Data-driven computing with trigonometric rational functions

Heather Wilber
University of Texas at Austin

Anil Damle
Cornell University

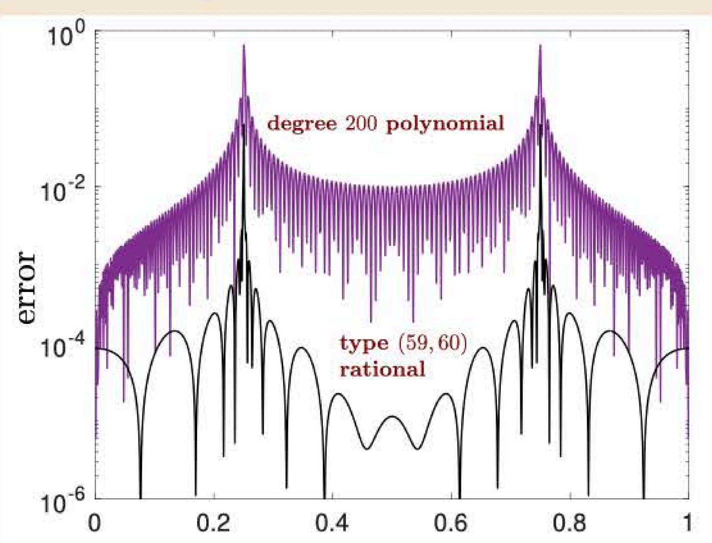
Alex Townsend
Cornell University

Signals with singularities

Reconstructing signals from noisy, incomplete or corrupted samples is challenging enough, but in many applications (e.g., [6, 8, 9, 10, 13]) an added complication arises. The underlying signal contains impulses, shocks, or sharp features that can cause traditional Fourier-based methods to underperform or fail.

We introduce a suite of tools for reconstructing and computing with such signals via data-driven rational approximation methods. **Our computing framework combines two complementary representations:(1) barycentric trigonometric rational functions, and (2) their Fourier transforms, which are short sums of complex exponentials.** Efficiently toggling between these representations lets us overcome computational and data-related challenges.

Why rationals?

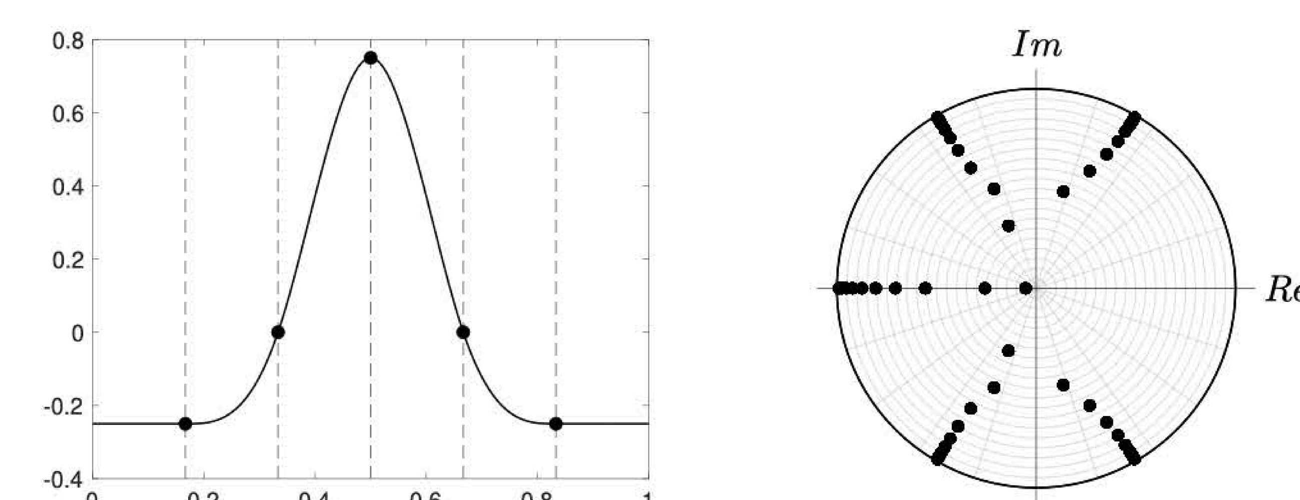


Rational functions are effective at representing signals in regions around singularities, whereas polynomial (e.g., Fourier methods) are hopeless. Automated, reliable tools for signal processing with rationals can help us make progress in applications involving these challenging regimes.

