Data-driven computing with trigonometric rational functions

Heather Wilber

April 21, 2022

Joint work with

Alex Townsend
Cornell University

Anil Damle
Cornell University

ODEN INSTITUTE
FOR COMPUTATIONAL ENGINEERING & SCIENCES

When are rationals useful?

When our toolbox is limited to the basic arithmetic operations $(+,\ -,\ \times,\ \div)$, the functions we can make are polynomials and rationals.

$$\sqrt{A} \qquad \exp(A) \qquad \text{sign}(\text{A})$$

Rationals appear in the fundamental things we do in numerical linear algebra.

Rational functions have excellent approximation power near singularities.

...and so much more!

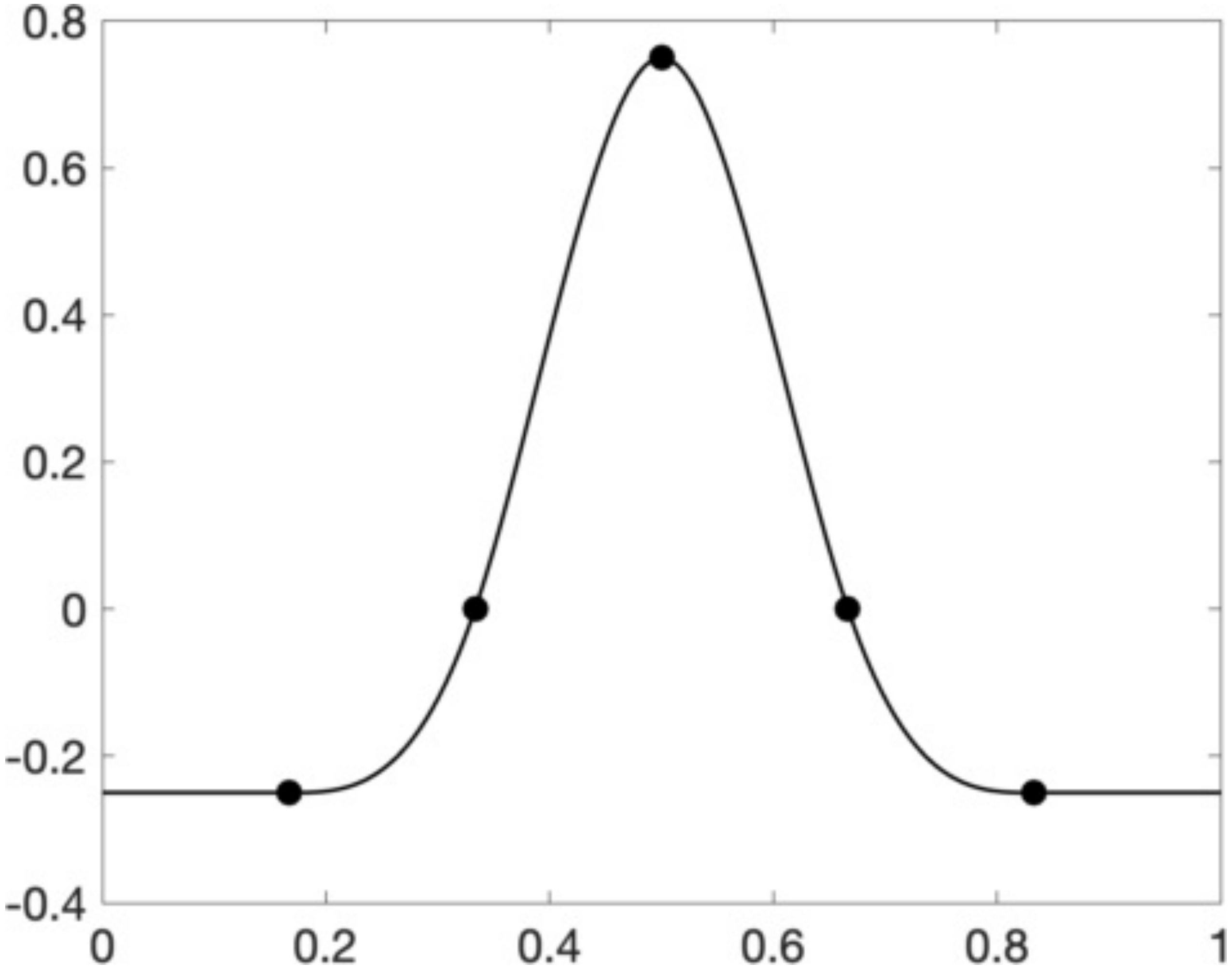
Applications in signal processing

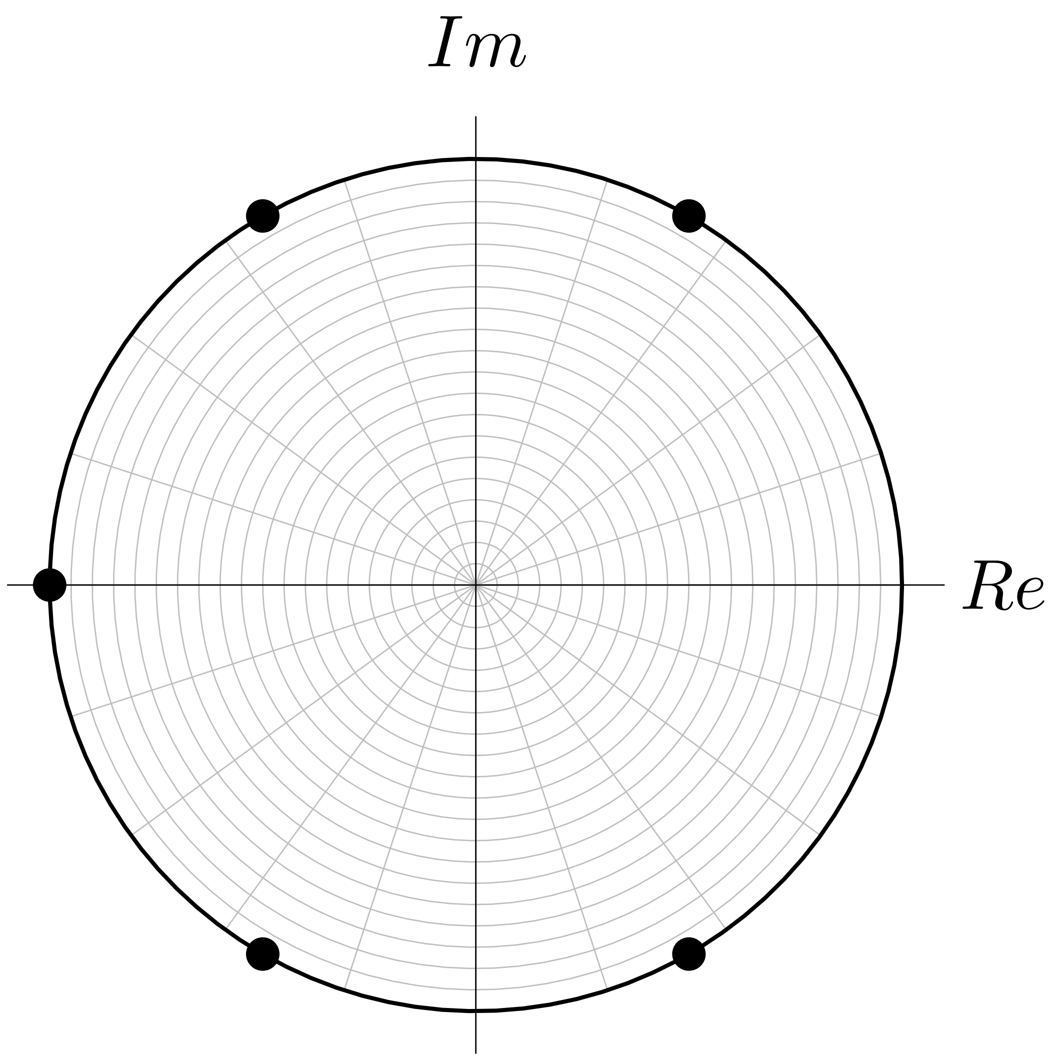
Rationals are useful for...

- recovering signals with slowly decaying spectral content.
  (approximations to signals with sharp features, rapid transitions)
- representing functions sparsely in both frequency and time domains.
- filtering noise.
- imputing missing data.
- extrapolation.
- identifying/locating singularities.

Applications in signal processing: When are rationals useful?

