Final project

December 15, 2025

1 Introduction

The aim of this paper is to propose AAA rational approximation as a method for interpolating or approximating smooth functions from equispaced samples on an interval. Although it is generally advantageous to approximate from larger datasets when available, whether equispaced or not, we demonstrate that this method often performs impressively even when the sampling grid is coarse. In most cases, it yields more accurate approximations than other existing methods.

1.1 Numerical Interpolation

In its basic form, the proposed method simply computes a AAA rational approximation to the data. Consequently, the resulting interpolant is a "numerical" one, meaning it is not mathematically exact at the sample points, but matches them to high precision. This relaxation is a crucial advantage for robustness. Using the Chebfun software, such a fit can be computed to a default relative accuracy of 10^{-13}.

1.2 Problem

However, the use of the AAA algorithm also encounters certain issues. First, because of the impossibility theorem, the performance of AAA on an equispaced grid may not always converge as rapidly as one might expect for stable algorithms. Furthermore, a principal drawback is the potential appearance of unwanted poles often referred to as "bad poles" or Froissart doublets within the domain of approximation. These spurious singularities, which can arise from rounding errors or under-resolution during the greedy selection process, result in infinite function values inside the interval, thereby destroying the accuracy of the interpolant locally and requiring specific cleanup procedures such as the AAA-Least Squares method.

2 Method Overview

In this paper, for comparison, they present several methods that are commonly used for handling equispaced nodes.

2.1 Polynomial least squares

It is well known that interpolation of n equispaced data values by a polynomial of degree $n-1$ leads to exponential instability, a manifestation of the Runge phenomenon, with errors growing at a rate of $O(2^n)$. To mitigate this, polynomial least-squares approximation with a degree $d < n-1$ is often employed. To effectively curtail exponential growth, d must be restricted , but in practice, larger degrees are often acceptable. A common strategy is to use an oversampling ratio $\gamma > 1$, where $d \approx n/\gamma$..

2.2 Fourier extension

The Fourier extension method approximates a non-periodic function f on $[-1, 1]$ by a Fourier series defined on a larger domain $[-T, T]$ (where $T > 1$). This approach avoids the Gibbs phenomenon associated with periodic extension. The fit is typically computed via least-squares or regularized least-squares.

2.3 Fourier series with corrections

For non-periodic functions, standard trigonometric interpolation converges slowly because of discontinuities at the boundaries. A powerful fix is to enrich the trigonometric approximation with correction terms—typically polynomials—that compensate for boundary effects. Originating in work by James Gregory, this approach can achieve algebraic or even super-algebraic convergence, depending on the order of the corrections. We benchmark against a least-squares scheme that augments a Fourier series with a polynomial component of degree about $\sqrt{n}$, using an oversampling factor of approximately 2.

2.4 Splines

Splines approximate functions using piecewise polynomials with continuity constraints. They are robust and avoid the oscillations typical of global polynomial interpolation. We consider standard cubic spline interpolants, where sample points serve as nodes separating cubic pieces with continuous first and second derivatives. While extremely stable, cubic splines typically exhibit a fixed algebraic convergence rate of $O(n^{-4})$ for smooth functions, which is slower than the spectral convergence potential of other methods.

2.5 Floater-Hormann Rational Interpolation

Floater and Hormann introduced a family of rational interpolants in barycentric form that are guaranteed to have no poles in the interpolation interval. The weights in the barycentric formula can be adjusted to any prescribed order of accuracy.

2.6 AAA

The AAA-algorithm is mentioned as in midtern project, and the formula is gaiven as:

$$r(x) = \frac{n(x)}{d(x)} = \frac{\sum_{j=1}^{m} \frac{w_j f(x_j)}{x - x_j}}{\sum_{j=1}^{m} \frac{w_j}{x - x_j}},$$

which is in the berrycentric form.

2.6.1 Bad Poles and Froissart Doublets

A key difficulty faced by the AAA algorithm is the appearance of undesirable singularities, commonly referred to as "bad poles." When approximating on an interval (for example, $[-1, 1]$), a pole p of the rational approximant $r(z)$ is classified as "bad" if it falls inside the approximation domain, that is, if $p \in [-1, 1]$.

To mitigate this, a post-processing "cleanup" step is often required. This involves detecting poles that satisfy $p \in [-1,1]$ (or those with negligible residues), removing them from the barycentric support, and re-calculating the weights using a linear least-squares fit based on the remaining valid poles (AAA-LS).

Algorithm: Spurious Pole Removal (Cleanup Phase)

1. **Decomposition :** Compute the poles p_j and residues c_j of the rational approximant $r(z)$:
$$r(z) = \sum_{j=1}^{m} \frac{c_j}{z - p_j} + d$$
2. **Identification:** Identify the set of indices $\mathcal{J}_{\text{bad}}$ corresponding to poles located inside the domain $[-1,1]$:
$$\mathcal{J}_{\text{bad}} = \{j \mid p_j \in [-1,1]\}$$
3. **Removal & Reconstruction:** Construct the cleaned rational function $\tilde{r}(z)$ by discarding the terms in $\mathcal{J}_{\text{bad}}$:
$$\tilde{r}(z) = d + \sum_{j \notin \mathcal{J}_{\text{bad}}} \frac{c_j}{z - p_j}$$

3 Numerical comparison

In this section, we present a comprehensive numerical comparison of the AAA algorithm against the five existing methods described in the previous chapter. We evaluate their performance on a set of test functions with varying mathematical properties, ranging from simple analytic functions to those with poles, branch points, and non-differentiable points.

