1 Introduction

Rational approximations of real or complex functions are used mainly in two kinds of applications. Sometimes they provide compact representations of functions that are much more efficient than polynomials for functions with poles or other singularities on or near the domain of approximation or on unbounded domains. Other times, their role is one of extrapolation: the extraction of information about poles or values or other properties of a function in regions of the real line or complex plane beyond where it is known a priori. For example, standard methods of acceleration of convergence of sequences and series, such as the eta and epsilon algorithms, are based on rational approximations [7, 19]. For a general discussion of the uses of rational approximation, see Chapter 23 of [61], and for theoretical foundations, see [18]. The *AAA* algorithm introduced in this paper can be used to find rational approximations either to a specified accuracy or of a specified rational type such as (m, m) or $(m-1, m)$, as is common, for example, in applications in model order reduction and system identification.

Working with rational approximations, however, can be problematic. There are various challenges here, one of which particularly grabs attention: spurious poles, also known as Froissart doublets, which can be regarded either as poles with very small residues or as pole-zero pairs so close together as to nearly cancel [20, 31, 33, 34, 35, 60]. Froissart doublets arise in the fundamental mathematical problem-i.e., in "exact arithmetic"-and are the reason why theorems on convergence of rational approximations, e.g., of Padé approximants along diagonals of the Padé table, typically cannot hold without the qualification of convergence in capacity rather than uniform convergence [7,53]. On a computer in floating-point arithmetic, they arise all the more often; we speak of numerical Froissart doublets, recognizable by residues on the order of machine precision. These difficulties are related to the fact that the problem of analytic continuation, for which rational approximation is the most powerful general technique, is ill-posed.

The AAA algorithm proposed in this paper offers a speed, flexibility, and robustness we have not seen in other algorithms; the name stands for "adaptive AntoulasAnderson." [1] (More recent material related to the Antoulas-Anderson method can be found in [45], and a discussion of related methods is given in section 11.) The algorithm combines two ideas. First, following Antoulas and Anderson [3] (although their presentation of the mathematics is very different), rational functions are represented in barycentric form with interpolation at certain support points selected from a set provided by the user. Second, the algorithm grows the approximation degree one by one, selecting support points in a systematic greedy fashion so as to avoid exponential instabilities. Numerical Froissart doublets usually do not appear, and if they do, they can usually be removed by one further solution of a least-squares problem.

Perhaps the most striking feature of the AAA algorithm is that it is not tied to a particular domain of approximation such as an interval, a circle, a disk, or a point. Many methods for rational approximation utilize bases that are domain dependent, whereas the AAA barycentric representation, combined with its adaptive selection of support points, avoids such a dependence. The algorithm works effectively with point sets that may include discretizations of disconnected regions of irregular shape, possibly unbounded, and the functions approximated may have poles lying in the midst of the sample points. Thus, the AAA algorithm is fast and flexible, but on the other hand, it does not claim to achieve optimality in any particular norm such as L^2 or L^∞. For such problems more specialized methods may be used, although as we shall mention in section 10 and have subsequently developed in [28], the AAA algorithm may still play a role in providing an initial guess or as the design pattern for a variant algorithm based, for example, on iterative reweighting.

2 Rational barycentric representations

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

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

where $m \geq 1$ is an integer, $z_1, \ldots, z_m$ are a set of real or complex distinct support points, $f_1, \ldots, f_m$ are a set of real or complex data values, and $w_1, \ldots, w_m$ 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 $z_1, \ldots, z_m$ is the monic polynomial of degree m with these numbers as roots,

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

If we define

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

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,

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

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. [2] The numerator $n(z)$ and denominator $d(z)$ are each of type $(m-1, m)$, so it is obvious from (2.1) that r is of type ($2m-1, 2m-1$); it is the cancellation of the factors $1/\ell(z)$ top and bottom that makes it actually of type ($m-1, m-1$). This is the paradox of the barycentric formula: it is of a smaller type than it looks, and its poles can be anywhere except where they appear to be (assuming the weights w_j in (2.1) are nonzero). The paradox goes further in that, given any set of support points $\{z_j\}$, there is a special choice of the weights $\{w_j\}$ for which r becomes a polynomial of degree $m - 1$. This is important in numerical computation, providing a numerically stable method for polynomial interpolation even in thousands of points; see [17, 25, 61] for discussion and references. However, it is not our subject here. Here we are concerned with the cases where (2.1) is truly a rational function, a situation exploited perhaps first by Salzer [54] and Schneider and Werner [56] and, most importantly, in subsequent years by Berrut and his collaborators [14, 15, 16, 17, 29, 47]. For further links to the literature, see section 11.

