

Developments in Barycentric Rational Interpolation

Lin Zejin

1 Introduction: polynomial interpolation in barycentric form

Interpolation is one of the central tools of numerical analysis: analysis problems are being solved by first choosing from the infinite complexity of an arbitrary function a few of its values and replacing it with an interpolant of these values.

This article reviews some of the recent advances in the practical use of interpolation in the form of the so-called *barycentric formula*.

In summary, Schneider and Werner [1] have been the first to determine barycentric representations of rational interpolants. Berrut and his students and colleagues develop it in [2, 3]. [4] gives a more efficient and stable algorithm based on the barycentric form, which is called the *AAA algorithm*.

2 Rational barycentric representations

The **barycentric formula** takes the form of a quotient of two partial fractions,

$$r(z) = \frac{n(z)}{d(z)} = \sum_{j=1}^m \frac{w_j f_j}{z - z_j} / \sum_{j=1}^m \frac{w_j}{z - z_j} \quad (2.1)$$

where $m \geq 1$ is an integer, z_1, \dots, z_m are a set of real or complex distinct support points, f_1, \dots, f_m are a set of real or complex data values, and w_1, \dots, w_m are a set of real or complex weights. As indicated in the equation, we let $n(z)$ and $d(z)$ stand for the partial fractions in the numerator and the denominator. When we wish to be explicit about step numbers, we write $r_m(z) = n_m(z)/d_m(z)$.

The node polynomial ℓ associated with the set z_1, \dots, z_m is the monic polynomial of degree m with these numbers as roots,

$$\ell(z) = \prod_{j=1}^m (z - z_j) \quad (2.2)$$

If we define

$$p(z) = \ell(z)n(z), \quad q(z) = \ell(z)d(z) \quad (2.3)$$

then p and q are each polynomials of degree at most $m - 1$. Thus, we have a simple link between the barycentric representation of r and its more familiar representation as a quotient of polynomials,

$$r(z) = \frac{p(z)/\ell(z)}{q(z)/\ell(z)} = \frac{p(z)}{q(z)} \quad (2.4)$$

This equation tells us that r is a rational function of type $(m-1, m-1)$, where, following standard terminology, we say that a rational function is of type (μ, ν) for integers $\mu, \nu \geq 0$ if it can be written as a quotient of a polynomial of degree at most μ and a polynomial of degree at most ν , not necessarily in lowest terms.

A key aspect of (2.1) is its interpolatory property. At each point z_j with $w_j \neq 0$, the formula is undefined, taking the form ∞/∞ (assuming $f_j \neq 0$). However, this is a removable singularity, for $\lim_{z \rightarrow z_j} r(z)$ exists and is equal to f_j . Thus, if the weights w_j are nonzero, (2.1) provides a type $(m-1, m-1)$ rational interpolant to the data f_1, \dots, f_m at z_1, \dots, z_m . Note that such a function has $2m-1$ degrees of freedom, so roughly half of these are fixed by the interpolation conditions and the other half are not.

We summarize the properties of barycentric representations developed in the discussion above by the following theorem.

Theorem 2.1 (rational barycentric representations). *Let z_1, \dots, z_m be an arbitrary set of distinct complex numbers. As f_1, \dots, f_m range over all complex values and w_1, \dots, w_m range over all nonzero complex values, the functions*

$$r(z) = \frac{n(z)}{d(z)} = \sum_{j=1}^m \frac{w_j f_j}{z - z_j} / \sum_{j=1}^m \frac{w_j}{z - z_j} \quad (2.5)$$

range over the set of all rational functions of type $(m-1, m-1)$ that have no poles at the points z_j . Moreover, $r(z_j) = f_j$ for each j .

Proof. By (2.4), any quotient n/d as in (2.5) is a rational function r of type $(m-1, m-1)$. Moreover, since $w_j \neq 0$, d has a simple pole at z_j and n has either a simple pole there (if $f_j \neq 0$) or no pole. Therefore, r has no pole at z_j .

