Learning from the square root function: rational approximation methods in computational mathematics

Heather Wilber
University of Texas at Austin

ODEN INSTITUTE
FOR COMPUTATIONAL ENGINEERING & SCIENCES

NSF

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.

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

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…

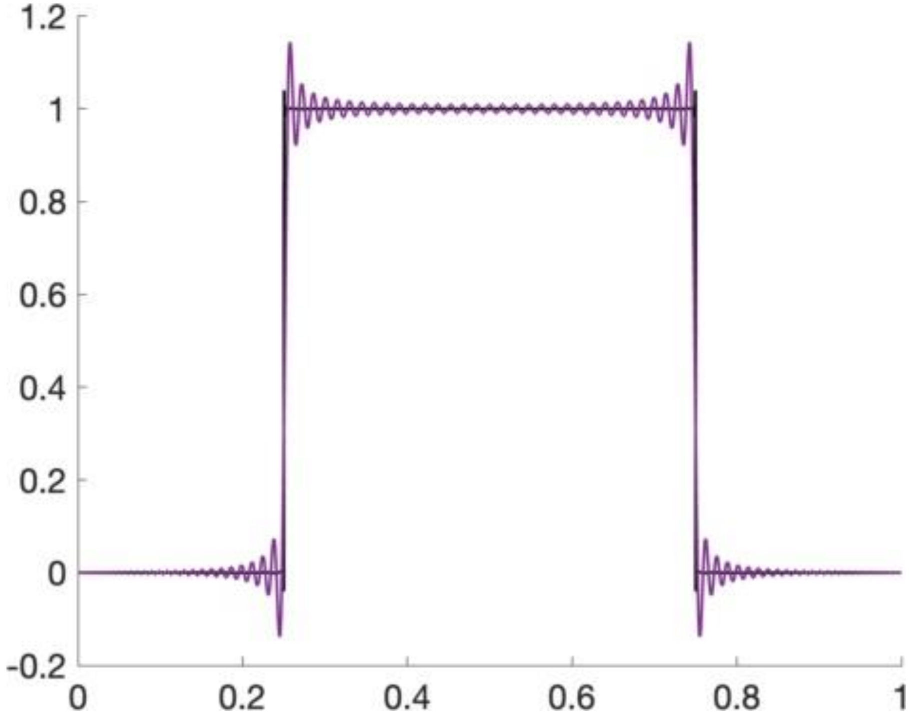
Eigendecompositions/Polar decomposition: (Nakatsukasa and Freund, 2015), (Saad, El-Guide, and Międlar), (Tang and Polizzi, 2014), (Güttel, 2010), (Ruhe, 1994 and many more…

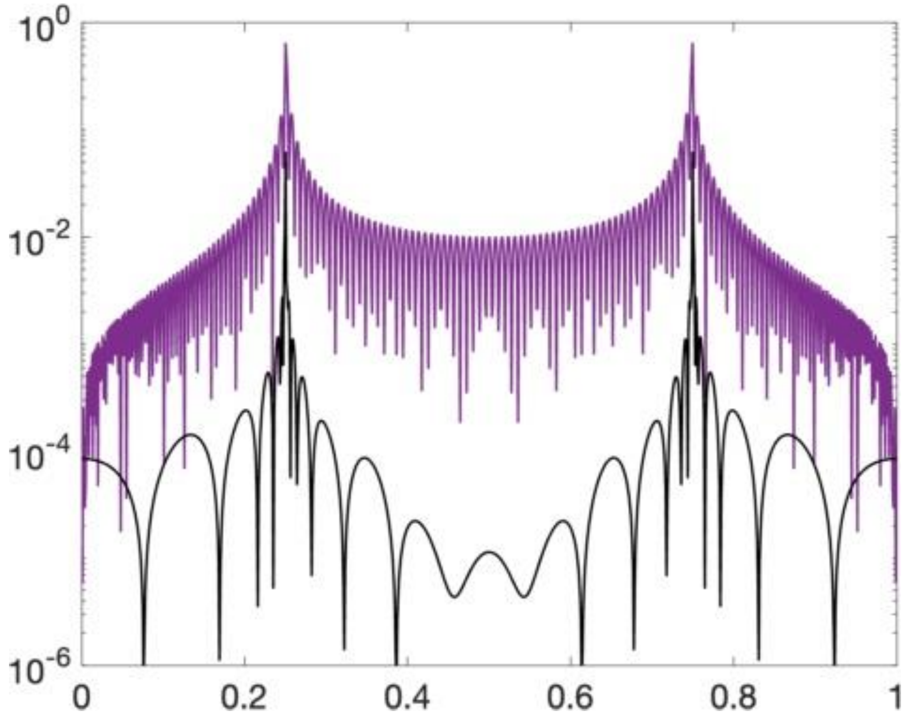
Solving linear systems/matrix equations: (Ruhe, 1994),(Druskin and Simoncini, 2011), (Sabino, 2008), (Kressner, Massei, and Robol, 2019), (Benner, Truhar, and Li, 2009), (W. And Townsend, 2018) many more…

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

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

Rational functions have excellent approximation power near singularities





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

...and so much more!

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.

Approximate $f(x) = \sqrt{x}$ on the interval $x \in [\beta, 1]$, $0 \leq \beta$.

Find $r_k(x)$ to minimize $\max_{x \in [\beta,1]} |r_k(x) - f(x)|$

The square root approximation problem gives us insight into many problems that involve computing with rational functions…

3 big ideas: many applications

- signal processing (event detection, filtering, denoising, reconstruction)
- numerical linear algebra (NLEVP, functions of matrices, low rank approximation, …)
- solving of PDEs
- quadrature, resolvent methods

(There are many more ideas and applications we won’t be talking about today!)

Approximate $f(x) = \sqrt{x}$ on the interval $x \in [\beta, 1]$, $0 \leq \beta$.

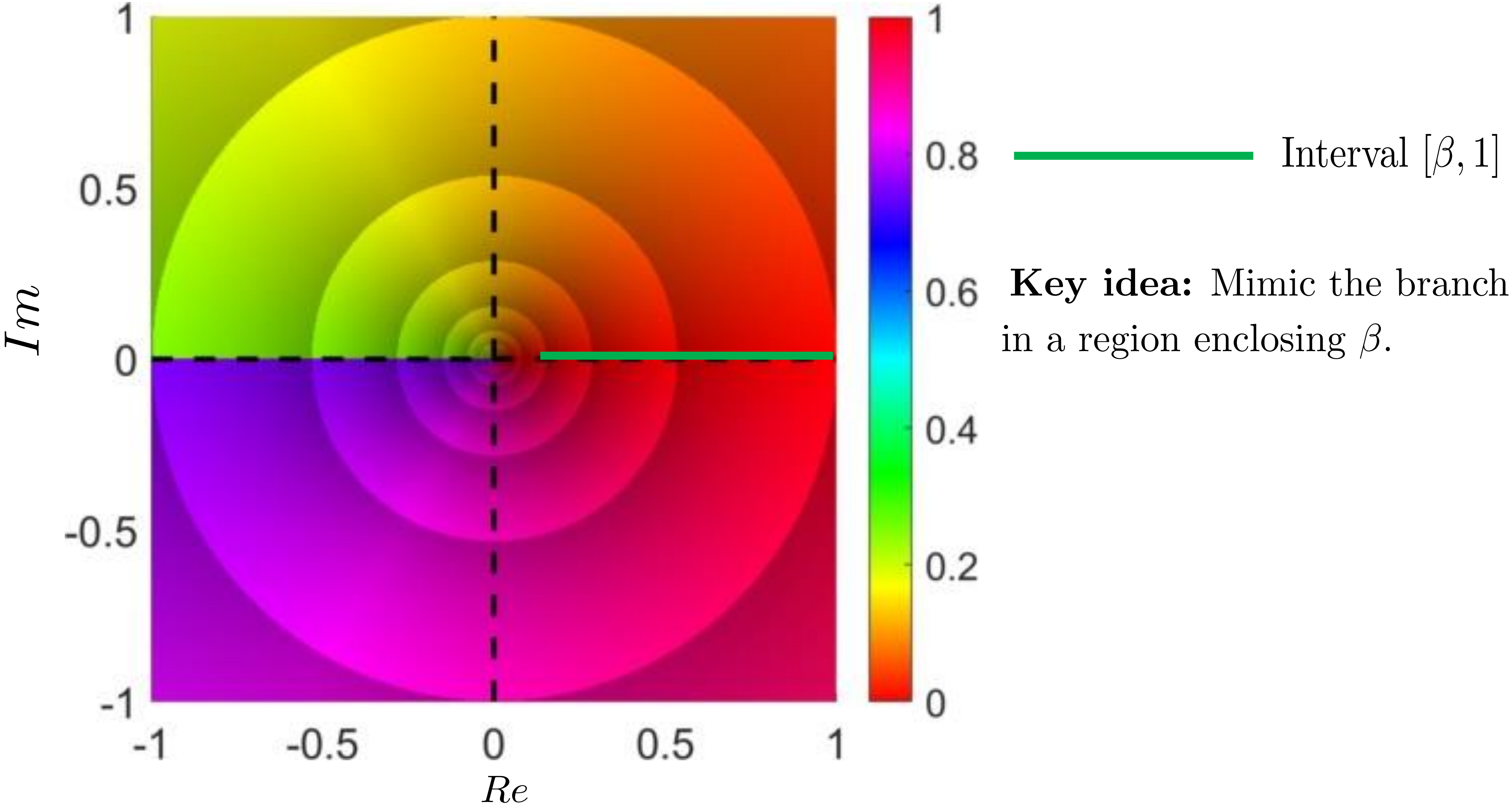
Find $r_k(x)$ to minimize $\max_{x \in [\beta,1]} |r_k(x) - f(x)|$

What is the main challenge?

And how can rationals help?

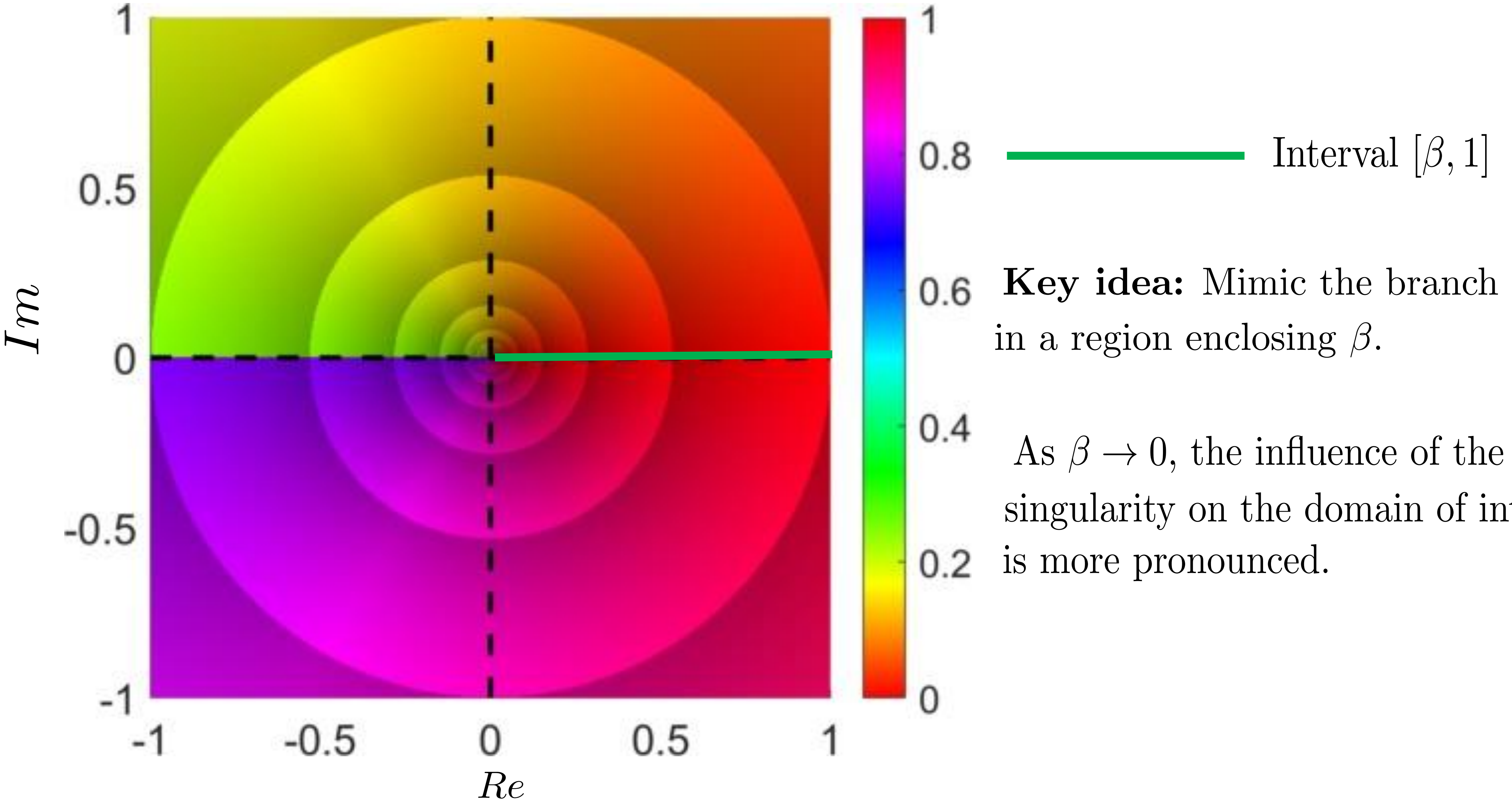
To see the issue, let’s venture into the complex plane...

