Journal of Scientific Computing (2022) 92:107
https://doi.org/10.1007/s10915-022-01963-z

TECHNICAL NOTE

Pole Recovery From Noisy Data on Imaginary Axis

Lexing Ying[1]

Received: 10 April 2022 / Revised: 14 July 2022 / Accepted: 21 July 2022

Abstract
This note proposes an algorithm for identifying the poles and residues of a meromorphic function from its noisy values on the imaginary axis. The algorithm uses Möbius transform and Prony's method in the frequency domain. Numerical results are provided to demonstrate the performance of the algorithm.

Keywords Rational approximation · Prony's method · Analytic continuation

Mathematics Subject Classification 30B40 · 93B55

1 Introduction

Let $g(z)$ be a meromorphic function of the form

$$g(z) = \sum_{j=1}^{N_p} \frac{r_j}{\xi_j - z}, \tag{1}$$

where the number of poles N_p, the pole locations $\{\xi_j\}$, and residues $\{r_j\}$ are all unknown, except that ξ_j are away from the imaginary axis $i\mathbb{R}$. The problem is to recover N_p, $\{\xi_j\}$ and $\{r_j\}$, given the noisy access of $g(z)$ along the imaginary axis $i\mathbb{R}$. Two access models are particularly relevant:

- the **random access model** where one can get noisy values of $g(z)$ at desired locations on $i\mathbb{R}$ and
- the **Matsubara model** where one can only get the noisy values of $g(z)$ at the Matsubara grid

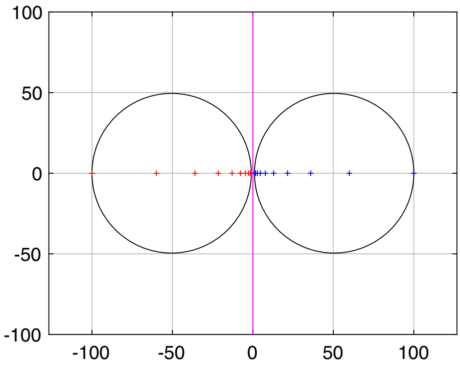
$$z_n = \begin{cases} 2n\frac{\pi i}{\beta}, & \text{for bosons,} \\ (2n+1)\frac{\pi i}{\beta}, & \text{for fermions.} \end{cases}$$

To make the problem numerically well-defined, we assume

✉ Lexing Ying
lexing@stanford.edu

1 Department of Mathematics, Stanford University, Stanford, CA 94305, United States

Fig. 1 The unknown poles are inside the two circles. The algorithm can access the noisy function values along the imaginary axis



- There exists constants $0 < a < b$ such that the poles $\{\xi_j\}$ reside within the union of the two disks of radius $\frac{b-a}{2}$ centered at $-\frac{b+a}{2}$ and $\frac{b+a}{2}$, respectively. See Fig. 1 for an illustration.

This assumption is quite natural because otherwise any algorithm is forced to sample extensively along the imaginary axis towards infinity.

There is also a matrix-valued version of this problem, where

$$G(z) = \sum_{j=1}^{N_p} \frac{R_j}{\xi_j - z}, \tag{2}$$

where $G(z)$ and R_j are matrices of size $N_b \times N_b$. The task is then to recover N_p, $\{\xi_j\}$ and $\{R_j\}$. A particularly important special case is where $R_j = v_j v_j^*$ for some $v_j \in \mathbb{C}^{N_b}$ [15].

The main contribution of this note is a simple algorithm based on conformal mapping and Prony's method [19]. Prony's method is generally considered to be unstable in the presence of noise [1]. However, in practice it has been widely used in signal processing and control theory, especially when certain prior can be imposed. In our problem, such prior information includes 1) the restriction on the locations of the poles and 2) whether the residues is real, positive, or positive definite.

This problem has many applications in scientific and engineering disciplines. One of the key examples is the reconstruction of spectral density from Matsubara Green's function [6]. It is also highly related to a couple of other well-studied problems, including rational function approximation and interpolation [2, 3, 5, 7, 12, 13, 16], Pade approximation [11], contractive analytic continuation [8, 9], approximation with exponential sums [4, 18], and hybridization fitting [15]. Recently, in [14], the author also applies Prony's method to the problem (1) with only a finite set of Matsubara data, which is a significantly harder problem.

The rest of the note is organized as follows. In Sect. 2, we describe the key ideas and the implementation of the algorithm in details. Several numerical examples are provided in Sect. 3.

2 Algorithm

2.1 Main Ideas

Let us consider the scalar case (1). Below we describe the algorithm as if one can manipulate continuous objects. The overall plan is

- Transforming the computational domain with a specific Möbius transform,

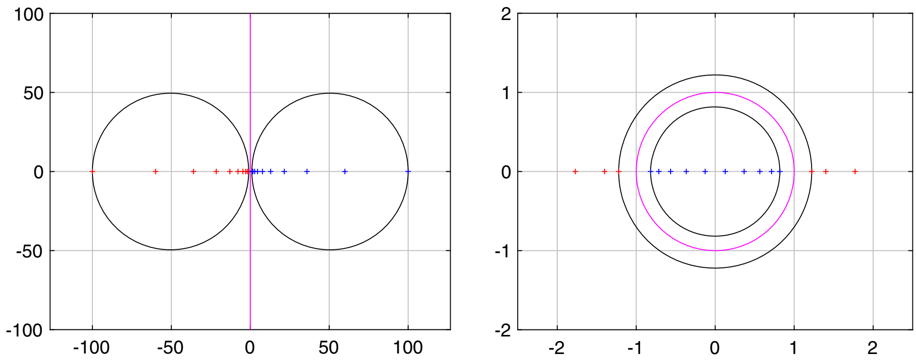


Fig. 2 Möbius transform. Left: the z plane. Right: the t plane

- Applying Prony's method to locate the poles in the transformed domain,
- Mapping back to get the pole locations in the original domain and computing the residues using the least square.

