

# Overview of Barycentric Rational Interpolation

## 1 Introduction

Rational interpolation is a classical extension of polynomial interpolation in which one seeks a rational function  $r(x) = P(x)/Q(x)$  that matches given data  $(x_j, f_j)$ ,  $j = 0, \dots, n$ . A powerful representation of such interpolants is given by the *barycentric formula*, which expresses  $r(x)$  as a ratio of two partial fractions. Specifically, for distinct nodes  $\{z_j\}$  and weights  $\{w_j\}$  one can write

$$r(x) = \frac{\sum_{j=0}^m \frac{w_j f_j}{x - z_j}}{\sum_{j=0}^m \frac{w_j}{x - z_j}},$$

yielding a rational function of type  $(m, m)$  that satisfies  $r(z_j) = f_j$  for each  $j$  when the weights are chosen appropriately. As  $x \rightarrow z_k$ , the dominant terms in numerator and denominator both behave like  $w_k f_k / (x - z_k)$  and  $w_k / (x - z_k)$ , respectively, yielding the limit  $r(z_k) = f_k$ . Thus the formula interpolates the data by construction.

This barycentric representation dates back to Taylor (1945) for polynomial interpolation and was recognized by Werner (1984) as a generally rational form [?]. In the mid-1980s, Schneider and Werner [?] gave the first explicit barycentric representation for rational interpolants. In the following decades, Berrut and colleagues developed more efficient and stable algorithms based on barycentric formulas [?, ?]. Our overview will cover these advances in barycentric rational interpolation and the underlying mathematical theory, and it culminates with a discussion of the AAA algorithm [?] as a recent high-performance method.

## 2 Historical Background

The barycentric interpolation formula was first discovered in the context of polynomials by W. Taylor in 1945. Decades later, W. Werner observed that the barycentric evaluation formula usually yields a rational interpolant, laying the groundwork for rational barycentric interpolation [?]. Schneider and Werner (1986) were the first to systematically derive barycentric formulas for rational interpolation [?]. Building on this, Berrut and Mittelmann (1997) and later Berrut et al. (2005) improved these algorithms and analyses [?, ?].

Together, these works showed that barycentric representations provide a concise and stable way to interpolate data with rational functions. For example, they allow one to avoid solving ill-conditioned Vandermonde systems. The modern view of rational interpolation therefore heavily relies on barycentric ideas, with the AAA algorithm as a recent example of this line of development [?].

### 3 Barycentric Rational Interpolation

#### 3.1 Formula and Interpolation Property

Consider data  $(z_1, f_1), \dots, (z_m, f_m)$  with distinct nodes  $z_j \in \mathbb{C}$ . A barycentric rational interpolant of type  $(m-1, m-1)$  can be written as

$$r(x) = \frac{\sum_{j=1}^m \frac{w_j f_j}{x - z_j}}{\sum_{j=1}^m \frac{w_j}{x - z_j}}, \quad (1)$$

for some nonzero weights  $w_1, \dots, w_m$ . One checks that  $r(z_k) = f_k$  by observing that, as  $x \rightarrow z_k$ , the terms in the numerator and denominator with index  $j = k$  dominate the sums, and their ratio tends to  $f_k$ . In effect, the barycentric formula guarantees the interpolation conditions without explicitly solving a system of equations.

The freedom in choosing the weights  $w_j$  means that many representations exist for the same interpolant. In the special case of polynomial interpolation, one can take  $w_j = 1/\prod_{i \neq j} (z_j - z_i)$ , and (??) reduces to the usual Lagrange form. In rational interpolation, the weights are typically determined by solving a linear system or optimization to satisfy extra conditions (such as minimal norm) [?].

Key properties of the barycentric representation include:

- **Interpolatory exactness:** By construction,  $r(z_j) = f_j$  for all nodes  $z_j$ . No Vandermonde matrix needs to be inverted, avoiding its numerical pitfalls.
- **Stability of evaluation:** The form (??) evaluates stably for  $x$  away from the poles of  $r$ . In fact, for polynomial interpolation it is known to be backward stable [?].
- **Update flexibility:** New support points can be added without recomputing the entire formula. This allows adaptive algorithms (like AAA) to build up an interpolant iteratively.
- **Pole adaptation:** The rational form can adapt to singular behavior in the data by placing poles where needed. The poles of  $r$  are determined by the zeros of the denominator in (??). One must watch out for spurious pole-zero pairs (Froissart doublets) which can occur if weights are not chosen carefully.

