Introduction to Scientific Computing

Midterm Project

Paper:

The AAA algorithm for rational approximation

Class: 建模所碩一

Student: 康芷俞

Date: October 31, 2025

Table of Contents

1 Problem Description

1.1 Background

In numerical analysis, approximating complex functions from discrete data is a fundamental task. Rational approximation—using ratios of polynomials—offers significant advantages over classical polynomial methods (e.g., Chebyshev or least-squares) by efficiently capturing poles, sharp transitions, and exponential behavior with much lower degrees.

1.2 The Problem Addressed by the Paper

Given discrete samples of a real- or complex-valued function $f(z)$ on a finite set $Z = \{z_j\}_{j=1}^{M} \subseteq \mathbb{F}$, where $\mathbb{F}$ denotes either $\mathbb{R}$ or $\mathbb{C}$, the objective is to construct a rational function

$$r(z) = \frac{n(z)}{d(z)}, \quad n, d \text{ being polynomials,}$$

that approximates $f(z)$ on Z with minimal residual error.

1.3 Why the AAA Algorithm Is Important

The AAA algorithm offers several key advantages:

- **Stability and efficiency:** Employs greedy selection of support points with computational complexity $\mathcal{O}(Mm^3)$, where m is typically modest.
- **Versatility:** Applicable to both complex- and real-valued functions over arbitrary discrete domains.
- **Broad connections:** The barycentric framework is conceptually linked to techniques used in RBF interpolation, elliptic PDE solvers, and Cauchy integral methods.

2 Method Overview

This section presents the core principles of the AAA algorithm, which combines barycentric representation, greedy support-point selection, and linearized least-squares computation.

2.1 Barycentric Representation

The rational approximant $r(z)$ is expressed in barycentric form as

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

where $\{z_j\}_{j=1}^m$ are distinct support points, $\{f_j\}_{j=1}^m$ are the corresponding function values, and $\{w_j\}_{j=1}^m$ are the barycentric weights. These quantities may be real or complex, and $m \geq 1$.

Let

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

be the node polynomial, and define

$$p(z) = \ell(z)n(z), \qquad q(z) = \ell(z)d(z).$$

Then

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

so $r(z)$ is a rational function of type $(m-1, m-1)$:

$$\deg(p) \leq m-1, \qquad \deg(q) \leq m-1.$$

Removable Singularities

Since each $w_j \neq 0$, every support point z_j is a removable singularity of $r(z)$:

$$\lim_{z\to z_j} r(z) = \lim_{z\to z_j} \frac{\frac{w_j f_j}{z-z_j} + \sum_{i\neq j} \frac{w_i f_i}{z-z_i}}{\frac{w_j}{z-z_j} + \sum_{i\neq j} \frac{w_i}{z-z_i}} = \lim_{z\to z_j} \frac{w_j f_j + (z-z_j)\sum_{i\neq j} \frac{w_i f_i}{z-z_i}}{w_j + (z-z_j)\sum_{i\neq j} \frac{w_i}{z-z_i}} = \frac{w_j f_j}{w_j} = f_j.$$

Hence, the barycentric form interpolates the sampled data exactly at all support points :

$$r(z_j) = f_j, \quad j = 1, \ldots, m.$$

2.2 The AAA algorithm

The AAA algorithm constructs the rational approximant $r(z)$ iteratively in barycentric form. At step m, it selects the next support point z_m using a greedy strategy and computes barycentric weights $\{w_j\}_{j=1}^m$ by solving a linear least-squares problem over the remaining sample set $Z^{(m)} = Z \setminus \{z_j\}_{j=1}^m$.

The goal is

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

which can be rewritten as a linear problem:

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

where the support points are chosen from $Z^{(m)}$ to ensure $n(z)$ and $d(z)$ are well-defined.

Greedy Selection

The next support point z_m is selected as the point in $Z^{(m-1)}$ that maximizes the residual:

$$z_m = \underset{z\in Z^{(m-1)}}{\operatorname{argmax}} |f(z) - r_{m-1}(z)|.$$

Here, $r_{m-1}(z)$ denotes the rational approximant constructed in the previous iteration.