Step 1. We introduce the following Möbius transform from $z \in \mathbb{C}$ to $t \in \mathbb{C}$

$$t = \frac{z - \sqrt{ab}}{z + \sqrt{ab}}, \quad z = -\sqrt{ab}\frac{t+1}{t-1}. \tag{3}$$

This transform maps

- the right half-plane $\mathbb{C}^+$ in z to the interior of the unit disk $\mathbb{D}$ in t,
- the left half-plane $\mathbb{C}^-$ in z to the exterior of $\mathbb{D}$ in t,
- the imaginary axis $i\mathbb{R}$ in z to the unit circle in t,
- the two circles centered at $-\frac{b+a}{2}$ and $\frac{b+a}{2}$ in z to two concentric circles with radius $\frac{\sqrt{b}-\sqrt{a}}{\sqrt{b}+\sqrt{a}}$ and $\frac{\sqrt{b}+\sqrt{a}}{\sqrt{b}-\sqrt{a}}$ in t (see Fig. 2 for an illustration).

The function $g(t) \equiv g(z(t))$ in the t space also enjoys a pole representation

$$g(z) = \sum_{j=1}^{N_p} \frac{w_j}{\tau_j - t} + \text{const},$$

with locations $\{\tau_j\}$ and residues $\{w_j\}$. Since $\{\tau_j\}$ are the images of the poles $\{z_j\}$ under the Möbius transform, finding $\{z_j\}$ is equivalent to locating $\{\tau_j\}$.

Step 2. Let us consider the integrals

$$\frac{1}{2\pi i} \int_{\partial\mathbb{D}} \frac{g(t)}{t^k} \frac{dt}{t} \tag{4}$$

for integer values of k. The integrals for negative and positive values of k give information about the poles inside $\mathbb{D}$ and the ones outside $\mathbb{D}$, respectively.

For any $k \le -1$,

$$
\begin{aligned}
\frac{1}{2\pi i}\int_{\partial\mathbb{D}} \frac{g(t)}{t^k}\frac{\mathrm{d}t}{t} &= \frac{1}{2\pi i}\int_{\partial\mathbb{D}} \left(\sum_{|\tau_j|<1} + \sum_{|\tau_j|>1}\right) \frac{w_j}{\tau_j - t} t^{|k|+1}\mathrm{d}t \\
&= \frac{1}{2\pi i}\sum_{|\tau_j|<1} w_j \int_{\partial\mathbb{D}} \frac{1}{\tau_j - t} t^{|k|-1}\mathrm{d}t = \frac{1}{2\pi i}\sum_{|\tau_j|<1} w_j \tau_j^{|k|-1} \\
&\quad \int_{\partial\mathbb{D}} \frac{1}{\tau_j - t}\mathrm{d}t = -\sum_{|\tau_j|<1} w_j \tau_j^{-(k+1)},
\end{aligned}
$$

where the second equality uses the fact that $\frac{w_j}{\tau_j - t}$ is analytic in $\mathbb{D}$ for $|\tau_j| > 1$ and the third equality uses the fact that $\frac{\tau_j^{|k|-1} - t^{|k|-1}}{\tau_j - t}$ is a polynomial hence analytic in $\mathbb{D}$. Therefore, the integrals (4) for $k \le -1$ contain information about the poles inside $\mathbb{D}$.

For any $k \ge 1$,

$$
\begin{aligned}
\frac{1}{2\pi i}\int_{\partial\mathbb{D}} \frac{g(t)}{t^k}\frac{\mathrm{d}t}{t} &= \frac{1}{2\pi i}\int_{\partial\mathbb{D}} \left(\sum_{|\tau_j|<1} + \sum_{|\tau_j|>1}\right) \frac{w_j}{\tau_j - t} \cdot \frac{1}{t^{k+1}}\mathrm{d}t \\
&= \frac{1}{2\pi i}\sum_{|\tau_j|>1} w_j \int_{\partial\mathbb{D}} \frac{1}{\tau_j - t} \cdot \frac{1}{t^{k+1}}\mathrm{d}t \\
&= \frac{1}{2\pi i}\sum_{|\tau_j|>1} w_j \int_{\partial\mathbb{D}} \frac{1}{\tau_j}\left(1 + \frac{t}{\tau_j} + \cdots\right) \frac{1}{t^{k+1}}\mathrm{d}t \\
&= \frac{1}{2\pi i}\sum_{|\tau_j|>1} w_j \frac{1}{\tau_j^{k+1}} \int_{\partial\mathbb{D}} \frac{1}{t}\mathrm{d}t = \sum_{|\tau_j|>1} w_j \tau_j^{-(k+1)},
\end{aligned}
$$

where the second equality uses the fact that for $|\tau_j| < 1$ the product $\frac{w_j}{\tau_j - t} \cdot \frac{1}{t^{k+1}}$ is analytic outside $\mathbb{D}$ with at least quadratic decay. The fourth step uses the fact that only the term with t^k in the power expansion gives a non-zero contribution. Hence the integrals (4) for $k \ge 1$ contain information about the poles outside $\mathbb{D}$.

Since the integral (4) is over the unit circle, it is closely related to the Fourier transform of the function $g(\theta) \equiv g(e^{i\theta})$:

$$
\frac{1}{2\pi i}\int_{\partial\mathbb{D}} \frac{g(t)}{t^k}\frac{\mathrm{d}t}{t} = \frac{1}{2\pi i}\int_0^{2\pi} g(\theta)e^{-ik\theta} i\,\mathrm{d}\theta = \frac{1}{2\pi}\int_0^{2\pi} g(\theta)e^{-ik\theta}\mathrm{d}\theta = \hat{g}_k. \tag{5}
$$

To recover the poles, we apply Prony's method using the Fourier coefficients. The discussion in the rest of this step is not new: it is part of the well-known AAK theory and an excellent write-up in a linear algebraic notation can be found in [10].

