



OPEN ACCESS

EDITED BY

Ali Haider Khan,
Lahore Garrison University, Pakistan

REVIEWED BY

Anas Bilal,
Hainan Normal University, China
Muhammad Sajid,
Air University, Pakistan

*CORRESPONDENCE

Fariba Karimi
✉ karimi@itis.swiss

RECEIVED 14 August 2024

ACCEPTED 07 May 2025

PUBLISHED 04 June 2025

CITATION

Karimi F, Neufeld E, Fallahi A, Kurtcuoglu V
and Kuster N (2025) Efficient Fourier base
fitting on masked or incomplete structured
data. *Front. Neuroimaging* 4:1480807.
doi: 10.3389/fnimg.2025.1480807

COPYRIGHT

© 2025 Karimi, Neufeld, Fallahi, Kurtcuoglu
and Kuster. This is an open-access article
distributed under the terms of the [Creative
Commons Attribution License \(CC BY\)](#). The
use, distribution or reproduction in other
forums is permitted, provided the original
author(s) and the copyright owner(s) are
credited and that the original publication in
this journal is cited, in accordance with
accepted academic practice. No use,
distribution or reproduction is permitted
which does not comply with these terms.

Efficient Fourier base fitting on masked or incomplete structured data

Fariba Karimi^{1,2*}, Esra Neufeld¹, Arya Fallahi^{1,2},
Vartan Kurtcuoglu³ and Niels Kuster^{1,2}

¹The Foundation for Research on Information Technologies in Society (IT²IS), Zurich, Switzerland,

²Department of Information Technology and Electrical Engineering, Swiss Federal Institute of
Technology (ETH), Zurich, Switzerland, ³The Interface Group, Institute of Physiology, University of
Zurich, Zurich, Switzerland

Introduction: Fourier base fitting for masked or incomplete structured data holds significant importance, for example in biomedical image data processing. However, data incompleteness destroys the simple unitary form of the Fourier transformation, necessitating the construction and solving of a linear system—a task that can suffer from poor conditioning and be computationally expensive. Despite its importance, suitable methodology addressing this challenge is not readily available.

Methods: In this study, we propose an efficient and fast Fourier base fitting method suitable for handling masked or incomplete structured data. The developed method can be used for processing multi-dimensional data, including smoothing and intra-/extrapolation, even when confronted with missing data.

Results: The developed method was verified using 1D, 2D, and 3D benchmarks. Its application is demonstrated in the reconstruction of noisy and partially unreliable brain pulsation data in the context of the development of a biomarker for non-invasive craniospinal compliance monitoring and neurological disease diagnostics.

Discussion: The study investigated the impact of different analytical and numerical performance improvement measures (e.g., term rearrangement, precomputation of recurring functions, vectorization) on computational complexity and speed. Quantitative evaluations on these benchmarks demonstrated that peak reconstruction errors in masked regions remained acceptable (i.e., below 10% of the data range for all investigated benchmarks), while the proposed computational optimizations reduced matrix assembly time from 843 s to 11 s in 3D cases, demonstrating a 75-fold speed-up compared to unoptimized implementations. Singular value decomposition (SVD) can optionally be employed as part of the solving-step to provide regularization when needed. However, SVD quickly becomes the performance limiting in terms of computational complexity and resource cost, as the number of considered Fourier modes increases.

KEYWORDS

Fourier-base fitting, image processing, reconstruction, brain deformation data, masked data

1 Introduction

Fast Fourier transform (FFT) is an efficient algorithm that transforms a signal from its original domain (usually time or space) to the frequency domain. In particular, it is used to compute the discrete Fourier transform (DFT) or its inverse (IDFT) of a signal. FFT is extensively employed in a wide range of applications such as filtering, efficient

convolution evaluation, wave propagation, and more. It can be interpreted as a special basis transformation to the orthonormal Fourier basis $v_k(\mathbf{r}) = e^{j\mathbf{k}\cdot\mathbf{r}}/C$ ($j = \sqrt{-1}$, C : normalization constant). When the spatial domain consists of a structured rectangular grid with L_i (typically an even number) equidistant coordinates along the i -axis with extent l_i , \mathbf{k} is equal to $(n_x \cdot 2\pi/l_x, n_y \cdot 2\pi/l_y, n_z \cdot 2\pi/l_z)$ with $-L_i/2 \leq n_i < L_i/2$. While the DFT is defined for regular grids and the FFT is applicable to data in that structured form, there is often a need to fit a Fourier basis to unstructured data, or to structured data known on a subset of the grid points (referred to here as “masked” data). Packages like PyNUFFT (Lin, 2018) are designed to handle unstructured data. However, PyNUFFT’s application to masked structured data (a special case of unstructured data) poses challenges, since PyNUFFT treats missing data as zero. This means that this package is not suited for extrapolating data based on its frequency domain representation, a common application in medical imaging.

One notable advantage of the Fourier basis lies in the simplicity of its application and similarity between the DFT and IDFT. They link the Fourier transform $\tilde{f}(\mathbf{k})$ to the underlying data $f(\mathbf{r})$ through $\tilde{f}(\mathbf{k}) = \sum_{\mathbf{r}} v_{\mathbf{k}}^*(\mathbf{r}) f(\mathbf{r})$ and $f(\mathbf{k}) = \sum_{\mathbf{r}} v_{\mathbf{k}}(\mathbf{r}) \tilde{f}(\mathbf{r})$, where $*$ denotes the complex conjugate. However, once the grid on which data is available ceases to be regular and complete, $v_{\mathbf{k}}$ stops being an orthogonal basis. In other words, $\sum_{\mathbf{r}} v_{\mathbf{k}}^*(\mathbf{r}) v_{\mathbf{k}'}(\mathbf{r})$ can be non-zero, even when $\mathbf{k} \neq \mathbf{k}'$. In fact, merely computing $\sum_{\mathbf{r}} v_{\mathbf{k}}^*(\mathbf{r}) f(\mathbf{r})$ no longer corresponds to determining the coordinates of f in the space spanned by the Fourier basis v , as the coordinate transformation matrix $\{v_{\mathbf{k}}(\mathbf{r})\}_{\mathbf{r},\mathbf{k}}$ is non-unitary and often non-square. Instead, (pseudo-)inversion of said matrix is required to establish the coordinate transformation.

It is often desirable to reduce the considered Fourier base vectors to the range $[-N_i, N_i]$ where $N_i < L_i/2$. Reducing the range of considered base vectors acts as a form of low-pass filtering, effectively smoothing the signal. Fitting such a reduced basis can be valuable, e.g., for obtaining a fitted approximation for inter- or extrapolation purposes.