A phase plot of the square root function



Key idea: Mimic the branch cut in a region enclosing β.

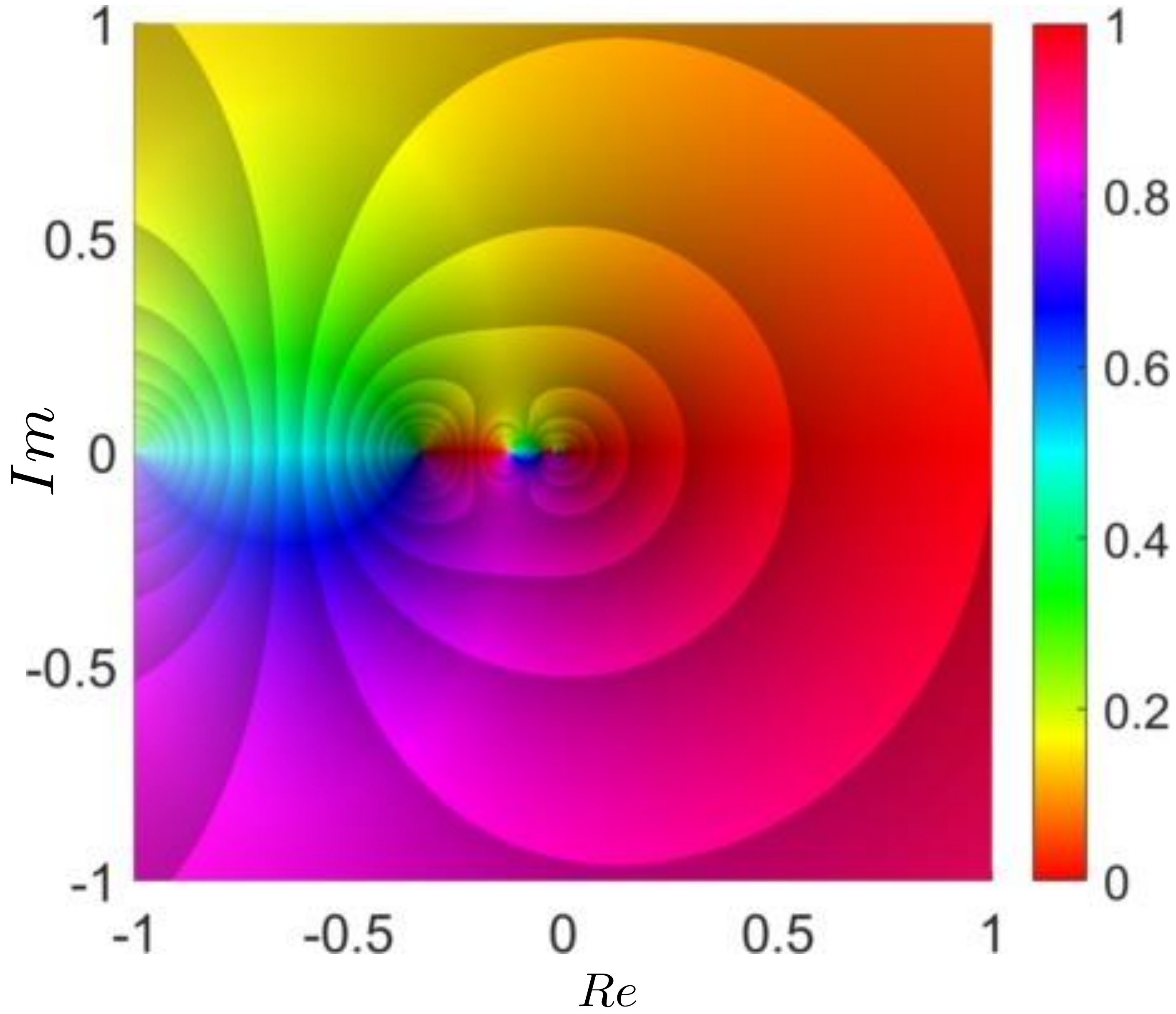
A phase plot of the square root function



Key idea: Mimic the branch cut in a region enclosing β.

As $\beta \to 0$, the influence of the singularity on the domain of interest is more pronounced.

A rational approximation to the square root



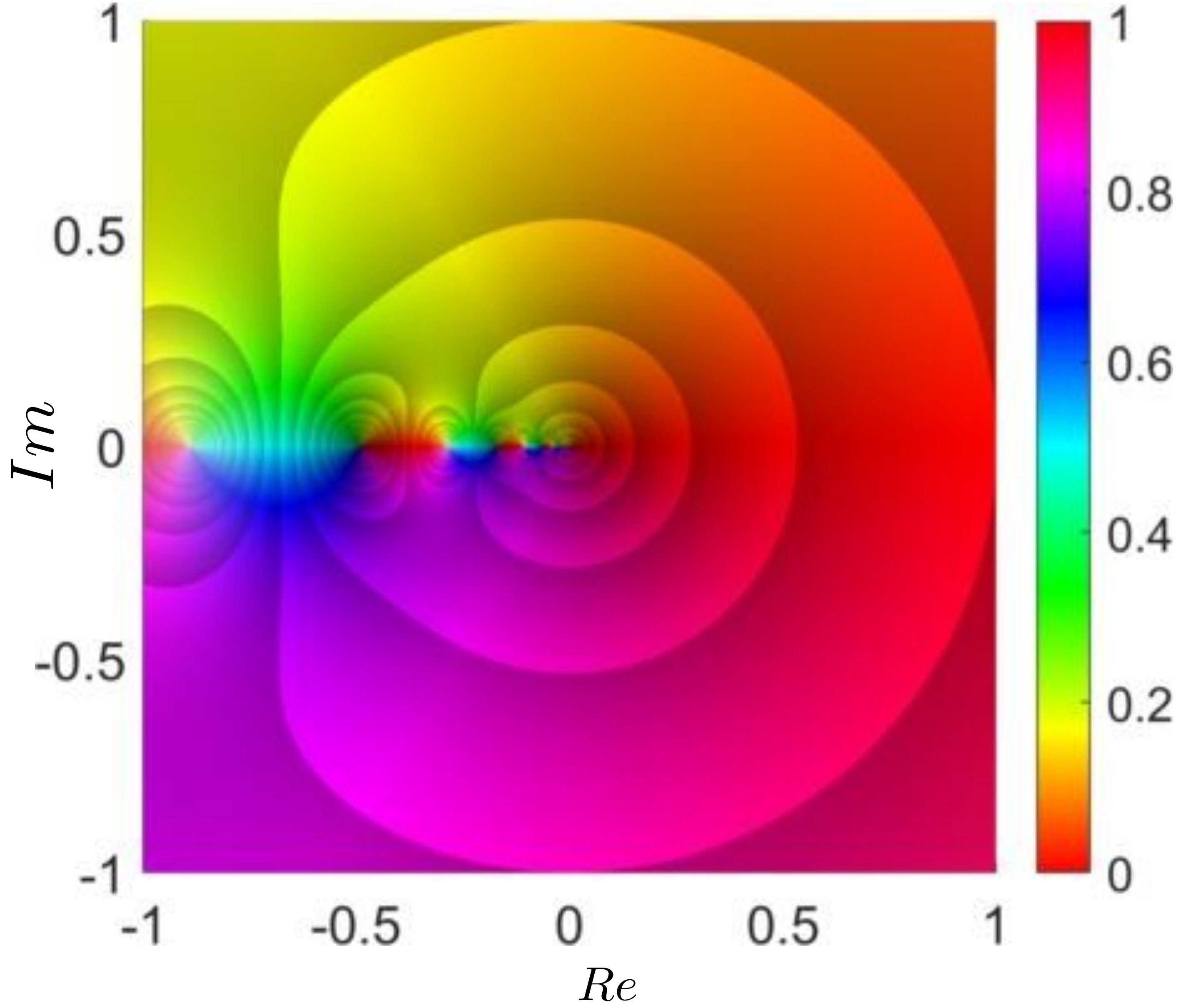
$\beta = .001$

5 poles

$$\max_{x\in[\beta,1]} |r_5(x) - \sqrt{x}|$$

$$\approx 5 \times 10^{-5}$$

A rational approximation to the square root



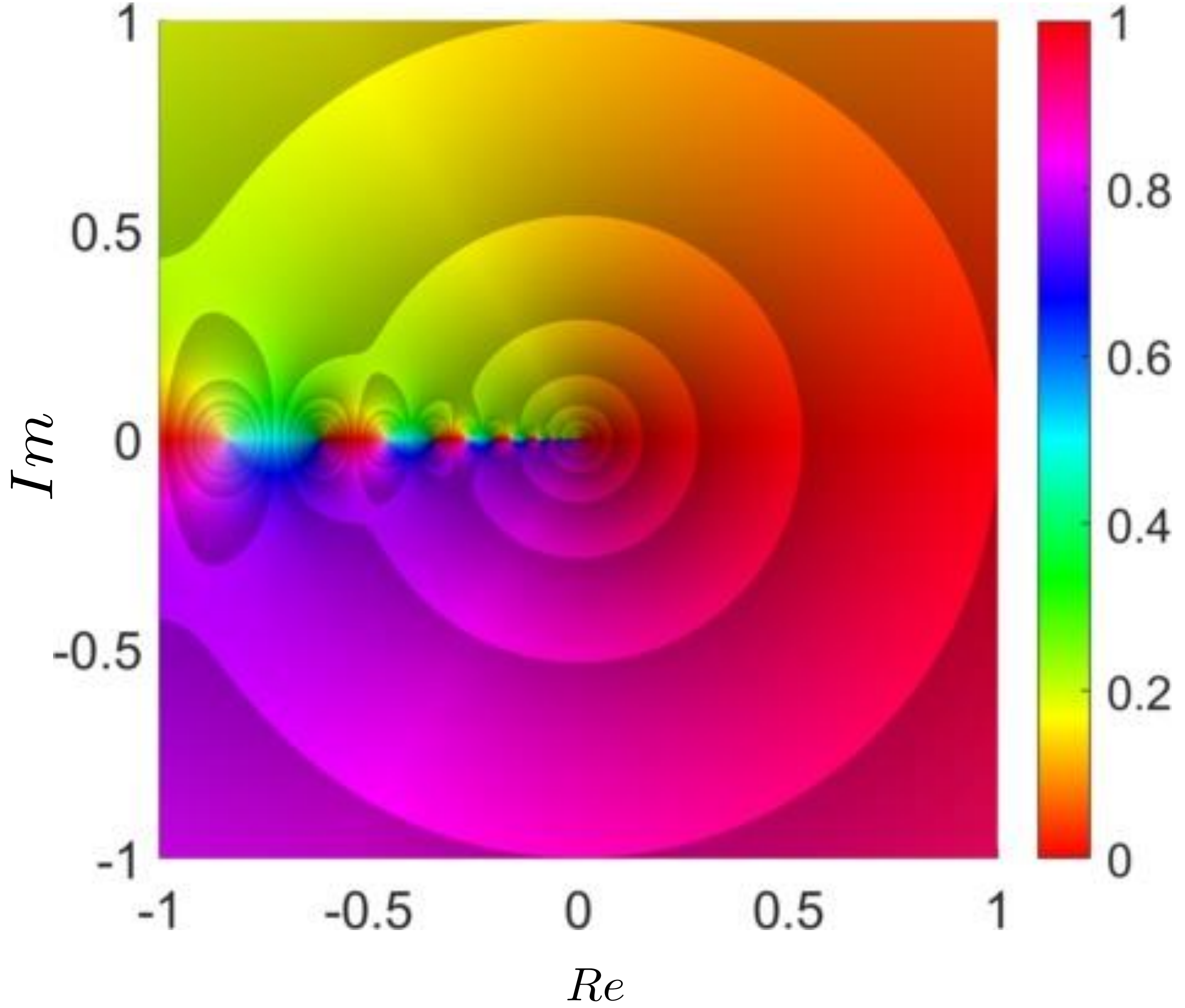
$\beta = .001$

10 poles

$$\max_{x \in [\beta,1]} |r_{10}(x) - \sqrt{x}|$$

$$\approx 2 \times 10^{-9}$$

A rational approximation to the square root



$\beta = .001$

20 poles

$$\max_{x\in[\beta,1]} |r_{20}(x) - \sqrt{x}|$$

$$\approx 4 \times 10^{-15}$$

Big idea 1: Cluster poles near singularities!

Let $r_k(x)$ be a rational function with $k-1$ zeros and k simple poles. Then,

$$r_k(x) = \sum_{j=1}^{k} \frac{r_j}{x - p_j}, \quad \text{where } \{p_j\}_{j=1}^k \text{ are the poles of } r_k, \text{ and } r_j = \text{res}(r_k, p_j).$$

Suppose we sample f at $\{x_1, \ldots, x_N\}$.