The five distinct test function is given below

1. $f_A(x) = \sqrt{1.21 - x^2}$: branch points at 1.1.
2. $f_B(x) = \sqrt{0.01 + x^2}$: branch points at $0.1i$.
3. $f_C(x) = \tanh(5x)$: poles on the imaginary axis.
4. $f_D(x) = \sin(40x)$: entire, oscillatory.
5. $f_E(x) = \exp(-1/x^2)$: C^∞ but not analytic.

3.1 Results and Discussion

The numerical results correspond to the convergence plots shown in Figure 2 of the paper. The comparison reveals the following key findings regarding the AAA algorithm's performance relative to the other five methods.

3.1.1 Superior Accuracy and Efficiency

The AAA algorithm consistently achieves the best performance among the tested methods. In most cases, it reaches a maximal error close to machine precision (10^{-13}) with fewer degrees of freedom (number of sample points n) than the competing algorithms.

3.1.2 Performance on Meromorphic Functions (f_C)

The most distinct advantage of AAA is observed for the function $f_C(x) = \tanh(5x)$.

- **Pole Detection:** This function has singularities (poles) on the imaginary axis. The AAA algorithm automatically locates these poles.
- **Rapid Convergence:** Once the sampling density is sufficient to resolve these poles, the error of the AAA approximation drops rapidly (exhibiting geometric convergence).
- **Comparison:** In contrast, polynomial-based methods (Poly-LS) converge very slowly because polynomials cannot efficiently approximate the sharp transition of the hyperbolic tangent function without an excessive number of degrees of freedom.

3.1.3 Performance on Functions with Branch Points (f_A, f_B)

For functions with branch points, such as f_A and f_B:

- AAA simulates the branch cuts by aligning poles along the cut in the complex plane.
- Even when singularities are very close to the interval (as in f_B), AAA remains stable and outperforms Fourier extension and polynomial methods.

3.1.4 Performance on Non-Analytic Functions (f_E)

For the function $f_E(x) = \exp(-1/x^2)$, which has an essential singularity at $x = 0$:

- Theoretical limitations dictate that no method can achieve exponential convergence; the best possible rate is root-exponential.
- The results show that AAA adapts to this limitation. It performs comparably to the Floater-Hormann rational interpolant, demonstrating that the method is robust even when the function lacks simple poles to exploit.

4 Convergence Properties

This section investigates the theoretical convergence behavior of the AAA algorithm on equispaced data. The discussion focuses on two main aspects: the algorithm's performance on functions without exploitable pole structures and its relationship with the theoretical "impossibility theorem" regarding equispaced approximations.

4.1 The Amber Function Test

To test the robustness of AAA in the absence of obvious singularities (poles) to exploit, the authors construct a special test function called the "Amber function," denoted as $A(x)$.

- **Construction:** $A(x)$ is constructed using a series with random-like coefficients determined by the binary expansion of π. It is designed to be analytic within an ellipse but possesses a natural boundary, meaning it has no isolated poles or branch points that can be easily "captured."
- **Result:** Numerical experiments show that even for this unstructured function, AAA remains stable. Its convergence rate becomes root-exponential, closely matching that of the Floater-Hormann rational interpolant.

- **Conclusion:** This demonstrates that AAA is a safe, general-purpose method. It accelerates convergence when poles are present but degrades gracefully to a standard robust convergence rate when they are not.

4.2 Reconciling with the Impossibility Theorem

A central theoretical issue is the "Impossibility Theorem", which states that no stable linear method can achieve exponential convergence on equispaced data. AAA appears to violate this by being both fast and stable. The paper explains this apparent contradiction through two factors:

4.2.1 Nonlinearity

The Impossibility Theorem applies strictly to linear methods. AAA is inherently nonlinear because the locations of the poles are determined adaptively based on the data, not fixed in advance.

4.2.2 Numerical vs. Mathematical Interpolation

More importantly, the theorem assumes that the method attempts to drive the error to zero as $n \to \infty$.

- AAA avoids the instability associated with exponential convergence by acting as a numerical interpolant.
- It stops adding degrees of freedom once the error reaches the machine precision floor.
- By not attempting to resolve the function below this noise floor, AAA effectively "truncates" the convergence process before the exponential instability can manifest.

5 Conclusion

This study establishes AAA rational approximation as a powerful and versatile tool for the interpolation of equispaced data, effectively overcoming the traditional stability issues associated with the Runge phenomenon. By treating the approximation problem through the lens of rational functions rather than polynomials, the method offers a paradigm shift in how equispaced data can be handled.

5.1 Summary of Contributions

The key contributions of this paper can be summarized as follows:

- **Robustness via Cleanup:** The introduction of the AAA-Least Squares (AAA-LS) cleanup procedure successfully resolves the issue of spurious poles (Froissart doublets) appearing within the interpolation interval. This ensures that the algorithm remains stable even when the sampling grid is coarse.
- **Structural Insight:** Unlike polynomial methods that blindly fit data, the AAA algorithm exploits the underlying analytic structure of the function. As demonstrated with meromorphic functions (e.g., $\tanh(5x)$), AAA achieves geometric convergence by automatically detecting and placing poles near the true singularities of the function.

- **Theoretical Reconciliation:** The paper provides a theoretical justification for the method's success in the face of the Impossibility Theorem. By operating as a nonlinear "numerical interpolant"—which stops refining once machine precision is reached—AAA avoids the exponential instability that plagues linear convergence schemes.

5.2 Final conclusion

Numerical comparisons against five established algorithms reveal that AAA is typically the fastest and most accurate method for smooth functions. While it may require parameter adjustment for noisy data and can be computationally more expensive for extremely high-degree approximations, its ability to combine the flexibility of rational functions with the stability of least-squares fitting makes it a superior choice.

In conclusion, for researchers and practitioners dealing with equispaced data, the AAA algorithm represents a robust, general-purpose default solver that often outperforms traditional splines and polynomial-based techniques.