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 $\lim_{z \to z_j} r(z)$ 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 $f_1, \ldots, f_m$ at $z_1, \ldots, z_m$. 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*

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

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 (see Theorem 4.4h and p. 553 of [44]). 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 $w_j \neq 0, p/\ell$ can be written in the partial fraction form of a numerator n as in (2.5).

If the support points $\{z_j\}$ have no influence on the set of functions described by (2.5), one may wonder, what is the use of barycentric representations? The answer is all about the numerical quality of the representation and is at the very heart of why the AAA algorithm is so effective. The barycentric formula is composed from quotients $1/(z - z_j)$, and for good choices of $\{z_j\}$, these functions are independent enough to make the representation well conditioned-often far better conditioned, in particular, than one would find with a representation $p(z)/q(z)$. As shown in section 11, they are also better conditioned than the partial fraction representations used by vector fitting, since in that case, the points $\{z_j\}$ are constrained to be the poles of r. □

The use of localized and sometimes singular basis functions is an established theme in other areas of scientific computing. Radial basis functions, for example, have excellent conditioning properties when they are composed of pieces that are well separated [30]. In [23], one even finds a barycentric-style quotient of two RBF sums utilized with good effect. Similarly, the method of fundamental solutions, which has had great success in solving elliptic PDEs such as Helmholtz problems, represents its functions as linear combinations of Hankel or other functions each localized at a singular point [8]. An aim of the present paper is to bring this kind of thinking to the subject of function theory. Yet another related technique in scientific computing is discretizations of the Cauchy integral formula, for example, by the trapezoidal rule on the unit circle, to evaluate analytic functions inside a curve. The basis functions implicit in such a discretization are singular, introducing poles and, hence, errors of size ∞ at precisely the data points where one might expect the errors to be 0 , but still the approximation may be excellent away from the curve [6, 48].

3 Core 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.

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 $m = 1, 2, 3, \ldots$, 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 $w_1, \ldots, w_m$ by solving a linear least-squares problem over the subset of sample points that have not been selected as support points,

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

Thus, at step m, we compute a rational function r of type $(\, m-1, m-1 \,)$, which generically will interpolate $f_1 = f(z_1), \ldots, f_m = f(z_m)$ at $z_1, \ldots, z_m$ (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

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

which in linearized form becomes

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

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 $z_1, \ldots, z_m$, so (3.3) would not make sense over all $z \in Z$. The weights $w_1, \ldots, w_m$ are chosen to solve the least-squares problem

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

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., $m \le M/2$.

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 $F^{(m)} = f\left(Z^{(m)}\right)$ as column vectors,

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

We seek a normalized column vector

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

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

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

where $A^{(m)}$ is the $(M-m) \times m$ Loewner matrix [2] (or Löwner in its original spelling)

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

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

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

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

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

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

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

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

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

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 $R\left(Z^{(m)}\right) = N/D$.

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 [47, sect. 2.3.3],

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

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 3.1. *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, $\mathrm{ar}(z) + b$ 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 $\sigma_{\min}\left(A^{(m)}\right) = \|fd - n\|_{Z^{(m)}}$ 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 $\sigma_{\min}\left(A^{(m)}\right) \leq \sigma_{\min}\left(A^{(m-1)}\right)$. □

One might ask, must the monotonicity be strict, with $\sigma_{\min}\left(A^{(m)}\right) < \sigma_{\min}\left(A^{(m-1)}\right)$ if $\sigma_{\min}\left(A^{(m-1)}\right) \neq 0$? So far as we are aware, the answer is no. An equality $\sigma_{\min}\left(A^{(m)}\right) = \sigma_{\min}\left(A^{(m-1)}\right)$ 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 $j = 1, 2, \ldots, m$, its complexity is $O\left(Mm^3\right)$ flops. This is usually modest since in most applications m is small.

Algorithm 1 AAA Algorithm

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

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

Initialize:

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

for $m = 1$ to $m_{\max}$ **do**
  Select next support point z_m where residual $|F - R|$ is largest
  Append z_m and $f_m = F(z_m)$ to support sets
  Remove z_m from candidate pool
  Update Cauchy matrix:
  $C \leftarrow [C \;\; 1/(Z - z_m)]$
  Construct Loewner matrix:
  $A = \mathrm{diag}(F) \cdot C - C \cdot \mathrm{diag}(f)$
  Compute SVD of A and select right singular vector w with smallest singular value
  Construct barycentric rational approximation:
  Numerator $N = C \cdot (w \cdot f)$
  Denominator $D = C \cdot w$
  $R_j = N_j / D_j$ for all j not in support points
  Compute max error: err $= \|F - R\|_\infty$
  Append err to errvec
  if err $\leq \varepsilon \cdot \|F\|_\infty$ **then**
    break
  end if
end for

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

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

Compute poles, residues, and zeros via generalized eigenvalue problems

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