Termination

The iteration is terminated once the nonlinear residual becomes sufficiently small. In practice, a relative tolerance of 10^{-13} with respect to $\max_{z\in Z}|f(z)|$ is typically used:

$$\|f(Z) - r(Z)\|_\infty \le 10^{-13} \max_{z\in Z} |f(z)|.$$

Matrix Formulation

Let $Z^{(m)} = (Z_1^{(m)}, \dots, Z_{M-m}^{(m)})^T, \quad F^{(m)} = f(Z^{(m)}) = (F_1^{(m)}, \dots, F_{M-m}^{(m)})^T, \quad w = (w_1, \dots, w_m)^T.$

The residual vector is

$$r_i = f(z_i^{(m)})d(z_i^{(m)}) - n(z_i^{(m)}) = \sum_{j=1}^{m} \frac{w_j(F_i^{(m)} - f_j)}{z_i^{(m)} - z_j}, \quad i = 1, \dots, M-m,$$

which can be expressed in matrix form as

$$r := \begin{bmatrix} r_1 \\ \vdots \\ r_{M-m} \end{bmatrix} = A^{(m)}w, \quad A_{i,j}^{(m)} = \frac{F_i^{(m)} - f_j}{z_i^{(m)} - z_j}.$$

Here $A^{(m)}$ is the *Loewner matrix*.

The goal is to minimize the residual norm $\|r\|_2 = \|A^{(m)}w\|_2$. To avoid the trivial solution, we impose the normalization constraint $\|w\|_2 = 1$, leading to the constrained least-squares problem:

$$\text{minimize } \|A^{(m)}w\|_2 \quad \text{subject to} \quad \|w\|_2 = 1.$$

This problem can be efficiently solved via the singular value decomposition (SVD):

$A^{(m)} = U\Sigma V^*$, $\quad w$ is taken as the right singular vector corresponding to the smallest singular value.

Values of Numerator and Denominator

We define

$$S_F = \operatorname{diag}(F_1^{(m)}, \dots, F_{M-m}^{(m)}), \quad S_f = \operatorname{diag}(f_1, \dots, f_m), \quad C = \begin{pmatrix} \frac{1}{Z_1^{(m)} - z_1} & \cdots & \frac{1}{Z_1^{(m)} - z_m} \\ \vdots & \ddots & \vdots \\ \frac{1}{Z_{M-m}^{(m)} - z_1} & \cdots & \frac{1}{Z_{M-m}^{(m)} - z_m} \end{pmatrix}.$$

Then the numerator and denominator evaluated at $Z^{(m)}$ can be expressed as

$$N := C(w \odot f), \quad D := Cw,$$

where $\odot$ denotes element-wise multiplication.

It follows that

$$A^{(m)} = S_F C - C S_f.$$

Computation of Poles and Zeros

After convergence, the final rational approximant $r(z) = n(z)/d(z)$ is obtained. The zeros of $d(z)$, which generally correspond to the poles of $r(z)$, can be computed by solving the $(m+1) \times (m+1)$ generalized eigenvalue problem in arrowhead form:

$$A\mathbf{v} = \lambda B\mathbf{v}, \quad A = \begin{pmatrix} 0 & w_1 & w_2 & \cdots & w_m \\ 1 & z_1 & & & \\ 1 & & z_2 & & \\ \vdots & & & \ddots & \\ 1 & & & & z_m \end{pmatrix}, \quad B = \begin{pmatrix} 0 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{pmatrix}.$$

Assuming nonzero weights, the problem has at least two infinite eigenvalues, while the remaining $m-1$ finite eigenvalues correspond to the zeros of $d(z)$, i.e., the poles of $r(z)$. Replacing w_j with $w_j f_j$ in the first row similarly gives the zeros of the numerator $n(z)$, which are the zeros of $r(z)$.

2.3 Supplementary Explanations

[Q]: Why is the condition $M \geq 2m$ required?

At iteration m, m support points $z_1, \ldots, z_m$ have been selected, leaving $M - m$ unsampled points. The Loewner matrix $A^{(m)} \in \mathbb{C}^{(M-m)\times m}$ defines a least-squares problem:

$$A^{(m)} w \approx 0.$$

For the system to be overdetermined and well-posed, the number of rows must be at least as large as the number of columns:

$$M - m \geq m \quad \Rightarrow \quad M \geq 2m.$$

If $M < 2m$, the system becomes underdetermined, and the SVD solution may not be unique or reliable.

[Q]: Why use the right singular vector corresponding to the smallest singular value?