Conversely, suppose r is a rational function of type $(m-1, m-1)$ with no poles at the points z_j , and write $r = p/q$, where p and q are polynomials of degree at most $m-1$ with no common zeros. Then q/ℓ is a rational function with a zero at ∞ and simple poles at the points z_j . Therefore, q/ℓ can be written in the partial fraction form of a denominator d as in (2.5) with $w_j \neq 0$ for each j . Similarly, p/ℓ is a rational function with a zero at ∞ and simple poles at the points z_j or a subset of them. Therefore, since $w_j \neq 0$, p/ℓ can be written in the partial fraction form of a numerator n as in (2.5). \square

3 Classical rational interpolation

The problem here is to find $r \in \mathcal{R}_{m,n}$ that interpolates the f_j , i.e., $p \in \mathcal{P}_m$ and $q \in \mathcal{P}_n$ such that

$$r(x_j) = \frac{p(x_j)}{q(x_j)} = f_j, \quad j = 0(1)N. \quad (3.1)$$

In the canonical representation, p and q together have $m+n+2$ coefficients, of which one may be set to 1 by dividing both polynomials by it. The $N+1$ interpolation conditions (3.1) thus are equally numerous as the coefficients when

$$N = m + n. \quad (3.2)$$

Classical rational interpolation in barycentric form

Schneider and Werner [1] have been the first to determine barycentric representations of rational interpolants. Their method uses a classical way of determining the Newton form of the interpolant before applying an algorithm of Werner to pass from the Newton to the barycentric form. In [2], the first and last authors have given a method for directly finding the corresponding weights u_j when $n \leq m$.

Theorem 3.1. *If a solution r of the classical rational interpolation problem (3.1)(3.2) with $n \leq m$ exists, then $\mathbf{u} = [u_0, u_1, \dots, u_N]$ is a vector of weights in one of its barycentric representations (2.1) iff \mathbf{u} belongs to the kernel of the $N \times (N+1)$ matrix*

$$\mathbf{A} := \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ x_0 & x_1 & x_2 & \cdots & x_N \\ x_0^2 & x_1^2 & x_2^2 & \cdots & x_N^2 \\ \vdots & \vdots & \vdots & & \vdots \\ x_0^{m-1} & x_1^{m-1} & x_2^{m-1} & \cdots & x_N^{m-1} \\ f_0 & f_1 & f_2 & \cdots & f_N \\ f_0 x_0 & f_1 x_1 & f_2 x_2 & \cdots & f_N x_N \\ f_0 x_0^2 & f_1 x_1^2 & f_2 x_2^2 & \cdots & f_N x_N^2 \\ \vdots & \vdots & \vdots & & \vdots \\ f_0 x_0^{n-1} & f_1 x_1^{n-1} & f_2 x_2^{n-1} & \cdots & f_N x_N^{n-1} \end{bmatrix}. \quad (3.3)$$

In order to save space, we introduce some notation: $\mathbf{V}_{P,Q}, P \leq Q$, will be the matrix made up of the first P rows of the transposed Vandermonde matrix corresponding to the $Q+1$ values x_0, \dots, x_Q , and $\mathbf{F}_Q = \text{diag}(f_0, \dots, f_Q) \in \mathbb{R}^{Q+1, Q+1}$ will be the diagonal matrix of values f_0, \dots, f_Q . Then \mathbf{A} in (3.3) may be written $\mathbf{A} = [\mathbf{V}_{m,N}^T, \mathbf{F}_N^T \mathbf{V}_{n,N}^T]^T$.

The proof of Theorem 3.1 consists in showing that the first m equations express that the degree of the denominator written in the form $\ell(x) \sum_{j=0}^N \frac{u_j}{x-x_j}$ in (2.1) is at most n , the last n equations that the numerator degree is at most m . Lemma 2.1 a) implies that r in (2.1) interpolates in x_j if $u_j \neq 0$; the latter is not necessary, though.

As a corollary, Theorem 3.1 delivers the kernel of the matrix $\mathbf{V}_{N-1,N}$: it is just the space spanned by the vector of the polynomial barycentric weights w_j .