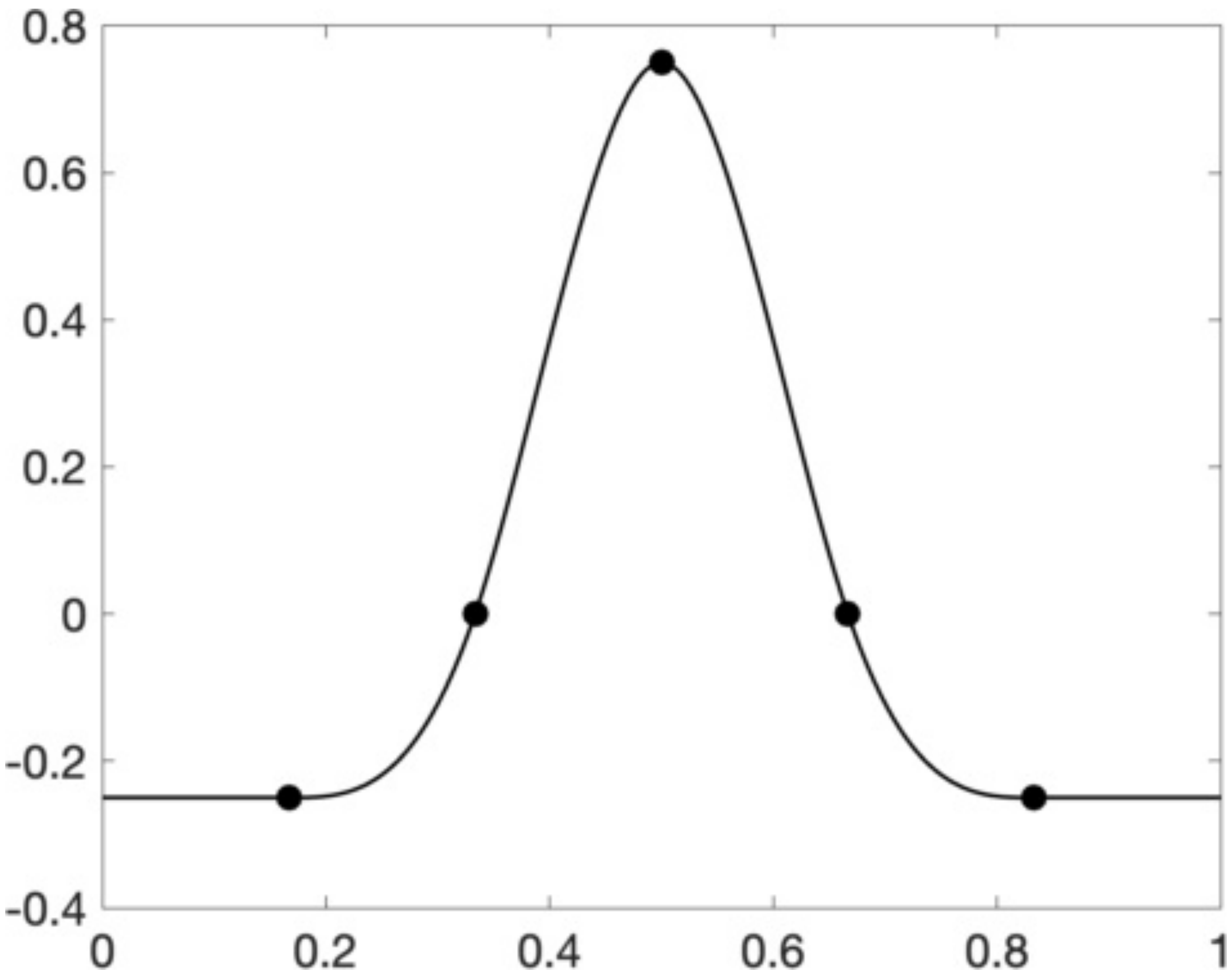
Example: Identifying singularities

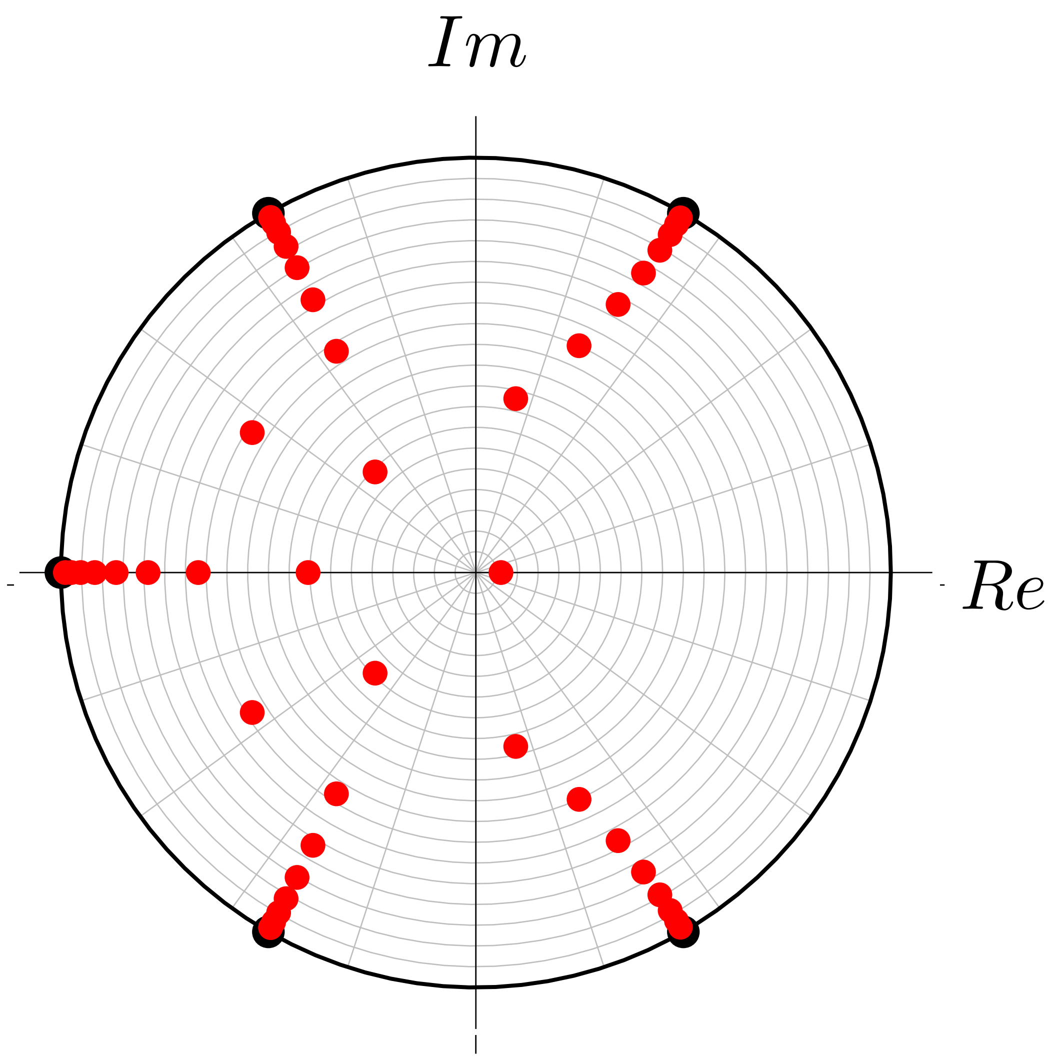




Applications in signal processing: When are rationals useful?

Example: Identifying singularities





Applications in signal processing: When are rationals useful?

Signal reconstruction: geophysics and seismology, biomedical monitoring, extrapolation/superresolution, filtering

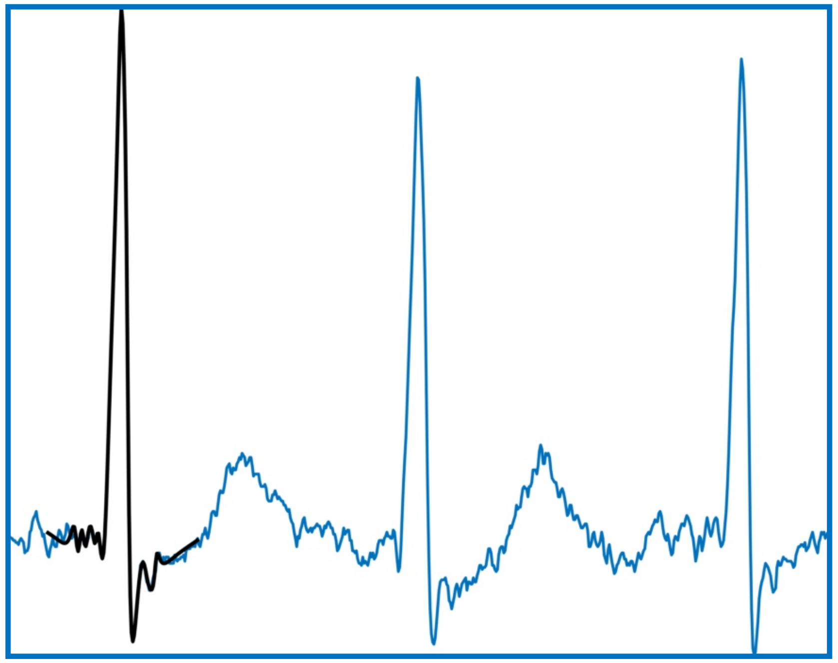
[Belykin and Monzón (2009), Moitra (2018), Fridli, Lósci and Schipp (2012), Vetterli, Marziliano, and Blu (2002)]

Feature extraction: abnormality detection, classification, parameter recovery

[Gilián (2016), Moitra (2018) , Peter and Plonka (2013), Potts and Tasche (2013)]

Related methods: wavelets, RBFs, splines

[De Boor, Debnath, Wendland,Unser and Blu, and many more]



Reconstructed ECG signal in REfit
(W., Damle, Townsend, 2022)

Data-driven rational approximation for signal reconstruction

GOAL: Develop software tools for working adaptively with trigonometric rational approximations to periodic functions.

- "Near-optimal" rational approximations
- Data-driven: no tuning parameters
- Works with noisy, under-resolved, missing data.
- Basic tools: algebraic operations (sums, products), differentiation, integration, filtering, rootfinding, polefinding, visualization, etc.

Regularized Prony's method

(Fourier domain)

⇄

The AAA algorithm

(time domain)

Trigonometric rational functions

f is periodic, real-valued, continuous on $[0,1)$, $\int_0^1 f(\theta)\mathrm{d}\theta = 0$.

We seek $r_m \approx f$, where

$$r_m(x) = \frac{p_{m-1}(x)}{q_m(x)} = \frac{\sum_{j=-(m-1)}^{m-1} a_j e^{2\pi i j x}}{\sum_{j=-m}^{m} b_j e^{2\pi i j x}}, \quad x \in [0,1).$$

r_m has $2m$ simple poles, $\{\eta_j, \overline{\eta}_j\}_{j=1}^m$, $0 \leq Re(\eta_j) < 1$.

Trigonometric rational functions in Fourier space

Key observation: The Fourier series of r_m can be efficiently represented by a short sum of complex, decreasing exponentials.

If $r_m(x) = \sum_{k=-\infty}^{\infty} (\hat{r}_m)_k e^{2\pi i k x}$, then for $k \geq 0$,

$$(\hat{r}_m)_k = R_m(k) := \sum_{j=1}^{m} \omega_j e^{\lambda_j k},$$

where $\lambda_j = 2\pi i \eta_j$, $Re(\eta_j) > 0$.