The poles inside $\mathbb{D}$ uses the Fourier coefficients of the negative frequencies. From the integrals with $k \le -1$, define the semi-infinite vector

$$
\hat{g}_- \equiv \begin{bmatrix} \hat{g}_{-1} \\ \hat{g}_{-2} \\ \vdots \end{bmatrix} \equiv \frac{1}{2\pi i}\int_{\partial\mathbb{D}} g(t) \begin{bmatrix} t^0 \\ t^1 \\ \vdots \end{bmatrix} \mathrm{d}t \equiv \begin{bmatrix} -\sum_{|\tau_j|<1} w_j \tau_j^0 \\ -\sum_{|\tau_j|<1} w_j \tau_j^1 \\ \vdots \end{bmatrix}
$$

Let us define S to be the shift operator that shifts the semi-infinite vector upward (and drops the first element). For any τ_j with $|\tau_j| < 1$,

$$S \begin{bmatrix} \tau_j^0 \\ \tau_j^1 \\ \vdots \end{bmatrix} = \begin{bmatrix} \tau_j^1 \\ \tau_j^2 \\ \vdots \end{bmatrix}, \quad \text{i.e.,} \quad (S - \tau_j) \begin{bmatrix} \tau_j^0 \\ \tau_j^1 \\ \vdots \end{bmatrix} = 0.$$

Since the operators $S - \tau_j$ all commute,

$$\prod_{|\tau_i|<1} (S - \tau_i) \begin{bmatrix} \tau_j^0 \\ \tau_j^1 \\ \vdots \end{bmatrix} = 0. \tag{6}$$

Since $\hat{g}_-$ is a linear combination of the semi-infinite vectors in (6),

$$\prod_{|\tau_i|<1} (S - \tau_i)\, \hat{g}_- = 0.$$

Introduce the polynomial $\prod_{|\tau_i|<1}(t - \tau_i) = p_0 t^0 + \cdots + p_d t^d$ with coefficients $\{p_i\}$, where the degree d is equal to the number of poles in $\mathbb{D}$. Then (6) becomes

$$p_0(S^0 \hat{g}_-) + \cdots + p_d(S^d \hat{g}_-) = 0, \quad \text{i.e.,} \quad \begin{bmatrix} \hat{g}_{-1} & \hat{g}_{-2} & \cdots & \hat{g}_{-(d+1)} \\ \hat{g}_{-2} & \hat{g}_{-3} & \cdots & \hat{g}_{-(d+2)} \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} p_0 \\ \cdots \\ p_d \end{bmatrix} = 0. \tag{7}$$

This implies that the number of poles in $\mathbb{D}$ is equal to the smallest positive integer d such that the matrix in (7) is rank deficient. In addition, $(p_0, \ldots, p_d)$ can be computed as a non-zero vector in the null-space of this matrix. Once $(p_0, \ldots, p_d)$ are available, the roots of

$$p(t) = p_0 t^0 + \ldots p_d t^d$$

are the poles $\{\tau_j\}$ inside $\mathbb{D}$.

The poles outside $\mathbb{D}$ uses the Fourier coefficients of the positive frequencies. From the integrals with $k \geq 1$, define the semi-infinite vector

$$\hat{g}_+ \equiv \begin{bmatrix} \hat{g}_1 \\ \hat{g}_2 \\ \vdots \end{bmatrix} \equiv \frac{1}{2\pi i} \int_{\partial\mathbb{D}} g(t) \begin{bmatrix} t^{-2} \\ t^{-3} \\ \vdots \end{bmatrix} \mathrm{d}t \equiv \begin{bmatrix} \sum_{|\tau_j|>1} w_j \tau_j^{-2} \\ \sum_{|\tau_j|>1} w_j \tau_j^{-3} \\ \vdots \end{bmatrix}$$

With the same shift operator S, for any τ_j with $|\tau_j| > 1$

$$S \begin{bmatrix} \tau_j^{-2} \\ \tau_j^{-3} \\ \vdots \end{bmatrix} = \begin{bmatrix} \tau_j^{-3} \\ \tau_j^{-4} \\ \vdots \end{bmatrix}, \quad \text{i.e.,} \quad (S - \tau_j^{-1}) \begin{bmatrix} \tau_j^{-2} \\ \tau_j^{-3} \\ \vdots \end{bmatrix} = 0.$$

Since the operators $S - \tau_j^{-1}$ all commute,

$$\prod_{|\tau_i|>1} (S - \tau_i^{-1}) \begin{bmatrix} \tau_j^{-2} \\ \tau_j^{-3} \\ \vdots \end{bmatrix} = 0. \tag{8}$$

Since $\hat{g}_+$ is a linear combination of the semi-infinite vectors in (8),

$$\prod_{|\tau_i|>1}(S-\tau_i^{-1})\hat{g}_+=0.$$

Introduce the polynomial $\prod_{|\tau_i|>1}(t-\tau_i^{-1})=p_0t^0+\cdots+p_dt^d$ with coefficients p_i, where the degree d is equal to the number of poles outside $\mathbb{D}$. Then (8) becomes

$$p_0(S^0\hat{g}_+)+\cdots+p_d(S^d\hat{g}_+)=0,\quad \text{i.e.,}\quad \begin{bmatrix}\hat{g}_1 & \hat{g}_2 & \cdots & \hat{g}_{d+1}\\ \hat{g}_2 & \hat{g}_3 & \cdots & \hat{g}_{d+2}\\ \vdots & \vdots & \vdots & \vdots\end{bmatrix}\begin{bmatrix}p_0\\ \cdots\\ p_d\end{bmatrix}=0. \tag{9}$$

This implies that the number of poles outside $\mathbb{D}$ is equal to the smallest value d such that the matrix in (9) is rank deficient. As before, $(p_0,\ldots,p_d)$ can be computed as a non-zero vector in the null-space of this matrix and the roots of

$$p(t)=p_0t^0+\ldots p_dt^d$$

are $\{\tau_j^{-1}\}$. Taking inverse of these roots gives the poles $\{\tau_j\}$ outside $\mathbb{D}$.

