Applied and Computational Harmonic Analysis 74 (2025) 101725

Contents lists available at ScienceDirect

Applied and Computational Harmonic Analysis

journal homepage: www.elsevier.com/locate/acha

Letter to the Editor

# Multidimensional unstructured sparse recovery via eigenmatrix ☆

Lexing Ying

*Department of Mathematics, Stanford University, Stanford, CA 94305, United States of America*

ARTICLE INFO

Communicated by Radu Balan

*MSC:*
30B40
65R32

*Keywords:*
Sparse recovery
ESPRIT
Eigenmatrix

ABSTRACT

This note considers the multidimensional unstructured sparse recovery problems. Examples include Fourier inversion and sparse deconvolution. The eigenmatrix is a data-driven construction with desired approximate eigenvalues and eigenvectors proposed for the one-dimensional problems. This note extends the eigenmatrix approach to multidimensional problems, providing a rather unified treatment for general kernels and unstructured sampling grids in both real and complex settings. Numerical results are provided to demonstrate the performance of the proposed method.

## 1. Introduction

This note considers the multidimensional unstructured sparse recovery problems of a general form. Let $X$ be the parameter space, typically a subset of $\mathbb{R}^d$ or $\mathbb{C}^d$, and $S$ be the sampling space. $G(s,x)$ is a kernel function defined for $s \in S$ and $x \in X$, and is assumed to be analytic in $x$. Suppose that

$$f(x) = \sum_{k=1}^{n_x} w_k \delta(x - x_k) \tag{1}$$

is the unknown multidimensional sparse signal, where $\{x_k\}_{1\le k\le n_x}$ are the spike locations and $\{w_k\}_{1\le k\le n_x}$ are the spike weights. The observed data of the problem is

$$u(s) := \int_X G(s,x) f(x) dx = \sum_{k=1}^{n_x} G(s,x_k) w_k \tag{2}$$

for $s \in S$.

Let $\{s_j\}$ be a set of $n_s$ unstructured samples in $S$ and $u_j := u(s_j)$ be the exact values. Suppose that we are only given the noisy observations $\tilde{u}_j := u_j(1+\sigma Z_j)$, where $Z_j$ are independently identically distributed (i.i.d.) random variables with zero mean and unit variance, and $\sigma$ is the noise magnitude. The task is to recover the spikes $\{x_k\}$ and weights $\{w_k\}$ from $\{\tilde{u}_j\}$.

This note focuses on the real case $X \subset \mathbb{R}^d$ since most multidimensional applications fit into this setting. Two important examples are listed as follows.

☆ This work is partially supported by NSF grants DMS-2011699 and DMS-2208163.

*E-mail address:* lexing@stanford.edu.

https://doi.org/10.1016/j.acha.2024.101725
Received 4 April 2024; Received in revised form 5 November 2024; Accepted 11 November 2024

Available online 19 November 2024

- Sparse deconvolution. For example, $G(s,x) = \frac{1}{\|s-x\|^\alpha}$ or $e^{-\|s-x\|}$, $X$ is a bounded domain, and $\{s_j\}$ are samples outside $X$ in $\mathbb{R}^d$.
- Fourier inversion. For example $G(s,x) = \exp(\pi i s \cdot x)$, $X = [-1,1]^d$ for example, and $\{s_j\}$ are samples in $\mathbb{R}^d$.

One example from the two-dimensional complex setting is

- Rational approximation. For example, $G(s,x) = \frac{1}{(s_1-x_1)(s_2-x_2)}$, $X = \mathbb{D}_1 \times \mathbb{D}_1$, and $S \subset (\mathbb{C} \setminus \mathbb{D}_1) \times (\mathbb{C} \setminus \mathbb{D}_1)$, where $\mathbb{D}_1$ is the unit disc in the complex plane.

The primary challenges of the current setup come from three sources. First, the kernel $G(s,x)$ can be quite general. Second, the samples $\{s_j\}$ are unstructured, which excludes the existing algorithms that exploit Cartesian sampling structure. Third, the sample values $\{\tilde{u}_j\}$ are noisy, which raises stability issues when the recovery problem is quite ill-posed.

### 1.1. *Contribution*

The recent paper [18] introduces the eigenmatrix for one-dimensional unstructured sparse recovery problems. This is a data-driven object that depends on $G(\cdot,\cdot)$, $X$, and the samples $\{s_j\}$. Constructed to have desired eigenvalues and eigenvectors, it turns the unstructured sparse recovery problem into an algebraic problem (such as a rootfinding or eigenvalue problem). The main features of the eigenmatrix are

- It assumes no special structure of the samples $\{s_j\}$.
- It offers a rather unified approach to these sparse recovery problems.

This note extends this approach to the multidimensional cases. We provide a rather unified treatment for general kernels and unstructured sampling grids in both real and complex settings.

### 1.2. *Related work*

There has been a long list of works devoted to the sparse recovery problems mentioned above. We refer to the paper [18] for the related work on the one-dimensional case. For the multidimensional Fourier inversion or superresolution, one well-studied approach is based on convex relaxation or $\ell_1$-minimization; see for example [8] and the references therein.

When the kernel takes the exponential form and the sampling grid is multidimensional Cartesian, there is a long list of works for Cartesian samples based on Prony type methods [1–3,5–7,9,11,13–15,17]. However, the topic is rather unexplored when either the kernel takes a general form or the sampling grid is unstructured. A closely related problem is the sparse moment problem; see, for example, [4] for a closely related work for the 2D case.

The rest of the note is organized as follows. Section 2 reviews the eigenmatrix for one-dimensional problems in the setting of the ESPRIT algorithm. Section 3 describes the extension to multidimensional problems. Section 4 presents several numerical experiments. Section 5 concludes with a discussion for future work.

## 2. Eigenmatrix with ESPRIT

This section provides a short review of the eigenmatrix approach. It was proposed in [18] for both Prony's method [10] and the ESPRIT algorithm [12]. The presentation here focuses on ESPRIT as it extends to multidimensional problems. On the other hand, it is not clear how to extend Prony's method since it depends on the characteristic polynomial, which is essentially a one-dimensional concept.

**Complex analytic case.** To simplify the discussion, assume that $X$ is the unit disc $\mathbb{D} \subset \mathbb{C}$. Define for each $x$ the column vector

$$\mathbf{g}(x) := \begin{bmatrix} G(s_1,x) \\ \vdots \\ G(s_{n_s},x) \end{bmatrix}.$$

The first step is to construct $M$ such that $M\mathbf{g}(x) \approx x\mathbf{g}(x)$ for $x \in \mathbb{D}$. Numerically, it is more robust to use the normalized vector $\hat{\mathbf{g}}(x) = \mathbf{g}(x)/\|\mathbf{g}(x)\|$ since the norm of $\mathbf{g}(x)$ can vary significantly depending on $x$. The condition then becomes

$$M\hat{\mathbf{g}}(x) \approx x\hat{\mathbf{g}}(x), \quad x \in \mathbb{D}.$$

We enforce this condition on a uniform grid $\{a_t\}_{1\le t\le n_a}$ of size $n_a$ on the boundary of the unit disk

$$M\hat{\mathbf{g}}(a_t) \approx a_t\hat{\mathbf{g}}(a_t).$$

Define the $n_s \times n_a$ matrix

$$\hat{G} = \begin{bmatrix} \hat{\mathbf{g}}(a_1) & \dots & \hat{\mathbf{g}}(a_{n_a}) \end{bmatrix}$$

with $\hat{\mathbf{g}}(a_t)$ as columns and also the $n_a \times n_a$ diagonal matrix $\Lambda = \mathrm{diag}(a_t)$. The previous condition can be written in a matrix form as

$$M\hat{G} \approx \hat{G}\Lambda.$$

$n_a$ is chosen so that the columns of $\hat{G}$ are numerically linearly independent (in practice, the condition number of $\hat{G}$ is bounded below $10^7$). When the columns of $\hat{G}$ are numerically linearly independent, we define the eigenmatrix as

$$M := \hat{G}\Lambda\hat{G}^+, \tag{3}$$

where the pseudoinverse $\hat{G}^+$ is computed by thresholding the singular values of $\hat{G}$. In practice, the thresholding value is chosen so that a small constant bounds the norm of $M$.

