hermite Rational Function Interpolation with Error Correction*

Erich L. Kaltofen[1,2], Clément Pernet[3], and Zhi-Hong Yang[1,2]

[1] Department of Mathematics, North Carolina State University,
Raleigh, North Carolina 27695-8205, USA
{kaltofen,zyang28}@ncsu.edu
[2] Department of Computer Science, Duke University,
Durham, North Carolina 27708-0129, USA
{kaltofen,zy99}@cs.duke.edu
https://users.cs.duke.edu/~elk27
[3] Laboratoire Jean Kuntzmann, Univ. Grenoble Alpes, CNRS,
38058 Grenoble Cedex 09, France
clement.pernet@univ-grenoble-alpes.fr
http://ljk.imag.fr/membres/Clement.Pernet/

Abstract. We generalize Hermite interpolation with error correction, which is the methodology for multiplicity algebraic error correction codes, to Hermite interpolation of a rational function over a field K from function and function derivative values.

We present an interpolation algorithm that can locate and correct $\leq E$ errors at distinct arguments $\xi \in \mathsf{K}$ where at least one of the values or values of a derivative is incorrect. The upper bound E for the number of such ξ is input. Our algorithm sufficiently oversamples the rational function to guarantee a unique interpolant. We sample $(f/g)^{(j)}(\xi_i)$ for $0 \leq j \leq \ell_i$, $1 \leq i \leq n$, ξ_i distinct, where $(f/g)^{(j)}$ is the j-th derivative of the rational function f/g, $f, g \in \mathsf{K}[x]$, $\mathrm{GCD}(f, g) = 1$, $g \neq 0$, and where $N = \sum_{i=1}^{n}(\ell_i + 1) \geq D_f + D_g + 1 + 2E + 2\sum_{k=1}^{E} \ell_k$; D_f is an upper bound for $\deg(f)$ and D_g an upper bound for $\deg(g)$, which are input to our algorithm. The arguments ξ_i can be poles, which is truly or falsely indicated by a function value ∞ with the corresponding $\ell_i = 0$. Our results remain valid for fields K of characteristic $\geq 1 + \max_i \ell_i$. Our algorithm has the same asymptotic arithmetic complexity as that for classical Hermite interpolation, namely $N(\log N)^{O(1)}$.

For polynomials, that is, $g = 1$, and a uniform derivative profile $\ell_1 = \cdots = \ell_n$, our algorithm specializes to the univariate multiplicity code decoder that is based on the 1986 Welch-Berlekamp algorithm.

Keywords: Hermite interpolation · Cauchy interpolation · Error correction codes · Multiplicity codes · List decoding

* This research was supported by the National Science Foundation under Grant CCF-1717100 (Kaltofen and Yang).

1 Introduction

Algebraic error correction codes are based on interpolating a polynomial f from its values $a_i = f(\xi_i)$ at distinct argument scalars ξ_i, when some of the inputs $\hat{a}_\lambda$ for the evaluations are incorrect, namely $\hat{a}_\lambda \neq a_\lambda$. The coefficients of f are from a field K, as are the arguments ξ_i and the list of correct and incorrect evaluations $\hat{a}_i$. The 1960 algorithm by Irving Reed and Gustave Solomon [16] reconstructs a polynomial f of degree $\leq D$ from $n = D + 1 + 2E$ values $\hat{a}_i$ when $\leq E$ of the values are incorrect, namely $|\{\lambda \mid \hat{a}_\lambda \neq f(\xi_\lambda)\}| \leq E$. The number of evaluations is optimal: for $n = D + 2E$ there may exist two polynomials that interpolate with $\leq E$ errors. The Reed-Solomon decoder generalizes to rational functions $f/g \in \mathsf{K}(x)$ with $n = D_f + D_g + 1 + 2E$, where $D_f \geq \deg(f)$, $D_g \geq \deg(g)$ [1]. Decoding can be performed by the extended Euclidean Algorithm [19,21] or by solving a linear system [7]. Lemma 3.2 in [7] shows that for $D_f = \deg(f)$, $D_g = \deg(g)$, $n - 1$ evaluations are always insufficient to correct E errors.

Multiplicity codes [17,15,4,5,11,13,3] generalize the Reed-Solomon problem to the Hermite interpolation problem with error correction. In classical (error-free) Hermite interpolation one reconstructs a polynomial (or rational function) from the values of the polynomial and its derivatives. The classical algorithm of divided differences can reconstruct f from $a_{i,j} = f^{(j)}(\xi_i)$, where $\xi_1, \ldots, \xi_n \in \mathsf{K}$ are distinct scalars, $f^{(j)}$ is the j-th derivative of f, and $0 \leq j \leq \ell_i$ with $(\ell_1 + 1) + \cdots + (\ell_n + 1) = D + 1$. We shall assume that the characteristic of the field K is either 0 or $\geq 1 + \max \ell_i$. For the case that $n = 1$, one has $f(x) = \sum_{0 \leq j \leq \ell_1} f^{(j)}(\xi_1)/j!\,(x - \xi_1)^j$. As in the Reed-Solomon decoding problem, one assumes that some inputs $\hat{a}_{i,j} \neq f^{(j)}(\xi_i)$. It is clear that not all evaluation profiles $(\ell_1, \ldots, \ell_n)$ with $\sum_{i=1}^{n}(\ell_i + 1) = D + 1 + 2E$ are decodable when there are $\leq E$ errors. For example, if $n = 1$ the oversampled derivatives cannot reveal all errors, since $f^{(j)}(x) = 0$ for $j \geq \deg(f) + 1$. Furthermore, if all $\hat{a}_{i,0}$ are erroneous, the constant coefficient of f is unrecoverable. Example 1 below shows that if one has $D = \deg(f)$, $n = D + 2E$ and $\ell_1 = \cdots = \ell_{2E} = 1$ and $\ell_{2E+1} = \cdots = \ell_{D+2E} = 0$, that is $N = D + 4E$, there may be a second polynomial of degree $\leq D$ that fits all but E evaluations.

The reason why, in the Hermite case, one may have to match every bad value with more than one additional good value, in contrast to the Reed-Solomon decoder, is apparent from the Birkhoff generalization of the Hermite interpolation problem. In Birkhoff interpolation one does not have all consecutive derivatives at a scalar ξ_i. Schoenberg [18] uses a matrix $\Theta = [\theta_{i,j}]_{1 \leq i \leq n, 0 \leq j \leq D} \in \{0,1\}^{n \times (D+1)}$ with exactly $(D + 1)$ 1-entries. If $\theta_{i,j} = 1$ then the evaluation $a_{i,j} = f^{(j)}(\xi_i)$ is input. For $\mathsf{K} = \mathbb{R}$ one asks for which Θ's one always gets a unique interpolant. The Pólya-Schoenberg Theorem states that for $n = 2$ there is a unique interpolant if and only if $\forall j, 0 \leq j \leq D-1$: $\sum_{0 \leq \mu \leq j}(\theta_{1,\mu} + \theta_{2,\mu}) \geq j+1$. If those Pólya conditions are violated, then either there are more than one solution or there is no solution. In the error correction setting, for example, in erasure codes, the errors invalidate evaluations, that is, set $\theta_{i,j} = 0$ whenever $\hat{a}_{i,j}$ is an error. Thus the remaining good points may constitute an (oversampled) Birkhoff problem that does not have one unique solution. For example,

for $f(x) = (x^2 - 1^2)(x^2 - 7^2)$ we have $f(x)' = 4x(x^2 - 5^2)$, so $f(-1) = f(1) = f(-7) = f(7) = 0$, $f'(0) = f'(-5) = f'(5) = 0$ is interpolated at 7 good values by the polynomials 0 and f (see also Example 2). Multiplicity code decoders also need to locate the erroneous locations. Our problem is more difficult: the correct values are those of a rational function, not of a polynomial.

The algorithms for error-correcting Hermite interpolation of polynomials and rational functions, in analogy to the Pólya conditions, interpolate a unique polynomial or rational function from its upper bounds for the degrees and the number of errors by sufficient oversampling. Suppose the profile of derivatives at each distinct argument ξ_i $(1 \le i \le n)$ is sorted: $\ell_1 \ge \cdots \ge \ell_n \ge 0$. Again, we assume that the characteristic of K is either 0 or $\ge \ell_1 + 1$. For a rational function $f/g \in \mathsf{K}(x)$ we input $D_f \ge \deg(f)$, $D_g \ge \deg(g)$, $\hat{a}_{i,j} \in \mathsf{K}$ for $1 \le i \le n$ and $0 \le j \le \ell_i$, and E such that for $\le E$ of all arguments ξ_i there is an error at least at one j:

$$E \ge |\{\, i \;\mid\; 1 \le i \le n \text{ and } \exists j, 0 \le j \le \ell_i \colon \hat{a}_{i,j} \ne (f/g)^{(j)}(\xi_i)\,\}|,$$

where $|\ldots|$ is the number of elements in the set. We use the fresh symbol $\infty = (f/g)(\xi_i)$ if $g(\xi_i) = 0$ and allow both false (non-pole) scalars $\hat{a}_{i,j} \in \mathsf{K}$ at such poles, as well as false poles $\hat{a}_{i,j} = \infty$ when $g(\xi_i) \ne 0$. We shall assume that the only $\hat{a}_{i,j} = \infty$ are at evaluations $j = 0$ and that then no derivatives are present, that is, $\ell_i = 0$. If $\hat{a}_{i,j} = \infty \ne \hat{a}_{i,k}$ for some $j \ne k$ then one of the values is erroneous, unless the characteristic of K is positive and $\le D_g$. In that case, the list of values $\hat{a}_{i,0}, \hat{a}_{i,1}, \ldots$ at ξ_i is pre-processed: see Remark 2. Note that without errors, f/g cannot be interpolated at a single argument ξ_1 when all derivative values are ∞.

Our algorithm recovers f/g if the number of evaluations, N, at n distinct ξ_i satisfies

$$N \stackrel{\text{def}}{=} \sum_{i=1}^{n}(\ell_i+1) = D_f + D_g + 1 + 2\sum_{i=1}^{E}(\ell_i+1) = D_f + D_g + 1 + 2E + 2\sum_{i=1}^{E}\ell_i \quad (1)$$

(see Theorem 1). The equation (1) implies $2E + 1 \le n$ (see (6)). Note that if $N <$ the right-side of (1), one needs to increase either n or $\ell_{E+1}, \ldots, \ell_n$ and sample more values. If $N >$ the right-side of (1), one can decrease $\ell_n, \ldots, \ell_{E+1}$ and/or n. If equality in (1) is achieved, further reduction of oversampling may be possible while preserving (1); see Remark 4. For polynomial interpolation we can set $D_g = 0$. In relation to Example 1: with $D = \deg(f), g = 1$, if $\ell_1 = \cdots = \ell_{2E+1} = 1$ and $\ell_{2E+2} = \cdots = \ell_{D+2E} = 0$ then f is recovered uniquely from $N = D + 4E + 1$ evaluations with $\le E$ errors. For $\ell_i = 0$ for all i, our algorithm specializes to rational function recovery with errors with $n = N = D_f + D_g + 1 + 2E$.

1.1 Comparison to Multiplicity Code Decoders

Multiplicity codes are based on Hermite polynomial interpolation with error correction, that is, $D_f = D$, $D_g = 0$. In [11] the following parameter settings

are used: $n = q$ and the field of scalars is $\mathsf{K} = \mathbb{F}_q$, a finite field of q elements. The number of derivatives is uniformly $\ell_1 = \cdots = \ell_q = s - 1$. There are $\leq E = (sq - D - 1)/(2s)$ indices λ_κ where at least one of the s derivative values $\hat{a}_{\lambda_\kappa, j}$ $(0 \leq j \leq s - 1)$ is an error. At each error index λ_κ, there can be as many as s errors, for a total of $(sq - D - 1)/2$ errors, the latter of which is the degree of the error locator polynomial in [11, Section 3.1]. Multiplicity codes then recover the code polynomial from the $N = sq$ values $\hat{a}_{i,j}$ for $1 \leq i \leq q$ and $0 \leq j \leq s - 1$, which agrees with the right-side of (1): $D + 1 + 2Es = sq$. Our decoders here allow for unequal ℓ_i.