Step 3. Once the poles inside and outside $\mathbb{D}$ in the t plane are ready, we take the union and apply (3) to get the poles $\{\xi_1,\ldots,\xi_{N_p}\}$ in the z plane. With the poles located, solving the least square problem

$$\sum_{j=1}^{N_p}\frac{r_j}{\xi_j-z}\approx g(z)$$

computes the residues $\{r_j\}$.

2.2 Implementation

To implement this algorithm, we need to take care of several numerical issues.

- The semi-infinite matrix in (7) and (9). In the implementation, we pick a value $d_{\max}$ that is believed to be the upper bound of the number of poles and form the matrix

$$H=\begin{bmatrix}\hat{g}_{-1} & \hat{g}_{-2} & \cdots & \hat{g}_{-d_{\max}}\\ \hat{g}_{-2} & \hat{g}_{-3} & \cdots & \hat{g}_{-(d_{\max}+1)}\\ \vdots & \vdots & \vdots & \vdots\\ \hat{g}_{-l} & \hat{g}_{-(l+1)} & \cdots & \hat{g}_{-(d_{\max}+l-1)}\end{bmatrix}\quad\text{or}\quad H=\begin{bmatrix}\hat{g}_1 & \hat{g}_2 & \cdots & \hat{g}_{d_{\max}}\\ \hat{g}_2 & \hat{g}_3 & \cdots & \hat{g}_{(d_{\max}+1)}\\ \vdots & \vdots & \vdots & \vdots\\ \hat{g}_l & \hat{g}_{l+1} & \cdots & \hat{g}_{(d_{\max}+l-1)}\end{bmatrix}, \tag{10}$$

  respectively for (7) and (9), with l satisfying $l\geq d_{\max}$. We find that in practice $l=d_{\max}$ is enough.
- The numerical estimation of the rank d in (7) and (9). To address this, let $s_1,s_2,\ldots,s_{d_{\max}}$ be the singular values of the matrix H. The numerical rank is chosen to be the smallest d such that s_{d+1}/s_1 is below the noise level.
- The computation of the vector p. We first compute the singular value decomposition (SVD) of

$$\begin{bmatrix}\hat{g}_{-1} & \hat{g}_{-2} & \cdots & \hat{g}_{-(d+1)}\\ \hat{g}_{-2} & \hat{g}_{-3} & \cdots & \hat{g}_{-(d+2)}\\ \vdots & \vdots & \vdots & \vdots\\ \hat{g}_{-l} & \hat{g}_{-(l+1)} & \cdots & \hat{g}_{-(d+l)}\end{bmatrix}\quad\text{or}\quad\begin{bmatrix}\hat{g}_1 & \hat{g}_2 & \cdots & \hat{g}_{d+1}\\ \hat{g}_2 & \hat{g}_3 & \cdots & \hat{g}_{d+2}\\ \vdots & \vdots & \vdots & \vdots\\ \hat{g}_l & \hat{g}_{l+1} & \cdots & \hat{g}_{d+l}\end{bmatrix},$$

respectively for (7) and (9). p is then chosen to be the last column of the V matrix.

- The matrix H in (10) requires the Fourier transform $\hat{g}_k$ from $k = -(d_{\max} + l - 1)$ to $(d_{\max} + l - 1)$.
  - In the **random access** model, we choose an even $N_s \geq 2(d_{\max} + l)$ and define for $n = 0, \ldots, N_s - 1$

$$t_n = \exp\left(i\frac{2\pi n}{N_s}\right), \quad z_n = -\sqrt{ab}\frac{t_n + 1}{t_n - 1}. \tag{11}$$

    Using samples $\{g(t_n)\}$ at the points $\{t_n\}$ corresponds to approximating (4) with the trapezoidal rule. The trapezoidal rule is exponentially convergent for smooth functions when the step size $h = \frac{2\pi}{N_s}$ is sufficient small. In the current setting, this corresponds to

$$h \ll \sqrt{\frac{a}{b}}, \quad \text{i.e.,} \quad N_s \gg \sqrt{\frac{b}{a}}.$$

    Applying the fast Fourier transform to $\{g(t_n)\}$ gives $\{\hat{g}_k\}$ for $k = -\frac{N_s}{2}, \ldots, \frac{N_s}{2} - 1$. Among them, $\hat{g}_{-(d_{\max}+l-1)}, \ldots, \hat{g}_{(d_{\max}+l-1)}$ are used to form the H matrix in (10).
  - In the **Matsubara** model, $g(z)$ is only given at the Matsubara grid

$$z_n = \begin{cases} 2n\frac{\pi i}{\beta}, & \text{for bosons,} \\ (2n+1)\frac{\pi i}{\beta}, & \text{for fermions.} \end{cases}$$

    Computing the integral (5) is not convenient in the t space since the images $t_n = \frac{z_n - \sqrt{ab}}{z_n + \sqrt{ab}}$ are not uniformly distributed. Instead, applying (3) shows that, in the z variable, the integral is

$$\frac{1}{2\pi i}\int_{+i\infty}^{-i\infty} g(z)\left(\frac{z-\sqrt{ab}}{z+\sqrt{ab}}\right)^{-(k+1)} \frac{2\sqrt{ab}}{(z+\sqrt{ab})^2}\mathrm{d}z \approx$$
$$\frac{-1}{\beta}\sum_{n\in\mathbb{Z}} g(z_n)\left(\frac{z_n-\sqrt{ab}}{z_n+\sqrt{ab}}\right)^{-(k+1)} \frac{2\sqrt{ab}}{(z_n+\sqrt{ab})^2},$$

    where the last step uses the trapezoidal quadrature on the Matsubara grid. The trapezoidal rule is exponentially convergent in the regime $a \gg \pi/\beta$. Since the last sum is over all integers, it needs to be truncated between $-N_m$ and N_m for some integer N_m. Noticing that the terms in the sum decay only quadratically, N_m is typically chosen to be quite large for a good accuracy.