In the course of our investigation into establishing a non-invasive surrogate for craniospinal compliance (CC) by measuring changes in head impedance during the cardiac cycle (Karimi et al., 2023; Spiegelberg et al., 2022), we were confronted with this issue. CC is a measure of intracranial volume buffering capacity. A drop in CC, as it may occur in pathological conditions such as hydrocephalus (Balédent et al., 2004), syringomyelia (Leung et al., 2016), Chiari I malformation (Terem et al., 2018), and cerebral small vessel disease (Perosa et al., 2022), often precedes a potentially life-threatening increase in intracranial pressure. Since direct continuous monitoring of CC is currently not possible, an indirect quantification with a surrogate derived from non-invasive brain motion assessment could play an important role in the clinical management of such conditions. Consequently, the quantification and interpretation of brain’s pulsatile motion not only contributes to understanding brain physiology, but can also play a role in the diagnosis and therapeutic decision-making for the above-mentioned neurological disorders.

To maximize the signal information content of a potential novel CC surrogate, we leveraged magnetic resonance imaging

(MRI) deformation data, illustrating how the brain pulsates during the cardiac cycle due to the exchange of blood and cerebrospinal fluid (CSF) between cranial and spinal compartments (Karimi et al., 2023). Deformation imaging is typically performed using Displacement Encoding with Stimulated Echoes (DENSE) (Soellinger et al., 2009; Adams et al., 2020; Karimi et al., 2023) or phase-contrast MRI (Enzmann and Pelc, 1992). Fourier basis fitting of the MRI data is essential for two distinct reasons (Karimi et al., 2023): first, it improves signal-to-noise ratio, which is of utmost importance considering the sensitivity of the surrogate signal prediction, the noisy nature of imaging data, and the subtlety of the features of interest. Second, Fourier basis fitting enables extrapolation of the data from the brain bulk volume. This is necessary because the displacements of primary relevance are located on the brain surface, where artifacts associated with flowing CSF and the high contrast to bone and air cavities make the displacement data unreliable, such that it has to be masked out. Similar challenges are common in the field of medical imaging, and beyond. The equivalent for masked structured data to the “FFT”-method should be fast and efficient both computationally and in terms of memory requirements, especially for processing large amounts of (e.g., 4D image) data. Since masking often results in an ill-conditioned Fourier decomposition matrix ($[A]$ in Equation 8), the required inversion can also become computationally challenging.

This work has three main goals:

- Develop a fast and robust Fourier base fitting method for masked structured data;
- Verify the method using 1D, 2D, and 3D benchmarks;
- Demonstrate its applicability to deformation data in order to support computation and interpretation of head impedance change over the cardiac cycle, utilizing the approach established in Karimi et al. (2023).

2 Methodology

2.1 Linear system derivation

Let $\hat{f}(\mathbf{r})$ be an approximation for $f(\mathbf{r})$ expressed in its Fourier basis as:

$$\begin{aligned} \hat{f}(\mathbf{r}) &= \sum_{n_z=-N_z}^{N_z} \sum_{n_y=-N_y}^{N_y} \sum_{n_x=-N_x}^{N_x} \{a_{n_x, n_y, n_z} e^{j(n_x k_{x0} x + n_y k_{y0} y + n_z k_{z0} z)}\} \\ &= \sum_n a_n e^{j\mathbf{k}_n \cdot \mathbf{r}} \end{aligned} \quad (1)$$

where N_i are the desired number of harmonics along the i -axis-direction, $k_{i0} = 2\pi/l_i$, and a_n are complex numbers. If the number of spatial points are larger than the number of coefficients, the system is overdetermined and a_n should be computed by minimizing the difference between $f(\mathbf{r})$ and $\hat{f}(\mathbf{r})$ using

(for instance) the least square approach:

$$a_n = \arg \min_{a_n} \|f(\mathbf{r}) - \hat{f}(\mathbf{r})\|^2 \quad (2)$$

This can result in complex-valued $\hat{f}(\mathbf{r})$. Therefore, to ensure a real-valued approximation, Equation 1 is modified as follows:

$$\hat{f}(\mathbf{r}) = \sum_n \left(a_n e^{jk_n \cdot \mathbf{r}} + a_n^* e^{-jk_n \cdot \mathbf{r}} \right) \quad (3)$$

Defining R as:

$$R = \sum_r \left(f(\mathbf{r}) - \sum_n \left(a_n e^{jk_n \cdot \mathbf{r}} + a_n^* e^{-jk_n \cdot \mathbf{r}} \right) \right)^2 \quad (4)$$

where the first summation is over all available points in 3D space, and expressing a_n in terms of real values $a_n = b_n + jc_n$, we obtain the following conditions by ensuring a zero gradient at minimum:

$$\frac{\partial R(\mathbf{r})}{\partial b_m} = \sum_r \left(f(\mathbf{r}) - \sum_n \left[a_n e^{jk_n \cdot \mathbf{r}} + a_n^* e^{-jk_n \cdot \mathbf{r}} \right] \right) \times \left(e^{jk_m \cdot \mathbf{r}} + e^{-jk_m \cdot \mathbf{r}} \right) = 0 \quad (5)$$

$$\frac{\partial R(\mathbf{r})}{\partial c_m} = \sum_r \left(f(\mathbf{r}) - \sum_n \left[a_n e^{jk_n \cdot \mathbf{r}} + a_n^* e^{-jk_n \cdot \mathbf{r}} \right] \right) \times \left(e^{jk_m \cdot \mathbf{r}} - e^{-jk_m \cdot \mathbf{r}} \right) = 0 \quad (6)$$

Multiplying Equations 5 and 6 by 0.5 and summing them yields:

$$\sum_r \left(e^{jk_m \cdot \mathbf{r}} \sum_n \left(a_n e^{jk_n \cdot \mathbf{r}} + a_n^* e^{-jk_n \cdot \mathbf{r}} \right) \right) = \sum_r f(\mathbf{r}) e^{jk_m \cdot \mathbf{r}} \quad (7)$$

In real-valued matrix-form, Equation 7 can be written as:

$$\begin{pmatrix} \vdots & \vdots & \ddots & \vdots \\ \cdots & \text{Re}\{S_{m+n} + S_{m-n}\} & -\text{Im}\{S_{m+n} - S_{m-n}\} & \cdots \\ \cdots & \text{Im}\{S_{m+n} + S_{m-n}\} & \text{Re}\{S_{m+n} - S_{m-n}\} & \cdots \\ \vdots & \vdots & \ddots & \vdots \end{pmatrix} \begin{pmatrix} \vdots \\ b_n \\ c_n \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ \text{Re}\{\sum_r f(\mathbf{r}) e^{jk_m \cdot \mathbf{r}}\} \\ \text{Im}\{\sum_r f(\mathbf{r}) e^{jk_m \cdot \mathbf{r}}\} \\ \vdots \end{pmatrix} \equiv [A][x] = [B] \quad (8)$$