Our main contribution is the generalization to Hermite interpolation of rational functions from such partially erroneous values, including the handling of arguments at roots of the denominator, that is, poles. An important idea behind Algorithm 5.1 is from the algorithm in [20] (as cited in [6]) for Hermite rational function interpolation, which in turn is based on Cauchy interpolation via the extended Euclidean algorithm. Our algorithm essentially performs Warner's algorithm, now on an unreduced fraction of polynomials, where both numerator and denominator are multiplied with the error locator polynomial, which the Cauchy interpolation algorithm computes (see Lemma 1). The Welch-Berlekamp decoder for Reed-Solomon codes [21] and its generalization to multiplicity code decoders [11, Section 3.1.1] also has our interpretation of solving such a Cauchy problem.

Because of derivatives, in the Hermite setting the roots of the error locator polynomial have multiplicities. With our assumption that $\hat{a}_{i,j} = \infty$ only if $j = \ell_i = 0$, we can prove that the N values in (1) are sufficient for unique recovery if there are $\leq E$ arguments ξ_i with some $\hat{a}_{i,j}$ being an error (see Theorem 1).

The half-GCD algorithm [14] and fast Hermite interpolation algorithms [2] then yield an arithmetic complexity of $N(\log N)^{O(1)}$. We note that the uniqueness of the interpolant for the error-free Hermite rational function problem implies uniqueness with errors when oversampled at N points (1), which yields a linear system for the coefficients of the unreduced numerator and denominator polynomials. Our approach computes a solution via the extended Euclidean algorithm and additionally optimizes the required polynomial division: see Remark 3. Our Algorithm 5.1 also diagnoses if no valid rational function interpolant exists, which can be used to perform list-decoding: see Remark 6.

2 Polynomial Hermite Interpolation

Let $n \geq 1$, $\xi_i \in \mathsf{K}$ for $1 \leq i \leq n$ be distinct values, $\ell_1 \geq \ell_2 \geq \cdots \geq \ell_n \geq 0$, $a_{i,j} \in \mathsf{K}$ for $1 \leq i \leq n$ and $0 \leq j \leq \ell_i$. Suppose that the characteristic of K is either 0 or $\geq \ell_1 + 1$. For $d = (\ell_1 + 1) + \cdots + (\ell_n + 1) - 1$ there exists a unique $f \in \mathsf{K}[x]$ with $\deg(f) \leq d$ such that $a_{i,j} = f^{(j)}(\xi_i)$ for $1 \leq i \leq n$ and $0 \leq j \leq \ell_i$ where $f^{(j)}(x)$ is the j-th derivative of $f(x) = c_d x^d + \cdots + c_1 x + c_0$ defined by

$$f^{(j)}(x) = \Big(\sum_{\delta=0}^{d} c_\delta x^\delta \Big)^{(j)} = \sum_{\delta=j}^{d} c_\delta \, \delta(\delta - 1) \cdots (\delta - j + 1) \, x^{\delta - j}. \tag{2}$$

We note that if K has any characteristic, the rational function field $\mathsf{K}(x)$ is a differential field with the derivative $'$ being a function satisfying $c' = 0$ for all $c \in \mathsf{K}$, $x' = 1$ and $(F+G)' = F'+G'$ and $(FG)' = F'G+FG'$ for all $F, G \in \mathsf{K}(x)$, which yields (2). See also Remark 2 below.

The algorithm of divided differences, which goes back to at least Guo Shoujing (1231–1316), computes the coefficients $\bar{c}_{i,j}$ forming the decomposition of $f(x)$ in mixed-shifted-basis representation

$$f(x) = \sum_{\nu=1}^{n} \sum_{\mu=0}^{\ell_\nu} \bar{c}_{\nu-1,\mu} \Big(\prod_{\kappa=1}^{\nu-1} (x-\xi_\kappa)^{\ell_\kappa+1} \Big) (x-\xi_\nu)^\mu.$$

For $1 \le i \le n$ and $0 \le j \le \ell_i$ the interpolant

$$H_{i,j}(x) = \sum_{\nu=1}^{i} \sum_{\mu=0}^{\ell'_\nu} \bar{c}_{\nu-1,\mu} \Big(\prod_{\kappa=1}^{\nu-1} (x-\xi_\kappa)^{\ell_\kappa+1} \Big) (x-\xi_\nu)^\mu$$

$$\text{with } \ell'_\nu = \ell_\nu \text{ for } \nu < i \text{ and } \ell'_i = j,$$

fits the values $a_{\nu,\mu}$ for $1 \le \nu \le i$ and $0 \le \mu \le \ell'_\nu$. We compute the next $H_{i,j+1}$ ($j < \ell_i$) or $H_{i+1,0}$ ($j = \ell_i$) to fit $a_{i,j+1}$ or $a_{i+1,0}$, respectively. For $j < \ell_i$ we have

$$H_{i,j+1}(x) = H_{i,j}(x) + \bar{c}_{i-1,j+1} G_{i,j+1}(x),$$

$$G_{i,j+1}(x) = \Big(\prod_{\kappa=1}^{i-1} (x-\xi_\kappa)^{\ell_\kappa+1} \Big) (x-\xi_i)^{j+1}.$$

Note that $G^{(\mu)}_{i,j+1}(\xi_\nu) = 0$ for $1 \le \nu \le i$ and $0 \le \mu \le \ell'_\nu$, so $H_{i,j+1}(x)$ interpolates all of $H_{i,j}$'s values. Finally, $G^{(j+1)}_{i,j+1}(\xi_i) = \big(\prod_{\kappa=1}^{i-1} (\xi_i - \xi_\kappa)^{\ell_\kappa+1} \big)\ (j+1)!$, which is $\neq 0$ by our assumption that the characteristic of K is 0 or $\ge \ell_1 + 1$. Therefore $H^{(j+1)}_{i,j+1}(\xi_i) = a_{i,j+1}$ has a unique solution $\bar{c}_{i-1,j+1}$. The case $H_{i+1,0}$ is similar.

Algorithms for computing the Hermite interpolant f in soft-linear arithmetic complexity go back to [2].

3 Rational Function Recovery

Our algorithms are a generalization of the Cauchy interpolation algorithm for rational functions, which is based on the extended Euclidean algorithm. We now state the key lemma, which goes back to Leopold Kronecker's algorithm for computing Padé approximants.

Lemma 1. *Let d and e be non-negative integers, and let $H(x) \in \mathsf{K}[x]$, K an arbitrary field,* $\deg(H) \le d+e$; *furthermore, let ξ_i, $1 \le i \le d+e+1$, be not necessarily distinct elements in K.*
1. *Define $r_0 = \prod_{i=1}^{d+e+1} (x-\xi_i)$ and $r_1(x) = H(x)$. Now let $r_\rho(x)$, $q_\rho(x) \in \mathsf{K}[x]$ be the ρ-th remainder and quotient respectively, in the Euclidean polynomial remainder sequence*

$$r_{\rho-2}(x) = q_\rho(x) r_{\rho-1}(x) + r_\rho(x),\ \deg(r_\rho) < \deg(r_{\rho-1}) \text{ for } \rho \ge 2.$$

In the exceptional case $H = 0$ the sequence is defined to be empty. Finally, let $s_\rho(x), t_\rho(x) \in \mathsf{K}[x]$ be the multipliers in the extended Euclidean scheme $s_\rho r_1 + t_\rho r_0 = r_\rho$, namely,

$$s_0 = t_1 = 0, \quad t_0 = s_1 = 1,$$
$$s_\rho = s_{\rho-2} - q_\rho\, s_{\rho-1}, \quad t_\rho = t_{\rho-2} - q_\rho\, t_{\rho-1} \quad \text{for } \rho \geq 2.$$

Then there exists an index $\gamma \geq 1$, such that $\deg(r_\gamma) \leq d < \deg(r_{\gamma-1})$ and

$$r_\gamma \equiv s_\gamma H \pmod{r_0} \quad \text{and} \quad \deg(s_\gamma) \leq e. \tag{3}$$

2. *Let $R(x), S(x) \in \mathsf{K}[x]$ be another solution of (3), namely*

$$R \equiv S\,H \pmod{r_0} \quad d \geq \deg(R) \text{ and } e \geq \deg(S). \tag{4}$$

Then $s_\gamma R = r_\gamma S$. If furthermore $\mathrm{GCD}(R, S) = 1$ then $R = c\,r_\gamma$, $S = c\,s_\gamma$ for some $c \in \mathsf{K} \setminus \{0\}$.

Proof. See [8, Lemma 1].

4 Error-Correcting Hermite Interpolation

Let $f(x) \in \mathsf{K}[x]$ be a univariate polynomial and D be an upper bound of $\deg(f)$. One is given a set of n distinct arguments $\xi_1, \ldots, \xi_n \in \mathsf{K}$, and for each argument ξ_i, one is given a row vector

$$\widehat{A}_{i,*} = [\hat{a}_{i,0}, \ldots, \hat{a}_{i,\ell_i}] \in \mathsf{K}^{1\times(\ell_i+1)}.$$

We call $\hat{a}_{i,j}$ an error if $\hat{a}_{i,j} \neq f^{(j)}(\xi_i)$, and we call $\widehat{A}_{i,*}$ error-free if $\hat{a}_{i,j} = f^{(j)}(\xi_i)$ for all $j = 0, \ldots, \ell_i$. Let $\{\lambda_1, \ldots, \lambda_k\} \subset \{1, \ldots, n\}$ be the set of indices where every row vector $\widehat{A}_{\lambda_1,*}, \ldots, \widehat{A}_{\lambda_k,*}$ has at least one error, and let $E \geq k$ (if all row vectors $\widehat{A}_{1,*}, \ldots, \widehat{A}_{n,*}$ are error-free, then let $E = k = 0$). As in Section 2 we assume that K is a field of characteristic 0 or $\geq 1 + \max_i \ell_i$. To uniquely recover $f(x)$, a condition $n \geq 2E + 1$ is necessary: if $n = 2E$, one can have for $f \in \mathsf{K}[x]$ with E errors $\hat{a}_{i,0} = f(\xi_i) + 1 \neq f(\xi_i)$ where $1 \leq i \leq E$, and for $f + 1$ with E errors $\hat{a}_{i,0} = f(\xi_i) \neq f(\xi_i) + 1$ where $E + 1 \leq i \leq 2E$, that is both f and $f + 1$ are valid interpolants with E errors. Without loss of generality, we assume that $\ell_1 \geq \cdots \geq \ell_n \geq 0$, and let

$$\widehat{A} = \begin{bmatrix} \widehat{A}_{1,*} \\ \vdots \\ \widehat{A}_{n,*} \end{bmatrix} \in \left(\mathsf{K}^{1\times(\ell_1+1)} \cup \cdots \cup \mathsf{K}^{1\times(\ell_n+1)}\right)^n$$

be the vector of those value row vectors. The total number of values in $\widehat{A}$ is $N = \sum_{i=1}^{n}(\ell_i + 1)$. We will show how to recover the polynomial $f(x)$ from the points $\xi_1, \ldots, \xi_n$ and the values in $\widehat{A}$ by the extended Euclidean algorithm if

$$n + \sum_{i=E+1}^{n} \ell_i = D + 1 + 2E + \sum_{i=1}^{E} \ell_i. \tag{5}$$

Note that the equality (5) is equivalent to $\sum_{i=1}^{n}(\ell_i+1) = D+1+2E+2\sum_{i=1}^{E}\ell_i$, which means $\widehat{A}$ has $N = D+1+2E+2\sum_{i=1}^{E}\ell_i$ values. Furthermore, the equality (5) implies that $n \geq 2E+1$, because recovery is unique and for $n \leq 2E$ we would have the ambiguous solution above; more explicitly, for $n \leq 2E$ we have the contradiction

$$n + \sum_{i=E+1}^{n} \ell_i \leq 2E + \ell_{E+1}E < D + 1 + 2E + \sum_{i=1}^{E} \ell_i. \tag{6}$$

We remark that the condition (5) can be relaxed to $n + \sum_{i=E+1}^{n}\ell_i \geq D + 1 + 2E + \sum_{i=1}^{E}\ell_i$, because in that case, one can decrease $\ell_n, \ell_{n-1}, \ldots, \ell_{E+1}$ successively to achieve the equality (5). In fact, even if the equality is satisfied, one may still be able to reduce the ℓ_i's on both sides so that the algorithm can recover $f(x)$ with fewer values (see Remark 4).