(Gaspard de Prony)

The parameters of R_m can be exactly recovered by observing $(\hat{r}_m)_0, \cdots, (\hat{r}_m)_{2m}$ (Prony's method)

$r_m \approx f$ can be constructed by solving the approximate interpolation problem $|\hat{f}_k - R_m(k)| \leq \epsilon \|f\|$, for $0 \leq k \leq N_\epsilon$. (Regularized Prony's method)

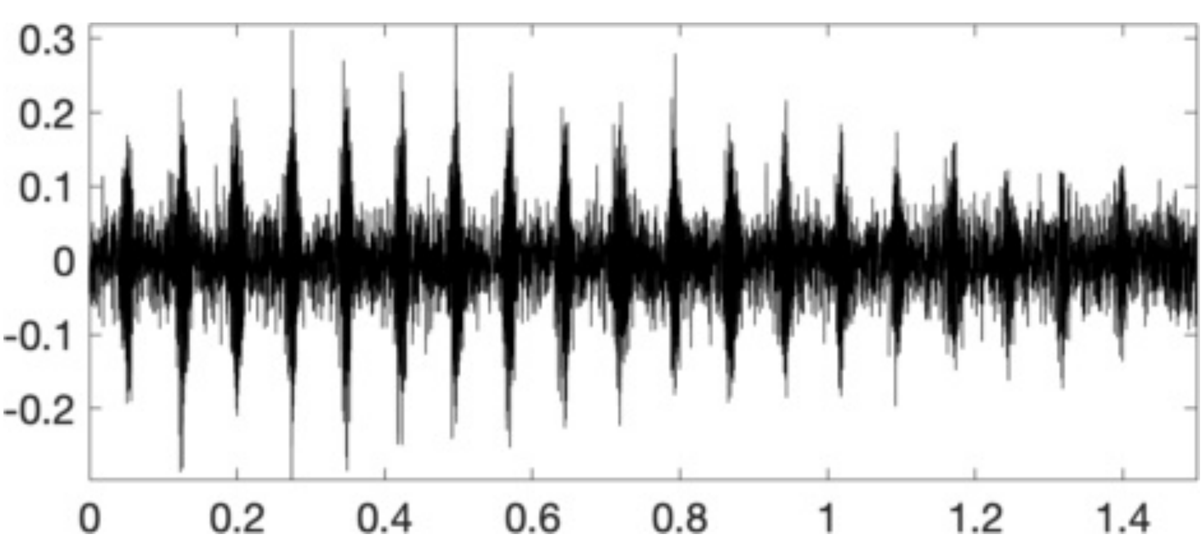
[Adamjan, Arov, and Krein (1971), Beylkin and Monzón (2005, 2009), Pototskaia and Plonka (2016), Potts and Tasche (2010)]

Exponential sum format

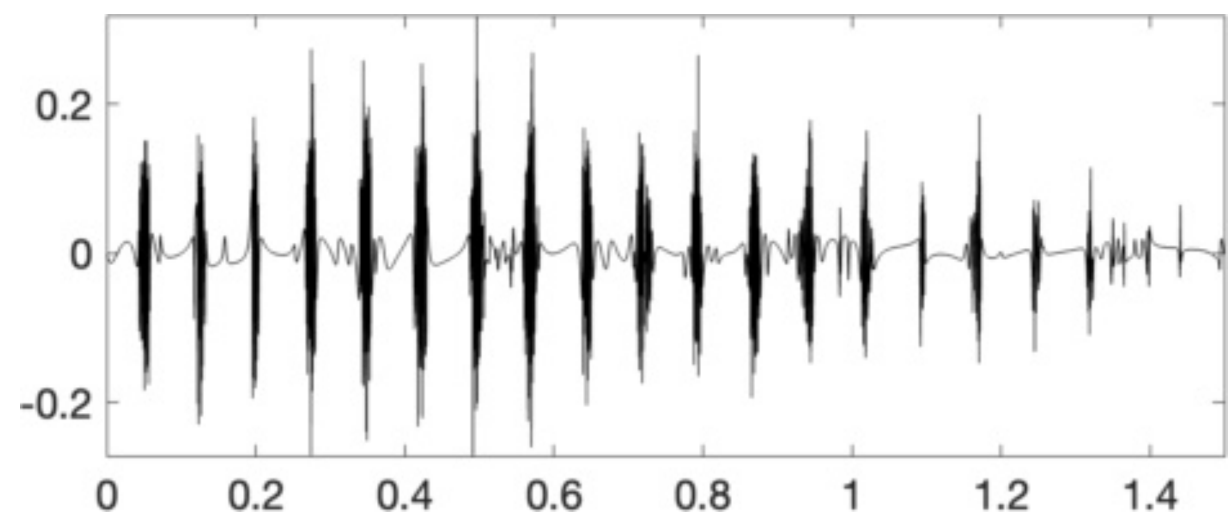
Advantage for reconstruction: Filter for Gaussian noise

Example: Extracting pulses in the Pacific Blue whale's song.

6001 noisy samples from a hydrophone



type $(245, 246)$ trigonometric rational



- Automatic construction in the presence of noise.
- Automatic denoising parameter detection.

[Peter & Plonka (2013), Potts & Tasche (2013), M. Vetterli, P. Marziliano, & T. Blu (2002)]

Exponential sum format

Advantage for postprocessing: Efficient recompression

"This formulation allows us to develop a numerical calculus that includes functions with singularities and sharp transitions…"

–Haut, Beylkin, Monzón (2012)

$$v_n \;+\; s_\ell \;=\; g_{n+\ell} \;\approx r_m$$

$$\mathcal{F}^{-1}(v_n) + \mathcal{F}^{-1}(s_\ell) \;=\; \sum_{j=1}^{n} \hat{\omega}_j e^{\hat{\lambda}_j k} \;+\; \sum_{j=1}^{\ell} \tilde{\omega}_j e^{\tilde{\lambda}_j k} \;\approx\; \sum_{j=1}^{m} \omega_j e^{\lambda_j k}$$

In theory, optimal "reduction" algorithms based on finite–rank Hankel operator properties.
In practice, we can usually find a stable solution in only $\mathcal{O}((n+\ell)^3)$ operations!

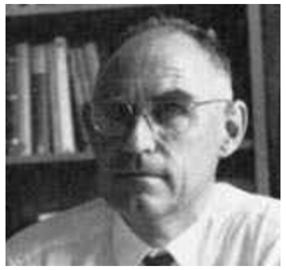
More advantages:

- Works for products, sums, convolutions, derivatives.
- Fast evaluation (on the grid) for derivatives and indefinite integrals.

[Adamjan, Arov, and Krein (1971), Beylkin and Monzon (2005), Haut, Beylkin and Monzón (2012)
Pototskaia and Plonka (2016)]

Trigonometric barycentric rational functions

$$r_m^{t,\gamma}(x) = \frac{n_{m-1}(x)}{d_m(x)} = \frac{\sum_{j=1}^{2m} \gamma_j f_j \cot\left(\pi(x-t_j)\right)}{\sum_{j=1}^{2m} \gamma_j \cot\left(\pi(x-t_j)\right)}, \qquad \sum_{j=1}^{2m} \gamma_j f_j = 0$$

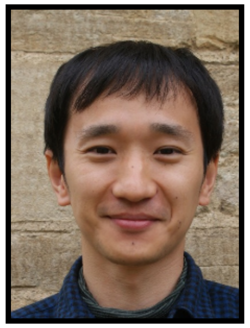


(P. Henrici)

(J.P. Berrut)

Key properties

- r_m is a type $(m-1, m)$ trigonometric rational.
- interpolates f at t_j: $r_m^{t,\gamma}(t_j) = f_j$.
- numerically stable evaluation for $x \in [0,1)$.



(Y. Nakatsukasa)

(L.N. Trefethen)

(O. Sète)

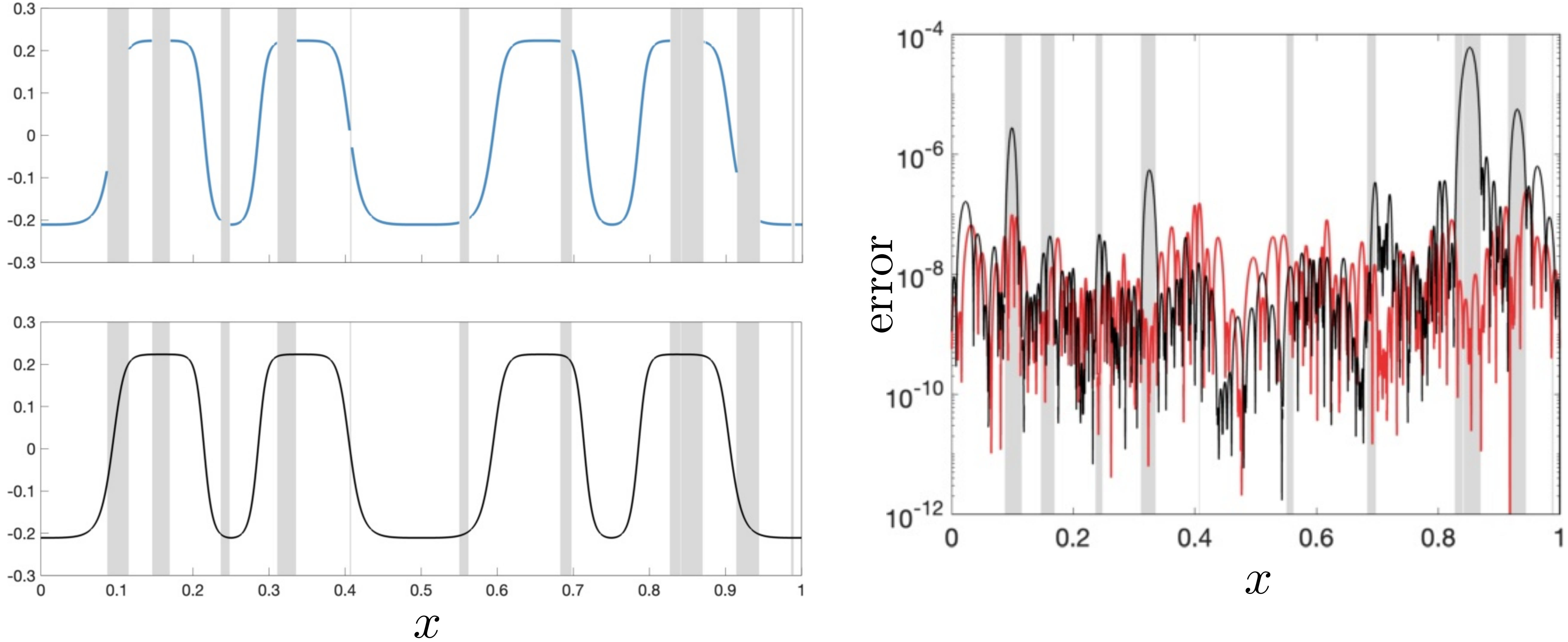
Construct via the PronyAAA algorithm

Key Idea: greedily build up an interpolant, one point at a time, choose weights via linearized least squares fit to data.

[Berrut (2005) , Berrut and Trefethen (2004), Henrici (1979), Higham (2004), Austin and Xu (2017), Nakatsukasa, Trefethen, & Sète (2018), Antoulas & Anderson (1986), Berrut (2005), Badoo(2021)]

PronyAAA algorithm

Advantage for reconstruction: Imputes missing data



AAA does not require equally-spaced or other grid-based sampling schemes.