- The least square solve for $\{r_j\}$. Using the z_n points in (11), we solve the following system

$$r = \arg\min_{x\in\mathbb{C}^{N_p}} \frac{1}{2}\|Ax - b\|^2, \quad A = \left[\frac{1}{\xi_j - z_n}\right]_{n,j}, \quad b = \begin{bmatrix} g(z_1) \\ \cdots \\ g(z_{N_s}) \end{bmatrix},$$

  The entries of r are the residues $\{r_j\}$.

We would like to comment that the implementation decisions made above are by no means the only choice. We refer to the excellent papers [4, 17, 18] for other possible choices.

2.3 Matrix-valued Version

Let us comment on the matrix-valued version (2). The algorithm remains essentially the same. Below we list the differences.

- $\hat{G}_k$ is now the matrix-valued Fourier coefficients from the samples $G(t_n) \equiv G(z(t_n))$.
- The SVD is applied to the $lN_b^2 \times (d+1)$ matrices

$$\begin{bmatrix} \mathrm{cv}(\hat{G}_{-1}) & \mathrm{cv}(\hat{G}_{-2}) & \cdots & \mathrm{cv}(\hat{G}_{-(d+1)}) \\ \mathrm{cv}(\hat{G}_{-2}) & \mathrm{cv}(\hat{G}_{-3}) & \cdots & \mathrm{cv}(\hat{G}_{-(d+2)}) \\ \vdots & \vdots & \vdots & \vdots \\ \mathrm{cv}(\hat{G}_{-l}) & \mathrm{cv}(\hat{G}_{-(l+1)}) & \cdots & \mathrm{cv}(\hat{G}_{-(d+l)}) \end{bmatrix} \text{ or } \begin{bmatrix} \mathrm{cv}(\hat{G}_{1}) & \mathrm{cv}(\hat{G}_{2}) & \cdots & \mathrm{cv}(\hat{G}_{d+1}) \\ \mathrm{cv}(\hat{G}_{2}) & \mathrm{cv}(\hat{G}_{3}) & \cdots & \mathrm{cv}(\hat{G}_{d+2}) \\ \vdots & \vdots & \vdots & \vdots \\ \mathrm{cv}(\hat{G}_{l}) & \mathrm{cv}(\hat{G}_{l+1}) & \cdots & \mathrm{cv}(\hat{G}_{d+l}) \end{bmatrix},$$

  where $\mathrm{cv}(\cdot)$ turns a matrix into a column vector.
- The least square problem is applied to

$$R = \arg\min_{X \in \mathbb{C}^{N_p \times N_b^2}} \frac{1}{2}\|AX - B\|^2, \quad A = \left[\frac{1}{\xi_j - z_n}\right]_{n,j}, \quad B = \begin{bmatrix} \mathrm{rv}(G(z_1)) \\ \cdots \\ \mathrm{rv}(G(z_{N_s})) \end{bmatrix},$$

  where $\mathrm{rv}(\cdot)$ turns a matrix into a row vector. Each row of R is then reshaped back to the $N_b \times N_b$ matrix R_j. In the special case of $R_j = v_j v_j^*$, v_j can be further constructed by applying a rank-1 approximation to R_j.

2.4 Special Cases and Extensions

Below we include a few comments concerning special cases and direct extensions.

- We have assumed that the poles reside in the two disks in the z plane. In many applications, it is known that the poles are actually on the real axis. In such a case, the Fourier coefficients $\hat{g}_k$ and hence the matrix H are real. Therefore, a real SVD can be used while determining the rank d and the coefficients $(p_0, \ldots, p_d)$. Finally, the roots of $p(z)$ are also real. These considerations can significantly improve the stability as shown in Sect. 3.
- We have not specified any noise model. If the noise model is known a prior, it is possible to denoise the values $g(z_n)$ before applying the algorithm described. Such a denoising step can potentially improve the accuracy and stability of pole locations.
- The algorithm can also be extended to the general setting, where the imaginary axis $i\mathbb{R}$ is replaced with any simple curve on the Riemann sphere. If the curve is smooth, the extension is straightforward as the trapezoidal quadrature can still be applied. When the curve is non-smooth, a special quadrature is needed for good accuracy.

3 Numerical Results

This section presents a few numerical examples. In all examples, $a = 1, b = 100$. The noise added to $g(z)$ is multiplicative:

$$g_{\text{noisy}} = g_{\text{exact}} \cdot (1 + \sigma N_{\mathbb{C}}(0, 1)).$$

This is a reasonable model since in many applications the magnitude of the noise is often proportional to the magnitude of the signal. For each example, we present the numerical results for both the random access model and the Matsubara model. For the random access model, $N_s = 1024$. For the Matsubara model, $N_m = 10^6$ and $\beta = 10\pi$.

Example 1 We first consider the case of complex pole locations. Within each circle, we place four poles and the residues $\{r_j\}$ are of unit order. Figure 3 plots the results at the noise level $\sigma = 0$, 10^{-6}, 10^{-5}, and 10^{-4}, where the left and right columns are for the random access and Matsubara models, respectively. The results show that

- At $\sigma = 0$, the algorithm gives perfect reconstruction at machine accuracy.
- At $\sigma = 10^{-6}$, the poles are accurately identified.
- At $\sigma = 10^{-5}$, the number of poles are correctly recovered, though the locations of the two poles far from $i\mathbb{R}$ are wrong.
- At $\sigma = 10^{-4}$, only the six poles close to $i\mathbb{R}$ are identified.