4.1 Error-correcting polynomial Hermite interpolation

Input: ▸ A field K, nonnegative integers $D, E \in \mathbb{Z}_{\geq 0}$;
- ▸ A set of distinct points $\{\xi_1, \ldots, \xi_n\} \subseteq \mathsf{K}$;
- ▸ A list of n row vectors $\widehat{A} = [\widehat{A}_{i,*}]_{1\leq i\leq n}$ where
  - ▸ $\ell_1 \geq \cdots \geq \ell_n \geq 0$; the characteristic of K is either 0 or $\geq \ell_1 + 1$;
  - ▸ $\widehat{A}_{i,*} = [\hat{a}_{i,0}, \ldots, \hat{a}_{i,\ell_i}]$;
  - ▸ $n + \sum_{i=E+1}^{n}\ell_i = D + 1 + 2E + \sum_{i=1}^{E}\ell_i$ ($\Longrightarrow n \geq 2E+1$).

Output: ▸ The interpolant $f(x) \in \mathsf{K}[x]$ and the error locator polynomial $\Lambda(x)$ in $\mathsf{K}[x]$ which satisfy
- ▸ $\deg(f) \leq D$;
- ▸ $\Lambda(x) = 1$ or $\Lambda(x) = \prod_{\kappa=1}^{k}(x-\xi_{\lambda_\kappa})^{\delta_\kappa}$ where $\xi_{\lambda_1}, \ldots, \xi_{\lambda_k}$ are distinct and $k \leq E$;
- ▸ row vector $\widehat{A}_{i,*}$ is error free if and only if $i \notin \{\lambda_1, \ldots, \lambda_k\}$;
- ▸ $\delta_\kappa = \ell_{\lambda_\kappa} + 1 - \min\{j \mid f^{(j)}(\xi_{\lambda_\kappa}) \neq \hat{a}_{\lambda_\kappa,j}\}$.
- ▸ Or a message indicating there is no such interpolant.

1. *If* $\hat{a}_{i,j} = 0$ *for all* $i = 1, \ldots, n$ *and* $j = 0, \ldots, \ell_i$, *then return* $f = 0$ *and* $\Lambda = 1$.
2. *Compute the Hermite interpolant* $H(x) \in \mathsf{K}[x]$ *of the data set* $\{(\xi_i;\ \hat{a}_{i,0}, \ldots, \hat{a}_{i,\ell_i}) \mid i = 1, \ldots, n\}$, *namely compute a polynomial* $H(x) \in \mathsf{K}[x]$ *such that* $H^{(j)}(\xi_i) = \hat{a}_{i,j}$.
   If $E = 0$ *and* $\deg(H) \leq D$, *then return* $f = H$ *and* $\Lambda = 1$. *If* $E = 0$ *and* $\deg(H) > D$, *then return a message indicating there is no such interpolant.*
3. *Let* $r_0 = (x-\xi_1)^{\ell_1+1}\cdots(x-\xi_n)^{\ell_n+1}$, $r_1 = H$, $s_0 = 0$, $s_1 = 1$, *and* $\rho = 2$.
   3a. *Compute the* ρ*-th Euclidean polynomial remainder* r_ρ *and the multiplier* s_ρ *in the extended Euclidean scheme* $s_\rho r_1 + t_\rho r_0 = r_\rho$, *namely*

$$r_\rho(x) = r_{\rho-2}(x) - q_\rho(x)r_{\rho-1}(x),\ \deg(r_\rho) < \deg(r_{\rho-1}),$$
$$s_\rho(x) = s_{\rho-2}(x) - q_\rho(x)s_{\rho-1}(x).$$

3b. *If* $\deg(r_\rho) \le D + E + \sum_{i=1}^{E} \ell_i$, *then let* $\gamma = \rho$ *and* go to Step 4.
3c. *Otherwise, let* $\rho = \rho + 1$ *and go to* Step 3a.
By the half-GCD algorithm, Step 3 can be performed in soft-linear arithmetic complexity.

4. *If* s_γ *divides* r_γ, *then factorize* s_γ *over* K; *if* s_γ *has* $\le E$ *distinct factors, then go to* Step 5. *Otherwise return a message indicating there is no such interpolant.*

5. *Compute* $f = r_\gamma/s_\gamma$. *If* $\deg(f) \le D$, *then return* f *and* $\Lambda = s_\gamma/\mathrm{lc}(s_\gamma)$, *where* $\mathrm{lc}(s_\gamma)$ *is the leading coefficent of* s_γ. *Otherwise return a message indicating there is no such interpolant.*
Step 3 computes (r_γ, s_γ) as in Lemma 1 with $d = D + E + \sum_{i=1}^{E} \ell_i$ and $e = E + \sum_{i=1}^{E} \ell_i$. We will prove that if there is an interpolant $f(x) \in \mathsf{K}[x]$ which satisfies the output specifications, then $r_\gamma/s_\gamma = (f\Lambda)/\Lambda = f$ (see Lemma 2) and $\Lambda = s_\gamma/\mathrm{lc}(s_\gamma)$ (see Lemma 3). Here $s_\gamma \neq 0$ because $\mathrm{GCD}(s_\gamma, t_\gamma) = 1$ for $\gamma \ge 2$.
On the other hand, if the polynomial r_γ/s_γ computed in Step 5 has degree $\le D$, then it satisfies the output specifications, which we will prove as a special case in Lemma 5. Therefore, we can check the validity of r_γ/s_γ without computing all the values $(r_\gamma/s_\gamma)^{(j)}(\xi_i)$ for $i = 1, \ldots, n$ and $j = 0, \ldots, \ell_i$.

Lemma 2. *With the notation as in Algorithm 4.1, if there is a polynomial* $f \in \mathsf{K}[x]$ *which satisfies the output specifications, then*

$$f\Lambda \equiv H\Lambda \pmod{r_0}. \tag{7}$$

Moreover, $r_\gamma/s_\gamma = (f\Lambda)/\Lambda = f$, *which implies the interpolant* f *is unique.*

Proof. Recall that $\Lambda(x) = (x - \xi_{\lambda_1})^{\delta_1} \cdots (x - \xi_{\lambda_k})^{\delta_k}$ is the error locator polynomial where

(i) $\xi_{\lambda_1}, \ldots, \xi_{\lambda_k}$ are the arguments with erroneous values, that is, for indices $\lambda_\kappa \in \{\lambda_1, \ldots, \lambda_k\}$, there exists $j \in \{0, \ldots, \ell_{\lambda_\kappa}\}$ such that $f^{(j)}(\xi_{\lambda_\kappa}) \neq \hat{a}_{\lambda_\kappa, j}$;
(ii) $\delta_\kappa = \ell_{\lambda_\kappa} + 1 - \min\{j \mid f^{(j)}(\xi_{\lambda_\kappa}) \neq \hat{a}_{\lambda_\kappa, j}\}$, $\kappa = 1, \ldots, k$.

Since $r_0 = (x - \xi_1)^{\ell_1 + 1} \cdots (x - \xi_n)^{\ell_n + 1}$, proving the equality (7) is equivalent to proving $(x - \xi_i)^{\ell_i + 1}$ divides $(f\Lambda - H\Lambda)$ for all $i = 1, \ldots, n$, which is again equivalent to proving the following equality:

$$(f\Lambda)^{(j)}(\xi_i) = (H\Lambda)^{(j)}(\xi_i) \text{ for all } i = 1, \ldots, n \text{ and } j = 0, \ldots, \ell_i. \tag{8}$$

If $i \notin \{\lambda_1, \ldots, \lambda_k\}$ then $f^{(j)}(\xi_i) = \hat{a}_{i,j} = H^{(j)}(\xi_i)$ for all $j = 0, \ldots, \ell_i$ and (8) follows immediately. For $\xi_{\lambda_\kappa} (1 \le \kappa \le k)$ and $j = 0, \ldots, \ell_{\lambda_\kappa}$,

$$(f\Lambda)^{(j)}(\xi_{\lambda_\kappa}) = \sum_{\tau=0}^{j} \binom{j}{\tau} f^{(j-\tau)}(\xi_{\lambda_\kappa}) \Lambda^{(\tau)}(\xi_{\lambda_\kappa}),$$

$$(H\Lambda)^{(j)}(\xi_{\lambda_\kappa}) = \sum_{\tau=0}^{j} \binom{j}{\tau} H^{(j-\tau)}(\xi_{\lambda_\kappa}) \Lambda^{(\tau)}(\xi_{\lambda_\kappa}).$$

Moreover,

$$\Lambda^{(\tau)}(\xi_{\lambda_\kappa}) = 0 \text{ if } \tau < \delta_\kappa, \tag{9}$$

$$f^{(j-\tau)}(\xi_{\lambda_\kappa}) = \hat{a}_{\lambda_\kappa, j-\tau} = H^{(j-\tau)}(\xi_{\lambda_\kappa}) \text{ if } \tau \geq \delta_\kappa. \tag{10}$$

The equality (9) holds because $\Lambda(x)$ has a factor $(x - \xi_{\lambda_\kappa})^{\delta_\kappa}$; the equality (10) follows from

$$j - \tau \leq \ell_{\lambda_\kappa} - \tau < \ell_{\lambda_\kappa} + 1 - \delta_\kappa = \min\{j \mid f^{(j)}(\xi_{\lambda_\kappa}) \neq \hat{a}_{\lambda_\kappa, j}\}.$$

Therefore, $f^{(j-\tau)}(\xi_{\lambda_\kappa})\Lambda^{(\tau)}(\xi_{\lambda_\kappa}) = H^{(j-\tau)}(\xi_{\lambda_\kappa})\Lambda^{(\tau)}(\xi_{\lambda_\kappa})$ for all $\tau = 0, \ldots, j$, and (8) is proved.

Let $d = D + E + \sum_{i=1}^{E} \ell_i$, $e = E + \sum_{i=1}^{E} \ell_i$, $R = f\Lambda$, and $S = \Lambda$. Then $\deg(r_0) = d + e + 1$, $\deg(H) \leq d + e$, $\deg(R) \leq d$, and $\deg(S) \leq e$. By Lemma 1, $r_\gamma / s_\gamma = R/S = f\Lambda/\Lambda = f$.

Lemma 3. *With the notation as in Algorithm 4.1, we have* $\Lambda = s_\gamma / \mathrm{lc}(s_\gamma)$.