REfit: barycentric + exponential

$$R_m(k) = \sum_{j=1}^{m} \omega_j e^{\lambda_j k}$$

$$\leftrightarrows$$

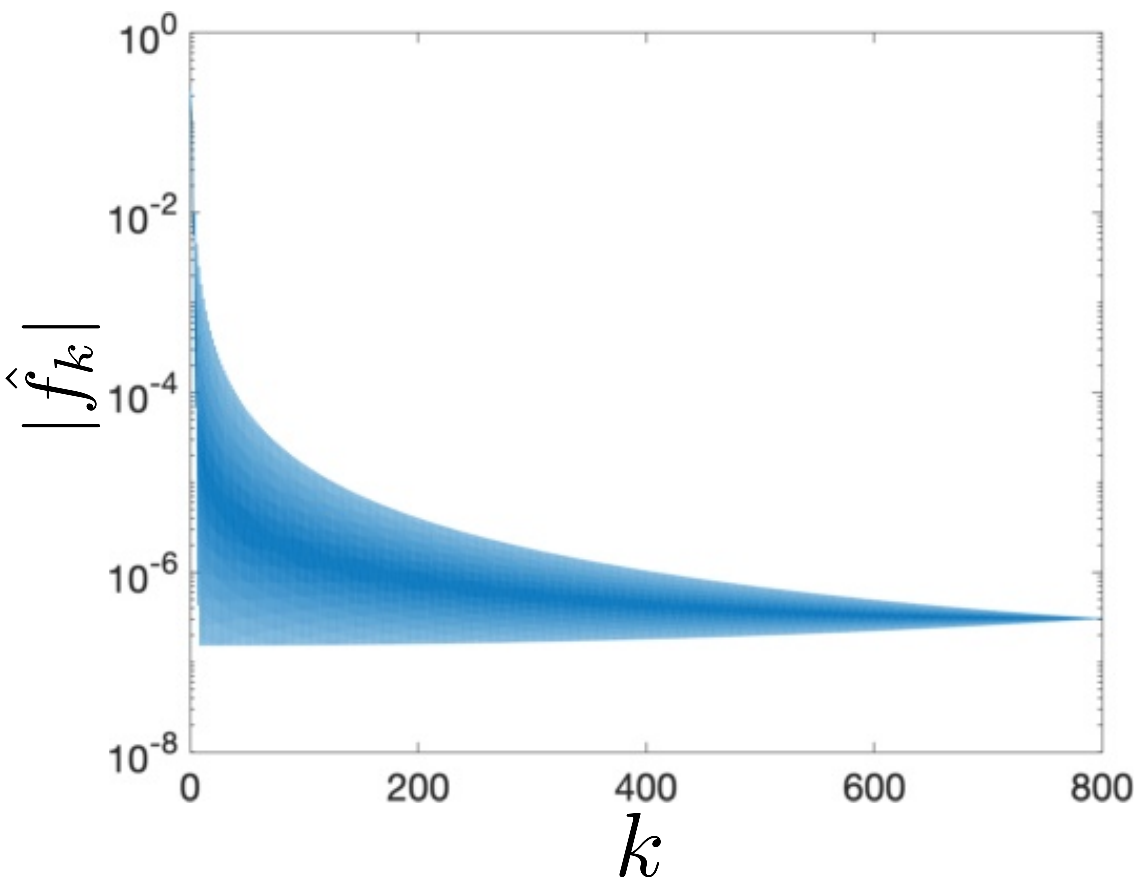
$$r_m^{t,\gamma}(x) = \frac{\sum_{j=1}^{2m} \gamma_j f_j \cot\left(\pi(x - t_j)\right)}{\sum_{j=1}^{2m} \gamma_j \cot\left(\pi(x - t_j)\right)}$$

Exponential sums	Barycentric form
Robustness to noise	Imputing missing data
Filtering and recompression	Differentiation (closed-form formula)
Pole symmetry preservation	Stable evaluation
convolution, cross-correlations	Rootfinding, identifying extrema

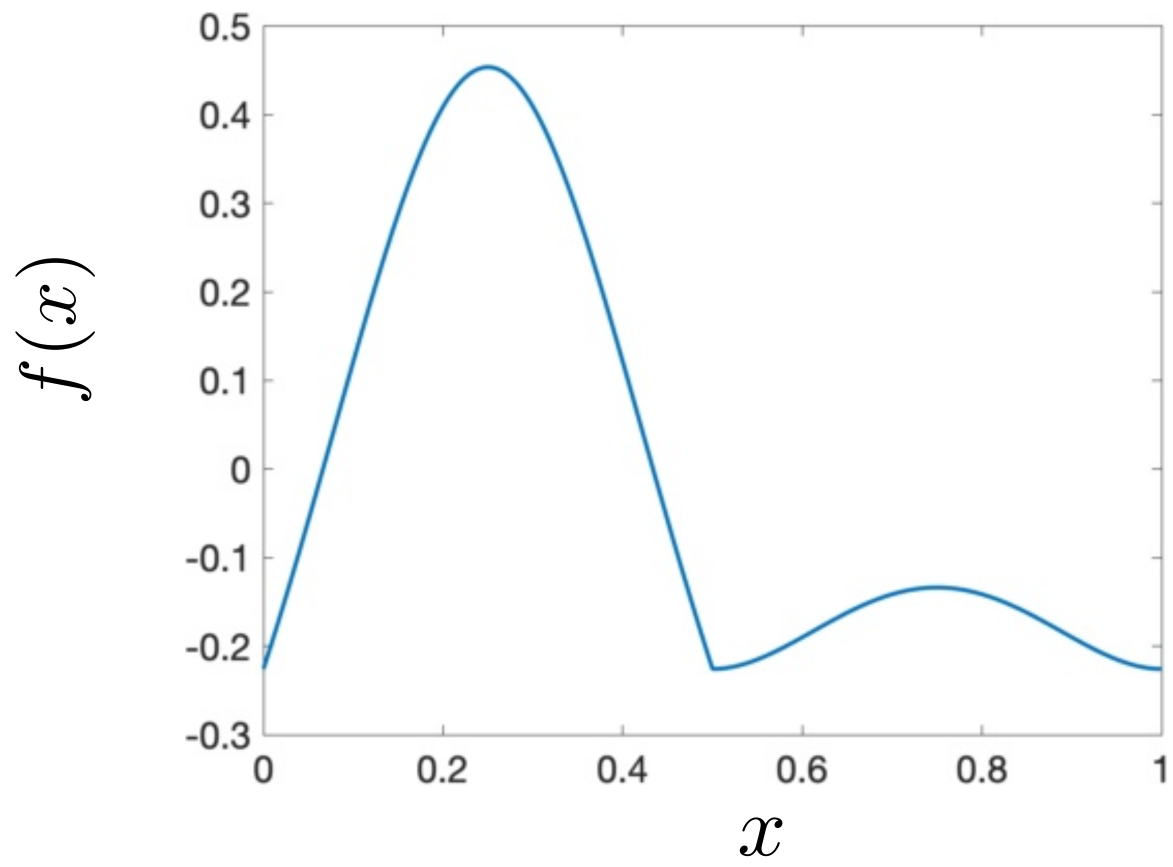
REfit: barycentric + exponential

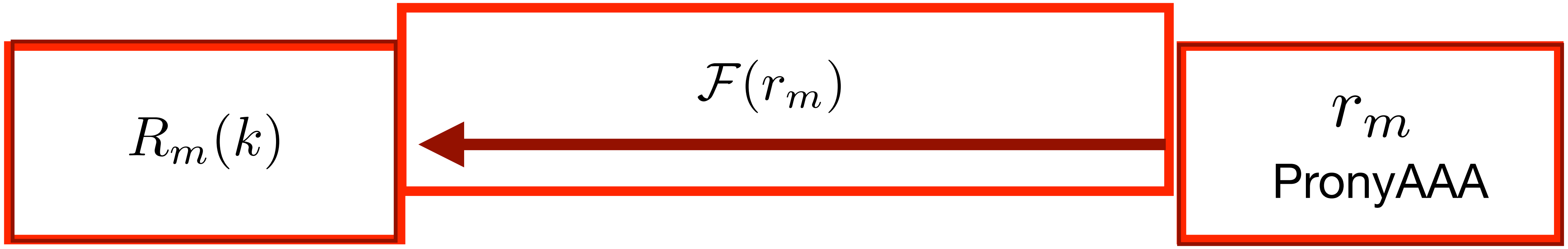
Problem: Fourier coefficients decay slowly, sample is underresolved...
How can I construct an exponential sum representation of $r_m \approx f$?

(Fourier space)



(Time)