**Real analytic case.** To simplify the discussion, assume that $X$ is the interval $[-1,1]$. For each $x$, again $\mathbf{g}(x) = [G(s_j,x)]_{1\le j\le n_s}$. The first step is to construct $M$ such that $M\mathbf{g}(x) \approx x\mathbf{g}(x)$ for $x \in [-1,1]$. Moving to the normalized vector $\hat{\mathbf{g}}(x) = \mathbf{g}(x)/\|\mathbf{g}(x)\|$, we instead aim for

$$M\hat{\mathbf{g}}(x) \approx x\hat{\mathbf{g}}(x), \quad x \in [-1,1].$$

This condition is enforced on a Chebyshev grid [16] $\{a_t\}_{1\le t\le n_a}$ of size $n_a$ on the interval $[-1,1]$

$$M\hat{\mathbf{g}}(a_t) \approx a_t\hat{\mathbf{g}}(a_t).$$

Introduce the $n_s \times n_a$ matrix $\hat{G} = [\hat{\mathbf{g}}(a_t)]_{1\le t\le n_a}$ with columns $\hat{\mathbf{g}}(a_t)$ as well as the $n_a \times n_a$ diagonal matrix $\Lambda = \mathrm{diag}(a_t)$. The condition now reads

$$M\hat{G} \approx \hat{G}\Lambda.$$

When the columns of $\hat{G}$ are numerically linearly independent, we define the eigenmatrix for the real analytic case as

$$M := \hat{G}\Lambda\hat{G}^+,$$

where the pseudoinverse $\hat{G}^+$ is computed by thresholding the singular values of $\hat{G}$.

**Combined with ESPRIT.** Define the vector

$$\tilde{\mathbf{u}} = \begin{bmatrix} \tilde{u}_1 \\ \vdots \\ \tilde{u}_{n_s} \end{bmatrix},$$

where $\tilde{u}_j$ are the noisy observations. Consider the matrix

$$\begin{bmatrix} \tilde{\mathbf{u}} & M\tilde{\mathbf{u}} & \dots & M^\ell\tilde{\mathbf{u}} \end{bmatrix}$$

with $\ell > n_x$, obtained from applying $M$ repetitively. Since $\tilde{\mathbf{u}} \approx \sum_k \mathbf{g}(x_k)w_k$ and $M\mathbf{g}(x) \approx x\mathbf{g}(x)$,

$$\begin{bmatrix} \tilde{\mathbf{u}} & M\tilde{\mathbf{u}} & \dots & M^\ell\tilde{\mathbf{u}} \end{bmatrix} \approx \begin{bmatrix} \mathbf{g}(x_1) & \dots & \mathbf{g}(x_{n_x}) \end{bmatrix} \begin{bmatrix} w_1 & & \\ & \ddots & \\ & & w_{n_x} \end{bmatrix} \begin{bmatrix} 1 & x_1 & \dots & (x_1)^\ell \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n_x} & \dots & (x_{n_x})^\ell \end{bmatrix}.$$

Let $\tilde{U}\tilde{S}\tilde{V}^*$ be the rank-$n_x$ truncated SVD of this left-hand side. The matrix $\tilde{V}^*$ satisfies

$$\tilde{V}^* \approx P \begin{bmatrix} 1 & x_1 & \dots & (x_1)^\ell \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n_x} & \dots & (x_{n_x})^\ell \end{bmatrix},$$

where $P$ is an unknown non-degenerate $n_x \times n_x$ matrix. Let $\tilde{Z}_U$ and $\tilde{Z}_D$ be the submatrices obtained by excluding the first column and the last column, respectively, i.e.,

$$\tilde{Z}_U \approx P \begin{bmatrix} 1 & \dots & (x_1)^{\ell-1} \\ \vdots & \ddots & \vdots \\ 1 & \dots & (x_{n_x})^{\ell-1} \end{bmatrix}, \quad \tilde{Z}_D \approx P \begin{bmatrix} x_1 & \dots & (x_1)^\ell \\ \vdots & \ddots & \vdots \\ x_{n_x} & \dots & (x_{n_x})^\ell \end{bmatrix}.$$

By forming $\tilde{Z}_D(\tilde{Z}_U)^+$ and noticing

$$\tilde{Z}_D(\tilde{Z}_U)^+ \approx P \begin{bmatrix} x_1 & & \\ & \ddots & \\ & & x_{n_x} \end{bmatrix} P^{-1},$$

one obtains the estimates $\{\tilde{x}_k\}$ for $\{x_k\}$ by computing the eigenvalues of $\tilde{Z}_D(\tilde{Z}_U)^+$.

With $\{\tilde{x}_k\}$ available, the least square solution of

$$\min_{\tilde{w}_k} \sum_j \left| \sum_k G(s_j, \tilde{x}_k)\tilde{w}_k - \tilde{u}_j \right|^2$$

gives the estimators $\{\tilde{w}_k\}$ for $\{w_k\}$.

## 3. Multidimensional case

This section extends the eigenmatrix approach to the multidimensional case. We start with the 2D real case and then move on to the higher dimensions. The main reason is that the 2D real case can be addressed with complex variables.

### *3.1. 2D real case*

The key idea is to reduce the 2D real case to the 1D complex case (see for example [4]). To simplify the discussion, assume first that $X$ is $[-1,1]^2$. Define for each $x = (x^1, x^2)$

$$\mathbf{g}(x) := \begin{bmatrix} G(s_1, x) \\ \vdots \\ G(s_{n_s}, x) \end{bmatrix} \quad \text{and} \quad \gamma(x) := x^1 + i x^2 \in \mathbb{C}.$$

The first step is to construct $M$ such that $M\mathbf{g}(x) \approx \gamma(x)\mathbf{g}(x)$ for $x \in [-1,1]^2$. Numerically, it is more robust to use the normalized vector $\hat{\mathbf{g}}(x) = \mathbf{g}(x)/\|\mathbf{g}(x)\|$. The condition then becomes

$$M\hat{\mathbf{g}}(x) \approx \gamma(x)\hat{\mathbf{g}}(x), \quad x \in [-1,1]^2.$$

We enforce this condition on a two-dimensional Chebyshev grid $\{a_t\}_{1\le t\le n_a}$ of size $n_a$ (i.e., with $\sqrt{n_a}$ points in each dimension) on the square,

$$M\hat{\mathbf{g}}(a_t) \approx \gamma(a_t)\hat{\mathbf{g}}(a_t).$$

Define the $n_s \times n_a$ matrix $\hat{G} = [\hat{\mathbf{g}}(a_t)]_{1\le t\le n_a}$ with $\hat{\mathbf{g}}(a_t)$ as columns and also the $n_a \times n_a$ diagonal matrix $\Lambda = \mathrm{diag}(\gamma(a_t))$. The previous condition can be written as

$$M\hat{G} \approx \hat{G}\Lambda.$$

Assuming that the columns of $\hat{G}$ are numerically linearly independent, we define

$$M := \hat{G}\Lambda\hat{G}^+,$$

where the pseudoinverse $\hat{G}^+$ is computed by thresholding the singular values of $\hat{G}$.