The minimization problem

$$\min_{\|w\|=1} \|A^{(m)} w\|$$

is solved via the SVD: $A^{(m)} = U\Sigma V^*$, where $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_m)$ with $\sigma_1 \geq \cdots \geq \sigma_m \geq 0$.

For any unit vector $w = \sum_{i=1}^{m} \alpha_i v_i$ with $\sum_i |\alpha_i|^2 = 1$,

$$\|A^{(m)} w\|^2 = \|\Sigma V^* w\|^2 = \sum_{i=1}^{m} \sigma_i^2 |\alpha_i|^2 \geq \sigma_m^2,$$

with equality if and only if $w = v_m$, the right singular vector associated with the smallest singular value σ_m. This choice minimizes the linearized residual $\|A^{(m)} w\|$, yielding the best approximation in the least-squares sense.

The AAA algorithm remains numerically stable even when σ_m is small or nearly degenerate, thanks to the barycentric representation which avoids the ill-conditioning inherent in polynomial quotient forms $p(z)/q(z)$.

3 AAA Approximation of $\tan(\frac{\pi z}{2})$ on a Spiral

The AAA algorithm approximates $\tan(\pi z/2)$ along a logarithmic spiral, converging in 12 iterations to a final error of 1.15×10^{-13}, with exponential decay observed starting from iteration 4. Pole recovery accuracy decreases with distance: $\sim 10^{-16}$ at ± 1, $\sim 10^{-7}$ at ± 3, and $\sim 10^{-3}$ at ± 5.

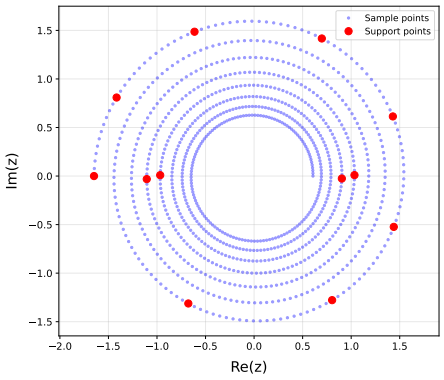

Fig 1: Sample Points and Support Points

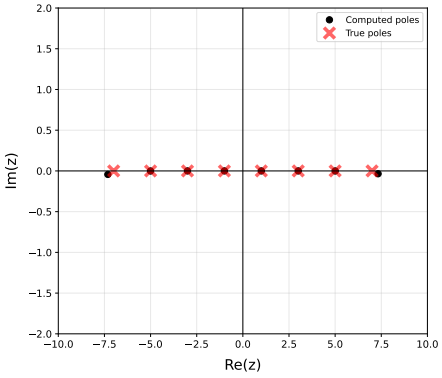

Fig 2: Poles of the Approximant

m	Error	m	Error
1	2.49e+01	7	5.87e-05
2	4.28e+01	8	1.29e-06
3	1.71e+01	9	3.57e-08
4	8.65e-02	10	6.37e-10
5	1.27e-02	11	1.67e-11
6	9.91e-04	12	1.15e-13

Table 1: Convergence history

True Pole	Computed Pole	Error
1.0	1.0000000000 -0.0000000000i	8.95e-16
-1.0	-1.0000000000 +0.0000000000i	2.54e-16
3.0	3.0000001004 -0.0000000409i	1.08e-07
-3.0	-3.0000000647 -0.0000000582i	8.70e-08
5.0	5.0026932457 -0.0005617593i	2.75e-03
-5.0	-5.0024386920 -0.0007471458i	2.55e-03

Table 2: Pole comparison

4 Conclusion

The AAA algorithm provides automatic rational approximation via barycentric representations with adaptive support point selection, requiring no user-specified parameters. It typically converges to the default tolerance of 10^{-13} in 10–50 iterations and performs robustly across diverse domains, including intervals, disks, and disconnected regions. Poles near the sampled region are often captured to 15-digit accuracy, while the barycentric representation ensures excellent numerical conditioning compared to polynomial quotient methods.

Limitations include potential spurious poles when approximating even or odd functions on real intervals, though the cleanup procedure typically removes numerical Froissart doublets. Functions with branch cuts may require many poles clustering along the cut, increasing computational cost to $\mathcal{O}(Mm^3)$, and pole accuracy degrades for singularities far from the sampled region.

Possible extensions noted in Section 10 include weighted norms, iterative reweighting for minimax approximation, and generalization to matrix-valued functions.