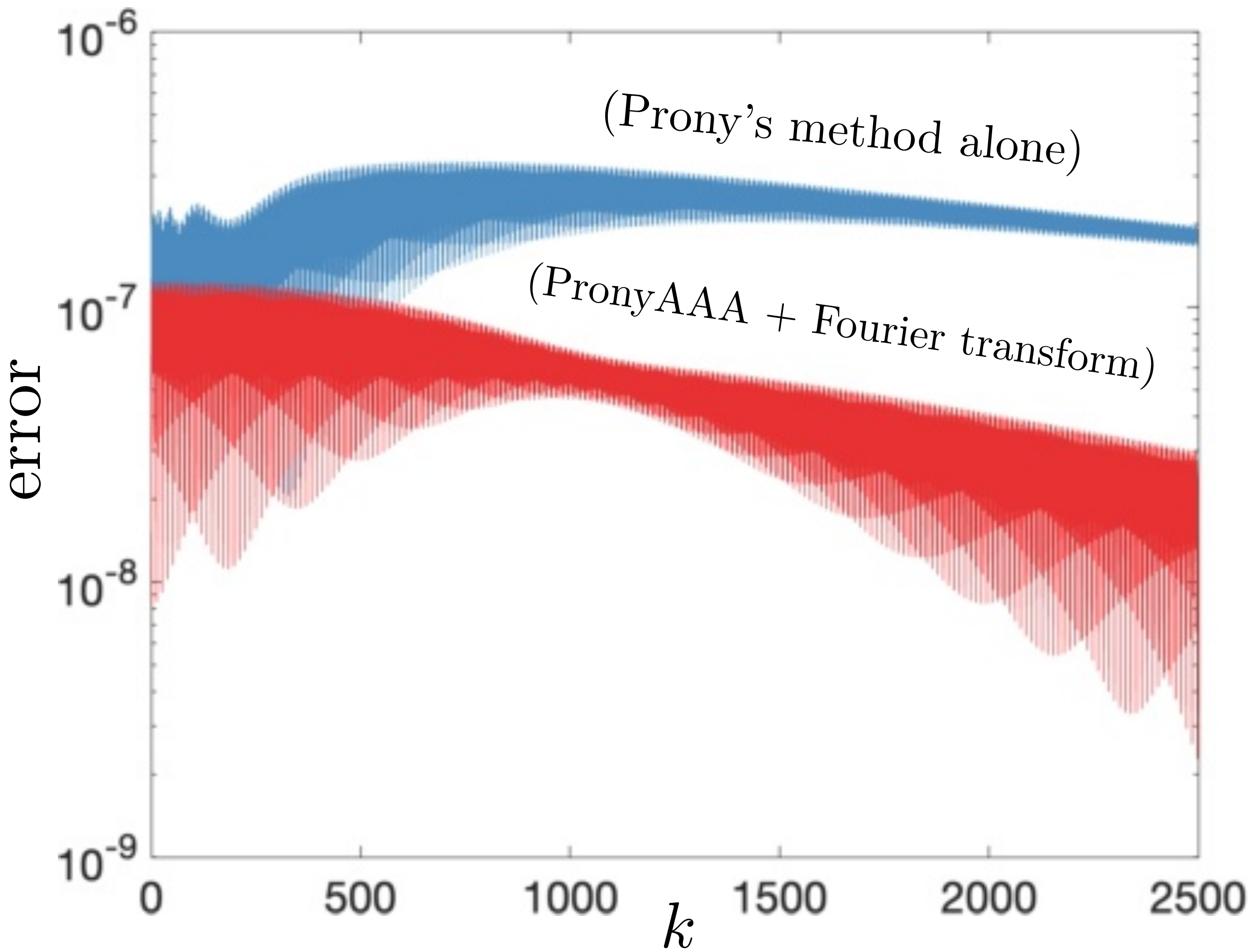




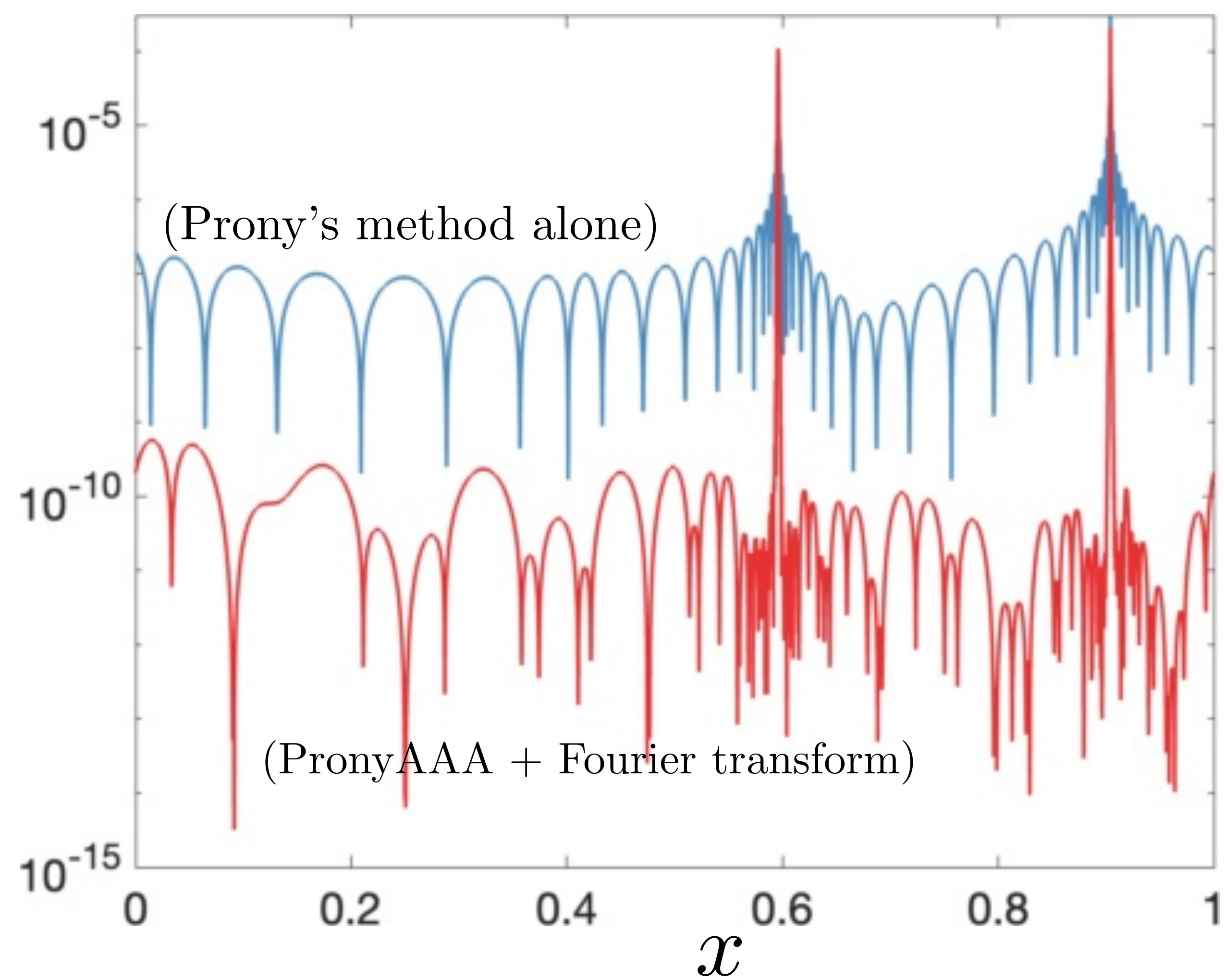
REfit: barycentric + exponential

Problem: Fourier coefficients decay slowly, sample is underresolved...
How can I construct an exponential sum representation of $r_m \approx f$?

(Fourier space)



(Time)