Proof. By Lemma 2, $r_\gamma = s_\gamma f$. On the other hand, $r_\gamma \equiv s_\gamma H \pmod{r_0}$. Therefore

$$s_\gamma f \equiv s_\gamma H \pmod{r_0}. \tag{11}$$

Let $(x - \xi_{\lambda_\kappa})^{\delta_\kappa}$ be a factor of Λ and denote $\epsilon_\kappa = \min\{\, j \mid f^{(j)}(\xi_{\lambda_\kappa}) \neq \hat{a}_{\lambda_\kappa, j}\}$, then

$$\delta_\kappa + \epsilon_\kappa = \ell_{\lambda_\kappa} + 1.$$

Since $(x - \xi_{\lambda_\kappa})^{\ell_{\lambda_\kappa}+1}$ is a factor of r_0, it follows from (11) that $(x - \xi_{\lambda_\kappa})^{\delta_\kappa + \epsilon_\kappa}$ divides $(f - H)s_\gamma$. In addition, $\epsilon_\kappa = \min\{\, j \mid f^{(j)}(\xi_{\lambda_\kappa}) \neq \hat{a}_{\lambda_\kappa, j} = H^{(j)}(\xi_{\lambda_\kappa})\}$ implies that

$$\mathrm{GCD}((x - \xi_{\lambda_\kappa})^{\delta_\kappa + \epsilon_\kappa}, f - H) = (x - \xi_{\lambda_\kappa})^{\epsilon_\kappa}.$$

Therefore $(x - \xi_{\lambda_\kappa})^{\delta_\kappa}$ divides s_γ, and so Λ divides s_γ.

Assume that $s_\gamma = \Lambda w$ for some $w \in \mathsf{K}[x]$, then $r_\gamma = s_\gamma f = f\Lambda w$, so the extended Euclidean scheme $s_\gamma r_1 + t_\gamma r_0 = r_\gamma$ becomes

$$f\Lambda w = H\Lambda w + t_\gamma r_0.$$

However, from Lemma 2, we know that $f\Lambda \equiv H\Lambda \pmod{r_0}$, which means there is $\tilde{t} \in \mathsf{K}[x]$ such that

$$f\Lambda = H\Lambda + \tilde{t}\, r_0.$$

Therefore $t_\gamma = \tilde{t}\, w$, and this leads to $w \in \mathsf{K}$ because $\mathrm{GCD}(s_\gamma, t_\gamma) = 1$. Since the leading coefficient of Λ is 1, we have $s_\gamma = \mathrm{lc}(s_\gamma)\Lambda$.

Note that Reed-Solomon decoding is a special case of our setting where $\ell_1 = \cdots = \ell_n = 0$ and $n = N = D+1+2E$. When $\ell_1 \geq 1$ and $n \leq D+2E$, our method requires $N = D + 1 + 2E + 2\sum_{i=1}^{E} \ell_i$ which is more than the values required by Reed-Solomon decoding. However, in some cases, the number of values we required is necessary for computing a unique interpolant $f(x)$, that is, there can be two valid interpolants if fewer values are given. We show this by the following example.

Example 1. Let $\mathsf{K} = \text{algclo}(\mathbb{Q}) \cap \mathbb{R}$ be the real algebraic closure of $\mathbb{Q}$, and assume that $2E \leq D-1$. Let $\xi_{2E+1}, \ldots, \xi_{D+2E}$ be D distinct points in K and

$$f(x) = \prod_{i=2E+1}^{D+2E} (x - \xi_i).$$

By Rolle's theorem, $f(x)'$ has $D-1$ distinct roots in K, which allows us to choose $2E$ distinct points $\xi_1, \ldots, \xi_{2E}$ from these roots. Moreover, all the points $\xi_1, \ldots, \xi_{2E}, \xi_{2E+1}, \ldots, \xi_{D+2E}$ are distinct. Now we have $f'(\xi_1) = \cdots = f'(\xi_{2E}) = 0$ and $f(\xi_{2E+1}) = \cdots = f(\xi_{D+2E}) = 0$ Let $\ell_1 = \cdots = \ell_{2E} = 1$, $\ell_{2E+1} = \cdots = \ell_{D+2E} = 0$ and $n = D + 2E$, then $N = \sum_{i=1}^{n} (\ell_i + 1) = D + 4E$. Suppose the N values are given as follows:

$$\left.\begin{array}{lll} \hat{a}_{i,0} = f(\xi_i) & \text{for } i = 1, \ldots, E, \\ \hat{a}_{i,0} = 0 & \text{for } i = E+1, \ldots, D+2E, \\ \hat{a}_{i,1} = 0 & \text{for } i = 1, \ldots, 2E. \end{array}\right\} \tag{12}$$

If the E errors are $\hat{a}_{1,0}, \ldots, \hat{a}_{E,0}$, then 0 is a valid interpolant; if the E errors are $\hat{a}_{E+1,0}, \ldots, \hat{a}_{2E,0}$ then f is a valid interpolant. Thus for the points $\xi_1, \ldots, \xi_{D+2E}$ and the $D + 4E$ values in (12), there are ≥ 2 valid interpolants. □

Example 2. As we have shown in Section 1, from Birkhoff problems with multiple solutions one obtains Hermite interpolation problems with errors that have multiple solutions. For instance, for the polynomial $f(x) = (x^2 - 1^2)^3 (x^2 - 7^2)^3$ we have $f^{(j)}(\xi) = 0$ for $\xi = \pm 1$, $\xi = \pm 7$ and $j = 0, 1, 2$, and $f'(\xi) = 0$ for $\xi = 0$ and $\xi = \pm 5$. Therefore with those $n = 7$ arguments ξ and $\ell_i = 2$ for $1 \leq i \leq 7$, one has both f and the zero polynomial as a solution with $E = 3$ errors at $N = 21 = \deg(f) + 1 + 2E + 2$ values. □

Example 3. If the field of scalars K has finite characteristic $\geq \ell_1 + 1$, our count (5) is optimal for higher derivatives. Let $n = 2E + 1$ and let $\ell_1 = \cdots = \ell_{2E+1} = p - 1$ for a prime number p which is the characteristic of the field of scalars K, whose cardinality is $|\mathsf{K}| \geq 2E + 2$, so that there exist $n + 1$ distinct elements ξ_i in K. Let $f(x) = (x - \xi_1)^p$. Then $f(\xi_1) = 0$ and $f^{(j)}(\xi_i) = 0$ for all $1 \leq i \leq n$ and $1 \leq j \leq \ell_i$. Therefore f and the zero polynomial interpolate all $(2E + 1)p - 2E$ zero values, and E errors cannot be unambiguously corrected from $N = (2E + 1) \deg(f)$ values. If one adds an $(N + 1)$'st value $f(\xi_{n+1})$ then $N + 1 = \deg(f) + 1 + 2E + 2E(p - 1) = (2E + 1)p + 1$ (cf. (5)) and Algorithm 4.1 interpolates a unique polynomial with $\leq E$ erroneous values. □

Remark 1. Let $E_{\text{tot}} \geq |\{\hat{a}_{i,j} \mid f^{(j)}(\xi_i) \neq \hat{a}_{i,j}\}|$ be a bound on the total number of errors. If all $\ell_i \leq 1$, we can prove that $N = 2D + 2E_{\text{tot}}$ is the optimal count in the case that $n \geq 2E_{\text{tot}} + 1$ which is necessary, and that $2E_{\text{tot}} \geq D - 1$ and that the characteristic of K is either 0 or $\geq D + 1$. For $D = 0$ we have $N = n = 2E_{\text{tot}} + 1$. We first show that the zero polynomial is the only interpolant of evaluations that yield 0 at any of $N - 2E_{\text{tot}}$ of the evaluations. If $E_0 \leq 2E_{\text{tot}}$ values $f(\xi_i)$ are removed, a non-zero polynomial of degree D can be zero at the remaining

$n - E_0$ values only if $n - E_0 \le D \iff E_0 \ge n - D$. There are $N - n \ge 0$ values of f', of which one removes $2E_{\text{tot}} - E_0 \le 2E_{\text{tot}} - (n - D)$ values. There remain $\ge N - n - (2E_{\text{tot}} - n + D) = D$ values of f' at distinct arguments, which are zero, which means $f' = 0$ and, by our assumption on the characteristic, $\deg(f) = 0$. Because $f(\xi_i) = 0$ at one of the $n \ge 2E_{\text{tot}} + 1$ arguments ξ_i, $f = 0$.

If there are $N = 2D + 2E_{\text{tot}} - 1$ values, we choose $n = 2E_{\text{tot}} + D$. We know from Example 1 that there exists a non-zero polynomial f and argument values ξ_i for $1 \le i \le n = D + 2E_{\text{tot}}$, such that $f(\xi_i) = 0$ for $i = 2E_{\text{tot}} + 1, \ldots, 2E_{\text{tot}} + D$ and $f'(\xi_i) = 0$ for $i = 1, \ldots, D - 1 \le 2E_{\text{tot}}$.

The number $N = 2D + 2E_{\text{tot}}$ of evaluations is $< D + 1 + 4E_{\text{tot}}$ for $2E_{\text{tot}} \ge D$. By Example 3, there is no unique interpolant for scalar fields K of positive characteristic $\le D$. We shall explain how to interpolate in characteristic 0 or $\ge D + 1$. Let $n' = N - n$ be the number for values for first derivatives. If the number of errors k_0 for the values $\hat{a}_{i,0}$ satisfies $k_0 \le (n - D - 1)/2$, then f can be computed by Reed-Solomon decoding from all $\hat{a}_{i,0}$. If $n \le D + 2k_0$ then $n' = N - n \ge D + 2(E_{\text{tot}} - k_0)$, where $(n' - D)/2 \ge E_{\text{tot}} - k_0 \ge$ the number of errors in $\hat{a}_{i,1}$. One can Reed-Solomon decode f' from all $\hat{a}_{i,1}$ and from it, by assumption on the characteristic of K, compute $f + C$ for an unknown constant C. A majority of the $n \ge 2E_{\text{tot}} + 1$ values $(f + C)(\xi_i)$ must equal $\hat{a}_{i,0}$, which determines the constant coefficient. □

5 The Rational Function Case

Let $f(x), g(x) \in \mathsf{K}[x]$, $g \ne 0$, $\deg(f) \le D_f$, $\deg(g) \le D_g$, $\text{GCD}(f, g) = 1$. One is given a set of n distinct arguments $\xi_1, \ldots, \xi_n \in \mathsf{K}$, and for each argument ξ_i, one is given a row vector

$$\widehat{A}_{i,*} = [\hat{a}_{i,0}, \ldots, \hat{a}_{i,\ell_i}] \in (\mathsf{K} \cup \{\infty\})^{1 \times (\ell_i + 1)}.$$

We call $\hat{a}_{i,j}$ an error if one of the two cases happens: ξ_i is not a pole of $(f/g)^{(j)}$ and $\hat{a}_{i,j} \ne (f/g)^{(j)}(\xi_i)$, or, ξ_i is a pole of $(f/g)^{(j)}$ and $\hat{a}_{i,j} \ne \infty$. Let $\{\lambda_1, \ldots, \lambda_k\} \subset \{1, \ldots, n\}$ be the set of indices where every row vector $\widehat{A}_{\lambda_1,*}, \ldots, \widehat{A}_{\lambda_k,*}$ has at least one error, and let $E \ge k$ (if all row vectors are error-free then let $E = k = 0$). Let $\widehat{A}$ be the list of these row vectors:

$$\widehat{A} = \begin{bmatrix} \widehat{A}_{1,*} \\ \vdots \\ \widehat{A}_{n,*} \end{bmatrix} \in \left((\mathsf{K} \cup \{\infty\})^{1 \times (\ell_1 + 1)} \cup \cdots \cup (\mathsf{K} \cup \{\infty\})^{1 \times (\ell_n + 1)} \right)^n.$$

We assume that $\ell_1 \ge \cdots \ge \ell_n \ge 0$, and the last n_∞ (n_∞ can be zero) rows of $\widehat{A}$ only have one value ∞ and all other rows of $\widehat{A}$ have values in K, that is, we have the input specifications:

- $\ell_1 \ge \cdots \ge \ell_{n - n_\infty} \ge 0$; if the characteristic p of K is > 0, then $p \ge \ell_1 + 1$ is required.

- $\hat{a}_{i,j} \neq \infty$ for all $i = 1, \ldots, n - n_\infty$ and $0 \leq j \leq \ell_i$;
- $\hat{a}_{i,0} = \infty$ and $\ell_i = 0$ for all $i = n - n_\infty + 1, \ldots, n$.