**Remark 1.** We claim that, for real analytic kernels $G(s,x)$, enforcing the condition at the Chebyshev grid $\{a_t\}$ is sufficient. To see this,

$$M\mathbf{g}(x) \approx M\left(\sum_t c_t(x)\mathbf{g}(a_t)\right) = \sum_t c_t(x)M\mathbf{g}(a_t) \approx \sum_t c_t(x)(\gamma(a_t)\mathbf{g}(a_t)) \approx \gamma(x)\mathbf{g}(x),$$

where $c_t(x)$ is the Chebyshev quadrature for $x$ associated with grid $\{a_t\}$. Here, the first and third approximations use the convergence property of the Chebyshev quadrature for the analytic functions $\mathbf{g}(x)$ and $\gamma(x)\mathbf{g}(x)$, and the second approximation directly comes from $M\mathbf{g}(a_t) \approx \gamma(a_t)\mathbf{g}(a_t)$.

Let $\tilde{\mathbf{u}} = [\tilde{u}_j]_{1\le j\le n_s}$ and form the following matrix by applying $M$ repetitively to $\tilde{\mathbf{u}}$

$$\begin{bmatrix} \tilde{\mathbf{u}} & M\tilde{\mathbf{u}} & \dots & M^\ell\tilde{\mathbf{u}} \end{bmatrix} \approx \begin{bmatrix} \mathbf{g}(x_1) & \dots & \mathbf{g}(x_{n_x}) \end{bmatrix} \begin{bmatrix} w_1 & & \\ & \ddots & \\ & & w_{n_x} \end{bmatrix} \begin{bmatrix} 1 & \gamma(x_1) & \dots & \gamma(x_1)^\ell \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \gamma(x_{n_x}) & \dots & \gamma(x_{n_x})^\ell \end{bmatrix}$$

with $\ell > n_x$. From the SVD, one can get estimates for $\{\gamma(x_k)\}$. Their real and imaginary parts give the approximate spikes $\{\tilde{x}_k\}$. Finally, the least square solve

$$\min_{\tilde{w}_k} \sum_j \left| \sum_k G(s_j, \tilde{x}_k)\tilde{w}_k - \tilde{u}_j \right|^2$$

gives the estimators $\{\tilde{w}_k\}$ for $\{w_k\}$.

**Remark 2.** Having the columns of $\hat{G}$ to be numerically linearly independent is essential. For example, the method fails on the kernel $G(s,x) = \ln|s-x|$, which is Green's function of the Laplace equation, as it violates the linear independence due to the mean value theorem. The following argument explains why. Suppose for now that for any $x \in [-1,1]^2$, $M\mathbf{g}(x) \approx \gamma(x)\mathbf{g}(x)$. Since $G(s,x) = \ln|s-x|$ satisfies the Laplace equation in $x$, the mean value theorem states that

$$\mathbf{g}(x) = ⨍ \mathbf{g}(b)db,$$

where the averaged integral is taken over a finite circle centered at $x$. Applying $M$ to both sides leads to

$$\gamma(x)\mathbf{g}(x) \approx M\mathbf{g}(x) = ⨍ M\mathbf{g}(b)db \approx ⨍ \gamma(b)\mathbf{g}(b)db. \tag{4}$$

However, the real and imaginary parts of $\gamma(x)\mathbf{g}(x)$, i.e., $x^1\mathbf{g}(x)$ and $x^2\mathbf{g}(x)$, are not harmonic functions in $x$. Therefore, (4) cannot hold for $G(s,x) = \ln|s-x|$ without introducing a finite error. Therefore, it is not possible to construct $M$ such that $M\mathbf{g}(x) \approx \gamma(x)\mathbf{g}(x)$ holds for all $x \in [-1,1]^2$. This argument also applies to any other kernel $G(s,x)$ that comes from the Green's function of a linear elliptic PDE.

### 3.2. 3D real case

To simplify the discussion, assume that $X$ is $[-1,1]^3$. Define for each $x$ the vector $\mathbf{g}(x) := [G(s_j,x)]_{1\le j\le n_s}$. The first step is to construct three eigenmatrices $M^1, M^2, M^3$ such that for $x = (x^1,x^2,x^3) \in [-1,1]^3$.

$$M^1\mathbf{g}(x) \approx x^1\mathbf{g}(x), \quad M^2\mathbf{g}(x) \approx x^2\mathbf{g}(x), \quad M^3\mathbf{g}(x) \approx x^3\mathbf{g}(x). \tag{5}$$

Introducing a product Chebyshev grid $\{a_t = (a_t^1, a_t^2, a_t^3)\}_{1\le t\le n_a}$ of size $n_a$, i.e., with ${n_a}^{1/3}$ points per dimension, we enforce the condition on this Chebyshev grid

$$M^1\hat{\mathbf{g}}(a_t) \approx a_t^1\hat{\mathbf{g}}(a_t), \quad M^2\hat{\mathbf{g}}(a_t) \approx a_t^2\hat{\mathbf{g}}(a_t), \quad M^3\hat{\mathbf{g}}(a_t) \approx a_t^3\hat{\mathbf{g}}(a_t). \tag{6}$$

Define the $n_s \times n_a$ matrix $\hat{G} = [\hat{\mathbf{g}}(a_t)]_{1\le t\le n_a}$ with $\hat{\mathbf{g}}(a_t)$ as columns and also $n_a \times n_a$ diagonal matrices $\Lambda^1 = \mathrm{diag}(a_t^1)$, $\Lambda^2 = \mathrm{diag}(a_t^2)$, and $\Lambda^3 = \mathrm{diag}(a_t^3)$. The previous condition can be written in a matrix form as

$$M^1\hat{G} \approx \hat{G}\Lambda^1, \quad M^2\hat{G} \approx \hat{G}\Lambda^2, \quad M^3\hat{G} \approx \hat{G}\Lambda^3.$$

When the columns of $\hat{G}$ are numerically linearly independent, this suggests the following choice of the eigenmatrix,

$$M^1 := \hat{G}\Lambda^1\hat{G}^+, \quad M^2 := \hat{G}\Lambda^2\hat{G}^+, \quad M^3 := \hat{G}\Lambda^3\hat{G}^+,$$

where the pseudoinverse $\hat{G}^+$ is computed by thresholding the singular values of $\hat{G}$.

**Remark 3.** The matrices $M^1$, $M^2$, and $M^3$ approximately commute since they share the same set of approximate eigenvalues and eigenvectors. Therefore, any product formed from them is approximately independent of the order.