where $S_{m+n} = \sum_r e^{j(k_m+k_n) \cdot \mathbf{r}}$ and $S_{m-n} = \sum_r e^{j(k_m-k_n) \cdot \mathbf{r}}$. In this system of equations, odd (respectively, even) rows express equations for b_n (c_n). Even though it may seem that the number of unknown variables is $N = 2(2N_x + 1)(2N_y + 1)(2N_z + 1)$, these are actually pairs of duplicates, as $b_{n_x, n_y, n_z} = b_{-n_x, -n_y, -n_z}$ and $c_{n_x, n_y, n_z} = -c_{-n_x, -n_y, -n_z}$. This arises from the fact that $\hat{f}(\mathbf{r})$ should

be real, necessitating $a_k = a_{-k}^*$. In other words, a reduced system of equations $[AA]$ and $[BB]$ can be constructed from $[A]$ and $[B]$, such that the dimensions are halved and $(2N_x + 1)(2N_y + 1)(2N_z + 1)$ unknown variables ($[xx]$) are apparent. $[x]$ is recovered from $[xx]$ as follows (notation: $a : b$ signifies integers from a to b ; $a : c : b$ signifies the same, but in steps of c):

$$\begin{aligned} x[i] &= \frac{1}{2} xx[i] & i &= 1 : \frac{N}{2} - 1 \\ x[i] &= xx[i] & i &= \frac{N}{2} \\ x[N-i] &= \frac{1}{2} xx[i] & i &= 1 : 2 : \frac{N}{2} - 1 \\ x[N-i+1] &= -\frac{1}{2} xx[i+1] & i &= 1 : 2 : \frac{N}{2} - 1 \end{aligned} \quad (9)$$

2.2 Efficient matrix computation

In the previous section, we obtained a system of linear equations, whose solution provides Fourier base fitting. In this section, we focus on the efficient computation of $[AA]$ and $[BB]$ in terms of both computational and memory resource requirements. Key considerations are the reduction of the number of function evaluations—particularly of expensive non-linear function evaluations, which are factored out, combined, and precomputed—and the enabling of vectorization. Let $M(x, y, z)$ be the 3D mask which has a value of one where data exist and zero elsewhere. We pre-evaluate the costly exponential functions $e^{jn_x k_{x0} x_{1D}}$, $e^{jn_y k_{y0} y_{1D}}$, and $e^{jn_z k_{z0} z_{1D}}$ for $n_x = -2N_x : 2N_x$, $n_y = -2N_y : 2N_y$, and $n_z = -2N_z : 2N_z$, where x_{1D} , y_{1D} , and z_{1D} are the one dimensional x , y , and z coordinates. By this, we do not need to compute any other exponential in later stages. Subsequently, we compute S_i as follows:

$$\begin{aligned} S_i &= \sum_r e^{jk_i \cdot \mathbf{r}} \\ &= \sum_{x_{1D}} \sum_{y_{1D}} \sum_{z_{1D}} M(x, y, z) e^{jk_{zi} z} e^{jk_{yi} y} e^{jk_{xi} x} \\ &= \sum_{x_{1D}} \sum_{y_{1D}} \left(\sum_{z_{1D}} M(x, y, z) e^{jk_{zi} z} \right) e^{jk_{yi} y} e^{jk_{xi} x} \\ &= \sum_{x_{1D}} \left(\sum_{y_{1D}} M_z(x, y) e^{jk_{yi} y} \right) e^{jk_{xi} x} \\ &= \sum_{x_{1D}} M_{y,z}(x) e^{jk_{xi} x} \end{aligned} \quad (10)$$

Similarly for the computation of $[BB]$ matrix elements:

$$\begin{aligned} BB[i] &= \sum_r f(x, y, z) e^{jk_i \cdot \mathbf{r}} \\ &= \sum_{x_{1D}} \sum_{y_{1D}} \sum_{z_{1D}} f(x, y, z) M(x, y, z) e^{jk_{zi} z} e^{jk_{yi} y} e^{jk_{xi} x} \\ &= \sum_{x_{1D}} \sum_{y_{1D}} F_z(x, y) e^{jk_{yi} y} e^{jk_{xi} x} \\ &= \sum_{x_{1D}} F_{y,z}(x) e^{jk_{xi} x} \end{aligned} \quad (11)$$

Note that Equations 10 and 11 take the form of a regular Fourier transformation, such that an FFT method could be employed. However, given that $N_i \ll L_i$ in our application-of-interest, it is preferable to use vectorized *tensordot* operations instead (see Section 4.1). Additionally, if $4N_i > L_i$, the number of required S computations can be reduced by considering $S_n = S_{n+p}$, where $p = (j_x \cdot L_x, j_y \cdot L_y, j_z \cdot L_z)$ —in other words, exploiting symmetry modulo L_i .

The code was implemented in Python 3.9 using the NumPy library. Pre-storage of required exponentials in 1D space and reducing 3D transformations to 1D ones (in Equations 10 and 11) results in important performance improvements in terms of computational complexity and memory requirements (see Section 4.1). Nested loops were avoided completely and *for*-loops were replaced whenever possible by vectorization and array operation. A custom least-square solving method was implemented to avoid the large memory overhead of commonly available Python implementations. This is necessary, as the first dimension of the $[A]$ matrix is equal to the number of spatial points, which can be very large for volumetric image data.

2.3 Solving and regularization

In the previous section, we obtained a linear system of equations ($[AA][xx] = [BB]$) that needs to be solved to obtain the correct Fourier basis. As $[AA]$ is frequently ill-conditioned, we used regularization based on singular value decomposition (SVD). Figure 1 shows the regularization pipeline: first, the SVD is obtained using the NumPy linear algebra package, i. e., $[AA] = USV^T$ where U and V are unitary matrices and S is diagonal. Subsequently, all singular values smaller than an adaptively adjusted threshold (see below) are zeroed in S . The regularized inverse of $[AA]$ is obtained as $[AA]_{reg}^{-1} = VS'^{-1}U^T$, where S' is the thresholded singular value matrix, and its inverse is easily obtained by inverting the diagonal elements. Subsequently, $[xx]$ is computed using $[xx] = [AA]_{reg}^{-1}[BB]$. To identify the singular value threshold, the maximum of relative error between $[BB] = [AA][xx]$ and $[BB]$ is computed. If it is larger than a user-specified acceptance criterion, the number of non-zero singular values is increased by a fixed factor (we used 1.02 in the applications below) and the procedure repeated.

Note that it is only necessary to compute S and AA , and to perform the SVD and matrix inversion once for a dataset with a constant mask, even if data is gathered at multiple time-steps.

2.4 Reconstruction from Fourier basis coefficients

For reconstruction, a variant of Equation 3 is used:

$$\hat{f}(\mathbf{r}) = 2Re \left\{ \sum_n \left(a_n e^{i\mathbf{k}_n \cdot \mathbf{r}} \right) \right\}$$

(12)

A summary of the main computational steps of the proposed method, including references to key equations and corresponding sections, is provided in Table 1.