For an arbitrary $\widehat{A}$, we process the inputs as is discussed in the following remark.

Remark 2. If for a location i one has $\hat{a}_{i,j} = \infty$ for all j, then a pole is indicated either truly or falsely. In this case we compress the list to a single value $\hat{a}_{i,0} = \infty$ and reset $\ell_i = 0$. For a true pole, and for characteristic of K either 0 or $\geq \deg(g) + 1$, all values are correct, but the additional $\hat{a}_{i,j} = \infty$ for $j \geq 1$ yield no additional information. In fact, $f(x) = 1/x^D$ cannot be interpolated from the values $f^{(j)}(0) = \infty$ for all $0 \leq j \leq D$ without errors. Our handling of poles is not a restriction of our algorithm, but is in the nature of the Hermite interpolation problem.

If for a location i, the list of values $\hat{a}_{i,0}, \hat{a}_{i,1}, \ldots$ is a mix of both elements $\in \mathsf{K}$ and ∞'s, we remove or truncate the list depending on the characteristic of K.

1. For characteristic of K either 0 or $\geq D_g + 1$, we have for $g(x) = (x - \alpha_1)^{\mu_1} \cdots (x - \alpha_\nu)^{\mu_\nu}$, α_i distinct $\in \mathrm{algclo}(\mathsf{K})$, that

$$\Big(\frac{f(x)}{g(x)}\Big)' = \frac{f(x)'}{g(x)} - \frac{f(x)g(x)'}{g(x)^2} = \frac{f(x)' \prod_i (x - \alpha_i) - f(x) \sum_i \mu_i \prod_{j \neq i} (x - \alpha_j)}{g(x) \prod_i (x - \alpha_i)}, \tag{13}$$

   where the right-side of (13) is a reduced rational function because no α_i is a root of the numerator: $f(\alpha_i) \neq 0$ because f/g is reduced, and $\mu_i \prod_{j \neq i} (\alpha_i - \alpha_j) \neq 0$ in K for all i by our assumption on the characteristic of K. Therefore, if the list $\hat{a}_{i,0}, \hat{a}_{i,1}, \ldots$ is a mix of both elements $\in \mathsf{K}$ and ∞'s, then some values in the list must be errors, so we remove the argument ξ_i and the list of values altogether. We also reduce the number of errors accordingly.
   Note that our algorithms do not account for error distributions and assume the worst case. For instance, if in a list of $\ell_i = 20$ values there is a single ∞, we do not treat ∞ as a likely error. If fact, if there is a burst of errors, that ∞ may be the correct value.
2. For positive characteristic $p \leq \deg(g)$, a mix of ∞'s and field element values may not indicate an error: for $\xi_1 = 0$ and $f/g = (cx^{p+1} + 1)/x^p$, $(f/g)' = c$ and has no pole at 0. For such a field, if $\hat{a}_{i,0} \neq \infty$ and $\hat{a}_{i,j} = \infty$ for some $j \geq 1$, then either $\hat{a}_{i,0}$ or $\hat{a}_{i,j}$ is an error, so we remove the argument ξ_i and the list of values altogether and adjust the number of errors. Otherwise, we truncate the list to a single value $\hat{a}_{i,0} = \infty$ and reset $\ell_i = 0$. □

Now we show how to recover the rational function f/g by the extended Euclidean algorithm with the following condition:

$$n + \sum_{i=E+1}^{n} \ell_i = n + \sum_{i=E+1}^{n-n_\infty} \ell_i = D_f + D_g + 1 + 2E + \sum_{i=1}^{E} \ell_i. \tag{14}$$

Let E_∞ be the number of false poles, namely, $E_\infty = |\{i \mid \hat{a}_{i,0} = \infty,\ g(\xi_i) \neq 0\}|$. Note that $n_\infty \leq D_g + E_\infty$. The condition (14) implies that $n - n_\infty \geq 2(E - E_\infty) + 1$, since otherwise we have the contradiction:

$$n_\infty + (n - n_\infty) + \sum_{i=E+1}^{n-n_\infty} \ell_i \leq (D_g + E_\infty) + 2(E - E_\infty) + (E - 2E_\infty)\ell_{E+1}$$
$$< D_f + D_g + 1 + 2E + \sum_{i=1}^{E} \ell_i.$$

The condition (14) can also be relaxed to $n + \sum_{i=E+1}^{n-n_\infty} \ell_i \geq D_f + D_g + 1 + 2E + \sum_{i=1}^{E} \ell_i$, because in that case, one can always adjust the ℓ_i's to achieve (14) (see Remark 4).

5.1 Error-correcting rational function Hermite interpolation

Input: ▸ A field K, nonnegative integers $D_f, D_g, E \in \mathbb{Z}_{\geq 0}$;
- ▸ A set of distinct points $\{\xi_1, \ldots, \xi_n\} \subset \mathsf{K}$.
- ▸ A list of n row vectors $\widehat{A} = [\widehat{A}_{i,*}]_{1 \leq i \leq n}$ and $n_\infty \in \mathbb{Z}_{\geq 0}$ where
  - ▸ $\ell_1 \geq \cdots \geq \ell_{n-n_\infty} \geq 0$, $\ell_{n-n_\infty+1} = \cdots = \ell_n = 0$;
  - ▸ the characteristic of K is either 0 or $\geq \ell_1 + 1$;
  - ▸ $\widehat{A}_{i,*} = [\hat{a}_{i,0}, \ldots, \hat{a}_{i,\ell_i}]$ and $\hat{a}_{i,j} \in \mathsf{K}$ for all $i = 1, \ldots, n - n_\infty$ and $j = 0, \ldots, \ell_i$;
  - ▸ $\hat{a}_{i,0} = \infty$ for all $i = n - n_\infty + 1, \ldots, n$;
  - ▸ $n + \sum_{i=E+1}^{n-n_\infty} \ell_i = D_f + D_g + 1 + 2E + \sum_{i=1}^{E} \ell_i$.

Output: ▸ The rational function $f/g \in \mathsf{K}(x)$ such that
- ▸ $f, g \in \mathsf{K}[x]$, $g \neq 0$, $\mathrm{GCD}(f, g) = 1$;
- ▸ $\deg(f) \leq D_f$ and $\deg(g) \leq D_g$;
- ▸ f/g produces errors in $\leq E$ row vectors of $\widehat{A}$.
- ▸ Or a message indicating there is no such function.

1. *If* $\hat{a}_{i,j} = 0$ *for all* $i = 1, \ldots, n$ *and* $j = 0, \ldots, \ell_i$, *then return* $f/g = 0$.
2. *Let* $I_\infty = \{n - n_\infty + 1, \ldots, n\}$ *and* $P_\infty(x) = \prod_{i \in I_\infty}(x - \xi_i)$.
3. *For* $i = 1, \ldots, n - n_\infty$ *and* $j = 1, \ldots, \ell_i$, *compute*

$$\hat{b}_{i,j} \overset{\text{def}}{=} \sum_{\tau=0}^{j} \binom{j}{\tau} \hat{a}_{i,\tau} P_\infty^{(j-\tau)}(\xi_i).$$

4. *Compute the polynomial Hermite interpolant* $\bar{H}(x)$ *of the data set* $\{(\xi_i;\ \hat{b}_{i,0}, \ldots, \hat{b}_{i,\ell_i}) \mid i = 1, \ldots, n - n_\infty\}$ (*namely* $\bar{H}^{(j)}(\xi_i) = \hat{b}_{i,j}$, *see* Section 2). *Let* $H(x) = \bar{H}(x)P_\infty(x)$.
5. *Let* $r_0(x) = P_\infty(x) \prod_{i=1}^{n-n_\infty} (x - \xi_i)^{\ell_i+1}$, $r_1 = H$, $s_0 = 0$, $s_1 = 1$ *and* $\rho = 2$.
   5a. *Compute the* ρ*-th Euclidean polynomial remainder* r_ρ *and the multiplier* s_ρ *in the extended Euclidean scheme* $s_\rho r_1 + t_\rho r_0 = r_\rho$, *namely*

$$r_\rho(x) = r_{\rho-2}(x) - q_\rho(x) r_{\rho-1}(x),\ \deg(r_\rho) < \deg(r_{\rho-1}),$$
$$s_\rho(x) = s_{\rho-2}(x) - q_\rho(x) s_{\rho-1}(x).$$

5b. *If* $\deg(r_\rho) \le D_f + n_\infty + E + \sum_{i=1}^{E} \ell_i$, *then let* $\gamma = \rho$ *and go to step 6.*

5c. *Otherwise, let* $\rho = \rho + 1$ *and go to Step 5a.*

Step 5 computes (r_γ, s_γ) as in Lemma 1 with $d = D_f + E + \sum_{i=1}^{E} \ell_i + n_\infty$ and $e = D_g + E + \sum_{i=1}^{E} \ell_i - n_\infty$. We will prove in Lemma 4 that if there are $f, g \in \mathsf{K}[x]$ satisfy the output specifications, then $r_\gamma/(s_\gamma P_\infty^2) = f/g$. Here we also have $s_\gamma \neq 0$ because $\mathrm{GCD}(s_\gamma, t_\gamma) = 1$ when $\gamma \ge 2$ and $s_1 = 1$.

6. *Compute* $\Gamma = \mathrm{GCD}(r_\gamma, s_\gamma)$ *and* $f/g = r_\gamma/(s_\gamma P_\infty^2)$ *with* $\mathrm{GCD}(f, g) = 1$.

6a. *If* $\deg(f) \le D_f$ *and* $\deg(g) \le D_g$, *compute* $k_1 = \big|\{i \mid 1 \le i \le n - n_\infty, \Gamma(\xi_i) = 0\}\big|$ *and* $k_2 = \big|\{i \mid n - n_\infty + 1 \le i \le n,\ g(\xi_i) \neq 0\}\big|$; *if* $k_1 + k_2 \le E$ *then return* f/g.

6b. *If* $\deg(f) > D_f$, *or* $\deg(g) > D_g$, *or* $k_1 + k_2 > E$, *return a message indicating there is no such function.*

We will prove in Lemma 5 that the rational function f/g returned by Step 6a satisfies the output specifications. Therefore, we can check the validity of f/g without computing all the values $f^{(j)}(\xi_i)$ and $g^{(j)}(\xi_i)$ for $i = 1, \ldots, n$ and $j = 0, \ldots, \ell_i$.

We will define the error locator polynomial $\Lambda(x)$ in Lemma 4, and then based on Lemma 6, we show how to compute f/g and $\Lambda(x)$ more efficiently other than reducing the fraction $r_\gamma/(s_\gamma P_\infty^2)$ and evaluating Γ and g (see Remark 3).