Example 2 Next we consider the case of real pole locations. Within each circle, there are 4 poles and the residues $\{r_j\}$ are again of unit order. Figure 4 summarizes the results at the noise level $\sigma = 0$, 10^{-5}, 10^{-4}, and 10^{-3}.

- At $\sigma = 0$, the algorithm gives perfect reconstruction.
- At $\sigma = 10^{-5}$, the poles are also recovered perfectly.
- At $\sigma = 10^{-4}$, the pole locations are recovered accurately, though with some errors for the two poles farthest away from $i\mathbb{R}$.
- At $\sigma = 10^{-3}$, only the six poles close to $i\mathbb{R}$ are identified.

A comparison with the previous example suggests that enforcing the real constraints significantly improves the stability and accuracy of the algorithm.

Example 3 Finally, we consider the matrix-valued version. The dimension N_b of the matrix R_j is set to be $N_b = 4$. When other parameters are fixed, larger values of N_b significantly improve the accuracy since each matrix entry effectively provides a new data point. With more data, the effective noise level goes down significantly.

Within each circle, there are again 4 poles and the residues $\{v_j\}$ (and equivalently $\{R_j\}$) are of unit order. Figure 5 summarizes the results at the noise level $\sigma = 0$, 10^{-4}, 10^{-3}, and 10^{-2}.

- At $\sigma = 0$, the algorithm again gives perfect reconstruction.
- At $\sigma = 10^{-4}$, the reconstruction is near perfect.
- At $\sigma = 10^{-3}$, the pole locations are recovered with good accuracy, though there are some errors for the two poles away from $i\mathbb{R}$.
- At $\sigma = 10^{-2}$, only the six poles close to $i\mathbb{R}$ are identified.

Noticing that the noise level in this example is much higher than the ones used in the previous examples, the results confirm that the matrix-valued version is easier, especially when N_b is large.

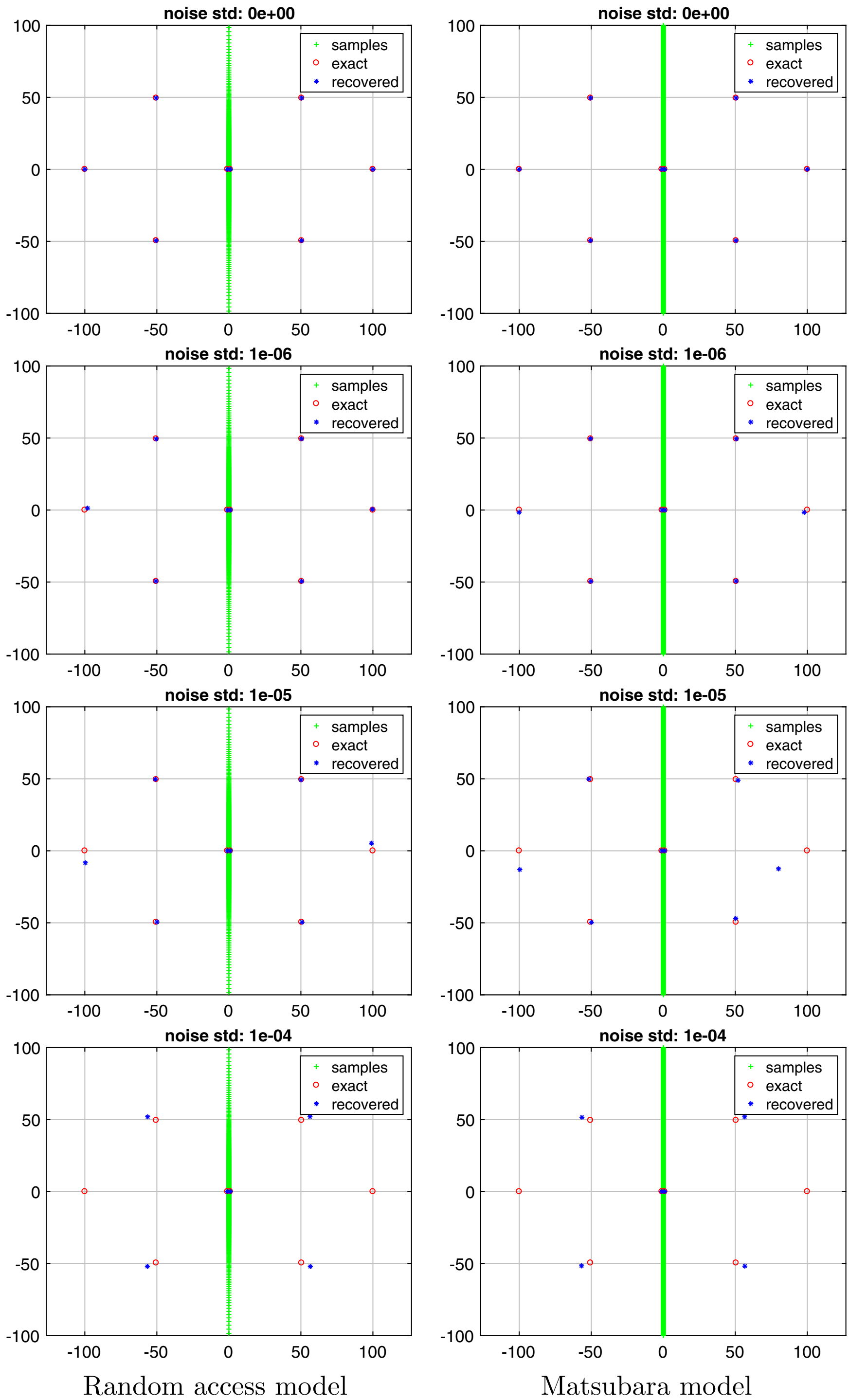


Fig. 3 Complex pole locations, with different levels of noise. Left: random access model. Right: Matsubara model