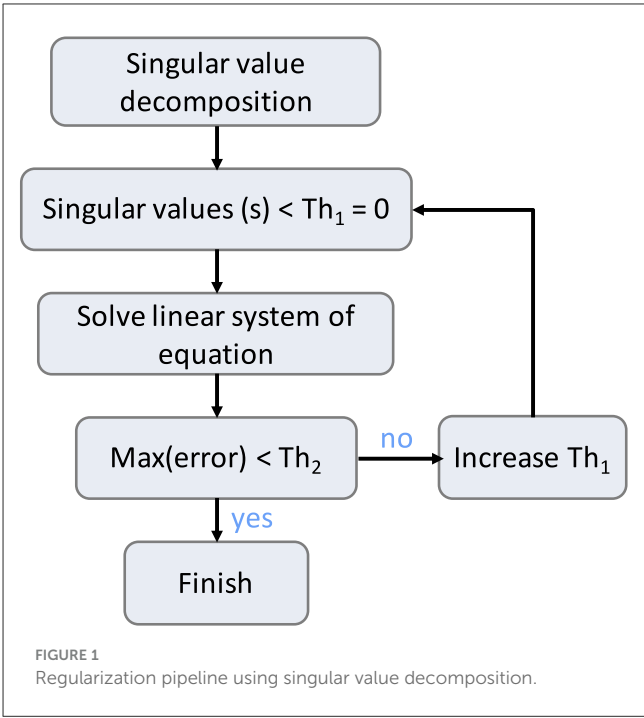


TABLE 1 Summary of the main computational steps in the proposed method, with references to key equations and sections.

Step	Description	Key equations/section
1	Fix the number of desired Fourier modes	Section 2.1 and Section 4.2
2	Precompute recurring exponential terms	Section 2.2
3	Assemble the (reduced) linear system matrix $[AA]$ and right-hand side vector $[BB]$ using the rearrangement and vectorization scheme.	Equations 8–11, Section 2.1
4	Solve the resulting linear system $[AA][xx] = [BB]$ either directly or via SVD-based regularization with adaptive thresholding.	Figure 1, Section 2.3
5	Reconstruct the spatial field from the estimated Fourier coefficients.	Equation 12, Section 2.4

3 Results

The developed method was tested on a Windows system with AMD Ryzen 9 7950X 16-Core processor and 128 GB of memory.

3.1 Verification benchmarks

To verify the proposed method, we defined three benchmark problems spanning from one to three spatial dimensions

TABLE 2 Implementation details and hyperparameter settings.

Category	Parameter	Value/description
Hardware	CPU	AMD Ryzen 9 7950X 16-Core processor
	GPU	NVIDIA GeForce RTX 3060
	RAM	128 GB
	OS	Windows 10
	Environment	Python 3.9, NumPy 1.26, JAX 0.4.20 (for SVD)
Fourier fitting	Number of Fourier modes (N_x, N_y, N_z)	$N_i = 11$ in each spatial direction for all benchmarks and $N_i = 4$ for deformation data
	Benchmark function parameters	$a = 5, b = 0.2, c = 1.5\pi$ (Ackley-type function)
	Grid resolution	200 (1D), 200×200 (2D), $200 \times 200 \times 200$ (3D)
	Padding	10 %
	Masking pattern	Structured block-wise masks (see Figures 2–8)
Regularization	Initial singular value truncation threshold	Singular values below 0.1
	Error-based refinement criterion	Triggered if $\max \ AA \cdot \text{coef} - BB\ > 3 \times 10^{-3}$
	Adjustment strategy	Increase number of retained singular values by 2% until error criterion is satisfied

based on the Ackley function (normalized to a $[0, 1]$ range), which is a common test function for global optimization methods. The full hardware configuration and implementation hyperparameters used for all benchmarks are summarized in Table 2.

3.1.1 1D benchmark

Let $f_{1D}(x)$ be defined as follows:

$$f_{1D}(x) = -a \exp(-b|x|) - \exp\left(\frac{1}{a} \cos(cx)\right) + a + \exp(1) \quad (13)$$

on $x: [-5, 5]$. The interval is discretized into 200 points and it is assumed that no data is available for $x: [-3.5, -2.5] \cup [-0.5, 0.5] \cup [2.5, 3.5]$. The reconstruction results (parameters: $a = 5, b = 0.2, c = 1.5\pi, N_x = 11$) are shown in Figure 2. Figure 3 displays the histogram of the reconstruction error for the domains inside and outside the mask. The quality metrics and compute times are specified in Table 3, which includes results for the 2D and 3D benchmarks as well.

3.1.2 2D benchmark

Let $f_{2D}(x, y)$ be defined as follows:

$$f_{2D}(x, y) = -a \exp\left(-b\sqrt{\frac{x^2 + y^2}{2}}\right) - \exp\left(\frac{\cos(cx) + \cos(cy)}{2}\right) + a + \exp(1) \quad (14)$$

on $x, y: [-5, 5] \times [-5, 5]$. The interval is discretized into 200×200 points and it is assumed that no data is available for $x, y: [-0.5, -0.5] \times [-0.5, 0.5] \cup [-0.5, 0.5] \times [2.0, 3.0] \cup [2.0, 3.0] \times [2.0, 3.0]$. The reconstruction results (parameters: $a = 5, b = 0.2, c = 1.5\pi, N_x = N_y = 11$) are shown in Figure 4. Figure 5 displays the histogram of the reconstruction error, normalized to the oscillation magnitude of the function, for the domains inside and outside the mask. Figure 6 shows corresponding results obtained using slower oscillations ($c = 0.8\pi$).