Lemma 4. *We use the notation of Algorithm 5.1 and assume there exists a rational function* $f/g \in \mathsf{K}(x)$ *which satisfies the output specifications. Let* $\xi_{\lambda_1}, \ldots, \xi_{\lambda_k}$ *be the arguments with erroneous values, that is, for indices* $\lambda_\kappa \in \{\lambda_1, \ldots, \lambda_k\}$, *there exists* $j \in \{0, \ldots, \ell_{\lambda_\kappa}\}$ *such that* $\hat{a}_{i,j}$ *is an error. For* $\lambda_\kappa \notin I_\infty$, *let* $\delta_\kappa = \ell_{\lambda_\kappa} + 1 - \min\{j \mid \hat{a}_{\lambda_\kappa, j}$ *is an error* $\}$. *Let*

$$\bar{\Lambda}(x) = \prod_{\lambda_\kappa \in \{\lambda_1, \ldots, \lambda_k\} \setminus I_\infty} (x - \xi_{\lambda_\kappa})^{\delta_\kappa},$$

$$\Lambda_\infty(x) = \prod_{\lambda_\kappa \in \{\lambda_1, \ldots, \lambda_k\} \cap I_\infty} (x - \xi_{\lambda_\kappa}),$$

$$g_\infty(x) = \prod_{1 \le \nu \le n,\ \nu \in I_\infty \setminus \{\lambda_1, \ldots, \lambda_k\}} (x - \xi_\nu). \tag{15}$$

Let $\Lambda(x) = \bar{\Lambda}(x)\Lambda_\infty(x)$ *and* $\bar{g} = g/g_\infty$. *Then*

$$f P_\infty \Lambda \equiv H \bar{g} \bar{\Lambda} \pmod{r_0}. \tag{16}$$

Moreover, $f/g = r_\gamma/(s_\gamma P_\infty^2)$, *which implies the interpolant* f/g *is unique.*

Proof. Note that $P_\infty = \Lambda_\infty g_\infty$, we have $H\bar{g}\bar{\Lambda} = \bar{H} P_\infty \bar{g} \bar{\Lambda} = \bar{H} g \Lambda$, hence (16) is equivalent to

$$f P_\infty \Lambda \equiv \bar{H} g \Lambda \pmod{r_0}. \tag{17}$$

By the same argument as in the proof of (7) in Lemma 2, proving (17) is equivalent to proving the following two equalities:

$$(fP_\infty\Lambda)(\xi_i) = (\bar{H}g\Lambda)(\xi_i) \text{ for } i \in I_\infty, \tag{18}$$

$$(fP_\infty\Lambda)^{(j)}(\xi_i) = (\bar{H}g\Lambda)^{(j)}(\xi_i) \text{ for } i \notin I_\infty,\ j = 0, \ldots, \ell_i. \tag{19}$$

Note that $\bar{H}g\Lambda = (\bar{H}\bar{g}\bar{\Lambda})P_\infty$, therefore both sides of the equation in (18) are equal to zero because $P_\infty(\xi_i) = 0$ for $i \in I_\infty$.

It remains to prove (19). For $i \notin I_\infty$ and $j = 0, \ldots, \ell_i$,

$$(fP_\infty\Lambda)^{(j)}(\xi_i) = \sum_{\tau=0}^{j} \binom{j}{\tau} (fP_\infty)^{(j-\tau)}(\xi_i)\Lambda^{(\tau)}(\xi_i)$$

$$(\bar{H}g\Lambda)^{(j)}(\xi_i) = \sum_{\tau=0}^{j} \binom{j}{\tau} (\bar{H}g)^{(j-\tau)}(\xi_i)\Lambda^{(\tau)}(\xi_i),$$

we show that either $\Lambda^{(\tau)}(\xi_i) = 0$ or $(fP_\infty)^{(j-\tau)}(\xi_i) = (\bar{H}g)^{(j-\tau)}(\xi_i)$ by considering the following three cases.

Case 1. $\xi_i \notin \{\xi_{\lambda_1}, \ldots, \xi_{\lambda_k}\}$, then for $j = 0, \ldots, \ell_i$,

$$(fP_\infty)^{(j)}(\xi_i) = \sum_{\sigma=0}^{j} \binom{j}{\sigma} f^{(\sigma)}(\xi_i) P_\infty^{(j-\sigma)}(\xi_i) \tag{20}$$

$$= \sum_{\sigma=0}^{j} \binom{j}{\sigma} \sum_{\mu=0}^{\sigma} \binom{\sigma}{\mu} \hat{a}_{i,\sigma-\mu} g^{(\mu)}(\xi_i) P_\infty^{(j-\sigma)}(\xi_i) \tag{21}$$

$$= (\bar{H}g)^{(j)}(\xi_i). \tag{22}$$

The equality (21) follows from

$$f^{(\sigma)} = (\,(f/g)\,g\,)^{(\sigma)} = \sum_{\mu=0}^{\sigma} \tbinom{\sigma}{\mu} (f/g)^{(\sigma-\mu)} g^{(\mu)}.$$

Case 2. $\xi_i = \xi_{\lambda_\kappa}$ for some $\kappa \in \{1, \ldots, k\}$ and $\tau < \delta_\kappa$, then $\Lambda^{(\tau)}(\xi_{\lambda_\kappa}) = 0$.

Case 3. $\xi_i = \xi_{\lambda_\kappa}$ for some $\kappa \in \{1, \ldots, k\}$ and $\tau \geq \delta_\kappa$, then $j - \tau < \min\{j \mid (f/g)^{(j)}(\xi_{\lambda_\kappa}) \neq \hat{a}_{\lambda_\kappa, j}\}$, and one can prove that $(fP_\infty)^{(j-\tau)}(\xi_{\lambda_\kappa}) = (\bar{H}g)^{(j-\tau)}(\xi_{\lambda_\kappa})$ as in (22).

Now (19) is proved, which completes the proof of (16). Let $R = fP_\infty\Lambda$ and $S = \bar{g}\bar{\Lambda}$, we rewrite (16) as

$$R \equiv SH \pmod{r_0}.$$

Let $d = D_f + E + \sum_{i=1}^{E} \ell_i + n_\infty$ and $e = D_g + E + \sum_{i=1}^{E} \ell_i - n_\infty$ we have $\deg(r_0) = d + e + 1$ by the input specifications of the Algorithm 5.1 (or the condition (14)). Moreover, $\deg(H) \leq d + e$, $\deg(R) \leq d$ and $\deg(S) \leq e$, by Lemma 1, we have $R/S = r_\gamma/s_\gamma$. Thus $f/g = R/(SP_\infty^2) = r_\gamma/(s_\gamma P_\infty^2)$.

Lemma 5. *Let* $\Gamma = \mathrm{GCD}(r_\gamma, s_\gamma)$ *and* $f/g = r_\gamma/(s_\gamma P_\infty^2)$ *with* $\mathrm{GCD}(f,g) = 1$ *be as in Step 6 of Algorithm 5.1. If* $\deg(f) \le D_f$, $\deg(g) \le D_g$ *and* $k_1 + k_2 \le E$, *then* f/g *satisfies the output specifications of Algorithm 5.1.*

Proof. It is sufficient to prove that f/g produces errors in $\le k_1$ row vectors of the list $[\widehat{A}_{1,*}, \ldots, \widehat{A}_{n-n_\infty,*}]$. By the extended Euclidean scheme $s_\gamma r_1 + t_\gamma r_0 = r_\gamma$,

$$r_\gamma \equiv s_\gamma H \pmod{P}, \tag{23}$$

where $H = r_1$ and $P = r_0$. Since $fP_\infty^2/g = r_\gamma/s_\gamma$ and $\Gamma = \mathrm{GCD}(r_\gamma, s_\gamma)$, (23) leads to

$$fP_\infty^2\Gamma \equiv gH\,\Gamma \pmod{P}. \tag{24}$$

By dividing P_∞, we get

$$fP_\infty\Gamma \equiv g\bar{H}\,\Gamma \pmod{\prod_{i=1}^{n-n_\infty}(x-\xi_i)^{\ell_i+1}}. \tag{25}$$

Therefore, if $\Gamma(\xi_i) \neq 0$, then $(fP_\infty)^{(j)}(\xi_i) = (g\bar{H})^{(j)}(\xi_i)$ for all $j = 0, \ldots, \ell_i$, and this equality expands to (20), (21), and (22). Because $P_\infty(\xi_i) \neq 0$ for all $i = 1, \ldots, n-n_\infty$, it follows from (20) and (21) that $f^{(j)}(\xi_i) = \sum_{\mu=0}^{j} \binom{j}{\mu} \hat{a}_{i,j-\mu} g^{(\mu)}(\xi_i)$ if $i \in \{1, \ldots, n-n_\infty\}$ and $\Gamma(\xi_i) \neq 0$. This means for f/g, the list $[\widehat{A}_{1,*}, \ldots, \widehat{A}_{n-n_\infty,*}]$ has at least $n - n_\infty - k_1$ error-free row vectors.

From Lemma 4 and Lemma 5, we conclude the correctness of the Algorithm 5.1 in the following theorem.

Theorem 1. *Let* D_f, D_g, E *and* $\ell_1 \ge \cdots \ge \ell_n$ *be nonnegative integers, and let* K *be a field of characteristic* $\ge \ell_1 + 1$. *For a set of* n *distinct points* $\{\xi_1, \ldots, \xi_n\} \subset \mathsf{K}$ *and a list of* n *row vectors* $\widehat{A} = [\widehat{A}_{i,*}]_{1 \le i \le n}$ *with* $\widehat{A}_{i,*} = [\hat{a}_{i,0}, \ldots, \hat{a}_{i,\ell_i}] \in (\mathsf{K} \cup \{\infty\})^{1 \times (\ell_i+1)}$, *if* $\widehat{A}$ *satisfies the input specifications of the Algorithm 5.1, then either there is a unique rational function interpolant* f/g *satisfying the output specifications and the Algorithm 5.1 will return it, or there is no such rational function interpolant and the Algorithm 5.1 will report the nonexistence.*

Lemma 6. *With the notation as in Algorithm 5.1 and Lemma 4, we have*

$$\mathrm{GCD}(r_\gamma, s_\gamma) = \bar{\Lambda} \cdot \mathrm{GCD}(\bar{g}, g_\infty),$$

and Λ_∞^2 *divides* r_γ.

Proof. We first prove that $\bar{\Lambda}$ divides s_γ and r_γ. Since $r_\gamma \equiv s_\gamma H \pmod{r_0}$, we have

$$r_\gamma \bar{g} \equiv s_\gamma H \bar{g} = s_\gamma (\bar{H} g) \Lambda_\infty \pmod{r_0}. \tag{26}$$

On the other hand, let $R = fP_\infty\Lambda$ and $S = \bar{g}\bar{\Lambda}$, as it is shown in the proof of Lemma 4 that

$$r_\gamma S = s_\gamma R, \tag{27}$$

which is $r_\gamma(\bar{g}\bar{\Lambda}) = s_\gamma f P_\infty \Lambda$. Because $\bar{\Lambda} \neq 0$, dividing $\bar{\Lambda}$ on both sides results in

$$r_\gamma \bar{g} \equiv s_\gamma (f P_\infty) \Lambda_\infty \pmod{r_0}. \tag{28}$$

Combining (26) and (28) leads to

$$s_\gamma (\bar{H} g) \Lambda_\infty \equiv s_\gamma (f P_\infty) \Lambda_\infty \pmod{r_0}. \tag{29}$$

Since $\bar{\Lambda}$ is a factor of r_0 and $\mathrm{GCD}(\bar{\Lambda}, \Lambda_\infty) = 1$, $\bar{\Lambda}$ divides $s_\gamma(\bar{H} g - f P_\infty)$. Let $(x - \xi_{\lambda_\kappa})^{\delta_\kappa}$ be a factor of $\bar{\Lambda}$, and let $\epsilon_\kappa = \min\{j \mid (f/g)^{(j)}(\xi_{\lambda_\kappa}) \neq \hat{a}_{\lambda_\kappa, j}\}$, using the same argument as in the proof of Lemma 3, one can prove that $(x - \xi_{\lambda_\kappa})^{\delta_\kappa}$ divides s_γ, and so $\bar{\Lambda}$ divides s_γ. Because $r_\gamma \equiv s_\gamma H \pmod{r_0}$, $\bar{\Lambda}$ also divides r_γ.

Now assume that $\mathrm{GCD}(r_\gamma, s_\gamma) = \bar{\Lambda} w$ for some $w \in \mathsf{K}[x]$. Let $v = \mathrm{GCD}(\bar{g}, g_\infty)$, then $\mathrm{GCD}(R, S) = \bar{\Lambda} v$. From (27), we have the reduced fractions:

$$\frac{r_\gamma/(\bar{\Lambda} w)}{s_\gamma/(\bar{\Lambda} w)} = \frac{R/(\bar{\Lambda} v)}{S/(\bar{\Lambda} v)}, \tag{30}$$

which implies that

$$r_\gamma / w = c(R/v),\ s_\gamma / w = c(S/v) \text{ for some } c \in \mathsf{K} \setminus \{0\}. \tag{31}$$

Combining the Euclidean scheme $s_\gamma r_1 + t_\gamma r_0 = r_\gamma$, we have

$$\frac{R - SH}{v} = \frac{r_\gamma - s_\gamma H}{c\, w} = \frac{t_\gamma r_0}{c\, w}, \tag{32}$$

and so $(R - SH)/r_0 = (t_\gamma v)/(c\, w)$. By Lemma 4, $(R - SH)/r_0$ is a polynomial, which implies that w divides $(t_\gamma v)$. But $\mathrm{GCD}(t_\gamma, w) = 1$ because s_γ and t_γ are relatively prime, therefore w divides v.