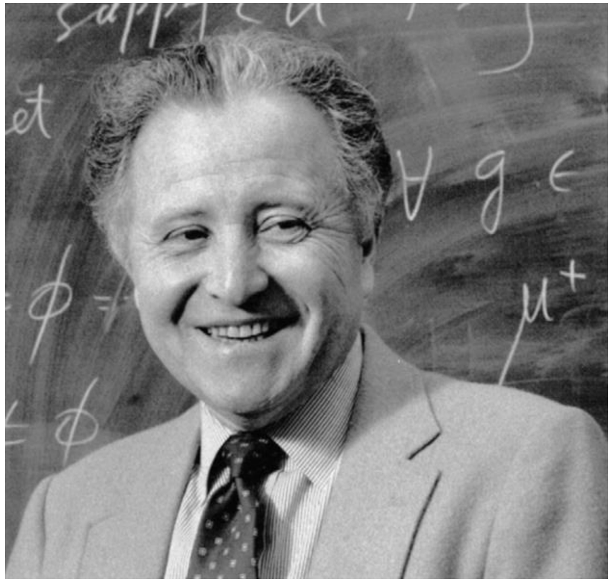
$$r_k(x_i) \approx f(x_i) \rightarrow \begin{bmatrix} \frac{1}{x_1 - p_1} & \cdots & \frac{1}{x_1 - p_k} \\ \frac{1}{x_2 - p_1} & \cdots & \frac{1}{x_2 - p_k} \\ \vdots & & \vdots \\ \vdots & & \vdots \\ \frac{1}{x_N - p_1} & \cdots & \frac{1}{x_N - p_k} \end{bmatrix} \begin{bmatrix} r_1 \\ \vdots \\ r_k \end{bmatrix} \approx \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ \vdots \\ f(x_N) \end{bmatrix}$$

Big idea 1: Cluster poles near singularities!

$$\begin{bmatrix} \frac{1}{x_1-p_1} & \cdots & \frac{1}{x_1-p_k} \\ \frac{1}{x_2-p_1} & \cdots & \frac{1}{x_2-p_k} \\ \vdots & & \vdots \\ \vdots & & \vdots \\ \frac{1}{x_N-p_1} & \cdots & \frac{1}{x_N-p_k} \end{bmatrix} \begin{bmatrix} r_1 \\ \vdots \\ r_k \end{bmatrix} \approx \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ \vdots \\ f(x_N) \end{bmatrix}$$

If we fix $\{p_1, \ldots, p_k\}$, this is a linear least squares problem!

Big idea 1: Cluster poles near singularities!



1964:

Root-exponential convergence rates

D.J. Newman proves that there is a sequence of rationals $\{r_1, r_2, \ldots\}$, where r_n has n poles and n zeros, that converges to $|x|$ on $[-1, 1]$ at the rate $\mathcal{O}(e^{-\sqrt{n}})$.

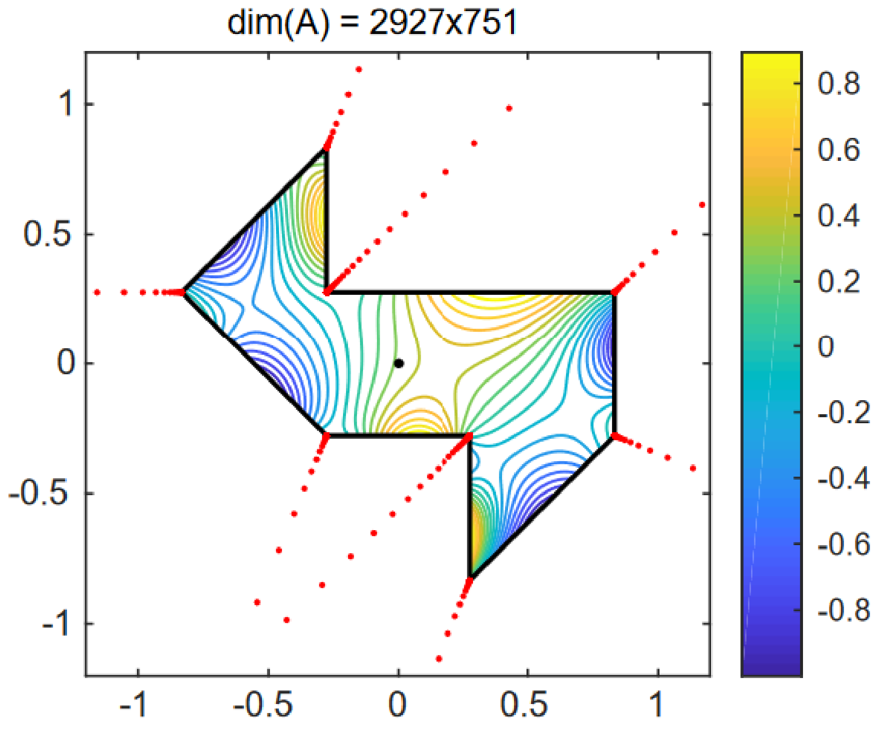
Since $|x| = \sqrt{x^2}$, for $x^2 \in [0, 1]$, this argument also implies that such a sequence exists for approximating $\sqrt{x}$ on $[0, 1]$.

Big idea 1: Cluster poles near singularities!

Gopal and Trefethen (2019): Lightning Laplace (Helmholtz, Biharmonic) solver.

(L.N. Trefethen)

(A. Gopal)



(Fairweather & Karageorghis, 1998), (Trefethen, 2020) (Trefethen, Nakatsukasa & Weideman,2021)

Big idea 1: Cluster poles near singularities!

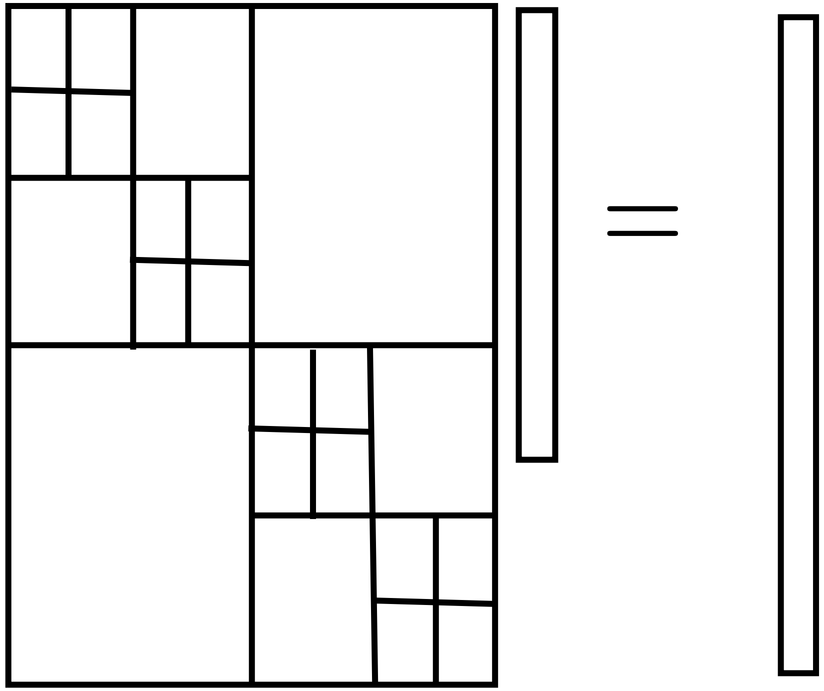
Hierarchical least squares and minimum norm solvers. W., Epperly (2022)

(E. Epperly)

Solves $Hx = b$ in $\mathcal{O}(m+n)k$, where $H \in \mathbb{C}^{m\times n}$ is a hierarchical semi-separable with off-diagonal blocks of rank $\leq k$.

Future work: generalized $\mathcal{H}^2$ solvers + specialized compression strategies.

Many additional applications!



(Xi, Xia, Cauley, Balakrishnan, 2014), (Chandrasekaran, Gu, Pals, 2006)

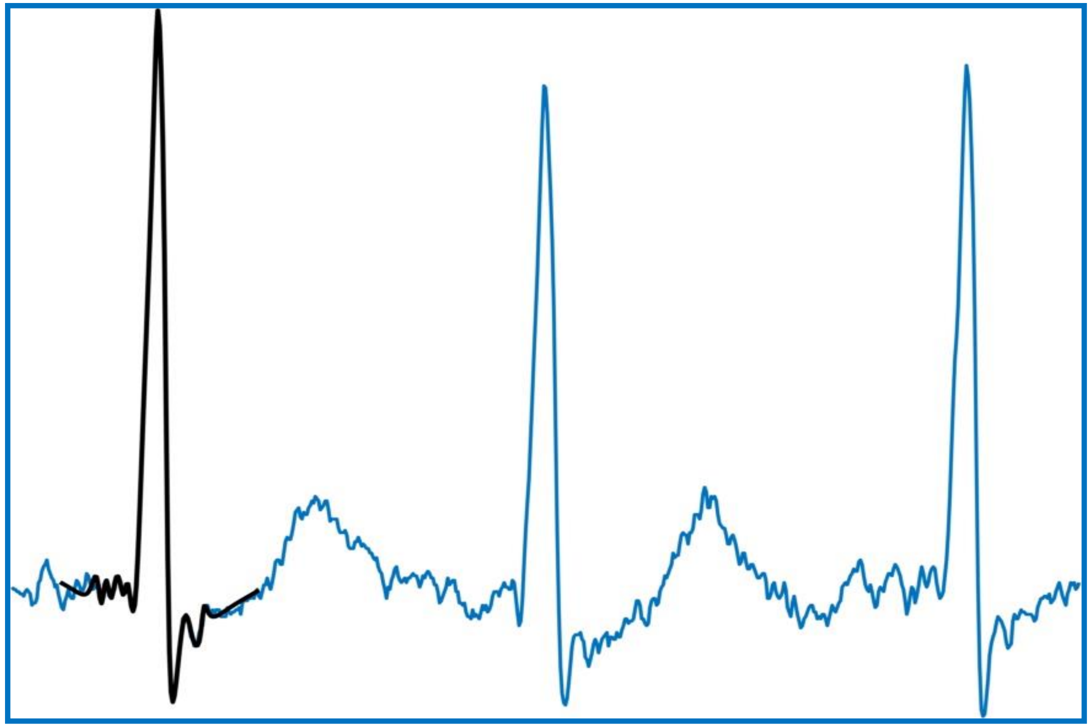
What happens if I don't know where the singularities are?

In many applications, we don't know where the singularities are.

The goal might involve detecting singularity locations/occurrences.

(A. Townsend)

(A. Damle)



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

Big idea 2: free-pole interpolation methods

$$r_n(x) = \frac{p_n(x)}{q_n(x)} \approx f(x) \qquad \Longrightarrow$$

for a collection of sample points X, minimize $\|f(X)q_n(X) - p_n(X)\|_2$

One option:

Barycentric rational interpolants

$+$

Greedy algorithm to pick interpolation points

Data-driven process

AAA, trigAAA, PronyAAA

(Antoulas & Anderson, 1986) (Nakatsukasa, Trefethen & Sete, 2018) (Baddoo, 2021), (Wilber, Damle & Townsend, 2022) (Related ideas from: Gutenknecht, Beylkin & Monzòn, Plonka, many more..)

Big idea 2: free-pole interpolation methods

$$r_n(x) = \frac{p_n(x)}{q_n(x)} \approx f(x) \qquad \Longrightarrow$$

for a collection of sample points X, minimize $\|f(X)q_n(X) - p_n(X)\|_2$

Another option:

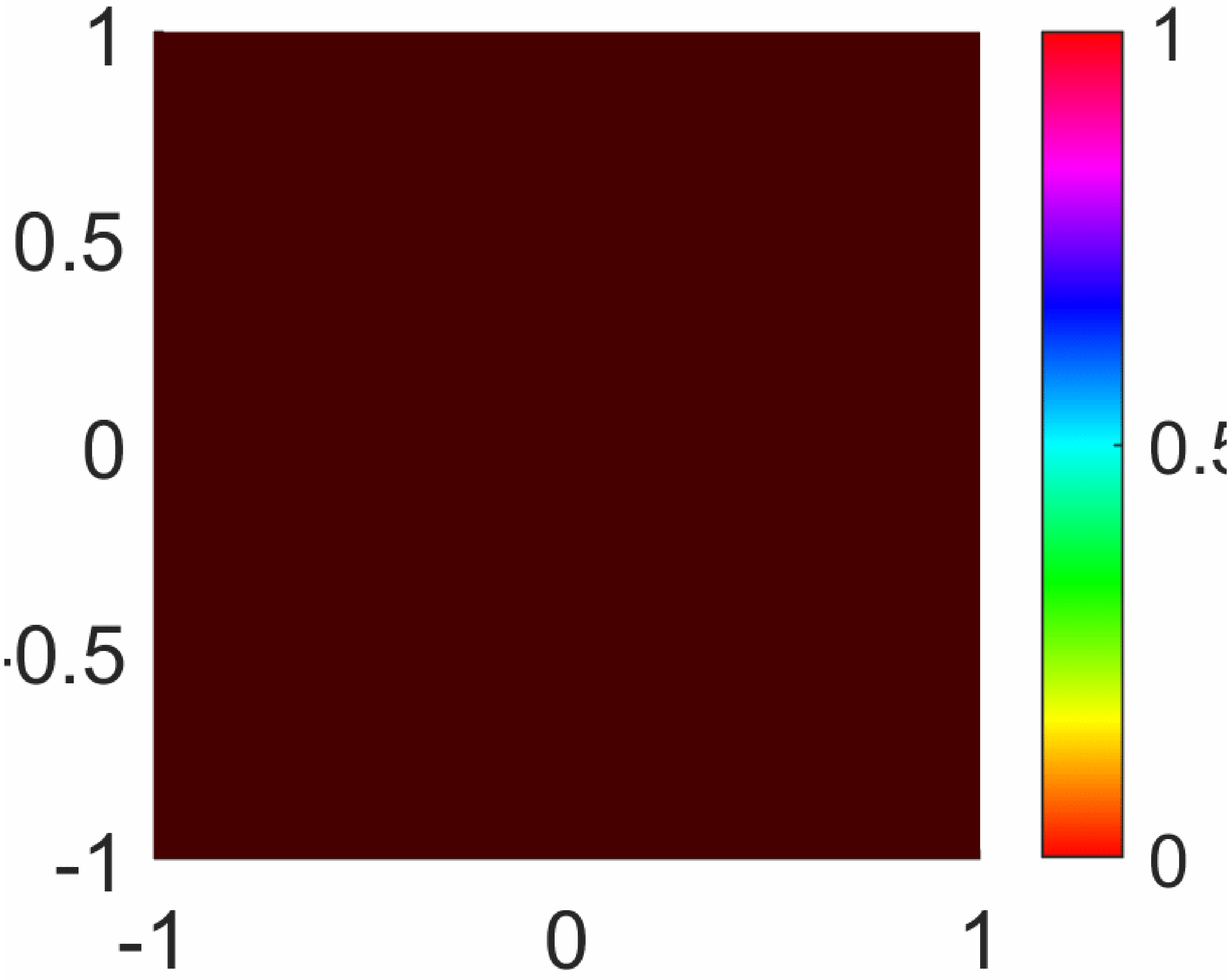
Prony's method + Fourier inversion

Data-driven process in Fourier space

REfit, Beylkin & Monzòn, Plonka

(Antoulas & Anderson, 1986) (Nakatsukasa, Trefethen & Sete, 2018) (Baddoo, 2021), (Wilber, Damle & Townsend, 2022) (Related ideas from: Gutenknecht, Beylkin & Monzòn, Plonka, many more..)

Big idea 2: free-pole interpolation methods

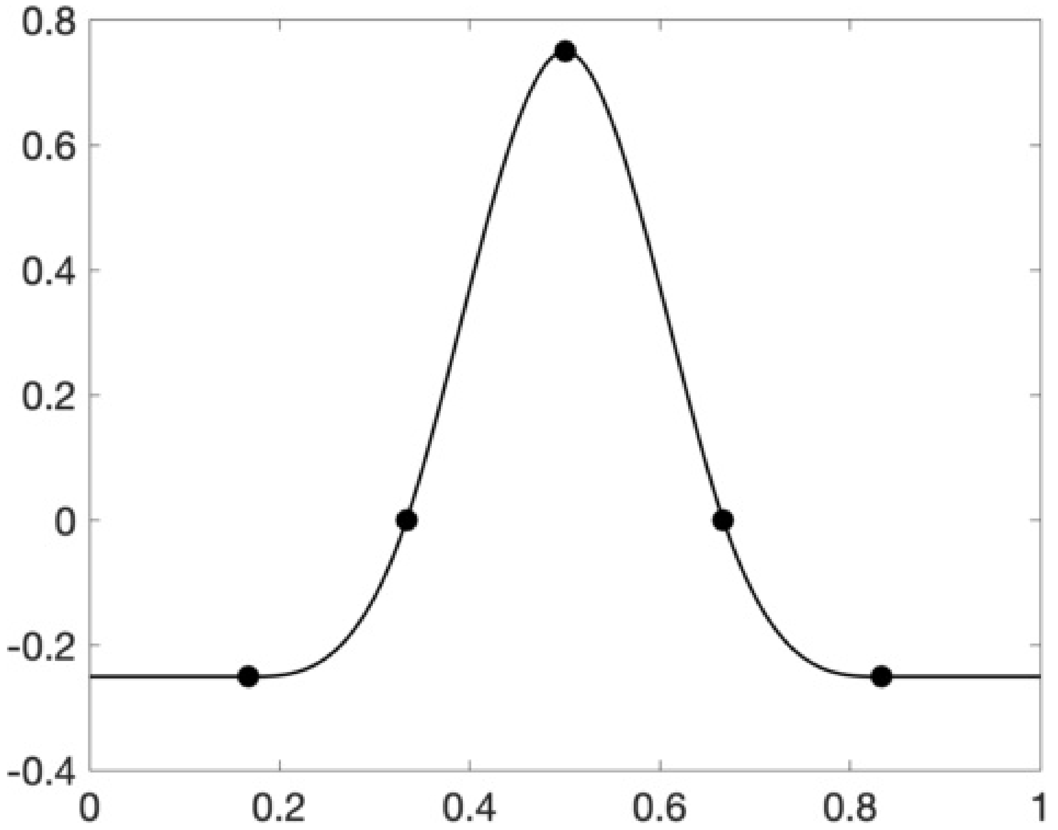


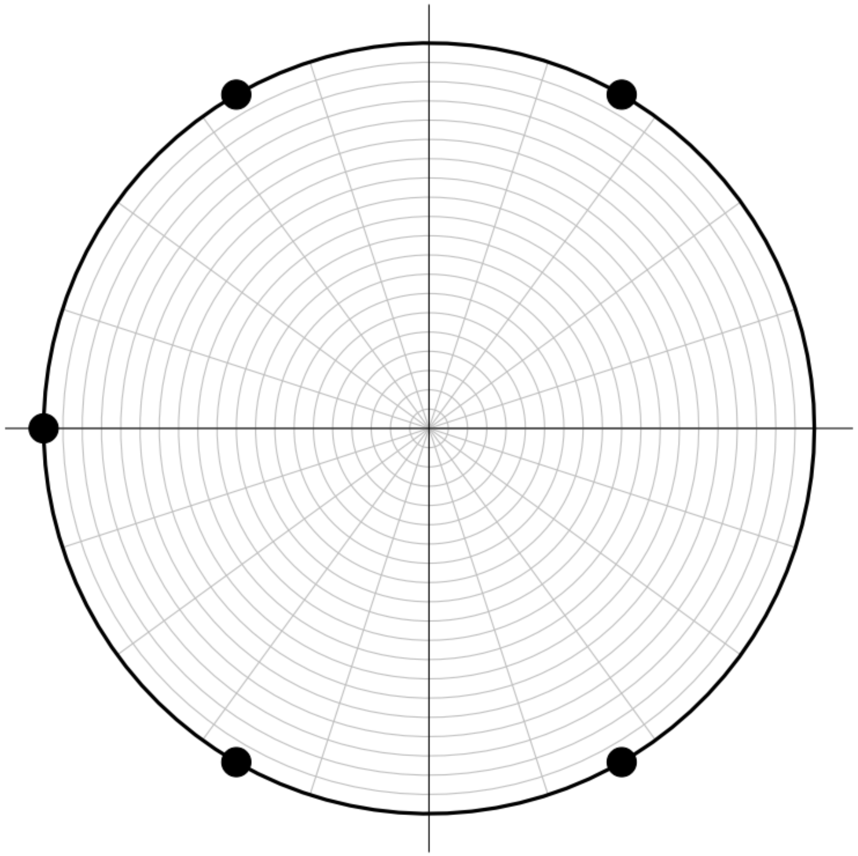
Applying AAA:
From 1 to 20 poles

At 20 poles,
max error on grid
$\approx 1 \times 10^{-14}$

Big idea 2: free-pole interpolation methods

Cubic Spline: Could you guess the knot locations?

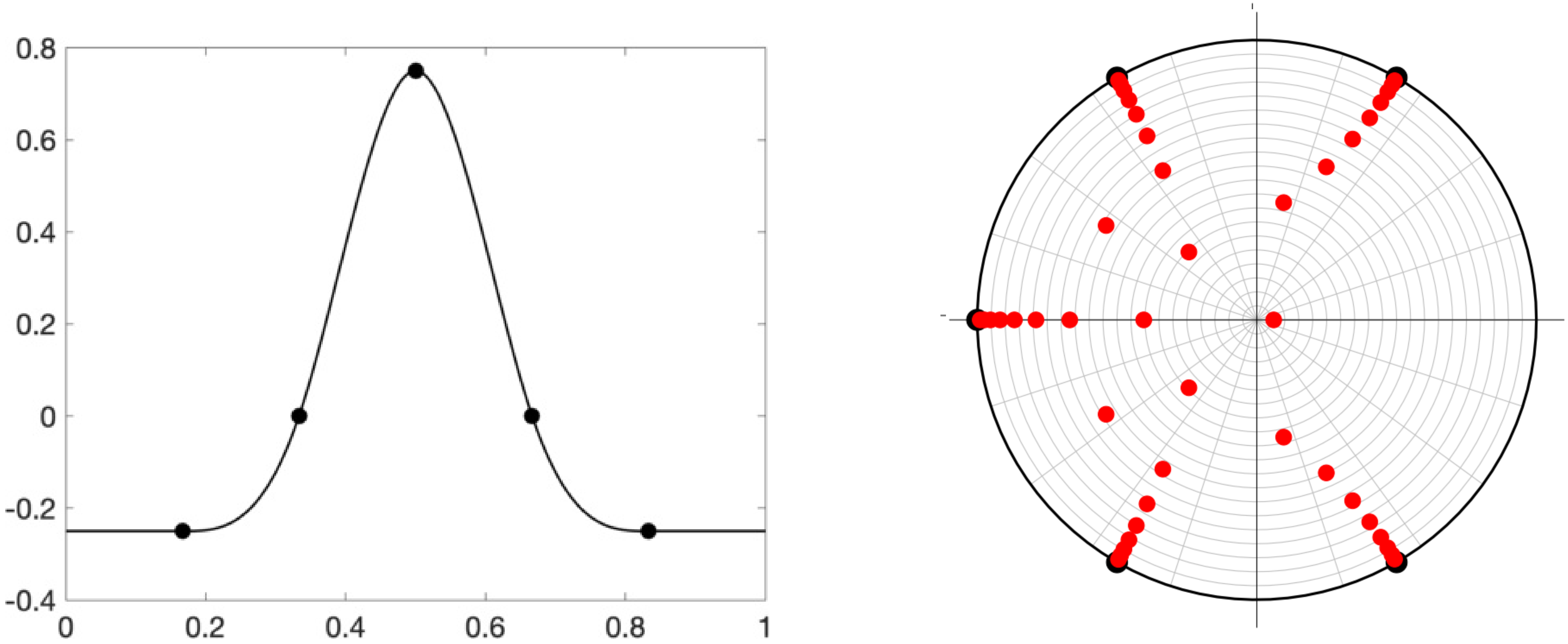




(Wilber, Damle & Townsend, 2022) (Beylkin & Monzòn, 2009)

Big idea 2: free-pole interpolation methods

Cubic Spline: Could you guess the knot locations?



(Wilber, Damle & Townsend, 2022) (Beylkin & Monzòn, 2009)

Big idea 2: free-pole interpolation methods

Data-driven rational approximations

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

<u>Feature extraction:</u> abnormality detection, classification, parameter recovery

<u>NLEVP, Reduced order modeling, dynamical systems</u>

(Antoulas & Anderson, 1986) (Nakatsukasa, Trefethen & Sete, 2018) (Baddoo, 2021), (Wilber, Damle & Townsend, 2022) (Related ideas from: Gutenknecht, Beylkin & Monzòn, Plonka, many more..)