#### 3.2 Derivation of the Barycentric Form

A derivation of (??) proceeds by starting from the requirement  $r(z_k) = f_k$ . Equivalently, one can write

$$r(x) = \frac{P(x)}{Q(x)}, \quad P(x) = \sum_{j=0}^m \alpha_j \prod_{i \neq j} (x - z_i), \quad Q(x) = \sum_{j=0}^m \beta_j \prod_{i \neq j} (x - z_i),$$

so that  $r(z_k) = \alpha_k/\beta_k = f_k$  for each  $k$ . This implies  $\alpha_k = f_k \beta_k$  for each  $k$ . Setting  $w_j = \beta_j$  and assuming  $\beta_j \neq 0$ , one finds  $\alpha_j = w_j f_j$ , and hence

$$r(x) = \frac{\sum_{j=0}^m w_j f_j \prod_{i \neq j} (x - z_i)}{\sum_{j=0}^m w_j \prod_{i \neq j} (x - z_i)}.$$

Expanding these products one checks that this is algebraically equivalent to (??). Thus the barycentric form is a natural parametrization of the general interpolant of type  $(m-1, m-1)$ .

#### 3.3 Classical Rational Interpolation Methods

Before barycentric methods became popular, rational interpolation was often implemented via continued fractions or by solving linear systems for polynomial coefficients. A direct approach is to impose

$$P(z_k) - f_k Q(z_k) = 0, \quad k = 1, \dots, m,$$

with  $\deg(P) \leq m - 1$ ,  $\deg(Q) \leq m - 1$ . This yields a homogeneous linear system for the coefficients of  $P$  and  $Q$ . In practice, one fixes a scale (say by setting the leading coefficient of  $Q$  to 1) and solves the resulting linear system. However, this can be ill-conditioned if the nodes  $z_k$  are poorly distributed.

Matrix formulations of barycentric interpolation provide more stable alternatives. Berrut and Mittelmann showed that one can compute the weights  $\{w_j\}$  by finding a null vector of a Loewner-like matrix derived from the data [?]. Concretely, let  $C$  be the Cauchy matrix with entries  $C_{ij} = 1/(z_i - z_j)$  (omitting  $i = j$ ), and form

$$A = \text{diag}(f)C - C \text{diag}(f).$$

Then one finds that  $A[w_1, \dots, w_m]^T = 0$  captures the interpolation conditions. Equivalently, computing the smallest singular vector of  $A$  yields the weights that best fit the data in a least-squares sense. This is essentially the approach taken inside the AAA algorithm (below).

A subtle issue is the existence of solutions. In some special cases (known as *unattainable points*), no rational interpolant of the specified degree exists [?]. Such cases are rare for generic data, and most analysis assumes that a solution of type  $(m - 1, m - 1)$  does exist.

## 4 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 [?]. 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 (??) 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.

## 5 The AAA Algorithm

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

1. *Barycentric representation.* The approximant is represented in the form (??) with an initially empty support set.
2. *Greedy support selection.* At each iteration, the algorithm evaluates the current approximation on a dense sample of points and picks the point  $z_{\text{new}}$  where the error  $|r(z) - f(z)|$  is largest. This point is added to the support set.
3. *Weight update via Loewner SVD.* Given the new support points  $z_1, \dots, z_k$ , the algorithm forms a Loewner matrix  $A$  by subtracting scaled Cauchy columns based on the function values. It then computes the smallest singular vector of  $A$  to obtain the new weights  $w_j$  that minimize the interpolation error.
4. *Convergence check.* If the maximum residual error falls below a tolerance  $\text{tol}$ , the iteration stops; otherwise, it continues until a preset maximum support size is reached.
5. *Cleanup step.* After convergence, the algorithm removes any numerical Froissart doublets by detecting negligible residues and simplifying the rational function.

The AAA algorithm automatically determines a suitable rational degree and support points without user input. Nakatsukasa *et al.* showed that AAA often achieves near-minimax approximation on complicated domains, outperforming many classical methods [?].