3.1.3 3D benchmark

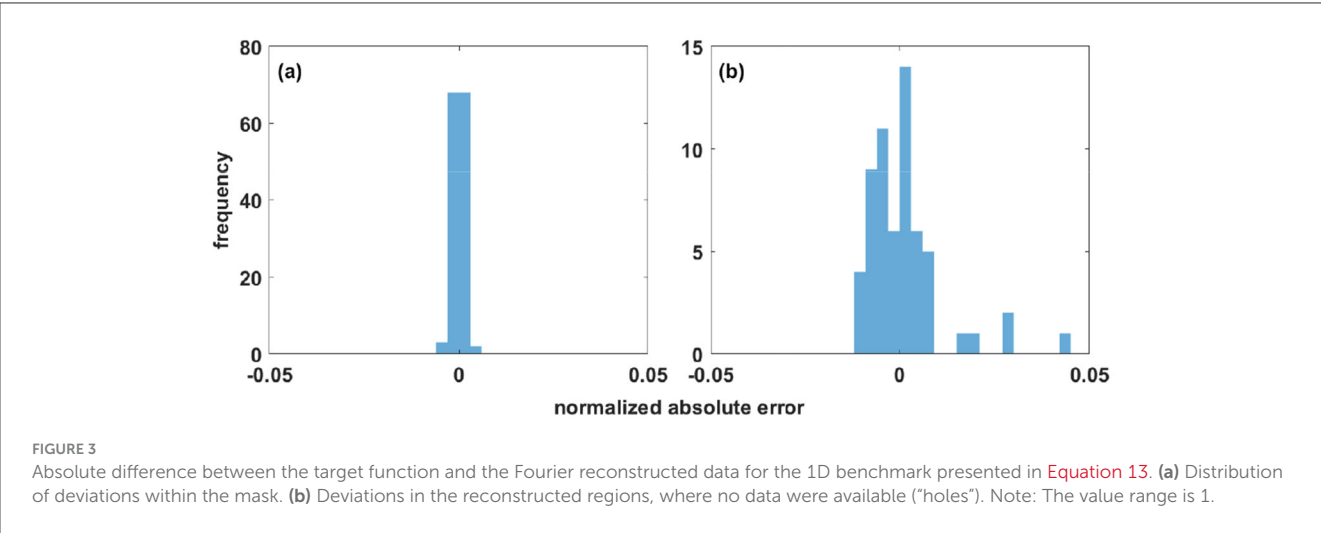
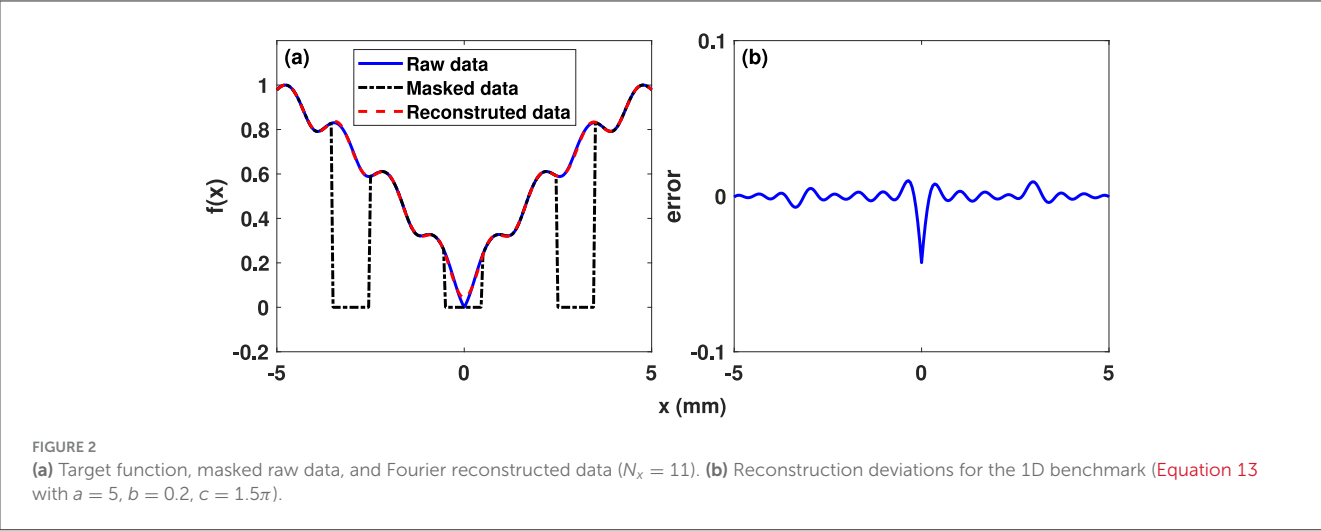
Let $f_{3D}(x, y, z)$ be defined as follows:

$$f_{3D}(x, y, z) = -a \exp\left(-b\sqrt{\frac{x^2 + y^2 + z^2}{3}}\right) - \exp\left(\frac{\cos(cx) + \cos(cy) + \cos(cz)}{3}\right) + a + \exp(1) \quad (15)$$

on $x, y, z: [-5, 5] \times [-5, 5] \times [-5, 5]$. The interval is discretized into $200 \times 200 \times 200$ points and it is assumed that no data are available for $x, y, z: [-0.5, -0.5] \times [-0.5, 0.5] \times [-0.5, 0.5] \cup [-0.5, 0.5] \times [2.0, 3.0] \times [-0.5, 0.5] \cup [2, 3] \times [2, 3] \times [-0.5, 0.5] \cup [-0.5, 0.5] \times [-0.5, 0.5] \times [2.0, 3.0]$. The reconstruction results (parameters: $a = 5, b = 0.2, c = 1.5\pi, N_x = N_y = N_z = 11$) are shown in Figure 7. Figure 8 displays the histogram of the reconstruction error, for the domains inside and outside the mask.

3.2 Application to deformation data

Deformation imaging reveals how the brain pulsates over a cardiac cycle due to transfer of blood and CSF between cranial and spinal compartments, which affects anatomical geometry and dielectric tissue properties, modulating the head impedance. In consequence, measurement of head impedance modulation can potentially be used to derive non-invasive CC surrogates. Deformation imaging reveals how the brain pulsates over a cardiac cycle due to transfer of blood and CSF between cranial and spinal compartments, which affects anatomical geometry and dielectric tissue properties, modulating the head impedance. In consequence, measurement of head impedance modulation can potentially be used to derive non-invasive CC surrogates. Experimental and modeling work by Spiegelberg et al. (2022) confirmed that head impedance variations can be measured and are at least partially of intracranial origin. A follow-up study Boraschi



et al. (2023) demonstrated posture-dependent changes in the non-invasively measured intracranial pulse waveform, providing further evidence that head impedance contains physiologically relevant information. These findings provide strong support for the feasibility of deriving meaningful, biophysically grounded surrogates of CC from surface electrode measurements. To translate this concept into a quantitative method, a computational biophysical model was developed Karimi et al. (2023) that relates small intracranial displacements and dielectric property changes to features of transient multi-contact head impedance variations. The computational model combines brain surface displacement fields over the cardiac cycle with electromagnetic simulations to derive resulting transient impedance variations, before deriving relationships between pulsation characteristics and the measurable signal—hence the necessity to extract surface motion fields from volumetric image data. However, deformation imaging is notoriously unreliable at the brain-CSF interface due to motion artifacts, dielectric-contrast-related noise, and flow-related imaging distortions. To improve the low signal-to-noise ratio of deformation data and replace unreliable data

TABLE 3 Summary of required computation times and reconstruction deviation metrics for the three benchmarks, providing information about speed and accuracy of the presented method.

Metric	1D	2D	3D
Fourier base fitting time	1.18 ms	49.5 ms	144'000 ms
Reconstruction time	0.19 ms	24.3 ms	1070 ms
Max. deviation in mask	0.003	0.05	0.04
Std. deviation in mask	0.002	0.01	0.007
Max. deviation outside mask	0.04	0.09	0.05
Std. deviation outside mask	0.01	0.04	0.01

The reconstruction deviations can be interpreted in comparison to the value range of 1.

near brain-CSF interfaces, first masking is applied. Masking excludes data outside the brain and within 3 mm of its borders (cortical surface and interface to ventricles). Subsequently, the Fourier reconstruction method is employed for smoothing and extrapolation-based reconstruction.