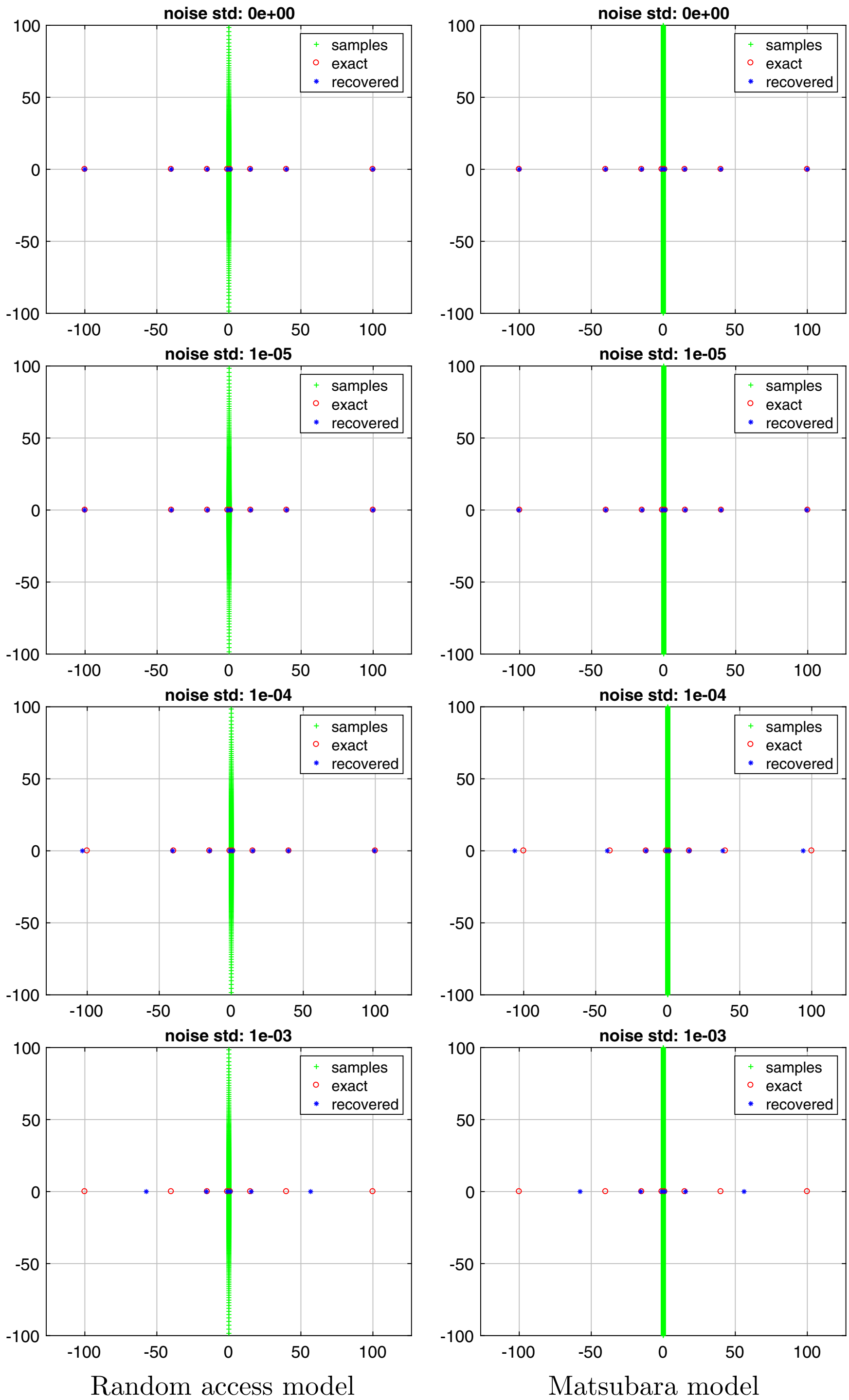


Fig. 4 Real pole locations, with different levels of noise. Left: random access model. Right: Matsubara model