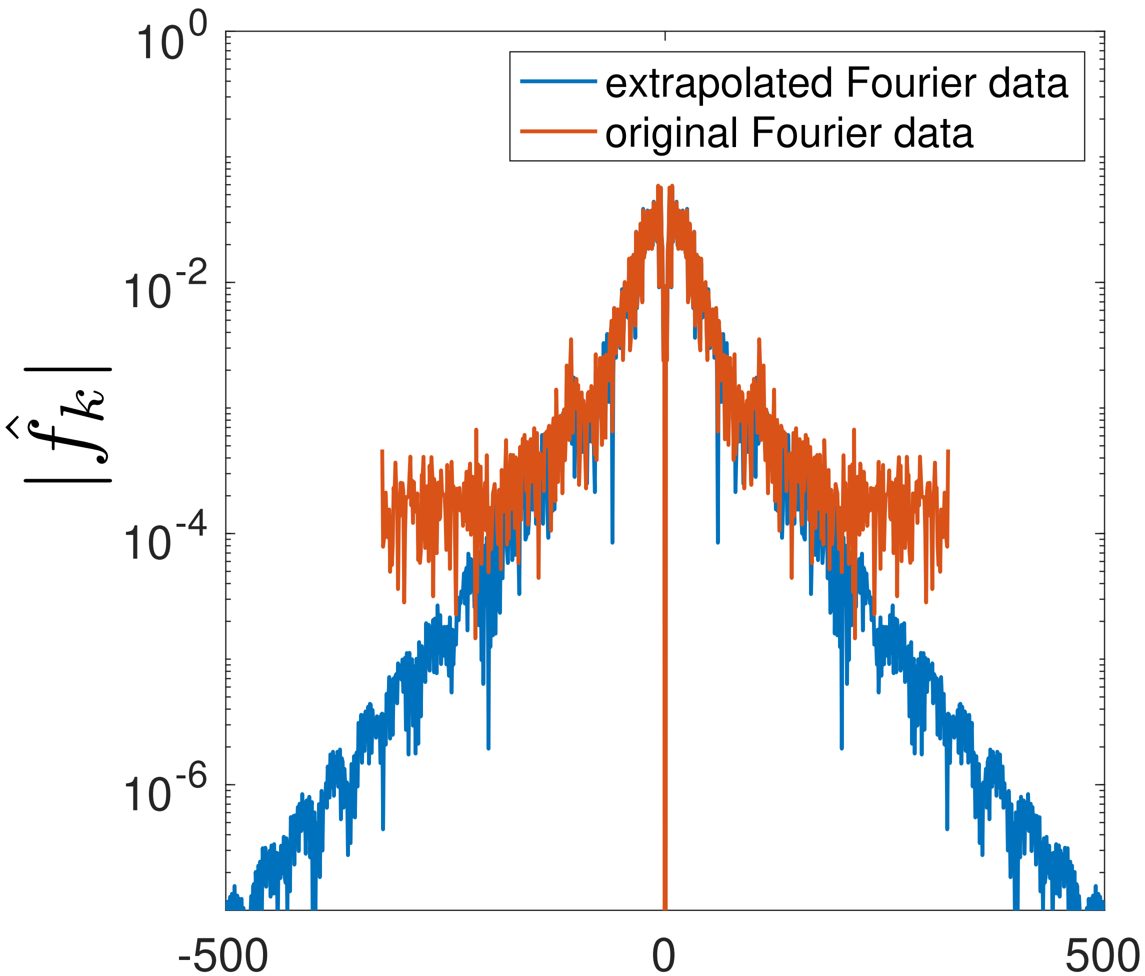


REfit: barycentric + exponential

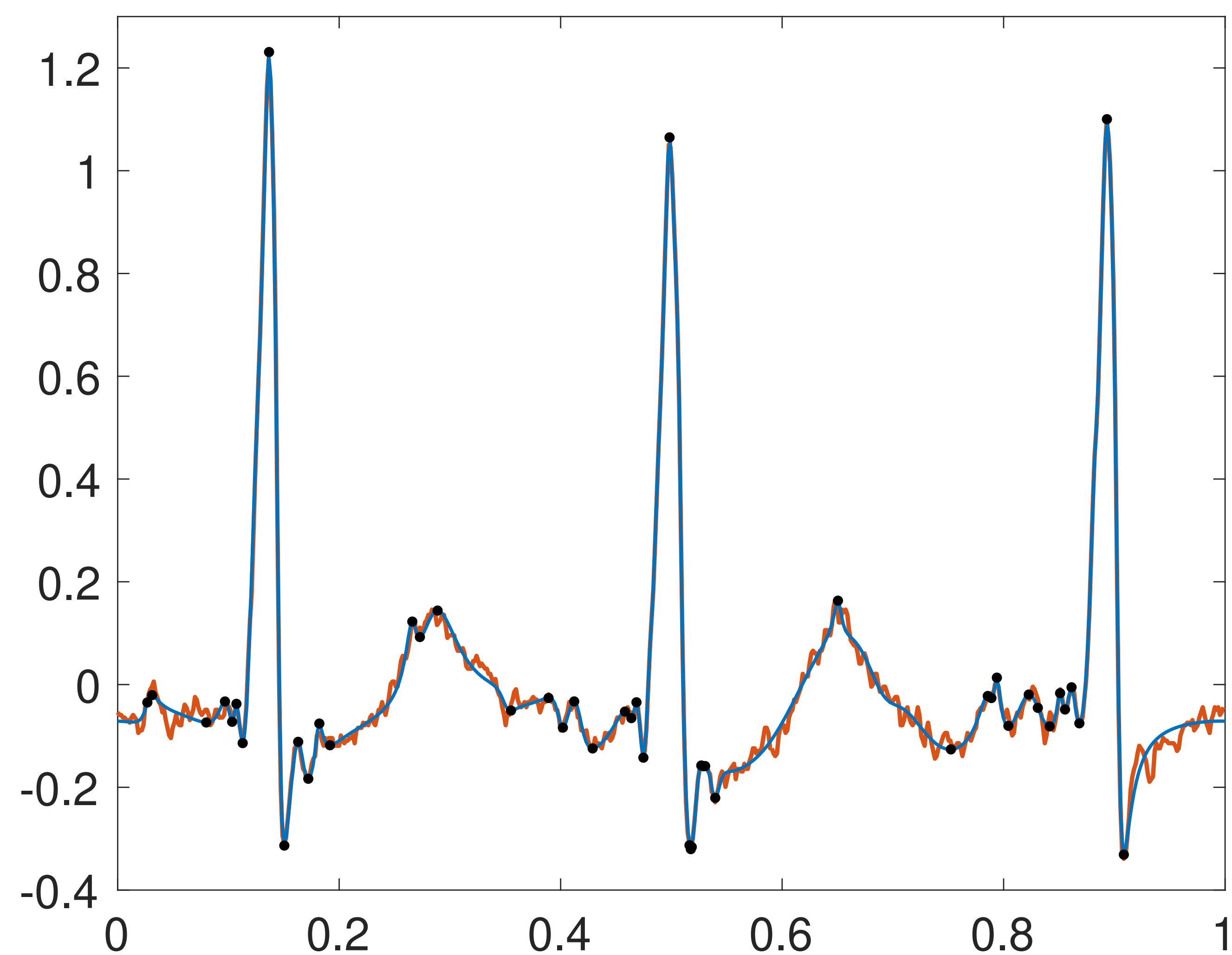
Problem: Noisy data, limited spatial resolution...

How can I construct a barycentric representation of $r_m \approx f$?

(Fourier space)



(time domain)



s_m → $\mathcal{F}^{-1}(s_m)$ → r_m

Thank you!

REfit for data-driven rational computing:
(open-source package for MATLAB)

My website:
heatherw3521.github.io

Other AMAZING rational approximation tools:
AAA in Chebfun:
www.chebfun.org (Nakatsukasa, Trefethen, Sète)

RKfit for rational Krylov subspace approximation:
guettel.com/rktoolbox/index.html (Berljafa, Güttel)

Begin Extra Slides

PronyAAA algorithm

Advantage for postprocessing: rootfinding

If $r_m^{t,\gamma}(\zeta_j) = 0$ and $\mu = e^{2\pi i \zeta_j}$, then $Ey = \mu By$, where

$$E = \left[\begin{array}{ccc|c} e^{2\pi i x_1} & & & i\omega_1 e^{2\pi i x_1} \\ & \ddots & & \vdots \\ & & e^{2\pi i x_{2m}} & i\omega_{2m} e^{2\pi i x_{2m}} \\ \hline f_1 & \cdots & f_{2m} & 0 \end{array}\right], B = \left[\begin{array}{ccc|c} 1 & & & i\omega_1 \\ & \ddots & & \vdots \\ & & 1 & i\omega_{2m} \\ \hline 0 & \cdots & 0 & 0 \end{array}\right].$$

There are $2m - 2$ finite, nonzero eigenvalues.

More advantages

- stable evaluation on $[0, 1)$ (stable interpolation/integration)
  [Higham (2004), Austin and Xu (2017)]
- fast evaluation of derivatives.
  [Berrut, Baltensperger, Mittelmann (2005)]

When are rationals useful?

Rationals appear in the fundamental things we do in numerical linear algebra.

Matrix function evaluation: (Gawlik, 2020), (Nakatsukasa and Gawlik, 2021), (Braess and Hackbusch, 2005, 2009) (Ward, 1977) (Gosea and Güttel, 2020) and many more…

Eigendecompositions/Polar decomposition: (Nakatsukasa and Freund, 2015), (Saad, El-Guide, and Międlar), (Tang and Polizzi, 2014), (Güttel, 2010), (Ruhe, 1994 and many more…

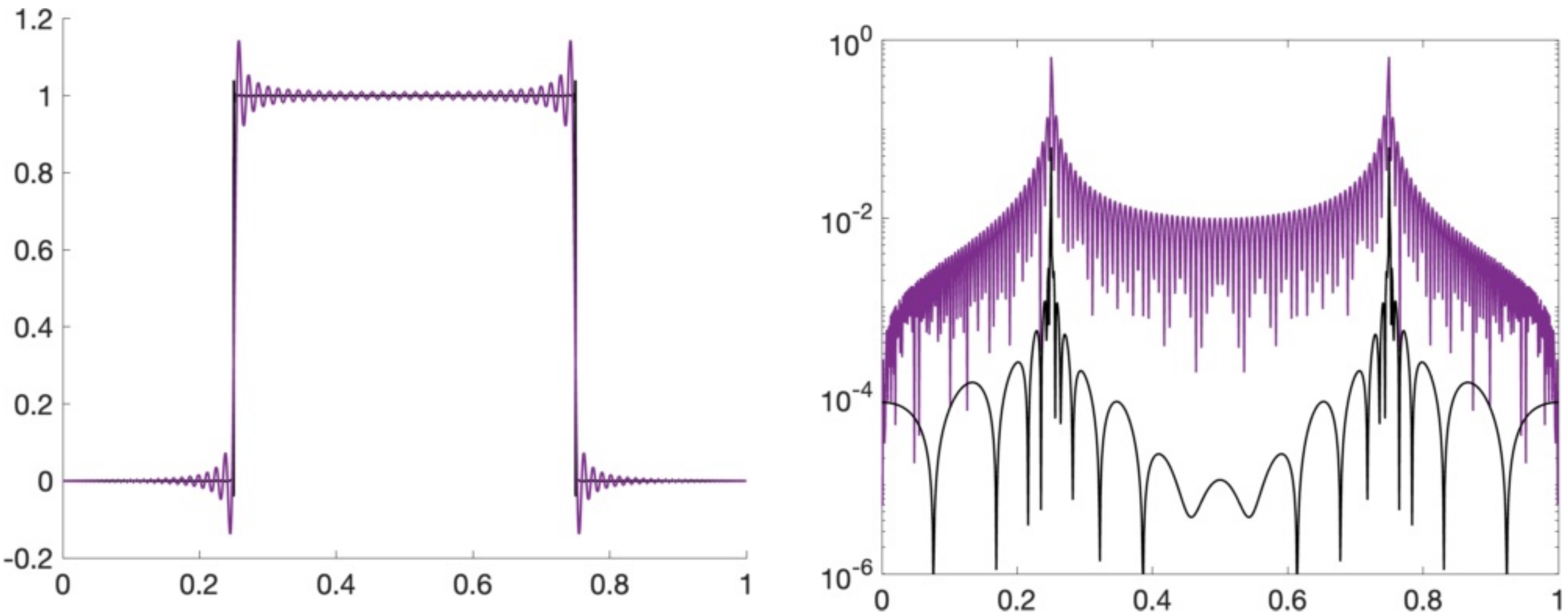
Solving linear systems/matrix equations: (Ruhe, 1994),(Druskin and Simoncini, 2011), (Sabino, 2008), (Kressner, Massei, and Robol, 2019), (Benner, Truhar, and Li, 2009), (W. And Townsend, 2018)many more…

Solving PDEs: (Haut, Beylkin and Monzòn 2015), (Trefethen and Tee, 2006), (Gopal and Trefethen, 2019) , (Haut, Babb, Martinsson, and Wingate, 2016), many more…

Quadrature, conformal mapping, analytic continuation, digital filter design, reduced order modeling… (See Approximation Theory and Practice, Ch. 23)

When are rationals useful?

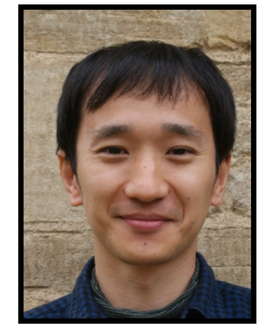
Rational functions have excellent approximation power near singularities



(purple = degree 200 polynomial, black = type $(59, 60)$ rational)

PronyAAA algorithm

Key Idea: greedily build up an interpolant, one point at a time.


(Y. Nakatsukasa)


(L.N. Trefethen)


(O. Sète)

Start with sampling locations $T = \{x_1, \ldots, x_N\}$.

Suppose the nodes are $t = \{t_1, \ldots, t_{2m}\} \subset T$

Determining the barycentric weights:

$$r_m^{t,\gamma}(x) = \frac{n_{m-1}(x)}{d_m(x)}$$

$$r_m^{t,\gamma}(x) d_m(x) = n_{m-1}(x)$$

$$\min_{\gamma \in \mathbb{C}} \sum_{x_j \in T \setminus t} \left(f(x_j) d_m(x_j) - n_{m-1}(x_j) \right)^2,$$

$$\text{s.t.} \sum_{j=1}^{2m} f(t_j)\gamma_j = 0, \quad \|\gamma\|_2 = 1.$$

Choosing the next interpolating point:

$$t_{2m+1} = \text{argmax}_{x \in T \setminus t} |r_m^{t,\gamma}(x_j) - f(x_j)|$$

[Nakatsukasa, Trefethen, & Sète (2018), Antoulas & Anderson (1986), Berrut (2005), Badoo(2021)]

Exponential sums to barycentric interpolants

$$R_m(k) = \sum_{j=1}^{m} \omega_j e^{\lambda_j k}$$

$$r_m(x) = \mathcal{F}^{-1}(R_m)(x)$$

$$r_m^{t,\gamma}(x) = \frac{\sum_{j=1}^{2m} \gamma_j f_j \cot\left(\pi(x - t_j)\right)}{\sum_{j=1}^{2m} \gamma_j \cot\left(\pi(x - t_j)\right)}$$

Theorem: (Damle, Townsend, W.) The type $(m-1, m)$ trigonometric rational $r_m = \mathcal{F}^{-1}(r_m)$ can be exactly recovered by a barycentric interpolant $r_m^{t,\gamma}$ for any set of distinct interpolating points $t = \{t_1, \ldots, t_{2m}\} \subset [0, 1)$.