Left: Errors are plotted on a logarithmic scale for polynomial (purple) and rational (black) approximations to a function with jump discontinuities. Polynomial errors can decay at most at an algebraic rate with respect to the distance from the singularity. In contrast, error for a rational approximation can decay exponentially fast [19].

Can you spot the knots?

Our methods are **data-driven**. They require no knowledge from the user about the number, locations, or types of singularities present. In fact, they can be used to identify and classify such features! In this example, we fit a rational approximation r to samples from a cubic spline (left). The locations of the knots in the spline are revealed by the clusters of poles of r (right, shown in the z-plane, $r(z) = r(e^{-2\pi i x})$.)



Left: A type (43, 44) trigonometric rational function $r(x)$ approximating a cubic spline on $[0, 1)$. The knots of the spline occur at the dotted lines. *Right:* Poles of $r(z)$, $z = e^{2\pi i x}$ with magnitude < 1 are plotted in the unit disk. They cluster toward points $e^{2\pi i x_k}$, where each x_k is the location of a knot.

Dual representations

Let $f : [0, 1) \to \mathbb{R}$ be a continuous periodic function of bounded variation, with zero mean value over $[0, 1)$. We observe (possibly noisy) samples of f at locations $\{x_j\}_{j=0}^{2N}$, and seek useful representations of f in both the time and frequency domains. By useful, we mean that the representations are **cheap to store**, **stable to evaluate**, and **conducive to efficient algorithms** for addition, convolution, differentiation, etc...

Direct construction methods

the time domain: barycentric interpolants

A trigonometric variant of the AAA algorithm [16] can be used to construct a type $(m-1, m)$ trigonometric rational in barycentric form [3,11]:

$$r(x) = \frac{\sum_{j=1}^{2m} \gamma_j f_j \cot(\pi(x - t_j))}{\sum_{j=1}^{2m} \gamma_j \cot(\pi(x - t_j))}, \qquad \sum_{j=1}^{2m} \gamma_j f_j = 0.$$

The trigonometric rational has m pairs of poles $\{\eta_j, \overline{\eta}_j\}$, where $Re(\eta_j) \in [0, 1)$. It also satisfies the interpolation property: $r(t_j) = f_j = f(t_j)$.

Advantages: impute missing data, high resolution in time domain, stable evaluation [1, 12], differentiation formula [2], rootfinding/locating extrema [16, 20].

the frequency domain: exponential sums

The trigonometric rational r has a Fourier series that might decay very slowly! However, its Fourier coefficients $\{\hat{r}_k\}_{k=-\infty}^{\infty}$ are given by the following [4,18]:

$$\hat{r}_k = R(k) := \begin{cases} \sum_{j=1}^{m} \omega_j e^{\alpha_j k}, & k \geq 0, \\ \sum_{j=1}^{m} \overline{\omega}_j e^{-\overline{\alpha}_j k}, & k < 0. \end{cases}$$

We only need to store the m pairs $\{(\omega_j, \alpha_j)\}$ to generate all the Fourier information for r. Using a stable version of Prony's method [4,17], we can directly construct $R \approx \mathcal{F}(f)$ from Fourier samples of f.

Advantages: robust to noise [20], near-optimal re-compression algorithm [4, 20], algebraic operations (e.g., sums/products), convolution, filtering.

Software

REfit
Rationals and Exponentials

Our algorithms are implemented in an open source software package in MATLAB called REfit. Inspired by Chebfun [7], the package uses two classes called `rfun` and `efun`. Rfun objects store barycentric rational representations, and efun objects store exponential sums in frequency space.

These objects can be manipulated using dozens of overloaded commands, like the following:

`+, -, .*, ./`	basic arithmetic
`diff(.), cumsum(.)`	differentiation, indefinite integration
`conv(.), corr(.,.)`	convolution, cross-correlation
`ft(.), ift(.,.)`	Fourier and inverse Fourier transforms

Whenever possible, we **automatically recompress** representations with near-optimal approximations after performing operations.