The algorithm given in [2] for computing the kernel of \mathbf{A} in (3.3) is much more efficient than computing the singular value decomposition of \mathbf{A} . It consists in triangulating (3.3) in two steps: one analytical, which leads to divided differences, the other numerical, through Gaussian elimination with column pivoting. Though, in contrast with p , r is a good approximation even with equidistant interpolation points for N large enough, Chebyshev nodes again lead to much better conditioned problems. Note, however, that these nodes must be reordered for a stable computation of the divided differences. The algorithm is then extremely stable, see the examples in [2].

As with the interpolating polynomial, the degrees of p and q may be smaller than m and n . This manifests itself in the kernel of \mathbf{A} having dimension larger.

A way of coping with this, suggested in [2], is to decrease n by 1, restart the computation, and repeat until a kernel of dimension 1 is obtained for some $n^* < n$. \mathbf{u} then yields the barycentric representation (2.1) of a reduced r , i.e., one in which the linear factors corresponding to common zeros of p and q have been simplified. We will denote it by r^* . If $u_k = 0$ for some k and $r^*(x_k) \neq f_k$, then x_k is an unattainable point and the problem does not have a solution. This is the first part

of the following theorem, due to Schneider and Werner [2], which allows an easy detection of two drawbacks of classical rational interpolation directly from the weights.

Theorem 3.2. *Let u be barycentric weights of a reduced rational interpolant $r^* = p^*/q^*$. Then*

- a) *a point x_k is unattainable iff $u_k = 0$;*
- b) *if $u_k \neq 0$ for all k , if the interpolation points have been ordered as $x_0 < x_1 < \dots < x_N$ and if $\text{sign } u_j = \text{sign } u_{j+1}$, then r^* has an odd number of poles between x_j and x_{j+1} .*

Proof. a) follows from the above discussion and Theorem 2.1, b) from noticing that q^* changes sign, thus has a zero, between x_j and x_{j+1} , and p^* does not, for otherwise r^* would not be reduced. \square

Stability and Conditioning

The barycentric representation is generally more numerically stable than solving for polynomial coefficients. Higham and others have proved that for polynomial interpolation the barycentric evaluation is backward stable [3]. For rational interpolation, similar arguments show that evaluation in barycentric form is stable when x is not too close to the poles of r .

The condition number of evaluating $r(x)$ depends on the distance to the nearest pole. If x is near a true pole of the function being approximated, rounding errors can be magnified. Nevertheless, in many practical settings the barycentric approach exhibits good stability and accuracy. In particular, because it never requires dividing by zero at the interpolation nodes, it avoids catastrophic cancellation in forming the interpolant itself.

One must also consider spurious pole-zero cancellations. In finite precision or noisy data, an interpolant may introduce small residue poles (Froissart doublets) that do not reflect true features of the underlying function. Modern implementations include cleanup steps to detect and remove these doublets, ensuring that the final rational approximant is as simple as possible.

4 AAA Algorithm

The Adaptive Antoulas–Anderson (AAA) algorithm is a recent method for rational approximation using the barycentric form [4]. It is an iterative procedure that constructs a rational approximant to given data or a function on a discrete set.

We begin with a finite sample set $Z \subseteq \mathbf{C}$ of $M \gg 1$ points. We assume a function $f(z)$ is given that is defined at least for all $z \in Z$. This function may have an analytic expression, or it may be just a set of data values.

The AAA algorithm takes the form of an iteration for $m = 1, 2, 3, \dots$, with r represented at each step in the barycentric form (2.5). At step m we first pick the next support point z_m by the greedy algorithm to be described below, and then we compute corresponding weights w_1, \dots, w_m by solving a linear least-squares problem over the subset of sample points that have not been selected as support points,

$$Z^{(m)} = Z \setminus \{z_1, \dots, z_m\} \tag{3.1}$$

Thus, at step m , we compute a rational function r of type $(m-1, m-1)$, which generically will interpolate $f_1 = f(z_1), \dots, f_m = f(z_m)$ at z_1, \dots, z_m (although not always, since one or more weights may turn out to be zero).