**Remark 4.** Enforcing the condition at the Chebyshev grid $\{a_t\}$ is again sufficient, i.e., $M^1\mathbf{g}(a_t) \approx a_t^1\mathbf{g}(a_t)$ implies $M^1\mathbf{g}(x) \approx x^1\mathbf{g}(x)$ for all $x \in [-1,1]^3$.

$$M^1\mathbf{g}(x) \approx M^1\left(\sum_t c_t(x)\mathbf{g}(a_t)\right) = \sum_t c_t(x)M^1\mathbf{g}(a_t) \approx \sum_t c_t(x)(a_t^1\mathbf{g}(a_t)) \approx x^1\mathbf{g}(x),$$

where $c_t(x)$ is the Chebyshev quadrature for $x$ associated with grid $\{a_t\}$. Here, the first and third approximations use the convergence property of the Chebyshev quadrature for analytic functions $\mathbf{g}(x)$ and $x^1\mathbf{g}(x)$, and the second approximation directly comes from $M^1\mathbf{g}(a_t) \approx a_t^1\mathbf{g}(a_t)$. The same derivation holds for $M^2$ and $M^3$.

With the eigenmatrices $M^1$, $M^2$, and $M^3$ available, the rest of the algorithm follows the approach in [1]. For any triple $\alpha = (\alpha^1, \alpha^2, \alpha^3)$, define

$$(x)^\alpha := (x^1)^{\alpha^1}(x^2)^{\alpha^2}(x^3)^{\alpha^3}, \quad M^\alpha := (M^1)^{\alpha^1}(M^2)^{\alpha^2}(M^3)^{\alpha^3}.$$

Since $M^1$, $M^2$, and $M^3$ approximately commute, the definition of $M^\alpha$ is approximately independent of the order. Consider the matrix

$$\begin{bmatrix}\tilde{\mathbf{u}} & \dots & M^\alpha\tilde{\mathbf{u}} & \dots & M^{(\ell,\ell,\ell)}\tilde{\mathbf{u}}\end{bmatrix} \tag{7}$$

with $\alpha = (\alpha^1, \alpha^2, \alpha^3)$ satisfies $0 \le \alpha^1, \alpha^2, \alpha^3 \le \ell$. The $(\ell+1)^3$ columns are ordered in the lexicographical order. We have the following approximation

$$\begin{aligned}&\begin{bmatrix}\tilde{\mathbf{u}} & \dots & M^{\alpha}\tilde{\mathbf{u}} & \dots & M^{(\ell,\ell,\ell)}\tilde{\mathbf{u}}\end{bmatrix}\\ &\approx \begin{bmatrix}\mathbf{g}(x_1) & \dots & \mathbf{g}(x_{n_x})\end{bmatrix}\begin{bmatrix}w_1 & & \\ & \ddots & \\ & & w_{n_x}\end{bmatrix}\begin{bmatrix}1 & \dots & (x_1)^{\alpha} & \dots & (x_1)^{(\ell,\ell,\ell)}\\ \vdots & \ddots & \vdots & \ddots & \vdots\\ 1 & \dots & (x_{n_x})^{\alpha} & \dots & (x_{n_x})^{(\ell,\ell,\ell)}\end{bmatrix}.\end{aligned}$$

Notice that one never needs to form the matrices $M^{\alpha}$ explicitly as the columns $M^{\alpha}\tilde{\mathbf{u}}$ can be computed effectively by progressively applying only the matrices $M^1$, $M^2$, and $M^3$.

Let $\tilde{U}\tilde{S}\tilde{V}^*$ be the rank-$n_x$ truncated SVD of this matrix. The matrix $\tilde{V}^*$ satisfies

$$\tilde{V}^* \approx P\begin{bmatrix}1 & \dots & (x_1)^{\alpha} & \dots & (x_1)^{(\ell,\ell,\ell)}\\ \vdots & \ddots & \vdots & \ddots & \vdots\\ 1 & \dots & (x_{n_x})^{\alpha} & \dots & (x_{n_x})^{(\ell,\ell,\ell)}\end{bmatrix},$$

where $P$ is an unknown non-degenerate $n_x \times n_x$ matrix.

For the first dimension, let $\tilde{Z}^1_U$ be the submatrix that excludes the columns $\alpha$ with $\alpha^1 = \ell$ and $\tilde{Z}^1_D$ be the submatrix that excludes the columns $\alpha$ with $\alpha^1 = 0$. Then

$$\tilde{Z}^1_U \approx P\begin{bmatrix}1 & \dots & (x_1)^{(\ell-1,\ell,\ell)}\\ \vdots & \ddots & \vdots\\ 1 & \dots & (x_{n_x})^{(\ell-1,\ell,\ell)}\end{bmatrix}, \quad \tilde{Z}^1_D \approx P\begin{bmatrix}(x_1)^{(1,0,0)} & \dots & (x_1)^{(\ell,\ell,\ell)}\\ \vdots & \ddots & \vdots\\ (x_{n_x})^{(1,0,0)} & \dots & (x_{n_x})^{(\ell,\ell,\ell)}\end{bmatrix}. \tag{8}$$

Introduce

$$B^1 := \tilde{Z}^1_D(\tilde{Z}^1_U)^+ \approx P\begin{bmatrix}x^1_1 & & \\ & \ddots & \\ & & x^1_{n_x}\end{bmatrix}P^{-1}. \tag{9}$$

For the second dimension, let $\tilde{Z}^2_U$ be the submatrix that excludes the columns $\alpha$ with $\alpha^2 = \ell$ and $\tilde{Z}^2_D$ be the submatrix that excludes the columns $\alpha$ with $\alpha^2 = 0$. Then

$$\tilde{Z}^2_U \approx P\begin{bmatrix}1 & \dots & (x_1)^{(\ell,\ell-1,\ell)}\\ \vdots & \ddots & \vdots\\ 1 & \dots & (x_{n_x})^{(\ell,\ell-1,\ell)}\end{bmatrix}, \quad \tilde{Z}^2_D \approx P\begin{bmatrix}(x_1)^{(0,1,0)} & \dots & (x_1)^{(\ell,\ell,\ell)}\\ \vdots & \ddots & \vdots\\ (x_{n_x})^{(0,1,0)} & \dots & (x_{n_x})^{(\ell,\ell,\ell)}\end{bmatrix}.$$

Introduce

$$B^2 := \tilde{Z}^2_D(\tilde{Z}^2_U)^+ \approx P\begin{bmatrix}x^2_1 & & \\ & \ddots & \\ & & x^2_{n_x}\end{bmatrix}P^{-1}.$$

For the third dimension, let $\tilde{Z}^3_U$ be the submatrix that excludes the columns $\alpha$ with $\alpha^3 = \ell$ and $\tilde{Z}^3_D$ be the submatrix that excludes the columns $\alpha$ with $\alpha^3 = 0$. Then

$$\tilde{Z}^3_U \approx P\begin{bmatrix}1 & \dots & (x_1)^{(\ell,\ell,\ell-1)}\\ \vdots & \ddots & \vdots\\ 1 & \dots & (x_{n_x})^{(\ell,\ell,\ell-1)}\end{bmatrix}, \quad \tilde{Z}^2_D \approx P\begin{bmatrix}(x_1)^{(0,0,1)} & \dots & (x_1)^{(\ell,\ell,\ell)}\\ \vdots & \ddots & \vdots\\ (x_{n_x})^{(0,0,1)} & \dots & (x_{n_x})^{(\ell,\ell,\ell)}\end{bmatrix}.$$