We now prove that $v = w$: suppose $v = w w^*$ with $\deg(w^*) \geq 1$. Since v divides g_∞, there exists a ξ_i, with $1 \leq i \leq n$ and $i \notin \{\lambda_1, \ldots, \lambda_k\}$, such that $g_\infty(\xi_i) = w^*(\xi_i) = 0$. We have the following contradiction:

$$\begin{aligned} 0 &= r_\gamma(\xi_i) - H(\xi_i) s_\gamma(\xi_i) && (33)\\ &= c f(\xi_i) \Lambda(\xi_i) \Lambda_\infty(\xi_i) (g_\infty / w^*)(\xi_i) && (34)\\ &\neq 0. && (35) \end{aligned}$$

The equation (33) follows from $r_\gamma = s_\gamma r_1 + t_\gamma r_0$; (34) is a consequence of (31) and $P_\infty(\xi_i) = 0$; since g_∞ in (15) has single roots, $(g_\infty/w^*)(\xi_i) \neq 0$, which leads to (35).

Finally, $f/g = r_\gamma/(s_\gamma P_\infty^2) = r_\gamma/(s_\gamma \Lambda_\infty^2 g_\infty^2)$ and $\mathrm{GCD}(g, \Lambda_\infty) = 1$, so Λ_∞^2 must be a factor of the numerator r_γ.

Remark 3. Instead of computing f/g by reducing the fraction $r_\gamma/(s_\gamma P_\infty^2)$ as in Step 6 of the Algorithm 5.1, we can compute $\bar{\Lambda}$ and Λ_∞ first, and then compute f/g. In other words, for computing Λ and f/g, we can replace the Step 6 of the Algorithm 5.1 with the following steps:

6a. *Compute* $\Gamma = \mathrm{GCD}(r_\gamma, s_\gamma)$, $\tilde{r} = r_\gamma/\Gamma$, *and* $\tilde{s} = s_\gamma/\Gamma$.
6b. *Compute* $w = \mathrm{GCD}(\Gamma, P_\infty)$, $u = P_\infty/w$ *and* $\bar{\Lambda} = \Gamma/w$.
6c. *Compute* $\tilde{u} = \tilde{r}/u$, $\Lambda_\infty = \mathrm{GCD}(\tilde{u}, P_\infty)$, $\tilde{f} = \tilde{u}/\Lambda_\infty$, $g_\infty = P_\infty/\Lambda_\infty$, *and* $\tilde{g} = \tilde{s}\, w\, g_\infty$.
6d. *Let* k_1 *and* k_2 *be the number of distinct factors of* $\bar{\Lambda}$ *and* Λ_∞ *respectively.*
6d(i). *If* $\deg(\tilde{f}) \le D_f$, $\deg(\tilde{g}) \le D_g$, *and* $k_1 + k_2 \le E$, *return* $\tilde{f}/\tilde{g}$, $\bar{\Lambda}$, *and* Λ_∞.
6d(ii). *Else, return a message indicating there are no* $f, g \in \mathsf{K}[x]$ *such that* $\deg(f) \le D_f$, $\deg(g) \le D_g$ *and* f/g *produces errors in* $\le E$ *row vectors of* $\widehat{A}$.

Proof. By Lemma 6, $\Gamma = \bar{\Lambda} \cdot \mathrm{GCD}(\bar{g}, g_\infty)$. Since $g_\infty(\xi_{\lambda_\kappa}) \neq 0$ (see (15)) for all $\lambda_\kappa \in \{\lambda_1, \ldots, \lambda_k\}$, we have $\mathrm{GCD}(\Lambda, g_\infty) = 1$, thus $\mathrm{GCD}(\Gamma, P_\infty) = \mathrm{GCD}(\bar{g}, g_\infty)$ and $\Gamma = \bar{\Lambda}\, w$.
From (31), we have

$$\tilde{r} = r_\gamma/\Gamma = c(R/\Gamma) \text{ and } \tilde{s} = s_\gamma/\Gamma = c(S/\Gamma) \text{ for some } c \in \mathsf{K} \setminus \{0\}.$$

Using the substitutions $R = f\Lambda P_\infty$, $S = \bar{g}\bar{\Lambda}$, $\Gamma = \bar{\Lambda}\, w$, and $u = P_\infty/w$, one can verify that $\tilde{u} = cf\Lambda_\infty$ and $\tilde{g} = cg$. Because f and g are relatively prime, we have $\mathrm{GCD}(f, g_\infty) = 1$, so $\mathrm{GCD}(\tilde{u}, P_\infty) = \Lambda_\infty$.

6 Further Remarks

Remark 4. As stated in the introduction, the sufficient conditions (1, 5, 14) for an interpolation profile of orders of derivatives $\ell_1 \ge \cdots \ge \ell_n$ at distinct arguments may oversample, because the ℓ_i's are on both sides and could be reduced simultaneously while preserving the conditions. Therefore, one can add the following "pre-processing data" Step 0 at the beginning of the Algorithm 4.1, which may reduce the number of values for recovering f and improve the efficiency of the algorithm.

0. *For every* $j = 0, 1, \ldots, \ell_1$, *let* $m_j = \max\{\, i \mid \hat{a}_{i,j}$ *is given as input*$\}$ *(the dimension of the* j*-th column of* $\widehat{A}$*, the number of inputs for the* j*-th derivative). Furthermore, let* $M_j = \sum_{\mu=0}^{j} m_\mu$, *which is the number of inputs up to the* j*-th derivative. Compute the minimal* β *such that* $M_\beta \ge D + 1 + 2(\beta+1)E$. *Let*

$$N^{[\mathrm{new}]} = D + 1 + 2(\beta+1)E \tag{36}$$

and

$$\ell_i^{[\mathrm{new}]} = \begin{cases} \beta & \text{for } 1 \le i \le N^{[\mathrm{new}]} - M_{\beta-1}, \\ \beta - 1 & \text{for } N^{[\mathrm{new}]} - M_{\beta-1} + 1 \le i \le m_\beta, \\ \ell_i & \text{for } i > m_\beta. \end{cases} \tag{37}$$

Now $m_\beta^{[\mathrm{new}]} = N^{[\mathrm{new}]} - M_{\beta-1}$ *and* $\sum_{i=1}^{n} (\ell_i^{[\mathrm{new}]} + 1) = N^{[\mathrm{new}]}$.

One can also add Step 0 at the beginning of Algorithm 5.1 by replacing D with $D_f + D_g$. Figure 1 shows how Step 0 removes redundant values. Recall we are given n distinct points $\xi_1, \ldots, \xi_n$, and for each point ξ_i, we are given a row vector of values: $\widehat{A}_{i,*} = [\hat{a}_{i,0}, \ldots, \hat{a}_{i,\ell_i}]$ with $\ell_1 \geq \cdots \geq \ell_n \geq 0$, and $\widehat{A}$ is the list of these row vectors

$$\widehat{A} = \begin{bmatrix} \widehat{A}_{1,*} \\ \vdots \\ \widehat{A}_{n,*} \end{bmatrix}.$$

$\widehat{A}$ is shown as the "staircase" in Figure 1, which has $D = 15$, $E = 2$, $n = 8$, $\ell_1 = 11$, $\ell_2 = 10$, $\ell_3 = \ell_4 = 8$, $\ell_5 = \ell_6 = 7$, $\ell_7 = 3$, $\ell_8 = 0$, $N = 62$, $\beta = 5$, $N^{[\text{new}]} = 40$, $\ell_1^{[\text{new}]} = \cdots = \ell_5^{[\text{new}]} = 5$, $\ell_6^{[\text{new}]} = 4$. Intuitively, Step 0 cuts $\widehat{A}$ by the red line and removes the right part, and the left part has $N^{[\text{new}]}$ values.

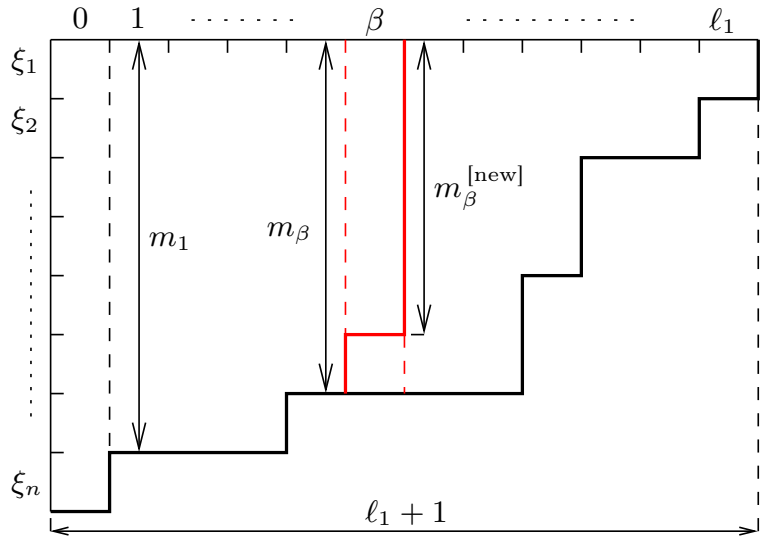

Fig. 1. Truncation by β

Lemma 7. *The β computed in the* Step 0 *above is no more than D.*

Proof. By the minimality of β, we have $M_{\beta-1} \leq D + 2\beta E$, and so

$$m_\beta = M_\beta - M_{\beta-1} \geq N^{[\text{new}]} - M_{\beta-1} \geq 2E + 1.$$

Moreover, $m_0 \geq \cdots \geq m_{\beta-1} \geq m_\beta \geq 2E + 1$, thus

$$(2E+1)\beta \leq \sum_{j=0}^{\beta-1} m_j = M_{\beta-1} \leq D + 2\beta E,$$

which concludes that $\beta \leq D$.

Remark 5. If we are given an error rate $1/q$ ($q \in \mathbb{Z}_{\geq 3}$) instead of an upper bound E on the number of errors, and we are also given bounds

(i) $D_f \geq \deg(f)$, $D_g \geq \deg(g)$
(ii) $\beta = \max\{j \mid (f/g)^{(j)}$ is available for evaluation$\}$,

then the Algorithm 5.1 can recover f/g for $q-2(\beta+1) = \eta > 0$ with $n = \left\lceil \frac{q\delta}{(\beta+1)\eta} \right\rceil$ distinct arguments and $N = \delta + 2(\beta+1)\left\lfloor \frac{\delta}{\eta} \right\rfloor$ values where $\delta = D_f + D_g(\beta+1)+1$. Cf. [9, Remark 1.1] and [10, Remark 1.1].

Remark 6. In the input specifications of the Algorithm 5.1, the number of values in $\widehat{A}$ is $C = D_f + D_g + 1 + 2\sum_{i=1}^{E}(\ell_i + 1)$ (see also (1)), which guarantees that the Algorithm 5.1 either returns a unique valid interpolant f/g or determines no such interpolant exists. If $E \geq 1$ and $\widehat{A}$ has $C - (\ell_n + 1)$ values, we can use Algorithm 5.1 on every $n-1$ row vectors of $\widehat{A}$ and with input $D_f, D_g, E-1$, to compute all possible rational functions f/g which satisfy:

- $f, g \in \mathsf{K}[x]$, $g \neq 0$, $\mathrm{GCD}(f, g) = 1$;
- $\deg(f) \leq D_f$ and $\deg(g) \leq D_g$;
- f/g produces errors in $\leq E$ row vectors of $\widehat{A}$.