The least-squares aspect of the algorithm is as follows. Our aim is an approximation

$$f(z) \approx \frac{n(z)}{d(z)}, \quad z \in Z \quad (3.2)$$

which in linearized form becomes

$$f(z)d(z) \approx n(z), \quad z \in Z^{(m)} \quad (3.3)$$

Note that in going from (3.2) to (3.3) we have replaced Z by $Z^{(m)}$, because $n(z)$ and $d(z)$ will generically have poles at z_1, \dots, z_m , so (3.3) would not make sense over all $z \in Z$. The weights w_1, \dots, w_m are chosen to solve the least-squares problem

$$\text{minimize } \|fd - n\|_{Z^{(m)}}, \quad \|w\|_m = 1 \quad (3.4)$$

where $\|\cdot\|_{Z^{(m)}}$ is the discrete 2-norm over $Z^{(m)}$ and $\|\cdot\|_m$ is the discrete 2-norm on m -vectors. To ensure that this problem makes sense, we assume that $Z^{(m)}$ has at least m points, i.e., $m \leq M/2$.

The greedy aspect of the iteration is as follows. At step m , the next support point z_m is chosen as a point $z \in Z^{(m-1)}$ where the nonlinear residual $f(z) - n(z)/d(z)$ at step $m-1$ takes its maximum absolute value.

Assuming the iteration is successful, it terminates when the nonlinear residual is sufficiently small; we have found it effective to use a default tolerance of 10^{-13} relative to the maximum of $|f(Z)|$. The resulting approximation typically has few or no numerical Froissart doublets, and if there are any, they can usually be removed by one further least-squares step to be described in section 5. (If the convergence tolerance is too tight, the approximation will stagnate and many Froissart doublets will appear.) In the core AAA algorithm, it is an approximation of type $(m-1, m-1)$.

It remains to spell out the linear algebra involved in (3.4). Let us regard $Z^{(m)}$ and $F^{(m)} = f(Z^{(m)})$ as column vectors,

$$Z^{(m)} = \left(Z_1^{(m)}, \dots, Z_{M-m}^{(m)} \right)^T, \quad F^{(m)} = \left(F_1^{(m)}, \dots, F_{M-m}^{(m)} \right)^T$$

We seek a normalized column vector

$$w = (w_1, \dots, w_m)^T, \quad \|w\|_m = 1$$

that minimizes the 2-norm of the $(M-m)$ -vector

$$\sum_{j=1}^m \frac{w_j F_i^{(m)}}{Z_i^{(m)} - z_j} - \sum_{j=1}^m \frac{w_j f_j}{Z_i^{(m)} - z_j}$$

that is,

$$\sum_{j=1}^m \frac{w_j \left(F_i^{(m)} - f_j \right)}{Z_i^{(m)} - z_j}$$

This is a matrix problem of the form

$$\text{minimize } \|A^{(m)}w\|_{M-m}, \quad \|w\|_m = 1 \quad (3.5)$$

where $A^{(m)}$ is the $(M-m) \times m$ Loewner matrix

$$A^{(m)} = \begin{pmatrix} \frac{F_1^{(m)} - f_1}{Z_1^{(m)} - z_1} & \cdots & \frac{F_1^{(m)} - f_m}{Z_1^{(m)} - z_m} \\ \vdots & \ddots & \vdots \\ \frac{F_{M-m}^{(m)} - f_1}{Z_{M-m}^{(m)} - z_1} & \cdots & \frac{F_{M-m}^{(m)} - f_m}{Z_{M-m}^{(m)} - z_m} \end{pmatrix} \quad (3.6)$$

We will solve (3.5) using the singular value decomposition (SVD), taking w as the final right singular vector in a reduced SVD $A^{(m)} = U\Sigma V^*$. (The minimal singular value of $A^{(m)}$ might be nonunique or nearly so, but our algorithm does not rely on its uniqueness.) Along the way it is convenient to make use of the $(M - m) \times m$ Cauchy matrix

$$C = \begin{pmatrix} \frac{1}{Z_1^{(m)} - z_1} & \cdots & \frac{1}{Z_1^{(m)} - z_m} \\ \vdots & \ddots & \vdots \\ \frac{1}{Z_{M-m}^{(m)} - z_1} & \cdots & \frac{1}{Z_{M-m}^{(m)} - z_m} \end{pmatrix} \quad (3.7)$$