The core AAA procedure can be implemented in roughly 40 lines of MATLAB. The listing below, taken from [?], provides the complete AAA code. It takes sample points  $Z$  and data  $F$  and returns the rational approximant  $r$  (as a function handle) along with its poles, residues, zeros, and weights:

```
function [r,pol,res,zer,z,f,w,errvec] = aaa(F,Z,tol,mmax)
% AAA rational approximation of data F on set Z
% [r,pol,res,zer,z,f,w,errvec] = aaa(F,Z,tol,mmax)
%
% Input: F = vector of data values (or function handle)
%        Z = vector of sample points
%        tol = relative tolerance (default 1e-13)
%        mmax = max allowed support size (default 100)
%
% Output: r = AAA approximant (function handle)
%         pol,res,zer = poles, residues, zeros
%         z,f,w = support points, values, weights
%         errvec = error at each iteration
M = length(Z);
if nargin<3, tol = 1e-13; end
if nargin<4, mmax = 100; end
if ~isfloat(F), F = F(Z); end
Z = Z(:); F = F(:);
SF = spdiags(F,0,M,M);
J = 1:M; z = []; f = []; errvec = []; R = mean(F);
for m = 1:mmax
    [~,j] = max(abs(F-R)); % select next support point
    z = [z; Z(j)]; f = [f; F(j)]; % include new support
    J(J==j) = []; % remove index from consideration
    C = [C, 1./(Z - Z(j))]; % update Cauchy matrix
    Sf = diag(f);
    A = SF*C - C*Sf; % Loewner matrix
    [~,~,V] = svd(A(J,:),0);
    w = V(:,m); % update weights
    N = C*(w.*f); D = C*w;
    R = F; R(J) = N(J)./D(J); % rational approximation values
    err = norm(F-R,inf);
    errvec = [errvec; err];
    if err <= tol*norm(F,inf)
        break;
    end
end
end
r = @(zz) feval(@rhandle,zz,z,f,w);
[pol,res,zer] = prz(r,z,f,w);
[r,pol,res,zer,z,f,w] = cleanup(r,pol,res,zer,z,f,w,Z,F);

function [pol,res,zer] = prz(r,z,f,w)
m = length(w);
B = eye(m+1); B(1,1) = 0;
E = [0 w.'; ones(m,1) diag(z)];
pol = eig(E,B); pol = pol(~isinf(pol));
dz = 1e-6*exp(2i*pi*(1:4)/4);
```

```

res = r(pol+dz)*dz.'/4;
E2 = [0 (w.*f).'; ones(m,1) diag(z)];
zer = eig(E2,B); zer = zer(~isinf(zer));

function r = rhandle(zz,z,f,w)
zv = zz(:);
CC = 1./(bsxfun(@minus, zv, z.'));
r = (CC*(w.*f))./(CC*w);
ii = isnan(r);
for k = 1:length(ii)
    if ii(k)
        r(k) = f(zv(k)==z);
    end
end
r = reshape(r,size(zz));

```

## 6 Conclusion

Barycentric rational interpolation unifies many ideas in numerical approximation. Historically rooted in Taylor’s barycentric formula, it was extended and refined by Schneider, Werner, and Berrut [?, ?, ?]. The barycentric representation provides an efficient and stable way to interpolate data with rational functions. Algorithms like AAA epitomize this approach by adaptively selecting support points and computing rational approximants with minimal user input [?]. The bibliography below provides references to the foundational literature.

## References

- [1] C. Schneider and W. Werner, “Some new aspects of rational interpolation,” *Math. Comput.*, vol. 47, pp. 285–299, 1986.
- [2] J.-P. Berrut and H. D. Mittelmann, “Matrices for the direct determination of barycentric weights of rational interpolation,” *J. Comput. Appl. Math.*, vol. 78, pp. 355–370, 1997.
- [3] J.-P. Berrut, R. Baltensperger, and H. D. Mittelmann, “Recent developments in barycentric rational interpolation,” in *Trends and Applications in Constructive Approximation*, G. de Bruin, D. Mache, and J. Szabados, Eds., Birkhäuser, 2005, pp. 27–51.
- [4] Y. Nakatsukasa, O. Sète, and L. N. Trefethen, “The AAA algorithm for rational approximation,” *SIAM J. Sci. Comput.*, vol. 40, no. 3, pp. A1494–A1522, 2018.