Big idea 2: free-pole interpolation methods

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 space) $\rightleftarrows$ The AAA algorithm (time/spatial)

Computing with rational functions and exponential sums

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

$$\xleftarrow{\texttt{ft}(r_m)}$$

$$\xrightarrow{\texttt{ift}(R_m)}$$

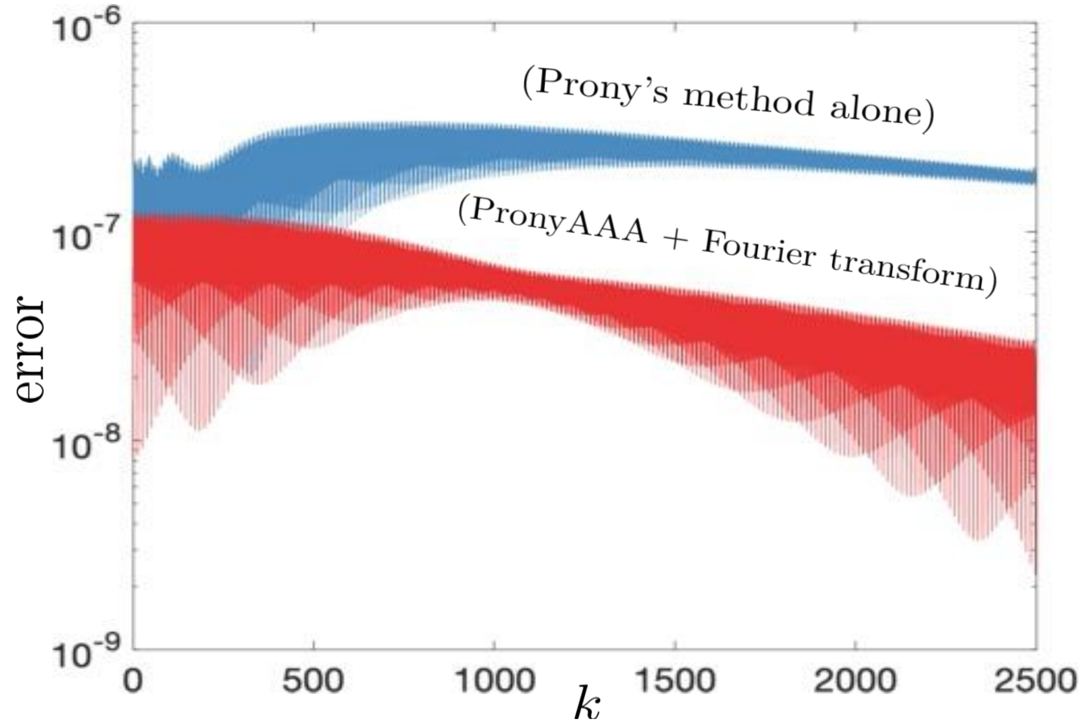
$$r_m^{t,\gamma}(x) = \frac{\sum_{j=1}^{2m} \gamma_j f_j \cot\left(\pi(x - t_j)\right)}{\sum_{j=1}^{2m} \gamma_j \cot\left(\pi(x - t_j)\right)}$$

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

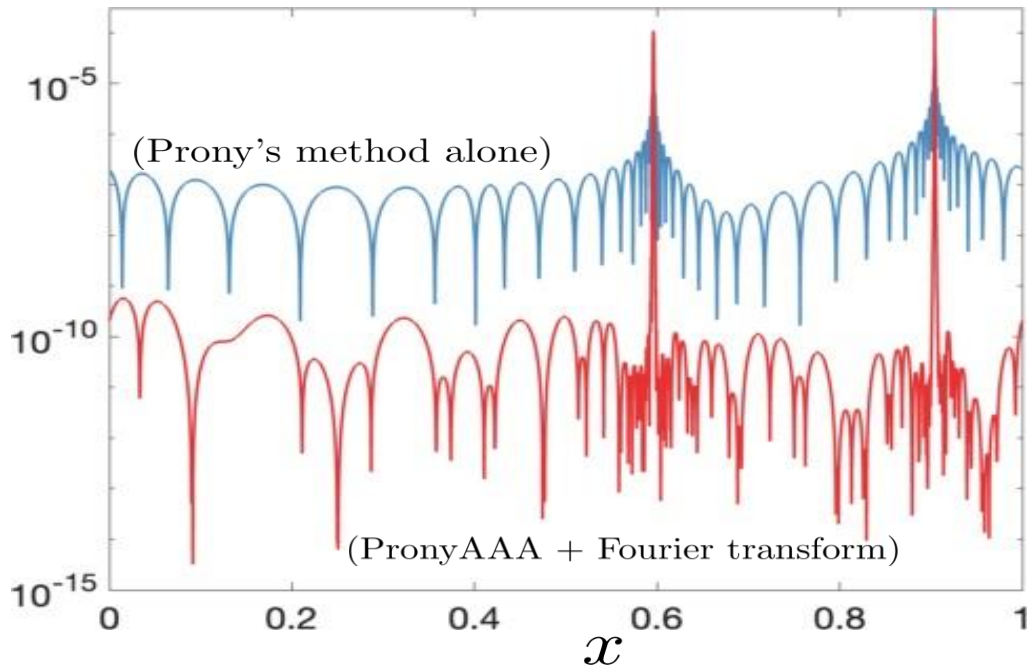
Computing with rational functions and exponential sums

$$R_m(k) = \sum_{j=1}^{m} \omega_j e^{\lambda_j k} \qquad \begin{array}{c} \xleftarrow{\quad\texttt{ft}(r_m)\quad} \\ \xrightarrow{\quad\texttt{ift}(R_m)\quad} \end{array} \qquad r_m^{t,\gamma}(x) = \frac{\sum_{j=1}^{2m} \gamma_j f_j \cot\left(\pi(x-t_j)\right)}{\sum_{j=1}^{2m} \gamma_j \cot\left(\pi(x-t_j)\right)}$$

(Fourier space)



(Time)



Computing with rational functions and exponential sums

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

$\texttt{ft}(r_m)$

$\texttt{ift}(R_m)$

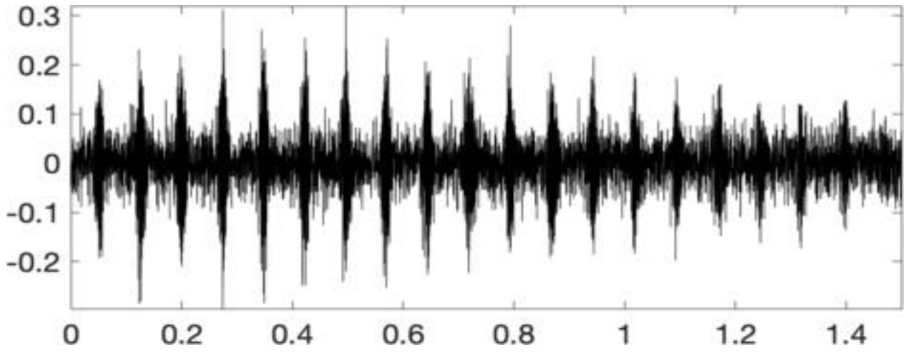
$$r_m^{t,\gamma}(x) = \frac{\sum_{j=1}^{2m} \gamma_j f_j \cot\left(\pi(x-t_j)\right)}{\sum_{j=1}^{2m} \gamma_j \cot\left(\pi(x-t_j)\right)}$$

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

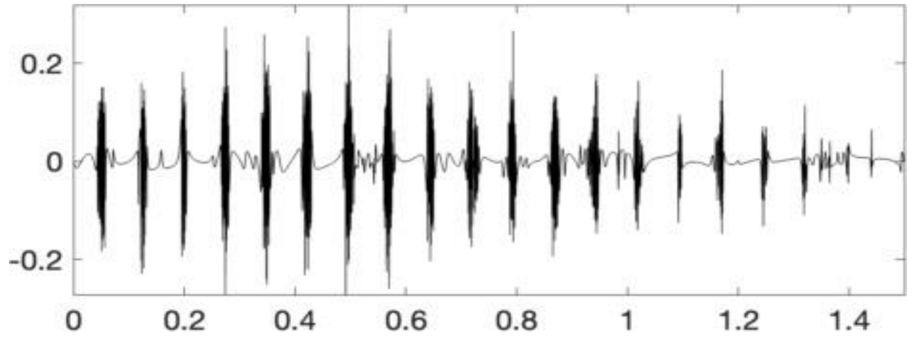
Data-driven computing with rational functions and exponential sums

Automatic denoising

6001 noisy samples from a hydrophone

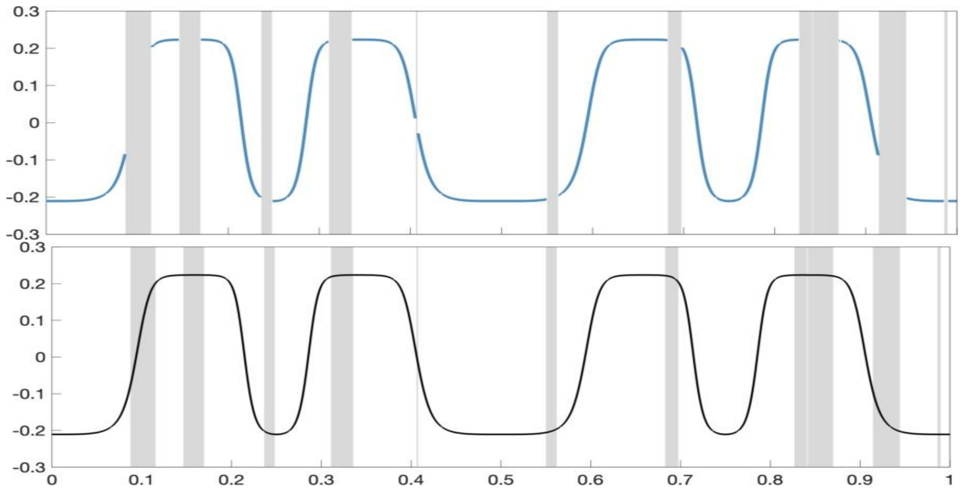


type (245, 246) trigonometric rational

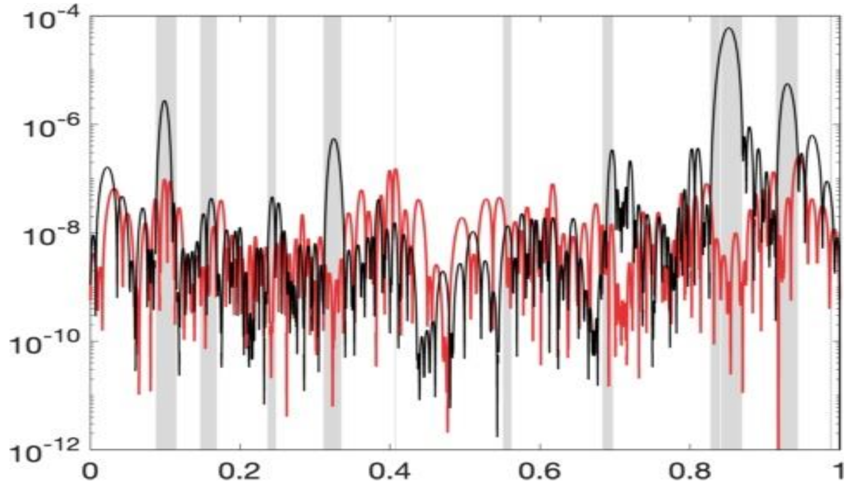


Impute missing data

3000 samples, 609 missing.



Error, type (34, 35) rational function



Big idea 3: closed-form approximations (via integration + quadrature)

When is it worth it to develop a closed-form solution?

- When closed-form “relatives” exist and can be studied.
- When the payout is big! Error analysis is valuable, solves related problems, etc.
- When the continuous problem really matters!

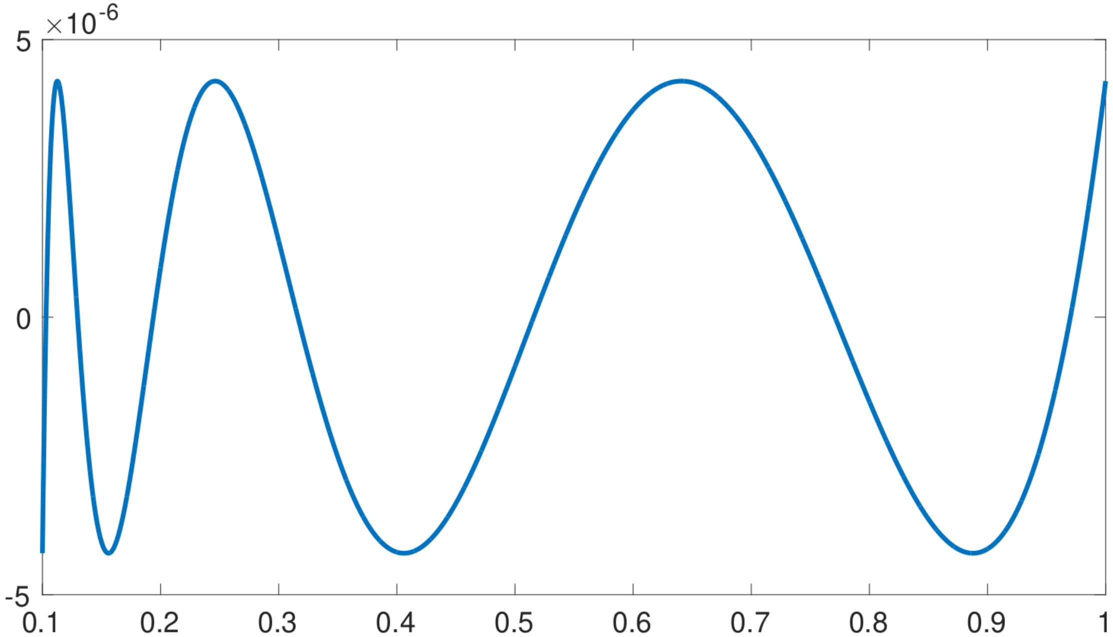
Example: The pth root function on $[0, 1]$.

The square root problem is linked to many important problems in computational mathematics...

(Y. I. Zolotarev)

The best relative rational approximation to $\sqrt{x}$ with k poles on the interval $[\beta, 1]$, $\beta > 0$.

relative error when $k = 3$



- Analysis of iterative solvers for matrix equations.
  [Druskin, Knizhnerman and Simoncini, Beckermann, Sabino, Penzl …]
- Efficient solvers for Sylvester and Ricatti matrix equations.
  [Simoncini, Palitta, Benner, Bujanović, Kürshcher, Saak, Breiten, Wong, Balakrishnan, Li, Truhar, Li, White, Bertram, Faßbender, Kressner, Massei, Robol, Lu, Wachspress, Mehrmann, Gugercin, Sorenson, Penzl, R.C. Smith …]

  W., Townsend (2018)

  W., Rubin, Townsend (2022)
- Singular value decay in matrices with displacement structure.
  [Beckermann, Townsend, Sabino, Rubin, W., …]
- Compression properties in tensors/tensor train compression.
  [Townsend, Shi, …]
- Fast solvers for certain linear systems $Xy = b$.
  [Martinsson, Rokhlin,Tygert, Chandrasekaran, Gu, Xia, Zhu, Xia, Xi, Gu, Beckermann, Kressner, W. , Epperly, W.]