whose columns define the basis in which we approximate. If we define diagonal left and right scaling matrices by

$$S_F = \text{diag}(F_1^{(m)}, \dots, F_{M-m}^{(m)}), \quad S_f = \text{diag}(f_1, \dots, f_m) \quad (3.8)$$

then we can construct $A^{(m)}$ from C using the identity

$$A^{(m)} = S_F C - C S_f \quad (3.9)$$

and once w is found with the SVD, we can compute $(M - m)$ -vectors N and D with

$$N = C(wf), \quad D = Cw \quad (3.10)$$

These correspond to the values of $n(z)$ and $d(z)$ at points $z \in Z^{(m)}$. (Since M will generally be large, it is important that the sparsity of S_F is exploited in the multiplication of (3.9).) Finally, to get an M -vector R corresponding to $r(z)$ for all $z \in Z$, we set $R = f(Z)$ and then $R(Z^{(m)}) = N/D$.

Algorithm 1 AAA Algorithm

Input: Function or data F , sample points $Z = \{z_1, \dots, z_M\}$, tolerance ε , maximum iterations

m_{\max}

Output: Rational approximant $r(z)$ in barycentric form, poles, zeros, residues, support points

1: Initialize:

- $R \leftarrow$ mean of F (initial approximation)
- Empty support point list: $z \leftarrow []$, $f \leftarrow []$, $C \leftarrow []$
- Error vector: $\text{errvec} \leftarrow []$

2: **for** $m = 1$ to m_{\max} **do**

3: Select next support point z_m where residual $|F - R|$ is largest

4: Append z_m and $f_m = F(z_m)$ to support sets

5: Remove z_m from candidate pool

6: Update Cauchy matrix:

7: $C \leftarrow [C \ 1/(Z - z_m)]$

8: Construct Loewner matrix:

9: $A = \text{diag}(F) \cdot C - C \cdot \text{diag}(f)$

10: Compute SVD of A and select right singular vector w with smallest singular value

11: Construct barycentric rational approximation:

12: Numerator $N = C \cdot (w \cdot f)$

13: Denominator $D = C \cdot w$

14: $R_j = N_j/D_j$ for all j not in support points

15: Compute max error: $\text{err} = \|F - R\|_{\infty}$

16: Append err to errvec

17: **if** $\text{err} \leq \varepsilon \cdot \|F\|_{\infty}$ **then**

18: **break**

19: **end if**

20: **end for**

21: Define rational approximant $r(z)$ using barycentric formula:

$$r(z) = \frac{\sum_j \frac{w_j f_j}{z - z_j}}{\sum_j \frac{w_j}{z - z_j}}$$

22: Compute poles, residues, and zeros via generalized eigenvalue problems

23: Optionally: remove Froissart doublets (close pole-zero pairs)

After the AAA algorithm terminates (assuming $w_j \neq 0$ for all j), one has a rational approximation $r(z) = n(z)/d(z)$ in barycentric form. The zeros of d , which are (generically) the poles of r , can be computed by solving an $(m+1) \times (m+1)$ generalized eigenvalue problem in arrowhead form,

$$\begin{pmatrix} 0 & w_1 & w_2 & \cdots & w_m \\ 1 & z_1 & & & \\ & & z_2 & & \\ 1 & & & \ddots & \\ \vdots & & & & \\ 1 & & & & z_m \end{pmatrix} = \lambda \begin{pmatrix} 0 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{pmatrix} \quad (3.11)$$

At least two of the eigenvalues of this problem are infinite, and the remaining $m - 1$ are the zeros of d . A similar computation with w_j replaced by $w_j f_j$ gives the zeros of $n(z)$.

The following proposition collects some elementary properties of the core AAA algorithm. We say "a" instead of "the" in view of the fact that in cases of ties in the greedy choice at each step, AAA approximants are not unique.

Proposition 4.1. *Let $r(z)$ be a AAA approximant at step m of a function $f(z)$ on a set Z (computed in exact arithmetic). The following statements refer to AAA approximants at step m , and a and b are complex constants.*

Affineness in f . For any $a \neq 0$ and b , $ar(z) + b$ is an approximant of $af(z) + b$ on Z .