Introduce

$$B^3 := \tilde{Z}^3_D(\tilde{Z}^3_U)^+ \approx P\begin{bmatrix}x^3_1 & & \\ & \ddots & \\ & & x^3_{n_x}\end{bmatrix}P^{-1}.$$

The definitions of $B^1$, $B^2$, and $B^3$ suggest that their eigenvalues approximate the three components of $\{x_k\}$, respectively. In order to avoid matching these components, it is more convenient to recover $P$ first. However, when $B^1$, $B^2$, or $B^3$ have degenerate eigenvalues, the individual eigenvalue decompositions might give incorrect answers. To robustly recover $P$, we choose three random unit complex numbers $\beta^1, \beta^2, \beta^3$ and introduce

$$B := \beta^1 B^1 + \beta^2 B^2 + \beta^3 B^3 \approx P\begin{bmatrix}\beta^1 x^1_1 + \beta^2 x^2_1 + \beta^3 x^3_1 & & \\ & \ddots & \\ & & \beta^1 x^1_{n_x} + \beta^2 x^2_{n_x} + \beta^3 x^3_{n_x}\end{bmatrix}P^{-1}. \tag{10}$$

With high probability, these eigenvalues are well-separated from each other. Therefore, computing the eigenvectors of $B$ gives $P$. From $P$ and $P^{-1}$, we can approximate $x^1_1,\dots,x^1_{n_x}$ from $B^1$, $x^2_1,\dots,x^2_{n_x}$ from $B^2$, and $x^3_1,\dots,x^3_{n_x}$ from $B^3$, respectively. Combining them gives the approximation $\{\tilde{x}_k\}$ for $\{x_k\}$. Finally, the least square solution of

$$\min_{\tilde{w}_k} \sum_j \left| \sum_k G(s_j,\tilde{x}_k)\tilde{w}_k - \tilde{u}_j \right|^2$$

leads to the estimators $\{\tilde{w}_k\}$ for $\{w_k\}$.

For the $d$-dimensional real case, the approach is similar.

### 3.3. *Multidimensional real case*

Assume that the domain $X$ is $[-1,1]^d$. We can choose a product Chebyshev grid to construct the eigenmatrices $M^1,\dots,M^d$, responsible for

$$M^1\mathbf{g}(x) \approx x^1\mathbf{g}(x) \quad \dots \quad M^d\mathbf{g}(x) \approx x^d\mathbf{g}(x).$$

A general multi-index power for $\alpha = (\alpha^1,\dots,\alpha^d)$ is defined in the same way

$$M^\alpha := (M^1)^{\alpha^1}\dots(M^d)^{\alpha^d}.$$

From the matrix

$$\left[\tilde{\mathbf{u}} \quad \dots \quad M^\alpha\tilde{\mathbf{u}} \quad \dots \quad M^{(\ell,\cdots,\ell)}\tilde{\mathbf{u}}\right]$$

with $(\ell+1)^d$ columns, we compute $\tilde{Z}^i_U$ and $\tilde{Z}^i_D$ for each dimension $i=1,\dots,d$. From the matrices $B^i := \tilde{Z}^i_D(\tilde{Z}^i_U)^+$ and $B := \beta^1B^1+\dots+\beta^dB^d$, we can then approximate the spike locations $\{x_k\}$.

### 3.4. *Multidimensional complex case*

The multidimensional dimensional complex case is similar. To simplify the notation, we consider two complex dimensions and assume $X = \mathbb{D}_1\times\mathbb{D}_1$. For each $x\in X$, define $\mathbf{g}(x) := [G(s_j,x)]_{1\le j\le n_s}$. Instead of the Chebyshev grid, we choose $\{a_t\}_{1\le t\le n_a}$ to be the product uniform grid over the two unit circles $\mathbb{S}_1\times\mathbb{S}_1$, i.e., with $\sqrt{n_a}$ uniform points on each circle. As before, we choose the matrices $M^1$ and $M^2$ by enforcing

$$M^1\hat{\mathbf{g}}(a_t) \approx a^1_t\hat{\mathbf{g}}(a_t), \quad M^2\hat{\mathbf{g}}(a_t) \approx a^2_t\hat{\mathbf{g}}(a_t) \tag{11}$$

on $\{a_t\}$. In fact, this ensures that

$$M^1\hat{\mathbf{g}}(x) \approx x^1\hat{\mathbf{g}}(x), \quad M^2\hat{\mathbf{g}}(x) \approx x^2\hat{\mathbf{g}}(x)$$

for any $x\in X$. To see this, notice that for $x=(x^1,x^2)$

$$\begin{aligned}
M^1\mathbf{g}(x^1,x^2) &= \frac{1}{(2\pi i)^2}\int_{\partial\mathbb{D}}\int_{\partial\mathbb{D}} \frac{M^1\mathbf{g}(z^1,z^2)}{(z^1-x^1)(z^2-x^2)}dz^1dz^2\\
&\approx \frac{1}{(2\pi i)^2}\sum_t \frac{M^1\mathbf{g}(a^1_t,a^2_t)}{(a^1_t-x^1)(a^2_t-x^2)}(ia^1_t\frac{2\pi}{\sqrt{n_a}})(ia^2_t\frac{2\pi}{\sqrt{n_a}})\\
&\approx \frac{1}{(2\pi i)^2}\sum_t \frac{a^1_t\mathbf{g}(a^1_t,a^2_t)}{(a^1_t-x^1)(a^2_t-x^2)}(ia^1_t\frac{2\pi}{\sqrt{n_a}})(ia^2_t\frac{2\pi}{\sqrt{n_a}})\\
&\approx \frac{1}{(2\pi i)^2}\int_{\partial\mathbb{D}}\int_{\partial\mathbb{D}} \frac{z^1\mathbf{g}(z^1,z^2)}{(z^1-x^1)(z^2-x^2)}dz^1dz^2\\
&= x^1\mathbf{g}(x^1,x^2)
\end{aligned}$$

and similarly for $M^2\mathbf{g}(x^1,x^2) \approx x^2\mathbf{g}(x^1,x^2)$. This argument relies on the analyticity of $\mathbf{g}(x^1,x^2)$, $x^1\mathbf{g}(x^1,x^2)$, and $x^2\mathbf{g}(x^1,x^2)$ with respect to $x_1$ and $x_2$. The first and third approximations are from the exponential convergence of the trapezoidal rule, while the second one is from the condition (11) enforced on $M^1$ and $M^2$. The two equalities are from the Cauchy integral formula.