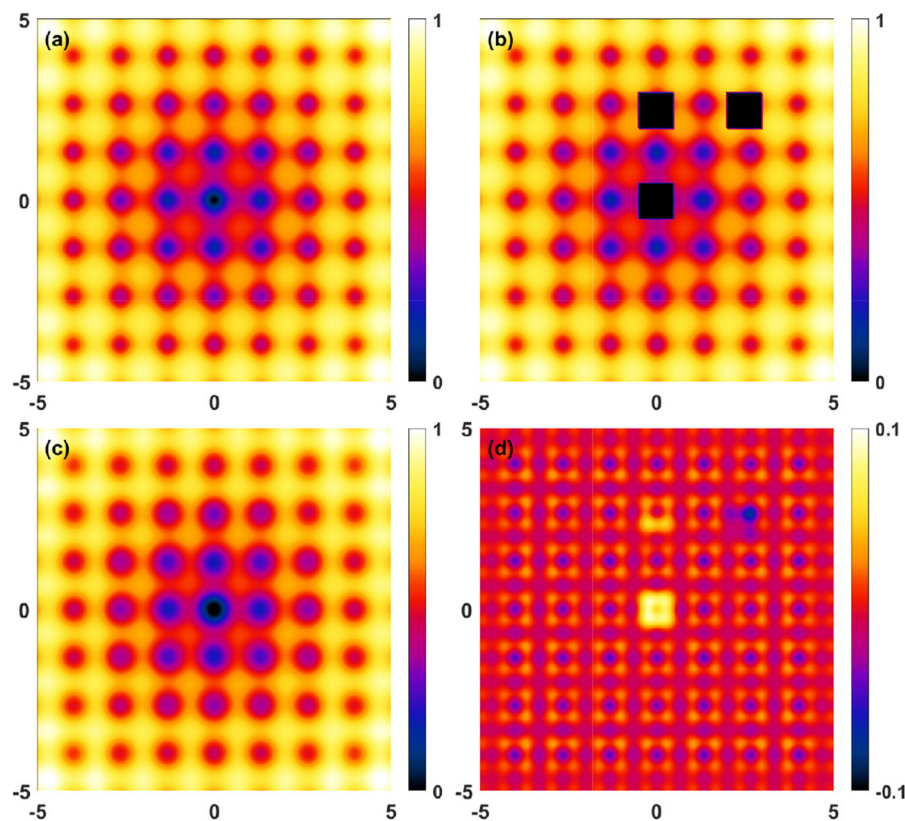


FIGURE 4

Reconstruction deviations for the 2D benchmark (Equation 14 with $a = 5$, $b = 0.2$, $c = 1.5\pi$). (a) Target function, (b) Masked raw data, (c) Fourier reconstructed ($N_x = N_y = 11$) data, (d) Reconstruction deviations. Note the different scale in (d).

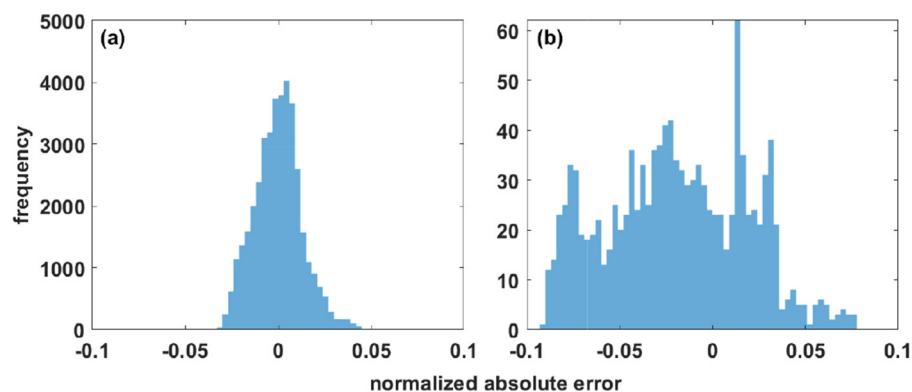


FIGURE 5

Normalized absolute difference between raw data and reconstructed data for the 2D benchmark presented in Equation 14, expressed in terms of oscillation range (i.e., 1). (a) Error within the mask. (b) Error within the holes.

The deformation data were acquired using a 7T MR scanner and the DENSE method (Adams et al., 2020), for right-left, anterior-posterior, and cranial-caudal gradient orientations (two opposite gradient polarities were recorded per direction). The resulting 4D data were large, comprising $320 \times 190 \times 320$ pixels, 20 snapshots over the cardiac cycle, and 6 polarities. Background noise

was reduced by subtracting the data from two opposite polarities. The Fourier basis fitting approach was applied for each time step during the cardiac cycle and each direction (right-left, anterior-posterior, and cranial-caudal). Figure 9 shows masked original deformation data and the smoothed and reconstructed one (setting $N_x = N_y = N_z = 4$) for one illustrative time step and brain slice.