  W., Beckermann, Kressner (2021)

  W., Epperly (2022)
- Optimal complexity solvers for some elliptic PDEs.
  [Olver, Townsend, Fortunato, W., Wright, Boullé, …]

  W., Wright, Townsend (2017)

  W., Townsend (2018)
- Matrix evaluation of sign, square root, absolute value, inversion functions.
  [Gawlik, Nakatsukasa, Hale, Higham, Trefethen, …]

  W., Chen, Martinsson (2022)
- Divide-and-conquer eigensolvers, polar decomposition algorithms.
  [Nakatsukasa, Freund (2016), …]
- Digital filters in signal processing.
  [Daniels, …]

The spectral fractional Poisson equation and pth root approximations

(P.G. Martinsson)

(K. Chen)

Let Ω be a bounded, simply connected, open subset of $\mathbb{R}^d$.

$$\mathcal{L}u = -\left(\frac{\partial^2 u}{\partial x_1^2} + \ldots, + \frac{\partial^2 u}{\partial x_d^2}\right)$$

Let $0 < \alpha < 1$. The spectral fractional Poisson equation is the BVP

$$\mathcal{L}^\alpha u = f,$$
$$u(x) = 0, \quad x \in \partial\Omega,$$

where $\mathcal{L}^\alpha : H_0^2(\Omega) \to L^2$. We will be interested in $\alpha = 1/p$, p pos. integer.

The spectral fractional Poisson equation

Let $\{(\lambda_k, e_k)\}_{k=1}^{\infty}$ be eigenvalue-eigenfunction pairs associated with $\mathcal{L}$ on Ω.

For all k, $\lambda_{k+1} > \lambda_k > 0$, and as $k \to \infty$, $\lambda_k \to 0$.

$$u(x) = \mathcal{L}^{-\alpha} f = \sum_{k=1}^{\infty} \lambda_k^{-\alpha} < e_k, f > e_k(x)$$

$$\boxed{\text{In other words, if } g(x) = x^{-\alpha}, \text{ then } u(x) = g(\mathcal{L}).}$$

"Diffusion of particles with spattering"

-C. Pozrikidis (The Fractional Laplacian)

[(Karnidakas, et. al.), (Pozrikidis) (Shen & Wang) (Harizanov) (Bonito &)]

The spectral fractional Poisson equation

Suppose we have a rational function $r_n(x) \approx x^{-1/p}$.

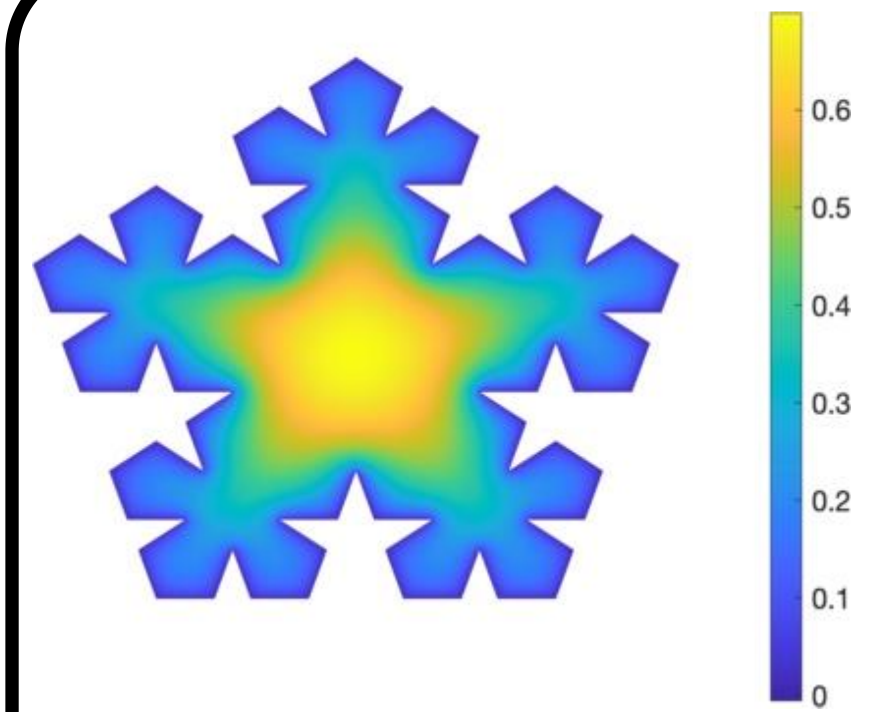
Suppose $r_n(x) = \sum_{j=1}^{n} \frac{\gamma_j}{x - p_j}$. Then,

$$u(x) \approx r_n(\mathcal{L})f = \sum_{j=1}^{n} \gamma_j (\mathcal{L} - p_j I)^{-1} f.$$

This means we can construct
$\hat{u} \approx u$ as $\hat{u} = \sum_{j=1}^{n} u_j$, where each u_j satisfies

$$(\mathcal{L} - p_j I) u_j = \gamma_j f,$$
$$u_j(x) = 0, \quad x \in \partial\Omega.$$

Fast direct solvers



$\Delta^{1/2} = 1$, via UltraSEM

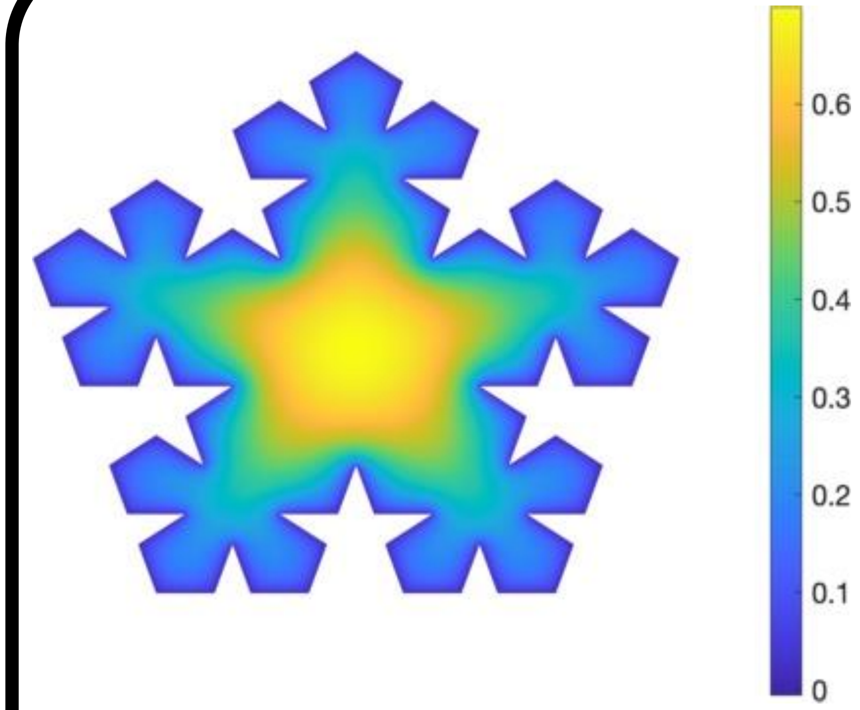
[(Bonito & Pasciak, 2015), (Aceto & Novati, 2017), (Martinsson, 2019) (Fortunato, Hale & Townsend, 2020)]

The spectral fractional Poisson equation

Key Ingredients:

1. Excellent fast and accurate solvers for shifted Laplace equations on complicated domains.

2. Excellent rational approximation to $x^{-1/p}$ on $[1, \infty)$
(Continuous problem, infinite domain)



$\Delta^{1/2} = 1$, via UltraSEM

[(Bonito & Pasciak, 2015), (Aceto & Novati, 2017), (Martinsson, 2019) (Fortunato, Hale & Townsend, 2020)]

How to solve it:

Transform to a finite interval:
Let $r_n(1/x) = y_n(x)$, where $x \in [0, 1]$

Now we must construct $y_n(x) \approx x^{1/p}$ on $[0, 1]$.

How to build such a rational function?

Sampling-based methods (fixed or free-pole):

Error blows up in locations off sampling grid as $x \to 0$.

Analytical construction:

Construct a contour integration problem on $[\beta, 1]$, Apply quadrature to form y_n.
If the contour + quadrature is chosen well, then y_n will behave well on $[0, \beta]$.

How to solve it: contour integration

$$f(x) = \frac{x}{2\pi i} \int_\gamma z^{-1} f(z)(z-x)^{-1} \mathrm{d}z, \qquad x \in [\beta, 1].$$

Apply a quadrature rule consisting of k weight-node pairs, $\{(w_j, z_j)\}_{j=1}^{k}$:

$$f(x) \approx \frac{x}{2\pi i} \sum_{j=1}^{k} \frac{-\gamma_j}{x - p_j}, \text{ where, } \gamma_j = w_j z_j^{-1} f(z_j) \mathrm{d}z_j.$$

Key Idea:

Choose the contour and quadrature points cleverly via conformal mapping.

[(Hale, Higham & Trefethen, 2007)]

How to solve it: contour integration

Let $f(x) = x^{1/2}$.

$$f(x) = \frac{x}{2\pi i}\int_{\gamma} z^{-1} f(z)(z-x)^{-1}\mathrm{d}z, \quad x \in$$

Let $v^2 = z$:

$$f(x) = \frac{x}{\pi i}\int_{\gamma_v} (v^2 - x)^{-1}\mathrm{d}v$$

Map conformally to a rectangle R via Schwarz-Christoffel mapping:

$$v = \beta^{1/2}\mathrm{sn}(t|q), \quad q = \beta^{1/2}$$

Apply trapezoidal quadrature rule.

The resulting rational approximation is the best relative approximation to $\sqrt{x}$ on $[\beta, 1]$!

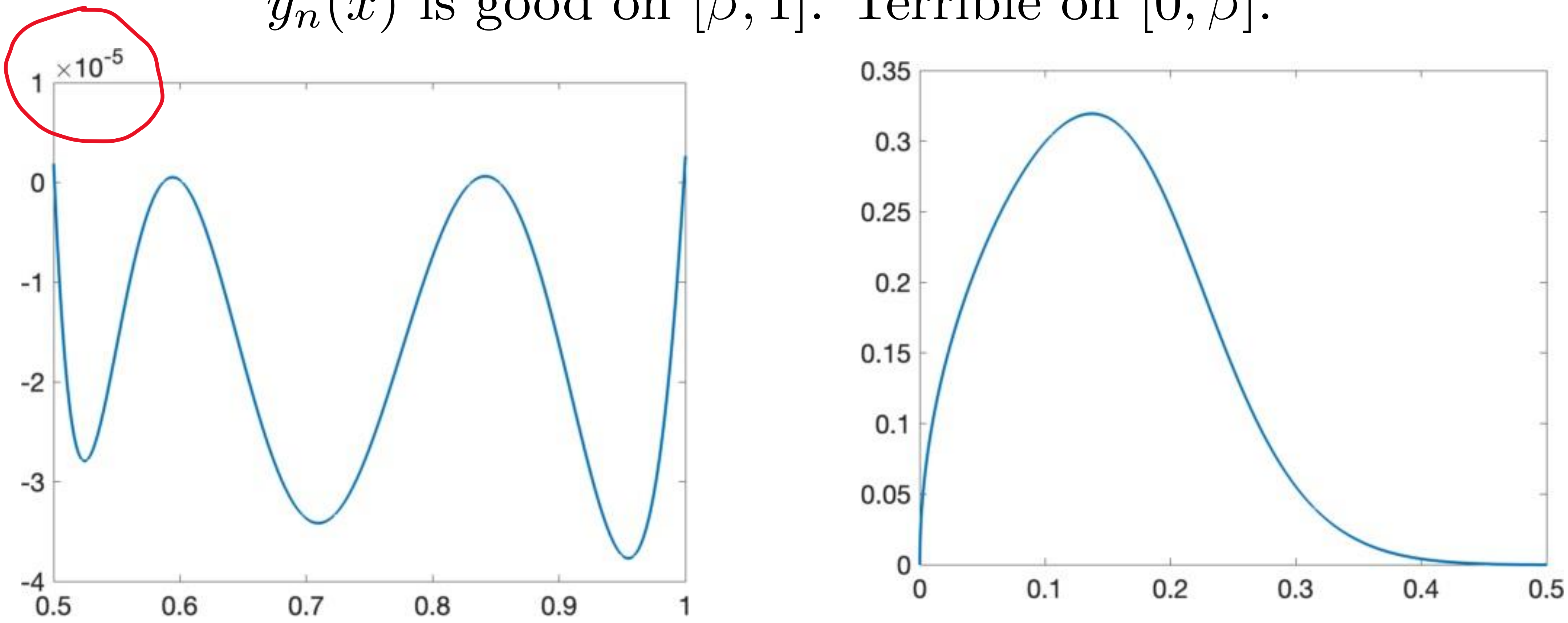
A blueprint for approximations to $x^{1/p}$ on $[0, 1]$?