Once $M^1$ and $M^2$ are ready, we define for any $\alpha=(\alpha^1,\alpha^2)$

$$M^\alpha := (M^1)^{\alpha^1}(M^2)^{\alpha^2}.$$

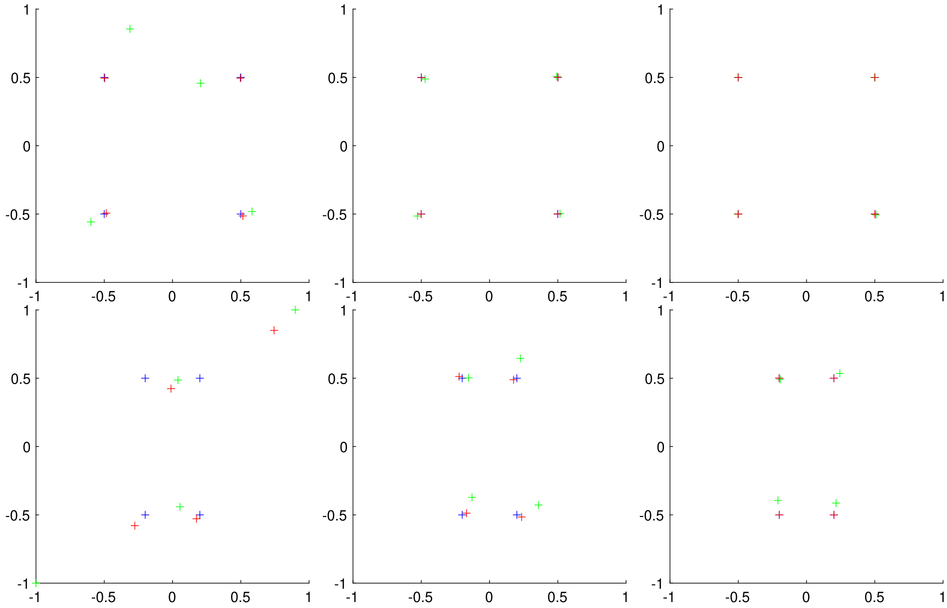

**Fig. 1.** Sparse deconvolution. $G(s,x)=\frac{1}{\|s-x\|}$. $X=[-1,1]^2$. $\{s_j\}$ are $n_s=1024$ random points in $[-2,2]^2$ outside $X$. Columns: $\sigma$ equals to $10^{-2}$, $10^{-3}$, and $10^{-4}$, respectively. Rows: the easy case (top) and the hard case (bottom). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

From the matrix

$$\begin{bmatrix}\tilde{\mathbf{u}} & \dots & M^{\alpha}\tilde{\mathbf{u}} & \dots & M^{(\ell,\ell)}\tilde{\mathbf{u}}\end{bmatrix}$$

the matrices $\tilde{Z}_U^t$ and $\tilde{Z}_D^t$ for $t=1,2$ can be computed as in (8). By introducing the matrices

- $B^t := \tilde{Z}_D^t(\tilde{Z}_U^t)^+$ for $t=1,2$ as in (9) and
- $B := \beta^1 B^1 + \beta^2 B^2$ as in (10),

we can approximate the spike locations $\{x_k\}$.

## 4. Numerical results

This section applies the eigenmatrix approach to the multidimensional unstructured sparse recovery problems mentioned above.

### *4.1. 2D real case*

In the examples, the Chebyshev grid is $32\times 32$, i.e., $n_a=32^2$. The spike weights $\{w_k\}$ are set to be 1 and the noises $\{Z_j\}$ are Gaussian. In each plot, blue, green, and red stand for the exact solution, the result before postprocessing, and the one after postprocessing, respectively.

**Example 1** *(Sparse deconvolution)*. The problem setup is

- $G(s,x)=\frac{1}{\|s-x\|}$.
- $X=[-1,1]^2$.
- $\{s_j\}$ are $n_s=1024$ random points in $[-2,2]^2$ outside $X$.

Fig. 1 summarizes the experimental results. The three columns correspond to noise levels $\sigma$ equal to $10^{-2}$, $10^{-3}$, and $10^{-4}$. Two tests are performed. In the first one (top row), $\{x_k\}$ are well-separated from each other. The plots show accurate recovery of the spikes from all $\sigma$ values. In the second one (bottom row), the spikes form two nearby pairs. The reconstruction at $\sigma=10^{-2}$ shows a noticeable error, while the results for $\sigma=10^{-3}$ and $\sigma=10^{-4}$ are accurate.

**Example 2** *(Fourier inversion)*. The problem setup is

- $G(s,x)=\exp(\pi i s\cdot x)$.

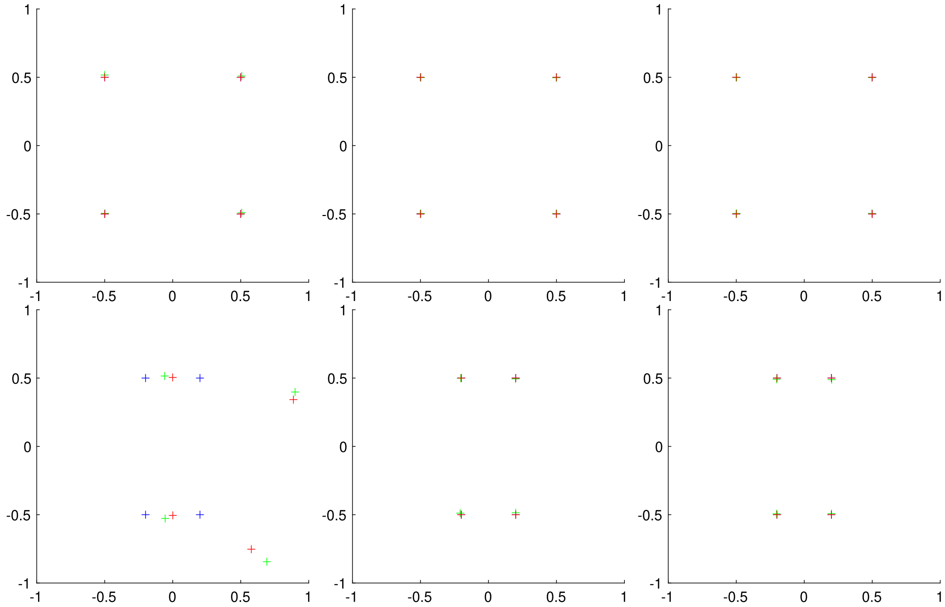

**Fig. 2.** Fourier inversion. $G(s,x) = \exp(\pi i s \cdot x)$. $X = [-1,1]^2$. $\{s_j\}$ are $n_s = 1024$ randomly chosen points in $[-8,8]^2$. Columns: $\sigma$ equals to $10^{-2}$, $10^{-3}$, and $10^{-4}$, respectively. Rows: the easy case (top) and the hard case (bottom).

- $X = [-1,1]^2$.
- $\{s_j\}$ are $n_s = 1024$ randomly chosen points in $[-8,8]^2$.

Fig. 2 summarizes the experimental results. The three columns correspond to $\sigma$ equal to $10^{-2}$, $10^{-3}$, and $10^{-4}$. Two tests are performed. In the first one (top row), $\{x_k\}$ are well-separated, and the reconstructions are accurate for all $\sigma$ values. In the second one (bottom row), the spikes form two nearby pairs. The reconstruction at $\sigma = 10^{-2}$ shows a noticeable error, but the ones for lower noises are accurate.

### 4.2. 3D real case

In the 3D examples, the Chebyshev grid is $16 \times 16 \times 16$, i.e., $n_a = 16^3$. The spike weights $\{w_k\}$ are set to be 1 and the noises $\{Z_j\}$ are still Gaussian.

**Example 3** *(Sparse deconvolution).* The problem setup is

- $G(s,x) = \frac{1}{\|s-x\|^{1/2}}$.
- $X = [-1,1]^3$.
- $\{s_j\}$ are $n_s = 8192$ random points in $[-2,2]^3$ outside $X$.

Fig. 3 summarizes the experimental results. The three columns correspond to noise levels $\sigma$ equal to $10^{-4}$, $10^{-5}$, and $10^{-6}$. Two tests are performed. In the first one (top row), $\{x_k\}$ are well-separated from each other. In the second one (bottom row), the spikes form two nearby pairs. The plots show accurate recovery of the spikes from all $\sigma$ values in both cases.