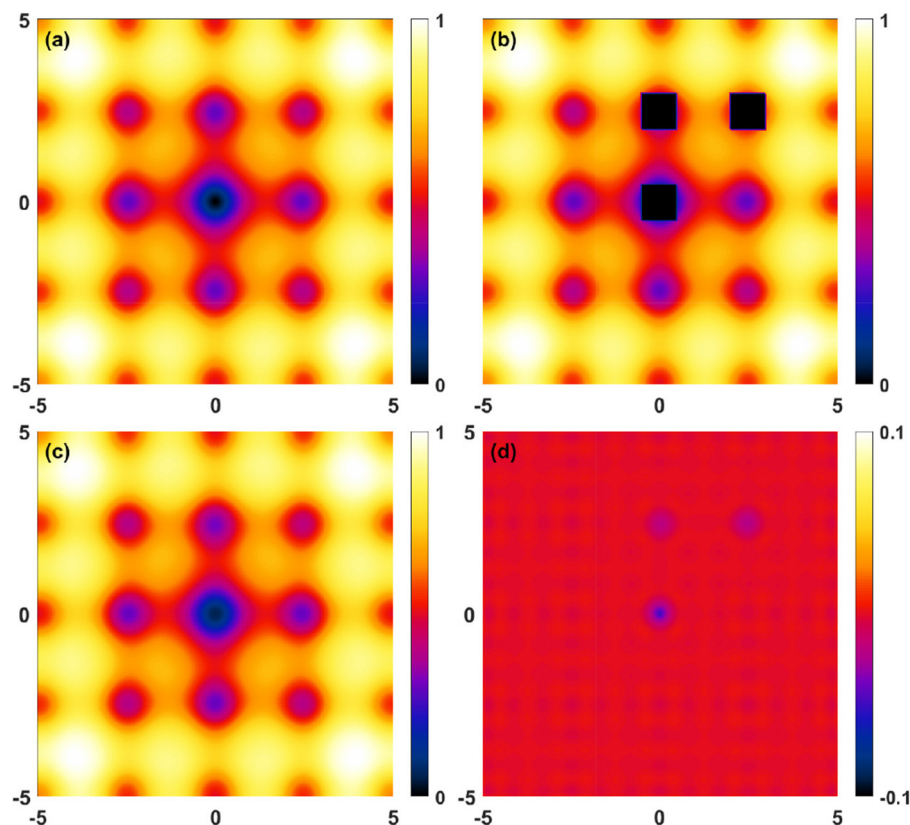


FIGURE 6

Reconstruction deviations for the 2D benchmark (Equation 14 with $a = 5$, $b = 0.2$, $c = 0.8\pi$). (a) Target function, (b) masked raw data, (c) Fourier reconstructed ($N_x = N_y = 11$) data, (d) reconstruction deviations. Note the different scale in this figure (d). In this case, the maximal absolute deviations were 0.006 and 0.053 within the masked domain and outside ("holes"), respectively, while the corresponding standard deviations were 0.001 and 0.009. The total value range is 1.

Fitting the Fourier basis took 1.9 s for computing the $[A]$ matrix (see Equation 8) and its SVD, which needs to be performed only once, regardless of how many imaging time steps and movement components need to be reconstructed. Computing the $[B]$ matrix and solving Equation 8 (i.e., base fitting) and reconstruction each took 0.5 s per time step and directional component.

Importantly, image-based motion reconstruction is only required during the development and validation of the CC monitoring approach—once the required relationships have been established, clinical applications of the non-invasive method will no longer require personalized deformation imaging, which is not readily available (certainly not in continuous monitoring scenarios). Instead, readily accessible head impedance recordings will be sufficient for measuring CC in individual patients.

4 Discussion

4.1 Complexity and timing

4.1.1 Exponential function evaluation

By precomputing the exponential functions, we reduce the number of corresponding evaluations from $(4N_x + 1) \cdot (4N_y + 1) \cdot (4N_z + 1) \cdot L_x \cdot L_y \cdot L_z + (2N_x + 1) \cdot (2N_y + 1) \cdot (2N_z + 1) \cdot L_x \cdot$

$L_y \cdot L_z$ to $(4N_x + 1) \cdot L_x + (4N_y + 1) \cdot L_y + (4N_z + 1) \cdot L_z$. The required number is further reduced if some of the axes have equal length. In addition, the operations can be efficiently vectorized in NumPy, involving the exponential of three outer products. Without precomputing the exponential functions, the matrix assembly time in our implementation increases by about 50 %.

4.1.2 Matrix computation

The calculations in Equation 10 involve $L_x \cdot L_y \cdot (4N_z + 1) + L_x \cdot (4N_y + 1) \cdot (4N_z + 1) + (4N_x + 1) \cdot (4N_y + 1) \cdot (4N_z + 1)$ scalar products, which is already an important complexity improvement when compared to the $L_x \cdot L_y \cdot L_z \cdot (4N_x + 1) \cdot (4N_y + 1) \cdot (4N_z + 1)$ complexity of a naive implementation. In addition, the $L_x \cdot L_y$ scalar products used to compute $M_z(x, y)$ involve the same vector $[e^{jk_z z_{1D}}]$ and N_x of the scalar products have the vector $[e^{jk_{yi} y_{1D}}]$ in common. Thanks to current CPU architectures and advanced compiler logics, these operations can be efficiently vectorized and parallelized as $4N_z + 1$, $4N_y + 1$, and $4N_x + 1$ tensor product operations (*tensordot* from NumPy) that each performs a reduction along a spatial direction of $L_x \cdot L_y \cdot L_z$, $L_x \cdot L_y \cdot (4N_z + 1)$, respectively $L_x \cdot (4N_y + 1) \cdot (4N_z + 1)$ -sized arrays. Similar considerations also apply to Equation 11.

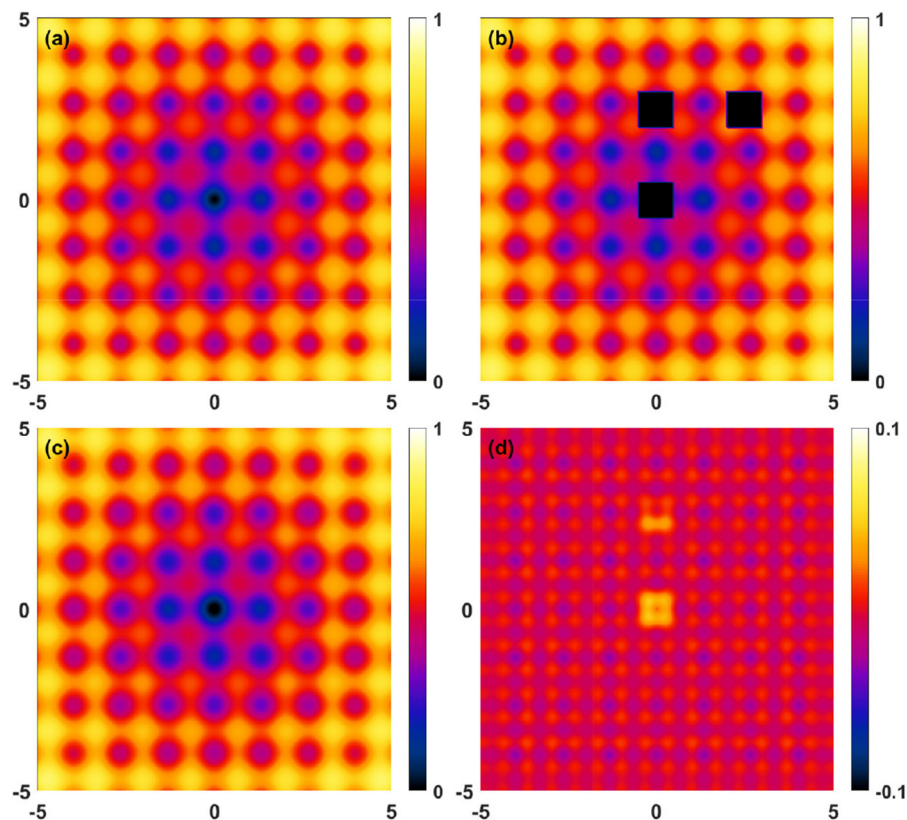


FIGURE 7

Reconstruction deviations for the 3D benchmark (Equation 15 with $a = 5$, $b = 0.2$, $c = 1.5\pi$). (a) Target function, (b) masked raw data, (c) Fourier reconstructed ($N_x = N_y = N_z = 11$) data, (d) reconstruction deviations. All results are shown on the $z = 0$ plane.