Combining representations

Fourier and inverse Fourier transforms

Our paradigm includes routines that build approximations directly from samples in either space, as well as specialized Fourier and inverse Fourier transforms that construct exponential sums in Fourier space directly from barycentric interpolants, and vice-versa.

These transform functions are crucial because (1) they allow us to shift between the frequency and time domains as needed for various downstream computations, and (2) they allow us to overcome issues (e.g., noise, undersampling) that could cause one of the direct construction methods to fail.

Exchanging exact recovery for stability

Given the poles and residues of r, the parameters in $R = \mathcal{F}(r)$ can be expressed explicitly [4]. Likewise, one can write down r in pole-residue form given R. However, exactly recovering the poles and residues in settings where the poles are clustered near a singularity is an ill-conditioned problem [14]. As a result, these formulas are too unstable to use directly. Finding good barycentric support (or interpolating) points adds yet another layer of complication to the computation of $r = \mathcal{F}^{-1}(R)$ in barycentric form. The transforms we develop are lossy due to regularization steps, but crucially, they are stable.

Fourier transform: Approximate poles and residues of r can be computed in only $\mathcal{O}(m^3)$ operations [16]. These are used to recover the parameters in the exponential sums R, along with a regularization step involving a small set (usually $\mathcal{O}(m)$) of Fourier coefficients.

Inverse Fourier transform: To construct $r = \mathcal{F}^{-1}(R)$ in barycentric form, we compute a sample of r using R and the FFT. Approximate poles of r, known from R, are used in a stability-based algorithm involving column-pivoted QR to choose good interpolating points. Then, a linearized least-squares fit to the samples finds the barycentric weights. The cost is $\mathcal{O}(N \log N + Nm^2)$, where N is the number of samples and is related to the approximate bandlimit of r.

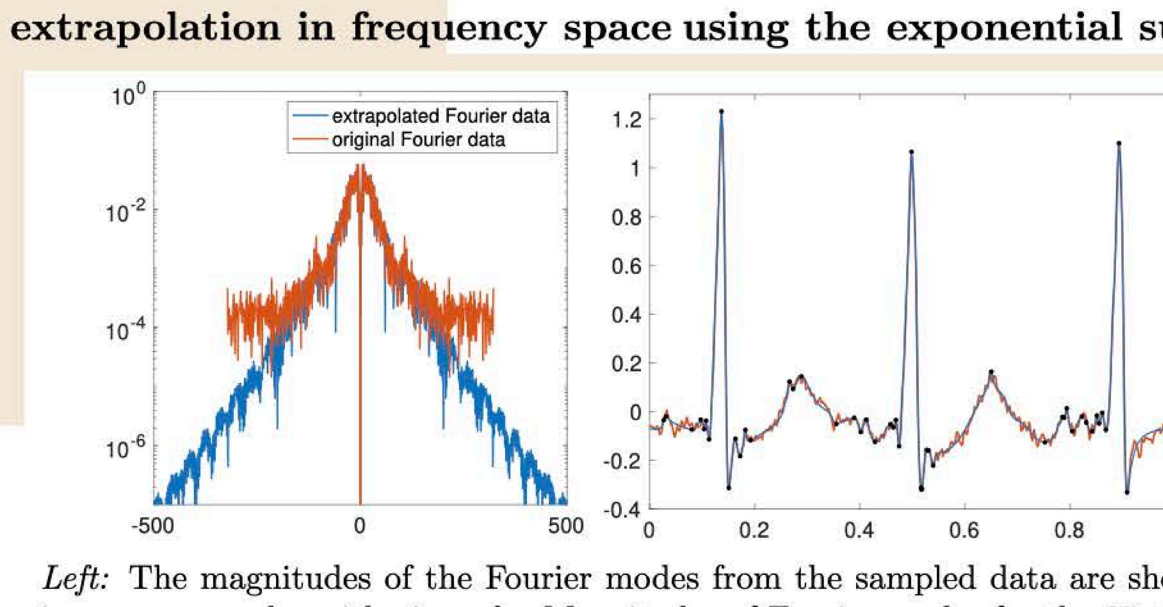
Example: ECG signal reconstruction

In this example, we fit a rational function to noisy ECG data [8, 9] taken from the PhysioNet MIT BIH arrhythmia database [15]. Having the rational in barycentric form is convenient for identifying local extrema, but we cannot apply AAA directly, as it is sensitive to noise. Even without noise, the signal is undersampled (645 samples). Without an enriched sample, AAA will fail to produce a stable interpolant.