**Example 4** *(Fourier inversion).* The problem setup is

- $G(s,x) = \exp(\pi i s \cdot x)$.
- $X = [-1,1]^3$.
- $\{s_j\}$ are $n_s = 8192$ randomly chosen points in $[-4,4]^3$.

Fig. 4 summarizes the experimental results. The three columns correspond to $\sigma$ equal to $10^{-3}$, $10^{-4}$, and $10^{-5}$. Two tests are performed. In the first one (top row), $n_x = 4$ and $\{x_k\}$ are well-separated from each other. In the second one (bottom row), the spikes again form two nearby pairs. The reconstructions are accurate for all $\sigma$ values.

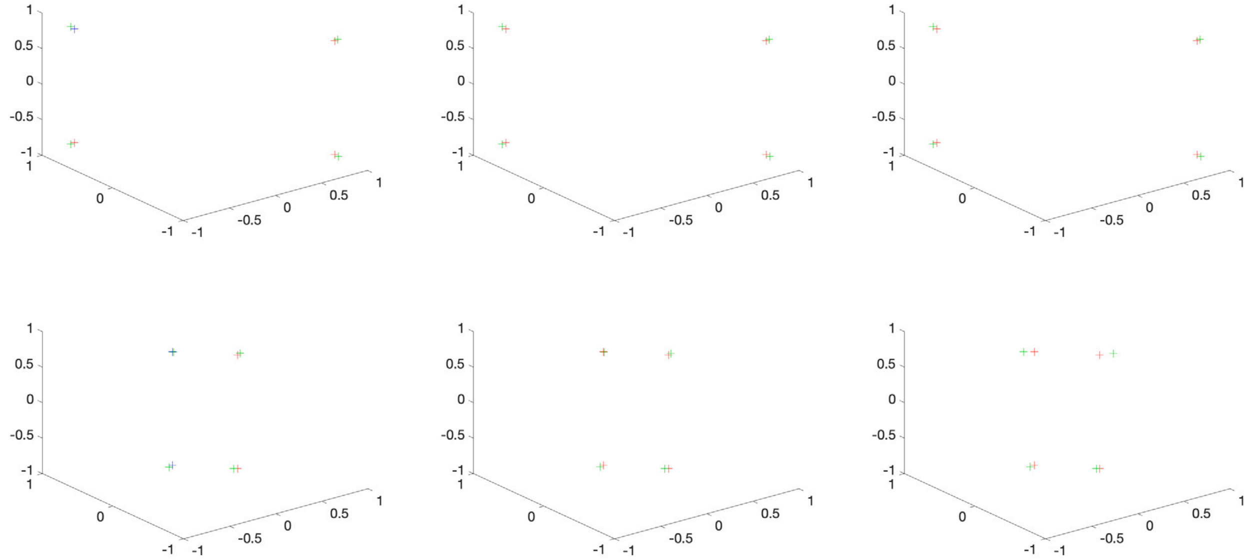

**Fig. 3.** Sparse deconvolution. $G(s,x) = \frac{1}{\|s-x\|^{1/2}}$. $X = [-1,1]^3$. $\{s_j\}$ are $n_s = 8192$ random points in $[-2,2]^3$ outside $X$. Columns: $\sigma$ equals to $10^{-4}$, $10^{-5}$, and $10^{-6}$, respectively. Rows: the easy case (top) and the hard case (bottom).

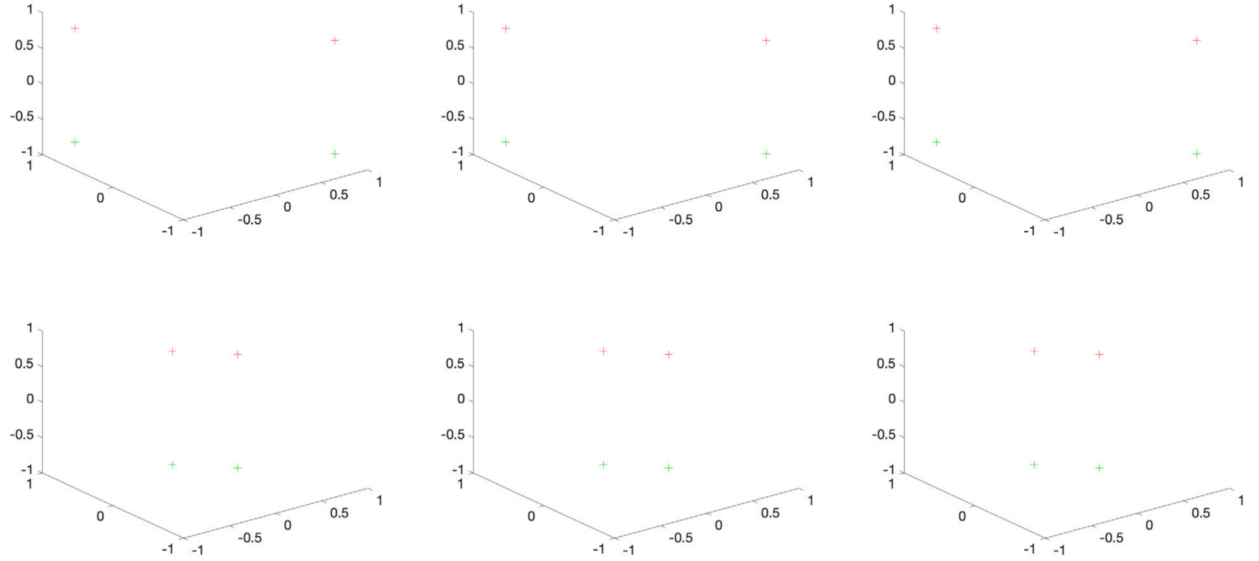

**Fig. 4.** Fourier inversion. $G(s,x) = \exp(\pi i s \cdot x)$. $X = [-1,1]^3$. $\{s_j\}$ are $n_s = 8192$ randomly chosen points in $[-4,4]^3$. Columns: $\sigma$ equals to $10^{-3}$, $10^{-4}$, and $10^{-5}$, respectively. Rows: the easy case (top) and the hard case (bottom).

### 4.3. 2D complex case

The product uniform grid over the circles is $32 \times 32$, i.e., $n_a = 32^2$. The spike weights $\{w_k\}$ are set to be 1 and the noises $\{Z_j\}$ are still Gaussian.

**Example 5** *(Rational approximation)*. The problem setup is

- $G(s,x) = \frac{1}{(s^1-x^1)(s^2-x^2)}$.
- $X = \mathbb{D}_1 \times \mathbb{D}_1$.
- $S = (\mathbb{C} \setminus \rho\mathbb{D}_1) \times (\mathbb{C} \setminus \rho\mathbb{D}_1)$, with $\rho = 1.2$. $\{s_j\}$ are $n_s = 1200$ random points in $S$.