Affineness in z . For any $a \neq 0$ and b , $r(az + b)$ is an approximant of $f(az + b)$ on $(Z - b)/a$.

Monotonicity. The linearized residual norm $\sigma_{\min}(A^{(m)}) = \|fd - n\|_{Z(m)}$ is a nonincreasing function of m .

Proof. These properties are straightforward and we do not spell out the arguments except to note that the monotonicity property follows from the fact that $A^{(m)}$ is obtained from $A^{(m-1)}$ by deleting one row and appending one column. Since the minimum singular vector for $A^{(m-1)}$ (padded with one more zero) is also a candidate singular vector of $A^{(m)}$, we must have $\sigma_{\min}(A^{(m)}) \leq \sigma_{\min}(A^{(m-1)})$. \square

One might ask, must the monotonicity be strict, with $\sigma_{\min}(A^{(m)}) < \sigma_{\min}(A^{(m-1)})$ if $\sigma_{\min}(A^{(m-1)}) \neq 0$? So far as we are aware, the answer is no. An equality $\sigma_{\min}(A^{(m)}) = \sigma_{\min}(A^{(m-1)})$ implies $f(z_m)d_{m-1}(z_m) = n_{m-1}(z_m)$, where z_m is the support point selected at step m . This in turn implies $d_{m-1}(z_m) = n_{m-1}(z_m) = 0$, since otherwise we could divide by $d_{m-1}(z_m)$ to find $f(z_m) - n_{m-1}(z_m)/d_{m-1}(z_m) = 0$, which would contradict the greedy choice of z_m . But so far as we know, the possibility $d_{m-1}(z_m) = n_{m-1}(z_m) = 0$ is not excluded.

Since the AAA algorithm involves SVDs of dimensions $(M - j) \times j$ with $j = 1, 2, \dots, m$, its complexity is $O(Mm^3)$ flops. This is usually modest since in most applications m is small.

Advantages of the AAA algorithm

The AAA algorithm uses a barycentric representation that provides better numerical conditioning and flexibility. This allows AAA to handle complex domains, including disconnected or irregular regions, with ease. Additionally, the AAA algorithm's greedy approach to selecting support points minimizes the risk of Froissart doublets (spurious pole-zero pairs). Another one of its key strengths is its adaptive selection of support points, which avoids exponential instabilities and ensures robust convergence without requiring user-specified initial parameters like pole locations or numbers. , a common issue in other methods.

5 Conclusion

The applications of the barycentric representation of rational interpolants brings interesting advances in infinitely smooth practical approximation. Its use in classical rational interpolation yields a very stable way of computing the interpolant and allows for a relatively simple detection of unattainable points and poles. The latter may also be easily monitored in the complex plane and their location optimized to yield new rational interpolants which approximate a given function with an error that diminishes with the number of the poles. AAA algorithm develops this idea further, allowing for a more stable way of computing rational approximants to a function defined on a support set, even not necessarily analytic.

References

- [1] C. Schneider and W. Werner. “Some New Aspects of Rational Interpolation”. In: *Mathematics of Computation* 47 (1986), pp. 285–299.
- [2] J.-P. Berrut and H. D. Mittelmann. “Matrices for the Direct Determination of the Barycentric Weights of Rational Interpolation”. In: *Journal of Computational and Applied Mathematics* 78 (1997), pp. 355–370.
- [3] J.-P. Berrut, R. Baltensperger, and H. D. Mittelmann. “Recent Developments in Barycentric Rational Interpolation”. In: *Trends and Applications in Constructive Approximation*. Ed. by M. G. de Bruin, D. H. Mache, and J. Szabados. Basel: Birkhäuser, 2005, pp. 27–51.
- [4] Yuji Nakatsukasa, Olivier Sète, and Lloyd N. Trefethen. “The AAA Algorithm for Rational Approximation”. In: *SIAM Journal on Scientific Computing* 40.3 (2018), A1494–A1522. DOI: 10.1137/16M1106122. eprint: <https://doi.org/10.1137/16M1106122>. URL: <https://doi.org/10.1137/16M1106122>.