To overcome these issues, we work in frequency space and construct a 35-term exponential sum R using the regularized Prony's method. This **automatically denoises the signal**. Then, we use our inverse Fourier transform to construct $r = \mathcal{F}^{-1}(R)$. Leveraging the barycentric form, we find local extrema. **Crucially, the signal is enriched via extrapolation in frequency space using the exponential sums.**

This can be viewed as a type of superresolution [5]. Most of this process is automatic; the entire procedure runs with 3 lines of code in MATLAB.

```
R = efun(data, 'tol', 1e-3);
r = ift(R);
extrema = [min(r); max(r)];
```



Left: The magnitudes of the Fourier modes from the sampled data are shown in orange on a logarithmic scale. Magnitudes of Fourier modes for the 35-term exponential sum approximant (R) are shown in blue. *Right:* The original signal (orange), along with the reconstructed rational approximant (r) (blue), andlocal extrema (black dots).

References

[1] Austin and Xu, *On the numerical stability of the second barycentric formula for trigonometric interpolation in shifted equispaced points*, IMA J. Numer. Anal. [2] Baltensperger *Some results on linear rational trigonometric interpolation*, Comp. Math. Appl. [3] Berrut and Trefethen, *Barycentric Lagarange interpolation*, SIAM Rev. [4] Beylkin and Monzón, *Nonlinear inversion of a band-limited Fourier transform*, Appl. and Comp. Harm. Anal. [5] Candès and Fernandez-Granda, *Toward a mathematical theory of super-resolution*, Comm. Pure Appl. Math. [6] Červený, Popov, and Pšenčík, *Computation of wave fields in inhomogeneous media–Gaussian beam approach*, Geophys. J. Int.

[7] Driscoll, Hale and Trefethen, *Chebfun Guide*, Pafnuty Publications, Oxford, 2014. [8] Fridli, Lócsi, and Schipp *Rational function systems in ECG processing*, in Int. Conf. Comp. Aided Sys. Theory, Springer, 2011, pp. 88–95. [9] Gilián, *ECG-based heart beat detection using rational functions*, in Conf. Dyadic Anal. Rel. Fields Appl., vol. 13, 2014 [10] Haut, Beylkin and Monzón, *Solving Burgers' equation using optimal rational approximations*, Appl. Comp. harm. Appl. [11] Henrici, *Barycentric formulas for interpolating trigonometric polynomials and their conjugates*, Num. Math. [12] Higham, *The numerical stability of barycentric Lagrange interpolation*, IMA J. Numer. Anal. [13] Martin, *A linear algorithm for ambient seismic noise double beamforming without explicit crosscorrelations*, Geophysics.

[14] Miller, *Stabilized numerical analytic prolongation with poles*, SIAM J. Appl. Math. [15] Moody, Mark and Goldberger, *Physionet: A research resource for studies of complex physiologic and biomedical signals*, in Computers and Cardiology 2000, vol. 27, IEEE, 2000. [16] Nakatsukasa, Sète and Trefethen, *The AAA algorithm for rational approximation*, SIAM J. Sci. Comput. [17] Potts and Tasche, *Parameter estimation for nonincreasing exponential sums by Prony-like methods*, Lin. Alg. Appl. [18] Stein and Shakarchi, *Complex Analysis*, vol. 2, Princeton University Press, 2010. [19] Trefethen, *Approximation theory and practice*, SIAM, 2013. [20] Wilber, Damle and Townsend, *Data-driven algorithms for signal processing with trigonometric rational functions*, SIAM J. Sci. Comput.