[(Hale, Higham & Trefethen, 2007)]

How to solve it: contour integration

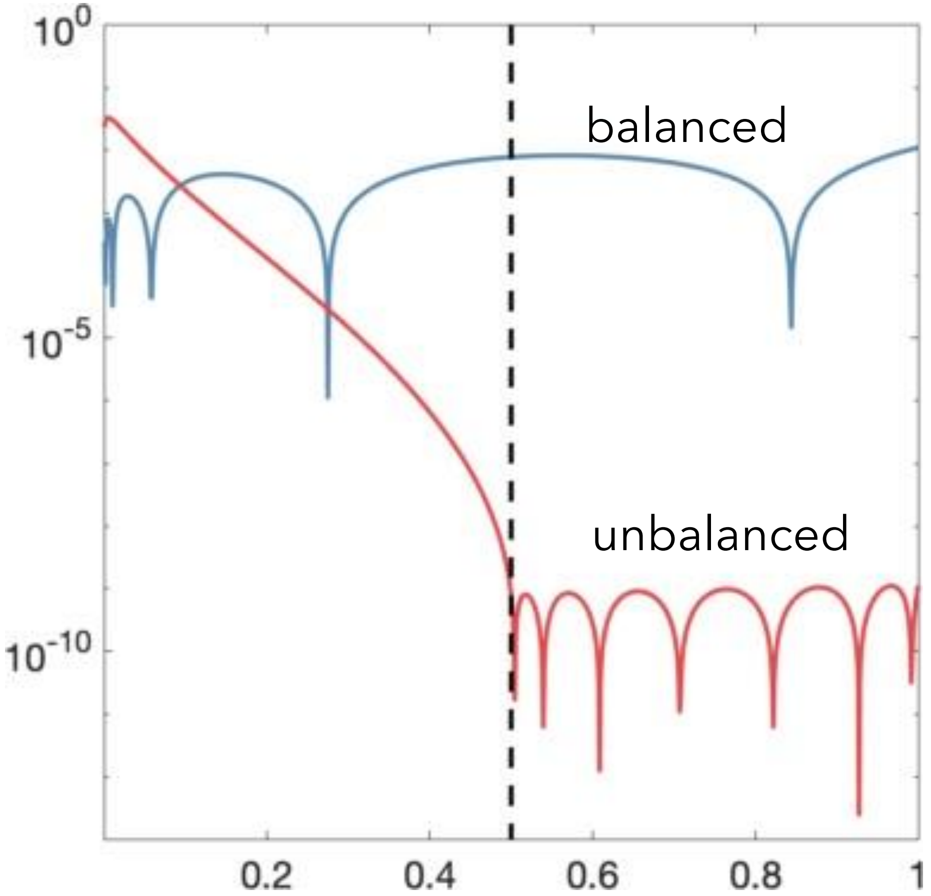
Bad extrapolation properties: $\beta = .5, \alpha = 1/2,$

$$y_n(x) \text{ is good on } [\beta, 1]. \text{ Terrible on } [0, \beta].$$



How to solve it: contour integration

Getting a good rational function :



$\beta = .5, \alpha = 1/2,$

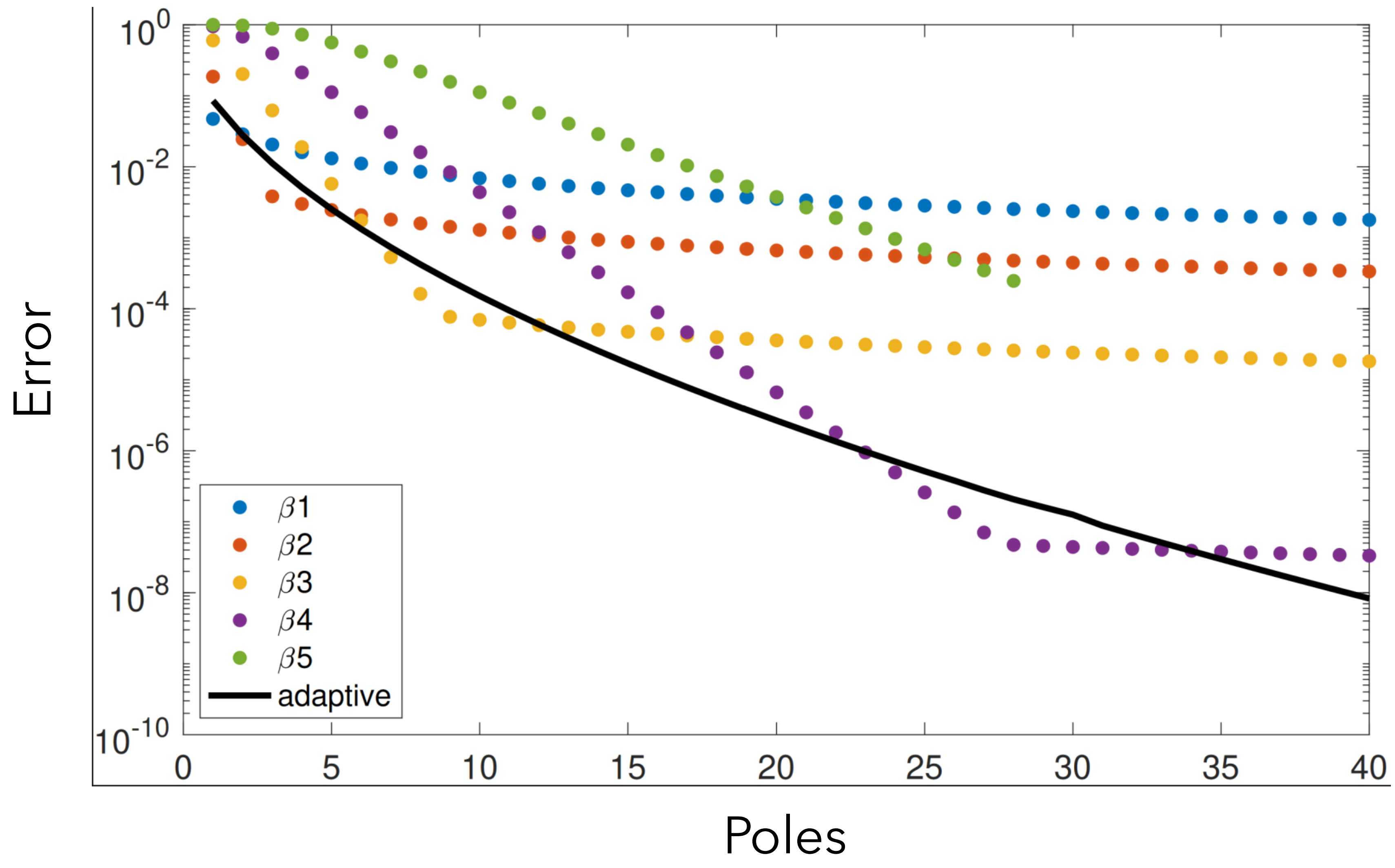
The error is $|y_n(x) - \sqrt{x}| < \eta(\beta)$ on $[0, \beta]$.

The error is $|y_n(x) - \sqrt{x}| < C\rho^{-n}$ on $[\beta, 1]$, where $\rho = \exp\left(\pi^2/2\log(4/\beta)\right)$

For a fixed n, choose β to balance out the error distribution. $\tilde{y}_n(x) = y_n[\beta](x)$

(Nakatsukasa & Gawlik, 2019), (Harizanov, 2022)

How to solve it: contour integration



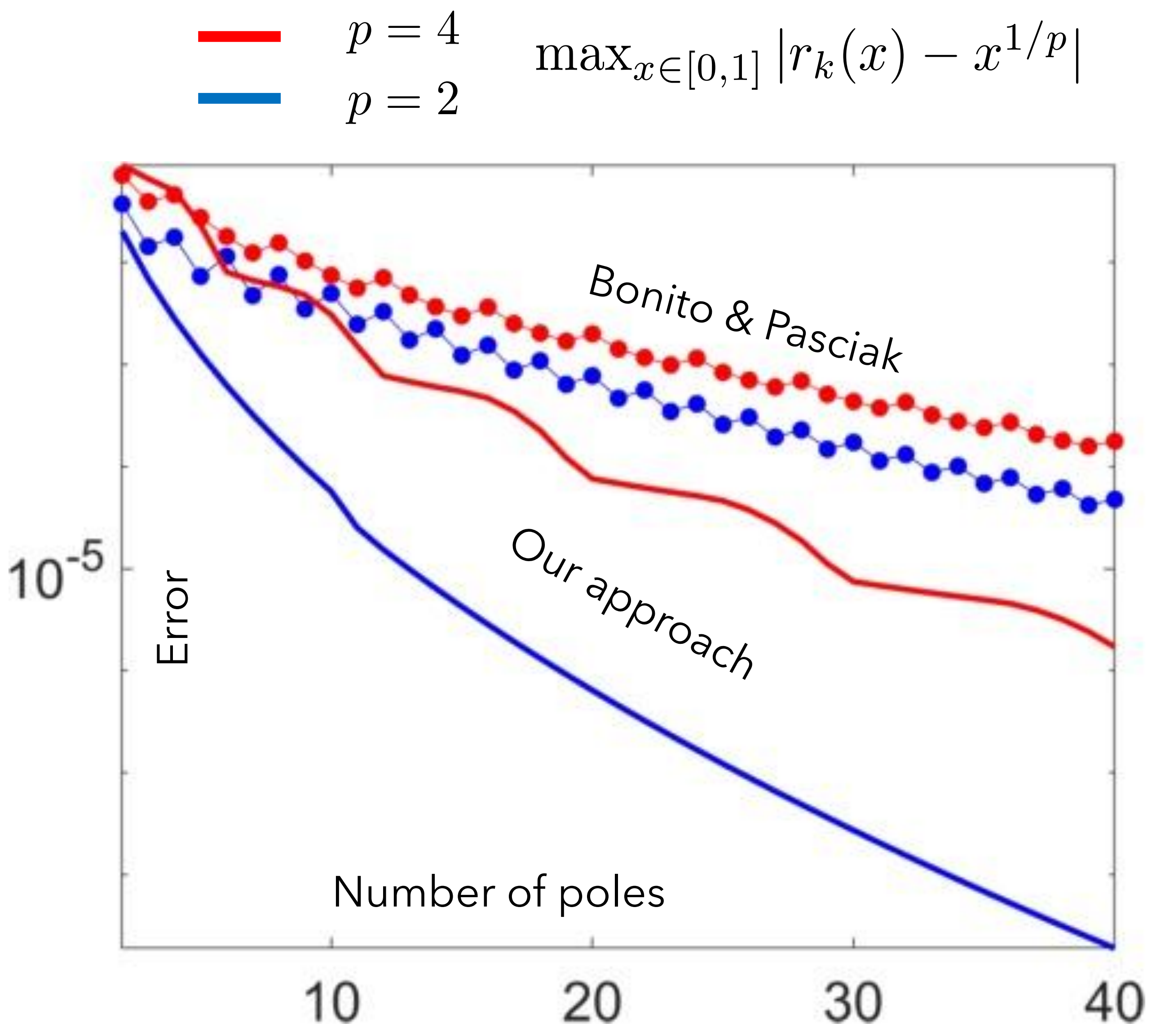
$\beta_1 = 10^{-1}$
$\beta_2 = 10^{-3}$
$\beta_3 = 10^{-6}$
$\beta_4 = 10^{-12}$
$\beta_5 = 10^{-24}$

(Nakatsukasa & Gawlik, 2019), (Harizanov, 2022)

How to solve it: contour integration

We extend the mapping + quadrature + balancing idea used for the square root approx. to construct rational approximations to $x^{1/p}$ on $[0,1]$.

Our strategy has closed-form expressions. Poles are always on the negative real line Exponential convergence rates on $[\beta, 1]$ and error on $[0,\beta]$ on is loosely bounded by $10\beta/p$. This suggests we can attain root-exponential convergence.