Exact recovery is an ill-conditioned problem: The choice of t matters greatly.

Idea 1: Apply $2m$ steps of PronyAAA. (chooses points via greedy residual minimization)

Can be numerically unstable. Loss of accuracy/poles occurring on the interval!

Exponential sums to barycentric interpolants

$$R_m(k) = \sum_{j=1}^{m} \omega_j e^{\lambda_j k}$$

$$r_m(x) = \mathcal{F}^{-1}(R_m)(x)$$

$$r_m^{t,\gamma}(x) = \frac{\sum_{j=1}^{2m} \gamma_j f_j \cot\left(\pi(x - t_j)\right)}{\sum_{j=1}^{2m} \gamma_j \cot\left(\pi(x - t_j)\right)}$$

Theorem: (Damle, Townsend, W.) The type $(m-1, m)$ trigonometric rational $r_m = \mathcal{F}^{-1}(r_m)$ can be exactly recovered by a barycentric interpolant $r_m^{t,\gamma}$ for any set of distinct interpolating points $t = \{t_1, \ldots, t_{2m}\} \subset [0, 1)$.

Exact recovery is an ill-conditioned problem: The choice of t matters greatly.

Idea 1: Apply $2m$ steps of PronyAAA. (chooses points via greedy residual minimization)

Can be numerically unstable. Loss of accuracy/poles occurring on the interval!

Idea 2: Be greedy about numerical stability instead!

(A new pivoting strategy for AAA based on column-pivoted QR + stabilization)

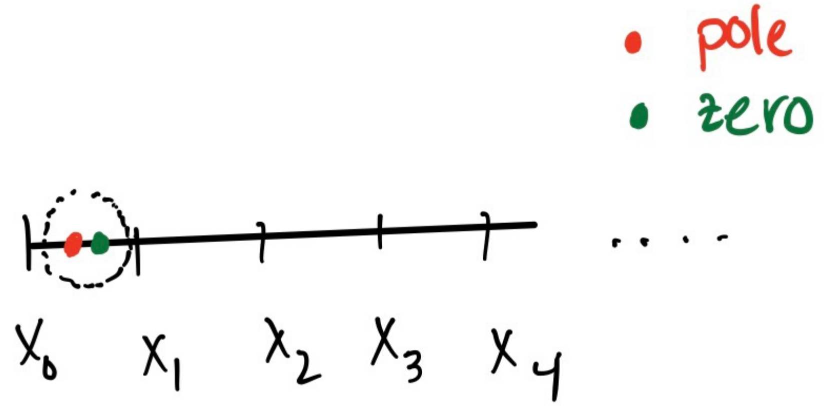
PronyAAA algorithm

$$r_m^{t,\gamma}(x) = \frac{\sum_{j=1}^{2m} \gamma_j f_j \cot\left(\pi(x - t_j)\right)}{\sum_{j=1}^{2m} \gamma_j \cot\left(\pi(x - t_j)\right)}$$

Where are the poles?

Nothing explicitly enforces that poles are located off $[0, 1)$.

Benign spurious poles: Can be eliminated easily with AAA cleanup routine.



Pernicious spurious poles: cannot be eliminated without strongly impacting accuracy.

Pernicious spurious poles appear when...

1. Data is not modeled well by type $(m-1, m)$ trigonometric rationals.

2. We demand too much accuracy (e.g., machine precision).

Prony's method

Given $(c_0, c_1, \ldots, c_{2M+1})$, recover

$$s_M(\ell) = \sum_{j=1}^{M} w_j e^{-\lambda_j \ell}, \quad \text{where } c_\ell = s(\ell) \text{ for } \ell \geq 0.$$

(Gaspard de Prony)

How can we find each λ_j?

Set $p(z) = \prod_{j=1}^{M}(z - \gamma_j)$, $\quad \gamma_j = e^{-\lambda_j}$. $\quad p(z) = \sum_{k=0}^{M} p_k z^k$ (Prony's polynomial)

If we can determine $p = (p_0, \ldots, p_M)$, then this becomes a rootfinding problem.

$$\text{For } \ell \geq 0, \quad \sum_{k=0}^{M} p_k s(k+\ell) = \sum_{j=1}^{M} w_j \sum_{k=0}^{M} p_k \gamma_j^{(k+\ell)} = \sum_{j=1}^{M} w_j \gamma_j^{\ell} \sum_{k=0}^{M} p_k \gamma_j^{k} = 0$$

$$\text{If} \quad H = \begin{pmatrix} c_0 & c_1 & \cdots & c_M \\ c_1 & c_2 & \cdots & c_{M+1} \\ \vdots & & & \vdots \\ c_M & c_{M+1} & \cdots & c_{2M} \end{pmatrix}, \quad \text{then } Hp = 0.$$

[Belykin & Monzon (2005, 2009) , Peter & Plonka (2013), Potts & Tasche (2013)]

barycentric to exponential sum

$$R_m(k) = \sum_{j=1}^{m} \omega_j e^{\lambda_j k}$$

$\longleftarrow \mathcal{F}(r_m^{t,\gamma})$

$$r_m^{t,\gamma}(x) = \frac{\sum_{j=1}^{2m} \gamma_j f_j \cot(\pi(x - t_j))}{\sum_{j=1}^{2m} \gamma_j \cot(\pi(x - t_j))}$$

Key Idea: Approximate λ_j, and use the "Prony principle".

- Find the poles of $r_m^{t,\gamma} \to$ approximate each λ_j.
- Evaluate $r_m^{t,\gamma}$ at $2N+1$ points $\to$ N Fourier coefficients.
- Solve $V\omega = s$, where s is an $\mathcal{O}(m)$ sample of coeffs.

[Miller (1970), Moitra (2016), Transtrum, Matcha and Sethna (2010)]

exponential sum to barycentric: CPQR-selected interpolation points

$$R_m(k) = \sum_{j=1}^{m} \omega_j e^{\lambda_j k}$$

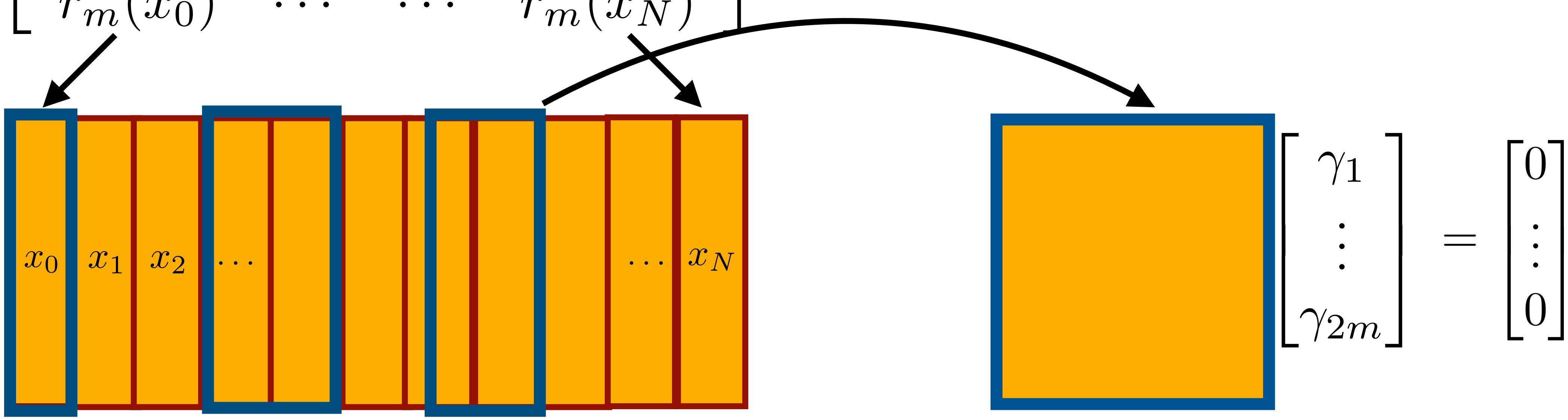
$$r_m(x) = \mathcal{F}^{-1}(R_m)(x)$$

$$r_m^{t,\gamma}(x) = \frac{\sum_{j=1}^{2m} \gamma_j f_j \cot\left(\pi(x - t_j)\right)}{\sum_{j=1}^{2m} \gamma_j \cot\left(\pi(x - t_j)\right)}$$

Observation: $d_m(\eta_j) = 0$ when $\eta_j = 2\pi i \lambda_j$.