Fig. 5 summarizes the experimental results. The two columns correspond to the noise level $\sigma$ equal to $10^{-2}$ and $10^{-4}$, respectively. At each noise level, two tests are performed. In the first one (top row), $\{x_k\}$ are well-separated from each other, while in the second

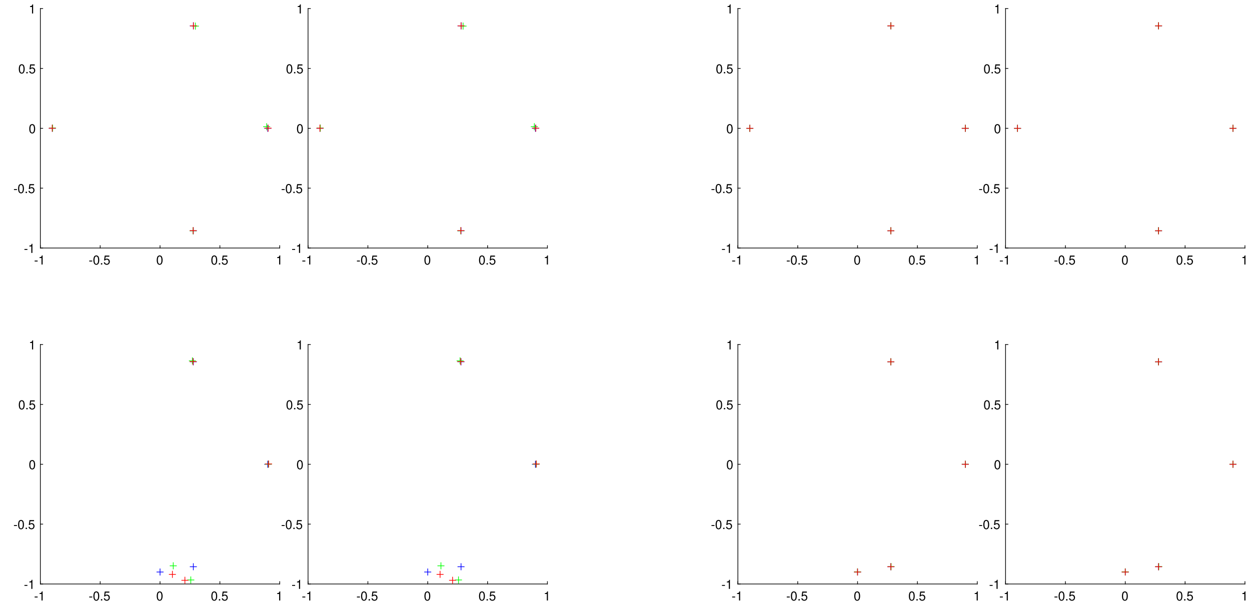

**Fig. 5.** Rational approximation. $G(s,x)=\frac{1}{(s^1-x^1)(s^2-x^2)}$. $X=\mathbb{D}_1\times\mathbb{D}_1$. $\{s_j\}$ are $n_s=1200$ random points in $S=(\mathbb{C}\setminus\rho\mathbb{D}_1)\times(\mathbb{C}\setminus\rho\mathbb{D}_1)$, with $\rho=1.2$. Columns: The two columns correspond to the noise level $\sigma$ equal to $10^{-2}$ and $10^{-4}$, respectively. Rows: the easy case (top) and the hard case (bottom). For each experiment, two plots show the component $x^1\in\mathbb{D}_1$ on the left and the component $x^2\in\mathbb{D}_1$ on the right.

one (bottom row), some spikes are close by. For each experiment, two plots are shown with the component $x^1\in\mathbb{D}_1$ on the left and $x^2\in\mathbb{D}_1$ on the right. With $\sigma=10^{-2}$, the recovered spike locations are accurate only for the well-separated case. At the lower noise $\sigma=10^{-4}$, the recovered results are accurate in both cases.

## 5. Discussions

This note extends the eigenmatrix approach to multidimensional unstructured sparse recovery problems. As a data-driven approach, it assumes no structure on the samples and offers a rather unified framework for such recovery problems. There are several directions for future work.

- A better understanding of the relationship between the size $n_a$ of the Chebyshev grid and the distribution of $\{s_j\}$.
- How does the accuracy of the eigenmatrix $M$ depend on $\{s_j\}$ and the singular value threshold?
- Providing error estimates for the examples in the numerical result section.
- The recovery algorithm presented above follows the ESPRIT algorithm and [1]. An immediate question is whether other methods (such as the MUSIC and the matrix pencil method) can be combined with the eigenmatrix approach.

## Data availability

No data was used for the research described in the article.

## References

[1] Fredrik Andersson, Marcus Carlsson, Esprit for multidimensional general grids, SIAM J. Matrix Anal. Appl. 39 (3) (2018) 1470–1488.
[2] Fredrik Andersson, Marcus Carlsson, Maarten V. De Hoop, Nonlinear approximation of functions in two dimensions by sums of exponential functions, Appl. Comput. Harmon. Anal. 29 (2) (2010) 156–181.
[3] Annie Cuyt, Wen-shin Lee, Sparse interpolation of multivariate rational functions, Theor. Comput. Sci. 412 (16) (2011) 1445–1456.
[4] Zhiyuan Fan, Jian Li, Efficient algorithms for sparse moment problems without separation, in: The Thirty Sixth Annual Conference on Learning Theory, 2023, pp. 3510–3565.
[5] Jouhayna Harmouch, Houssam Khalil, Bernard Mourrain, Structured low rank decomposition of multivariate Hankel matrices, Linear Algebra Appl. 542 (2018) 162–185.
[6] Stefan Kunis, Thomas Peter, Tim Römer, Ulrich von der Ohe, A multivariate generalization of Prony's method, Linear Algebra Appl. 490 (2016) 31–47.
[7] Thomas Peter, Gerlind Plonka, Robert Schaback, Prony's method for multivariate signals, PAMM 15 (1) (2015) 665–666.
[8] Clarice Poon, Gabriel Peyré, Multidimensional sparse super-resolution, SIAM J. Math. Anal. 51 (1) (2019) 1–44.
[9] Daniel Potts, Manfred Tasche, Parameter estimation for multivariate exponential sums, Electron. Trans. Numer. Anal. 40 (204–224) (2013) 94.
[10] R. Prony, Essai experimental et analytique, J. Éc. Polytech. (1795) 24–76.
[11] Stephanie Rouquette, Mohamed Najim, Estimation of frequencies and damping factors by two-dimensional esprit type methods, IEEE Trans. Signal Process. 49 (1) (2001) 237–245.

[12] Richard Roy, Thomas Kailath, Esprit-estimation of signal parameters via rotational invariance techniques, IEEE Trans. Acoust. Speech Signal Process. 37 (7) (1989) 984–995.

[13] Joseph J. Sacchini, William M. Steedly, Randolph L. Moses, Two-dimensional Prony modeling and parameter estimation, IEEE Trans. Signal Process. 41 (11) (1993) 3127–3137.

[14] Souleymen Sahnoun, Konstantin Usevich, Pierre Comon, Multidimensional esprit for damped and undamped signals: algorithm, computations, and perturbation analysis, IEEE Trans. Signal Process. 65 (22) (2017) 5897–5910.

[15] Tomas Sauer, Prony's method in several variables, Numer. Math. 136 (2017) 411–438.

[16] Endre Süli, David F. Mayers, An Introduction to Numerical Analysis, Cambridge University Press, 2003.

[17] Filiep Vanpoucke, Marc Moonen, Yannick Berthoumieu, An efficient subspace algorithm for 2-d harmonic retrieval, in: Proceedings of ICASSP'94. IEEE International Conference on Acoustics, Speech and Signal Processing, 1994, pp. IV–461.

[18] Lexing Ying, Eigenmatrix for unstructured sparse recovery, Appl. Comput. Harmon. Anal. 71 (2024) 101653.