Summary: Three big strategies for constructing rational approximations

When you know where the singularity lives + have access to samples: **Pole clustering + linear fit to data!**

When you want to know where the singularity lives + have access to samples: **Pole free interpolation methods!**

When you need a continuous or closed-form solution: **Contour integration + quadrature!**

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)

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)

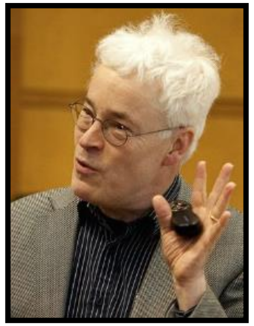
$r_m \approx f$ can be constructed by solving the approximate interpolation problem $|\hat{f}_k - R_m(k)| \leq \epsilon \|f\|$, for $0 \leq k \leq N_\epsilon$. (Regularized Prony's method)

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

The AAA 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 $r_n(x_k) = f(x_k)$ for $\{x_0, \ldots, x_n\}$

Determining the barycentric weights:

Here, $r_n(x) = \frac{p_n(x)}{q_n(x)}$. Get weights by minimizing $\|f(X)q_n(X) - p_n(X)\|_2$

Choosing the next interpolating point:

$$x_{n+1} = \text{argmax}_{x \in T \setminus \{x_0, \ldots, x_n\}} |r_n(x) - f(x)|$$

PronyAAA algorithm

Advantage for postprocessing: rootfinding

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

tages

- 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…

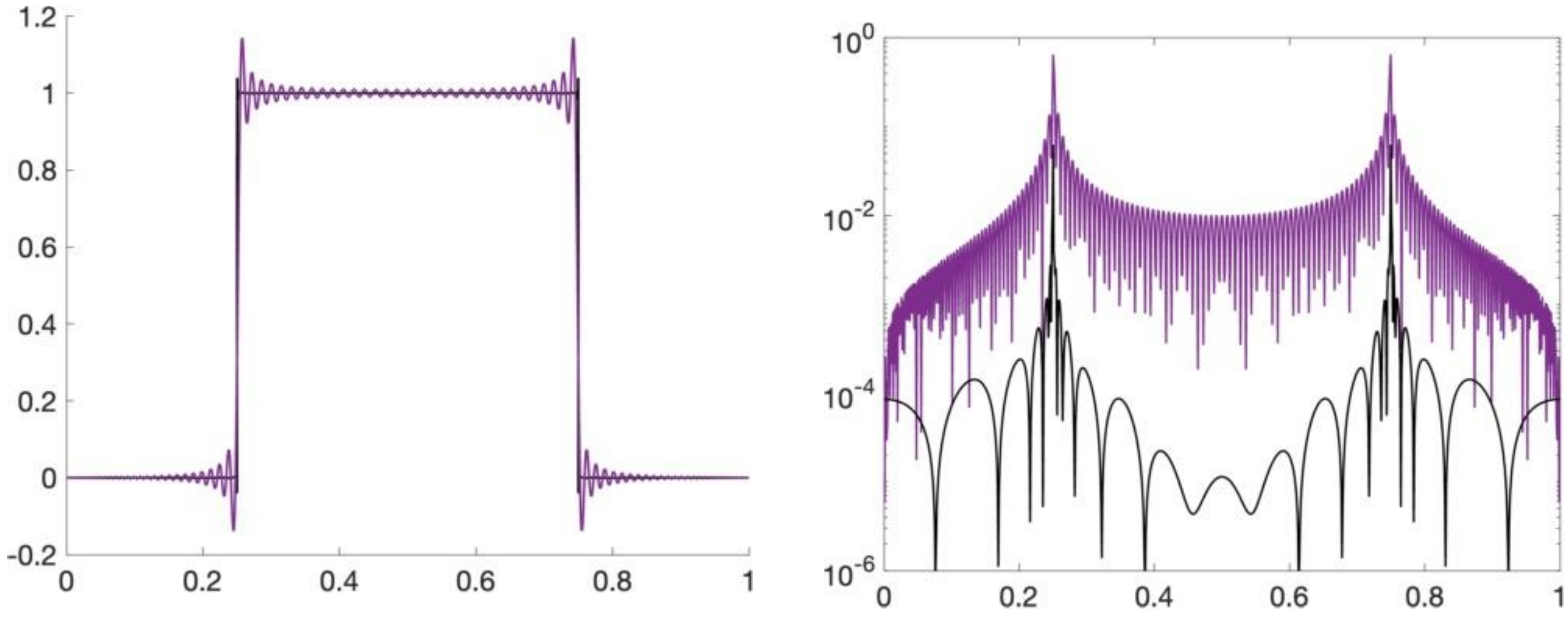
Solving linear systems/matrix equations: (Ruhe, 1994),(Druskin and Simoncini, 2011), (Sabino, 2008), (Kressner, Massei, and Robol, 2019), (Benner, Truhar, and Li, 2009), (W. And Townsend, 2018)many more…

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

<u>Key Idea:</u> **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 $t = \{t_1, \ldots, t_{2m}\} \subset T$

<u>Determining the barycentric weights:</u>

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

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

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

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

<u>Choosing the next interpolating point:</u>

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

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

Exponential sums to barycentric interpolants

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

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

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

<u>Theorem:</u> (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.

<u>Idea 1:</u> 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)

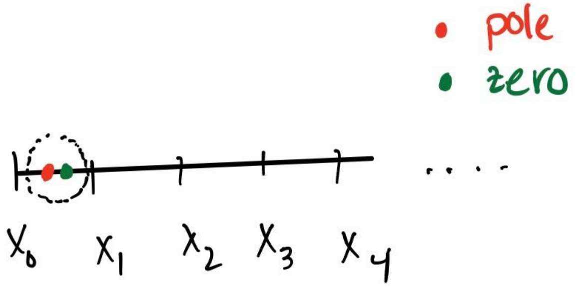
PronyAAA algorithm

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

e 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

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

How can we find each λ_j?

(Gaspard de Prony)

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)

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

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

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

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

barycentric to exponential sum

$$R_m(k) = \sum_{j=1}^{m} \omega_j e^{\lambda_j k} \xleftarrow{\mathcal{F}(r_m^{t,\gamma})} r_m^{t,\gamma}(x) = \frac{\sum_{j=1}^{2m} \gamma_j f_j \cot\left(\pi(x - t_j)\right)}{\sum_{j=1}^{2m} \gamma_j \cot\left(\pi(x - t_j)\right)}$$

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

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

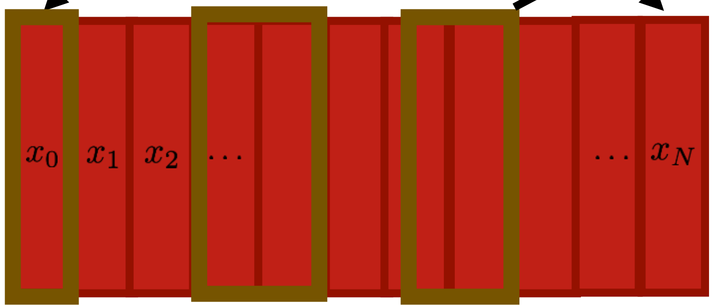
$$r_m(x) = \mathcal{F}^{-1}(R_m)(x)$$

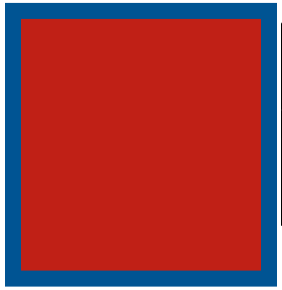
$$\boxed{r_m^{t,\gamma}(x) = \frac{\sum_{j=1}^{2m} \gamma_j f_j \cot\left(\pi(x - t_j)\right)}{\sum_{j=1}^{2m} \gamma_j \cot\left(\pi(x - t_j)\right)}}$$

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.

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





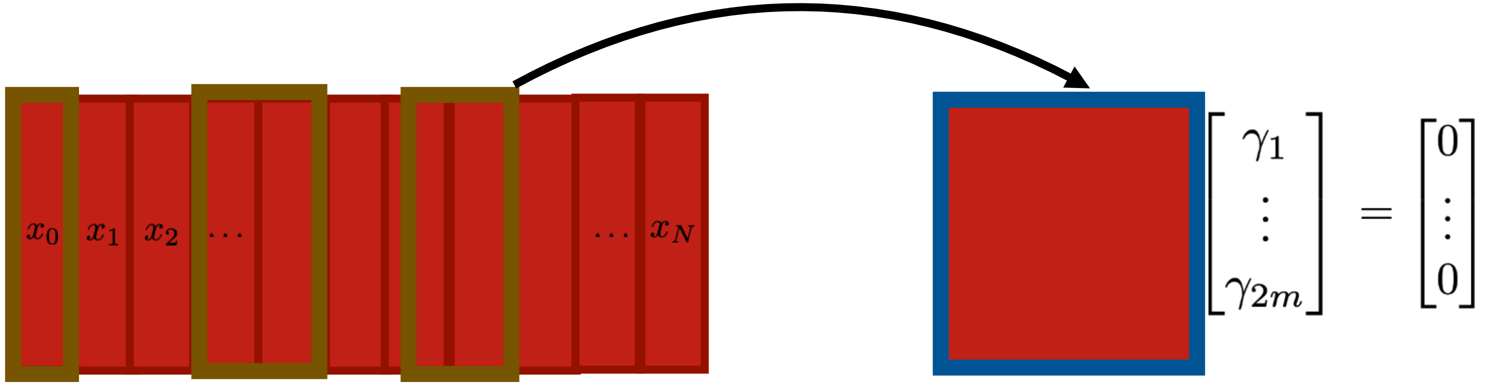
$$\begin{bmatrix} \gamma_1 \\ \vdots \\ \gamma_{2m} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

exponential sum to barycentric: CPQR-selected interpolation points

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

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

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



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

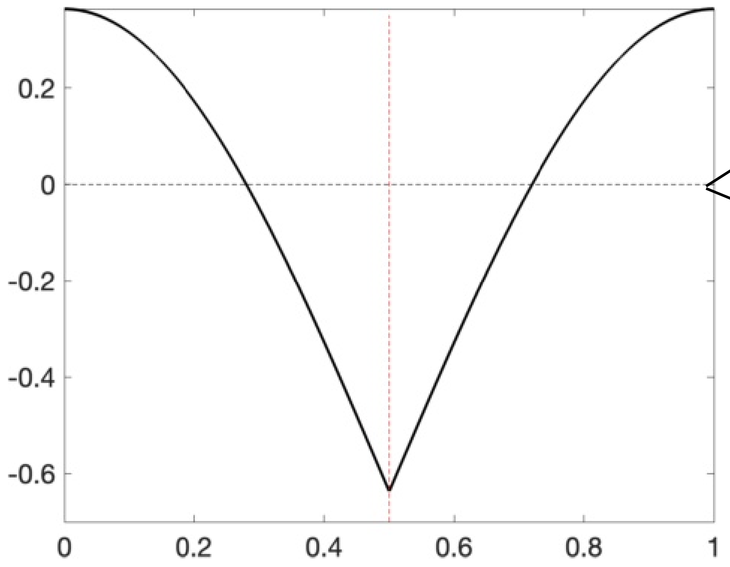
Column-pivoted QR (CPQR) [Golub & Businger (1965), Chandrasekaran & Ipsen (1994), Gu & Eisenstat (1996)]

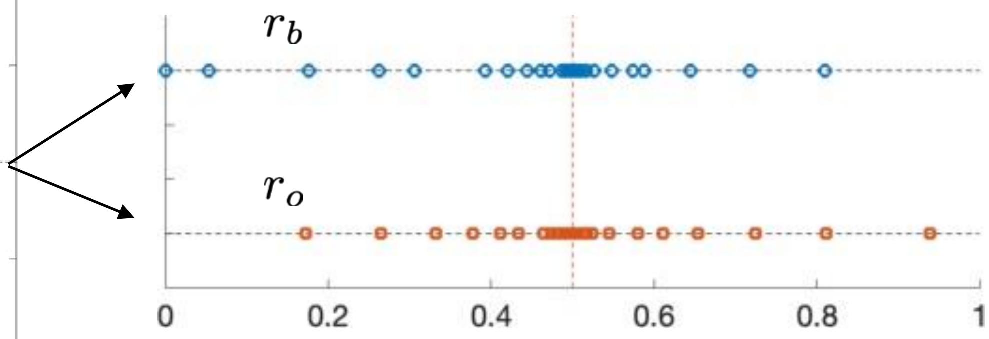
1. CPQR to choose candidates for barycentric nodes.
2. Regularization procedure: Constrained optimization to subselect from candidate nodes + find weights $\gamma = \{\gamma_1, \ldots, \gamma_{2m}\}$.

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.

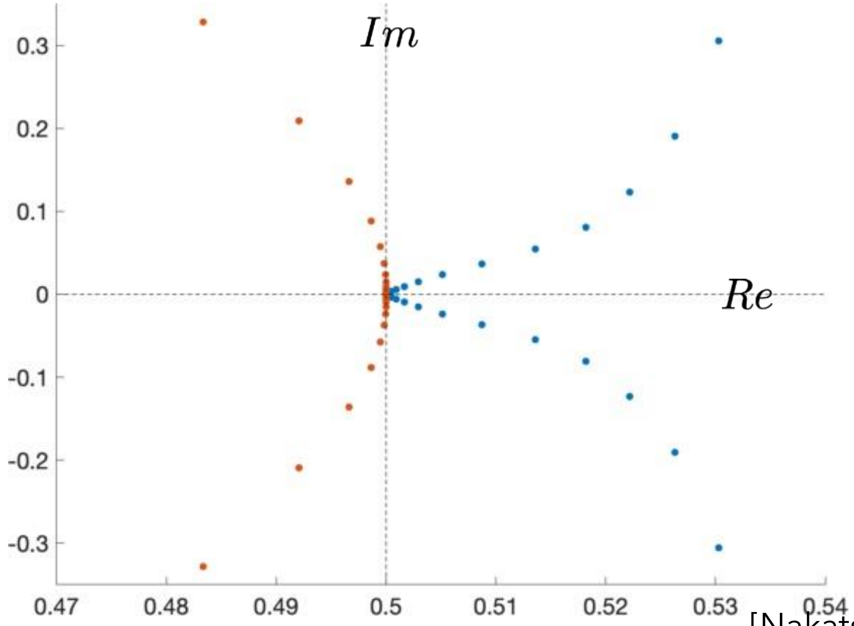




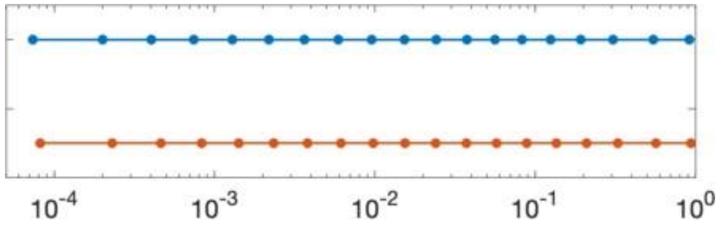
AAA-selected and CPQR-selected poles

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

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.



[Nakatsukasa , Weideman & Trefethen (2021)]