Let $T = \{x_0, x_1, \ldots, x_N\}$ be sample locations. Let $\{\eta_1, \eta_2, \ldots, \eta_{2m}\}$ be the poles of r_m.

$$\left[\begin{array}{cccc} \ell_{1,0} & \cdots & \cdots & \ell_{1,N} \\ \vdots & & & \vdots \\ \ell_{2m,0} & \cdots & \cdots & \ell_{2m,N} \\ \hline r_m(x_0) & \cdots & \cdots & r_m(x_N) \end{array}\right], \quad \ell_{j,k} = \cot(\pi\eta_j - \pi x_k)$$

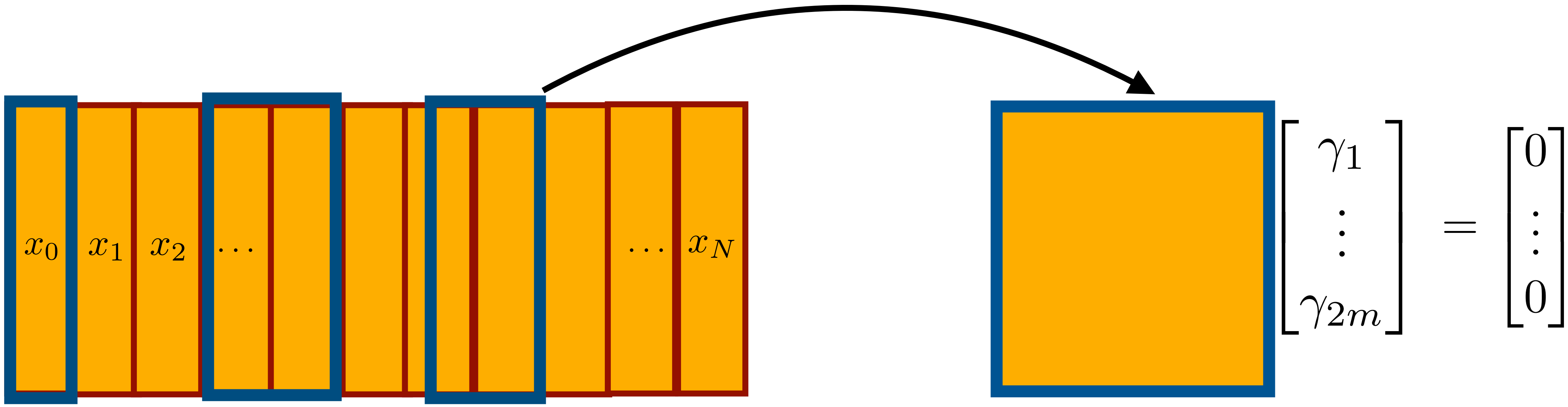


exponential sum to barycentric: CPQR-selected interpolation points

$$R_m(k) = \sum_{j=1}^{m} \omega_j e^{\lambda_j k}$$

$$r_m(x) = \mathcal{F}^{-1}(R_m)(x)$$

$$r_m^{t,\gamma}(x) = \frac{\sum_{j=1}^{2m} \gamma_j f_j \cot\left(\pi(x - t_j)\right)}{\sum_{j=1}^{2m} \gamma_j \cot\left(\pi(x - t_j)\right)}$$



Greedily select columns to form the most well-conditioned submatrix.

Column-pivoted QR (CPQR) [Golub & Busigner (1965), Chandrasekaran & Ipsen (1994), Gu & Eisenstat (1996)]

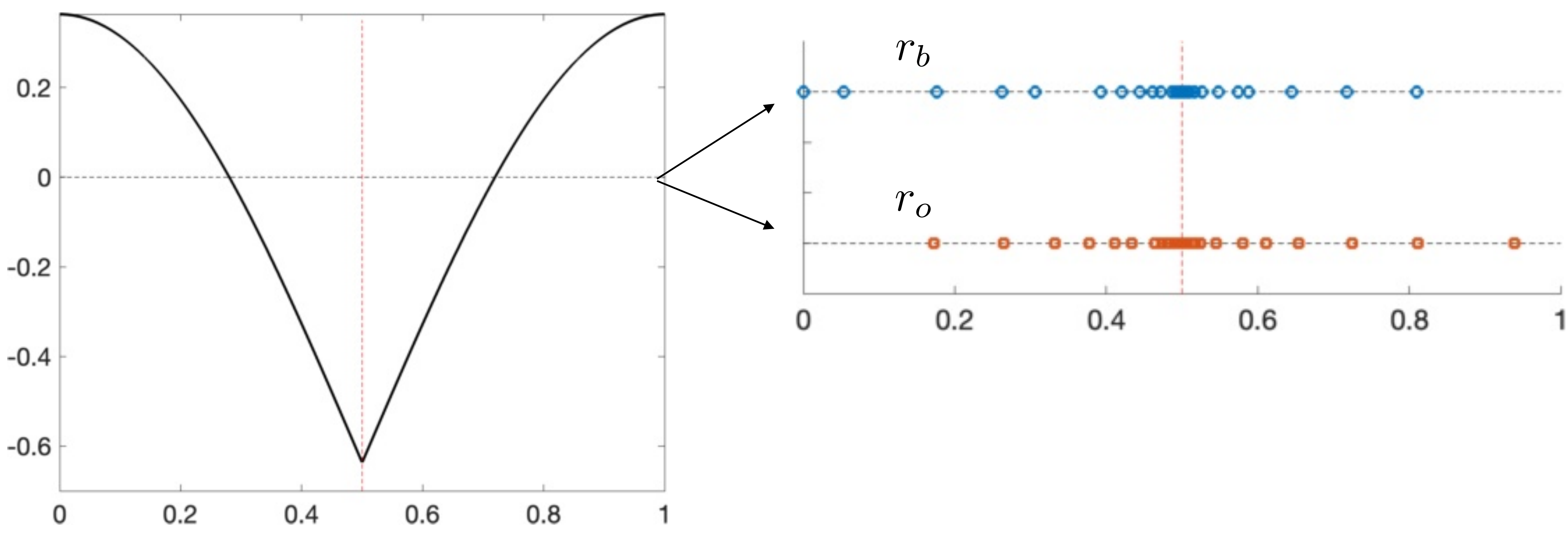
1. CPQR to choose candidates for barycentric nodes.
2. Regularization procedure: Constrained optimization to subselect from candidate nodes + find weights $\gamma = \{\gamma_1, \dots, \gamma_{2m}\}$.

AAA-selected and CPQR-selected interpolation points

Example: $$f(x) = |\sin(\pi(x - 1/2))| - \pi/2$$

r_b = apply PronyAAA to data directly.

r_o = apply Prony's method to Fourier coefficients to get R_o, then compute $\mathcal{F}^{-1}(R_o) = r_o$ using CPQR-selected barycentric nodes.

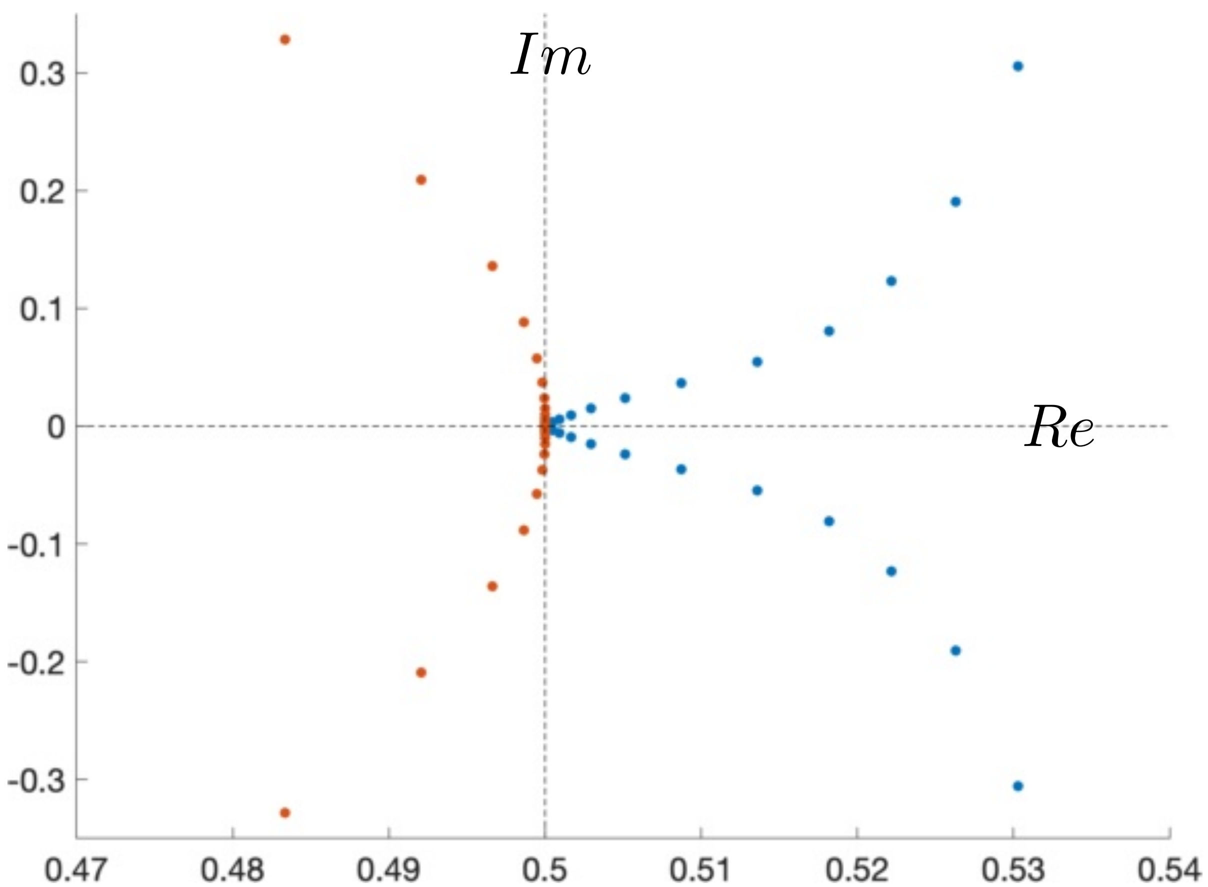


AAA-selected and CPQR-selected poles

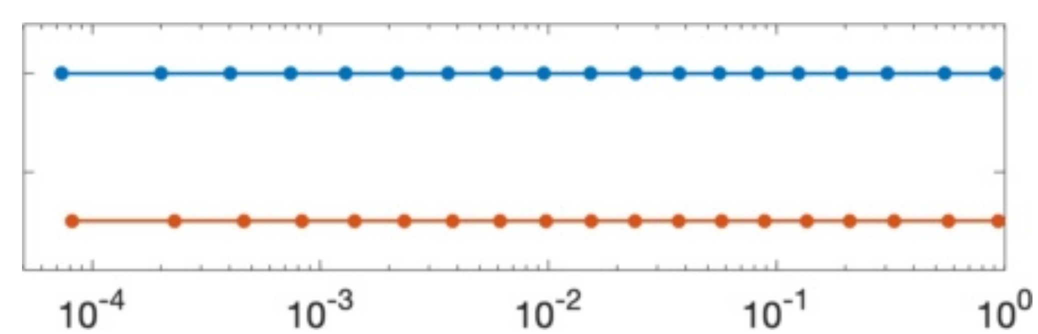
$$f(x) = |\sin(\pi(x - 1/2))| - \pi/2$$

r_b = apply PronyAAA to data directly.

r_o = apply Prony's method to Fourier coefficients to get R_o, then compute $\mathcal{F}^{-1}(R_o) = r_o$ using CPQR-selected barycentric nodes.



Very different pole configurations, similar clustering properties.



distances from pole to singularity

[Nakatsukasa , Weideman & Trefethen (2021)]