This is because for every such rational function f/g, there is $\mu \in \{1, \ldots, n\}$ for which f/g produces errors in $\leq E-1$ row vectors of the list $\widehat{A} - \widehat{A}_{\mu,*} \stackrel{\text{def}}{=} [\widehat{A}_{1,*}, \ldots, \widehat{A}_{\mu-1,*}, \widehat{A}_{\mu+1,*}, \ldots, \widehat{A}_{n,*}]$ (if $\mu = 1$ or n, consider $\widehat{A}_{0,*}$ and $\widehat{A}_{n+1,*}$ as empty row vectors). Moreover, the list $\widehat{A} - \widehat{A}_{\mu,*}$ has $C - (\ell_n + 1) - (\ell_\mu + 1)$ values which are sufficient to recover f/g with the input bounds D_f, D_g and $E-1$, because $C - (\ell_n + 1) - (\ell_\mu + 1)$ is equal to:

$$\begin{cases} D_f + D_g + 1 + 2\displaystyle\sum_{i=1, i\neq\mu}^{E}(\ell_i + 1) + (\ell_\mu - \ell_n), & \text{if } 1 \leq \mu \leq E, \\ D_f + D_g + 1 + 2\displaystyle\sum_{i=1}^{E-1}(\ell_i + 1) + (2\ell_E - \ell_\mu - \ell_n), & \text{if } E+1 \leq \mu \leq n. \end{cases}$$

This method can be generalized to situations where $\widehat{A}$ has $C - \sum_{i=n}^{n-n_0}(\ell_i + 1)$ values and n_0 is a small constant compared to E. For the polynomial case with a uniform derivative profile, that is, $D_g = 0$ and $\ell_1 = \cdots = \ell_n$, [4] and [12] give algorithms to list-decode derivative (or multiplicity) codes by solving differential equations.

7 Conclusion

Interpolation algorithms go back to ancient Chinese mathematicians. Algorithms that also can tolerate errors in the evaluations appeared as error correction algebraic codes in the early 1960s. Table 1 gives a brief history. Our paper completes

Table 1. A brief history of univariate interpolation.

	Polynomial	*Rational Function*
at values	Sun-Tsu/Lagrange, Guo Shoujing/Newton	Cauchy
at values of derivatives	Hermite, Birkhoff	Padé/Kronecker, Warner 1974
at values with errors	Reed and Solomon [16]	Beelen, Høholdt, Nielsen, Wu [1]
at values of derivatives with errors	Multiplicity codes: Rosenbloom and Tsfasman [17]	This paper

the second column by giving an error correction interpolation algorithm of nearly linear arithmetic complexity for a rational function from values at its derivatives.

Note added August 10, 2020: page 3, before first unnumbered displayed formula "$E \geq \ldots :$ " "at one" $\longleftrightarrow$ "one"
page 3, after first unnumbered displayed formula "$E \geq \ldots :$ " added "where $|\ldots|$ is the number of elements in the set"

Note added August 14, 2020: page 10, Remark 1: $E_{\text{tot}} \longleftrightarrow E$ and added first sentence. Also added the last paragraph on how to decode.

Note added September 15, 2020: Added Padé/Kronecker to Table 1.

References

1. Beelen, P., Høholdt, T., Nielsen, J.S.R., Wu, Y.: On rational interpolation-based list-decoding and list-decoding binary Goppa codes. IEEE Trans. Inf. Theory **IT-59**(6), 3269–3281 (2013), URL: http://arxiv.org/abs/1211.0122
2. Chin, F.Y.: A generalized asymptotic upper bound for fast polynomial evaluation and interpolation. SIAM J. Comput. **5**(4), 682–690 (1976)
3. Coxon, N.: Fast systematic encoding of multiplicity codes. J. Symbolic Comput. **94**, 234–254 (2019)
4. Guruswami, V., Wang, C.: Optimal rate list decoding via derivative codes. In: Goldberg, L.A., Jansen, K., Ravi, R., Rolim, J.D.P. (eds.) Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, pp. 593–604. Springer (2011)
5. Guruswami, V., Wang, C.: Linear-algebraic list decoding for variants of Reed-Solomon codes. IEEE Transactions on Information Theory **59**(6), 3257–3268 (2013), URL: https://sites.math.rutgers.edu/~sk1233/part2.pdf
6. Gustavson, F.G., Yun, D.Y.Y.: Fast computation of the rational Hermite interpolant and solving Toeplitz systems of equations via the extended Euclidean algorithm. In: Ng, E.W. (ed.) Proceedings of the International Symposium on Symbolic and Algebraic Computation. pp. 58–64. EUROSAM '79, Springer, Berlin, Heidelberg (1979)
7. Kaltofen, E., Pernet, C., Storjohann, A., Waddell, C.A.: Early termination in parametric linear system solving and rational function vector recovery with error correction. In: Burr, M. (ed.) ISSAC '17 Proc. 2017 ACM Internat. Symp. Symbolic

Algebraic Comput. pp. 237–244. Association for Computing Machinery, New York, N. Y. (2017), URL: http://users.cs.duke.edu/~elk27/bibliography/17/KPSW17.pdf
8. Kaltofen, E., Trager, B.M.: Computing with polynomials given by black boxes for their evaluations: Greatest common divisors, factorization, separation of numerators and denominators. J. Symbolic Comput. **9**(3), 301–320 (1990), URL: http://users.cs.duke.edu/~elk27/bibliography/90/KaTr90.pdf
9. Kaltofen, E., Yang, Z.: Sparse multivariate function recovery from values with noise and outlier errors. In: Kauers, M. (ed.) ISSAC 2013 Proc. 38th Internat. Symp. Symbolic Algebraic Comput. pp. 219–226. Association for Computing Machinery, New York, N. Y. (2013), URL: http://users.cs.duke.edu/~elk27/bibliography/13/KaYa13.pdf
10. Kaltofen, E., Yang, Z.: Sparse multivariate function recovery with a high error rate in evaluations. In: Nabeshima, K. (ed.) ISSAC 2014 Proc. 39th Internat. Symp. Symbolic Algebraic Comput. pp. 280–287. Association for Computing Machinery, New York, N. Y. (2014), URL: http://users.cs.duke.edu/~elk27/bibliography/14/KaYa14.pdf
11. Kopparty, S.: Some remarks on multiplicity codes. In: Barg, A., Musin, O.R. (eds.) Discrete Geometry and Algebraic Combinatorics: AMS Spec. Session. Contemporary Mathematics, vol. 625, pp. 155–176 (2014), uRL: https://sites.math.rutgers.edu/~sk1233/multcode-survey.pdf
12. Kopparty, S.: List-decoding multiplicity codes. Theory of Computing **11**(1), 149–182 (2015), URL: https://sites.math.rutgers.edu/~sk1233/part2.pdf
13. Kopparty, S., Saraf, S., Yekhanin, S.: High-rate codes with sublinear-time decoding. Journal of the ACM (JACM) **61**(5), 1–20 (2014)
14. Moenck, R.T.: Fast computation of GCDs. In: Proc. 5th ACM Symp. Theory Comp. pp. 142–151 (1973)
15. Nielsen, R.R.: List decoding of linear block codes. Ph.D. thesis, Technical University of Denmark (2001)
16. Reed, I.S., Solomon, G.: Polynomial codes over certain finite fields. Journal of the Society for Industrial and Applied Mathematics **8**(2), 300–304 (1960)
17. Rosenbloom, M.Y., Tsfasman, M.A.: Codes for the m-metric. Problemy Peredachi Informatsii **33**(1), 55–63 (1997)
18. Schoenberg, I.J.: On Hermite-Birkhoff interpolation. J. Math. Analysis and Applic. **16**, 538–543 (1967), URL: https://doi.org/10.1016/0022-247X(66)90160-0
19. Sugiyama, Y., Kasahara, M., Hirasawa, S., Namekawa, T.: A method for solving key equation for decoding Goppa codes. Information and Control **27**(1), 87–99 (1975)
20. Warner, D.D.: Hermite interpolation with rational functions. Ph.D. thesis, University of California, San Diego (1974)
21. Welch, L.R., Berlekamp, E.R.: Error correction of algebraic block codes. US Patent 4,633,470 (1986), filed 1983; see http://patft.uspto.gov/

A Appendix

Notation (in alphabetic order):
$\hat{a}_{i,j}$ the input value for the j-th derivative of f, or an error, at the i-th point
$\widehat{A}_{i,*} = [\hat{a}_{i,0}, \ldots, \hat{a}_{i,\ell_i}]$, the row vector of values for the i-th point ξ_i

Notation continued (in alphabetic order):

$\widehat{A}$	$= [\widehat{A}_{1,*}, \ldots, \widehat{A}_{n,*}]^T$, the collection of all input values
$\hat{b}_{i,j}$	$= \sum_{\tau=0}^{j} \binom{j}{\tau} \hat{a}_{i,\tau} P_\infty^{(j-\tau)}(\xi_i)$the value for the j-th derivative of H at the i-th point
β	the minimal integer such that there are $\geq D + 1 + 2E + 2\beta E$ values for derivatives of order $\leq \beta$
c_j	the coefficient of x^j in f
D	an upper bound of the degree of the polynomial interpolant
D_f	an upper bound of the degree of the numerator of the rational interpolant
D_g	an upper bound of the degree of the denominator interpolant
δ_κ	$= \ell_{\lambda_\kappa} + 1 - \min\{\, j \mid \hat{a}_{\lambda_\kappa,j}$ is an error$\}$
E	an upper bound on the number of errors in the input values to the algorithm
ξ_i	the i-th interpolation point
ξ_{λ_κ}	$1 \leq \kappa \leq k$, are the points with erroneous values, namely, $\exists j$ s.t. $\hat{a}_{\lambda_\kappa,j}$ is an error
ϵ_κ	$= \min\{\, j \mid \hat{a}_{\lambda_\kappa,j}$ is an error$\} = \ell_{\lambda_\kappa} + 1 - \delta_\kappa$
f	polynomial interpolant or numerator of the rational interpolant for the correct values
g	the denominator of the rational interpolant for the correct values
$\bar{g}$	a factor of g indicating true non-poles
g_∞	a factor of g indicating true poles
H	the polynomial Hermite interpolant for all input values (including $\leq E$ errors)
I_∞	$= \{i \mid \exists j$ s.t. $\hat{a}_{i,j} = \infty\}$
k	the actual number of points with erroneous input values
K	a field
ℓ_i	the highest derivative order at the i-th point
Λ	the error locator polynomial
$\bar{\Lambda}$	$= \prod_{\kappa \in \{1,\ldots,k\}, \lambda_\kappa \notin I_\infty} (x - \xi_{\lambda_\kappa})^{\delta_\kappa}$
Λ_∞	$= \prod_{\kappa \in \{1,\ldots,k\}, \lambda_\kappa \in I_\infty} (x - \xi_{\lambda_\kappa})$
m_j	the number of input values for the j-th derivative of f
M_j	the number of input values for up to the j-th derivative of f
n	the number of distinct points
n_∞	degree of P_∞
N	the number of the input values
P_∞	$= \prod_{\exists j \text{ s.t. } \hat{a}_{i,j} = \infty} (x - \xi_i)$, the polynomial indicating all poles
r_0	$= (x - \xi_1)^{\ell_1 + 1} \cdots (x - \xi_n)^{\ell_n + 1}$
r_γ	the γ-th remainder of the Euclidean polynomial remainder sequence $r_0, r_1, \ldots$
s_γ	the Bézout coefficient of r_1 in the γ-th extended Euclidean scheme: $s_\gamma r_1 + t_\gamma r_0 = r_\gamma$
t_γ	the Bézout coefficient of r_0 in the γ-th extended Euclidean scheme: $s_\gamma r_1 + t_\gamma r_0 = r_\gamma$