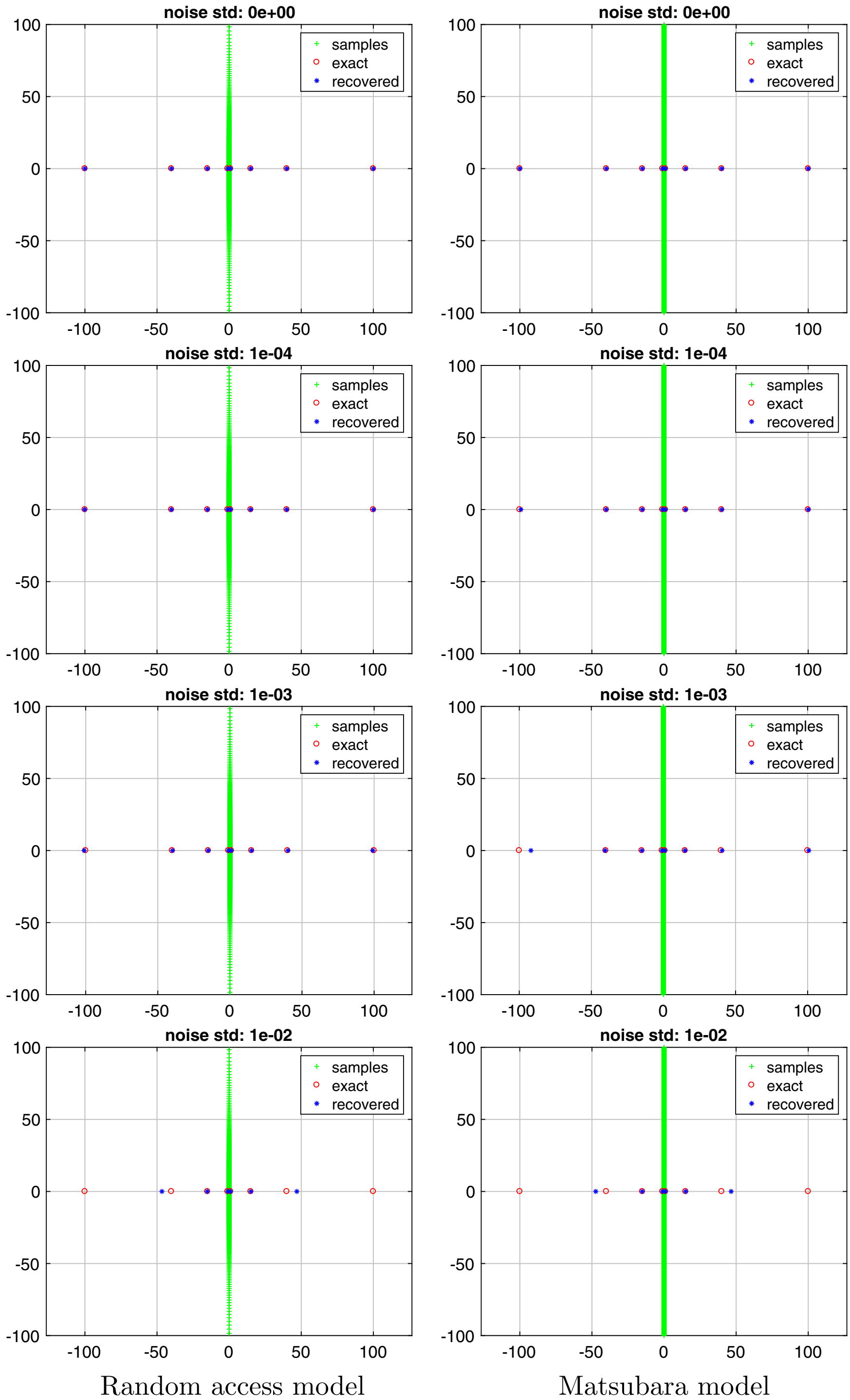


Fig. 5 Matrix case with real poles, with different levels of noise. Left: random access model. Right: Matsubara model

Acknowledgements The work is partially supported by National Science Foundation under award DMS-2011699. We thank Anil Damle, Zhen Huang, and Lin Lin for discussions. We also thank the reviewers for comments and suggestions.

Funding The authors have not disclosed any funding.

Data Availability Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Ankur Moitra, Super-resolution, extremal functions and the condition number of vandermonde matrices. In: proceedings of the forty-seventh annual ACM Symposium on Theory of Computing, pp. 821–830 (2015)
2. Berljafa, Mario, Guttel, Stefan: The RKFIT algorithm for nonlinear rational approximation. SIAM J. Sci. Comput. **39**(5), A2049–A2071 (2017)
3. Berrut, Jean-Paul., Trefethen, Lloyd N.: Barycentric Lagrange interpolation. SIAM Rev. **46**(3), 501–517 (2004)
4. Beylkin, Gregory, Monzón, Lucas: On approximation of functions by exponential sums. Appl. Comput. Harmon. Anal. **19**(1), 17–48 (2005)
5. Beylkin, Gregory, Monzón, Lucas: Nonlinear inversion of a band-limited Fourier transform. Appl. Comput. Harmon. Anal. **27**(3), 351–366 (2009)
6. Bruus, Henrik, Flensberg, Karsten: Many-body quantum theory in condensed matter physics: an introduction. OUP Oxford (2004)
7. Derevianko, Nadiia, Plonka, Gerlind: Exact reconstruction of extended exponential sums using rational approximation of their fourier coefficients. Anal. Appl. **20**(03), 543–577 (2022)
8. Fei, Jiani, Yeh, Chia-Nan., Gull, Emanuel: Nevanlinna analytical continuation. Phys. Rev. Lett. **126**(5), 056402 (2021)
9. Fei, Jiani, Yeh, Chia-Nan., Zgid, Dominika, Gull, Emanuel: Analytical continuation of matrix-valued functions: Carathéodory formalism. Phys. Rev. B **104**(16), 165111 (2021)
10. Gerlind Plonka and Vlada Pototskaia, Application of the aak theory for sparse approximation of exponential sums, arXiv preprint arXiv:1609.09603 (2016)
11. Gonnet, Pedro, Guttel, Stefan, Trefethen, Lloyd N.: Robust Padé approximation via svd. SIAM Rev. **55**(1), 101–117 (2013)
12. Gustavsen, Bjorn, Semlyen, Adam: Rational approximation of frequency domain responses by vector fitting. IEEE Trans. Power Delivery **14**(3), 1052–1061 (1999)
13. Heather Wilber, Anil Damle, and Alex Townsend, Data-driven algorithms for signal processing with rational functions, arXiv preprint arXiv:2105.07324 (2021)
14. Lexing Ying, Analytic continuation from limited noisy matsubara data, arXiv preprint arXiv:2202.09719 (2022)
15. Mejuto-Zaera, Carlos, Zepeda-Núñez, Leonardo, Lindsey, Michael, Tubman, Norm, Whaley, Birgitta, Lin, Lin: Efficient hybridization fitting for dynamical mean-field theory via semi-definite relaxation. Phys. Rev. B **101**(3), 035143 (2020)
16. Nakatsukasa, Yuji, Sète, Olivier, Trefethen, Lloyd N.: The AAA algorithm for rational approximation. SIAM J. Sci. Comput. **40**(3), A1494–A1522 (2018)
17. Potts, Daniel, Tasche, Manfred: Parameter estimation for exponential sums by approximate prony method. Signal Process. **90**(5), 1631–1642 (2010)
18. Potts, Daniel, Tasche, Manfred: Parameter estimation for nonincreasing exponential sums by Prony-like methods. Linear Algebra Appl. **439**(4), 1024–1039 (2013)
19. Prony, R.: Essai experimental et analytique. J. Ecole Polytechnique **1**, 24–76 (1795)