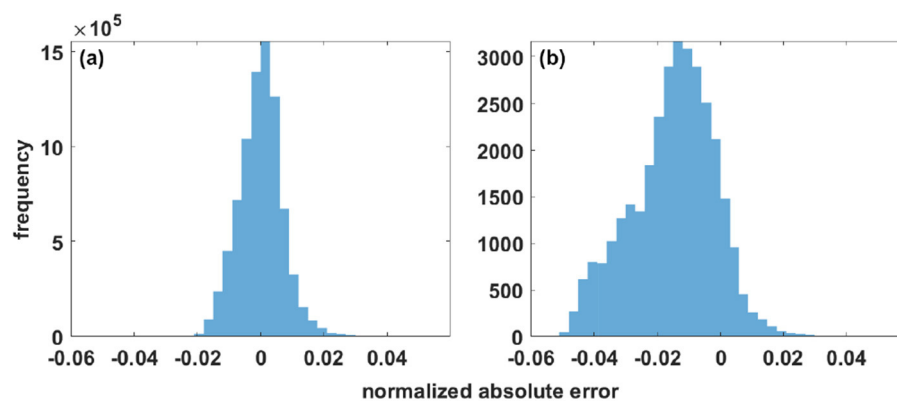


FIGURE 8

Normalized absolute difference between raw data and reconstructed data for the 3D benchmark presented in Equation 15, expressed in terms of oscillation range (i.e., 1). (a) Error within the mask. (b) Error within the holes.

4.1.3 Solving

We resort to SVD to ensure robust operation even when the matrix conditioning prevents direct solving. SVD complexity scales with the third order of the matrix dimension, i.e., $((2N_x + 1)(2N_y + 1)(2N_z + 1))^3$. Note, however, that this only

depends on the number of considered Fourier base vectors, which is typically much smaller than the number of points in the spatial domain. As mentioned earlier, SVD only needs to be performed once, regardless of how many imaging time steps and movement components need to be reconstructed.

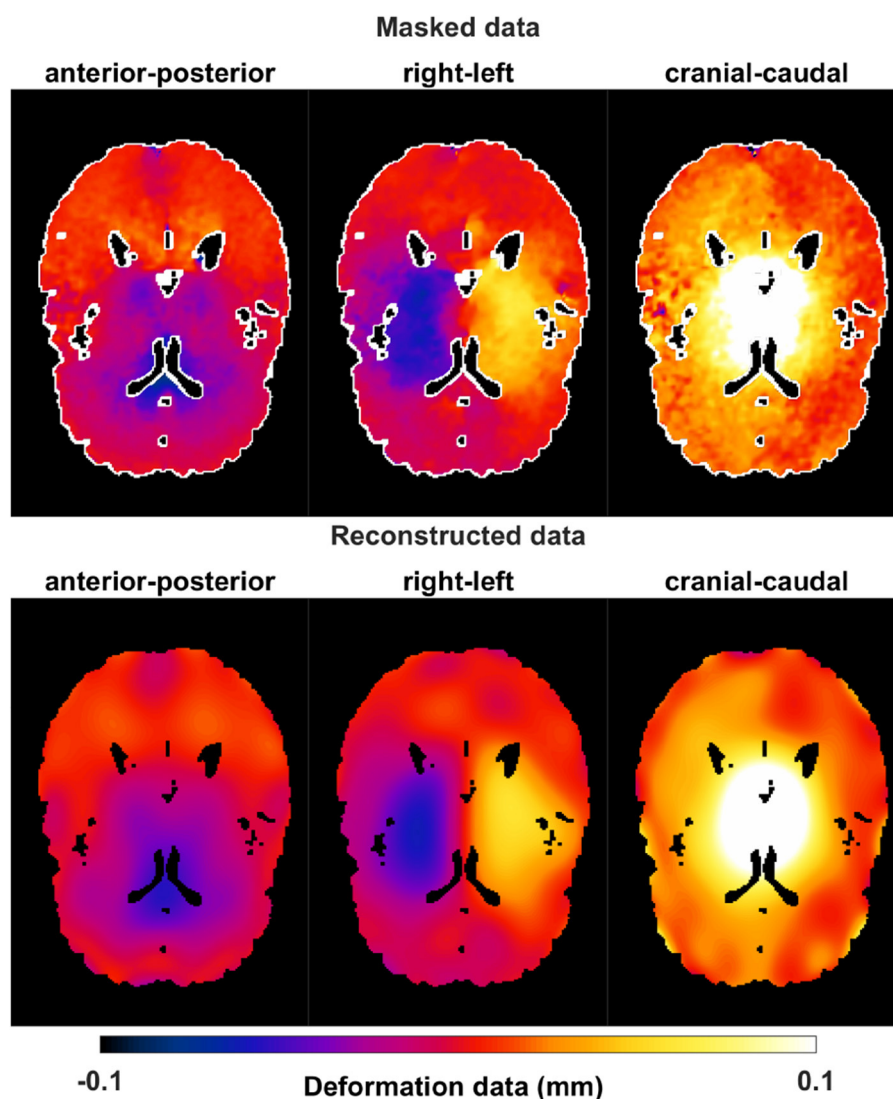


FIGURE 9

Masked deformation data and reconstructed one using Fourier base fitting (with $N_x = N_y = N_z = 4$) for one time step and one slice in the brain for right-left, anterior-posterior, and cranial-caudal components. The white parts in the masked data are CSF close-by regions which need to be extrapolated.

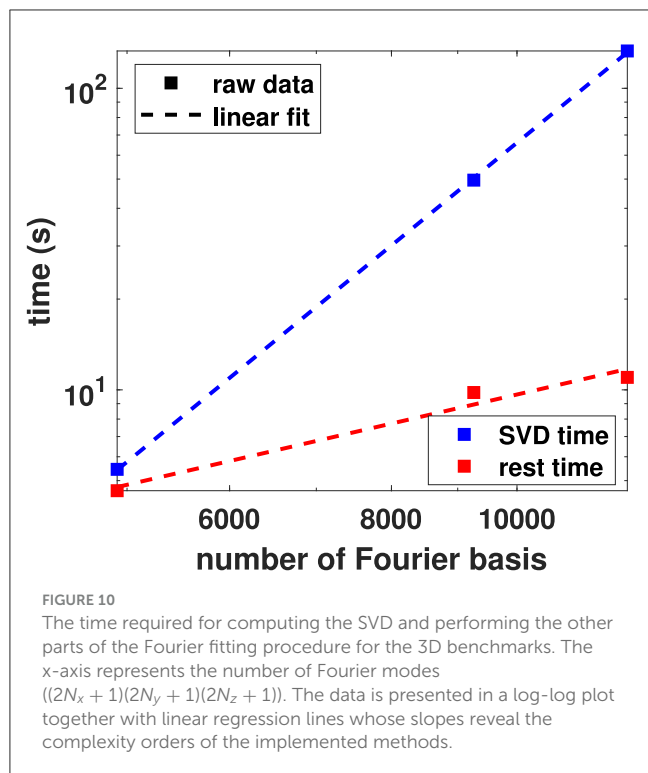
4.1.4 Repeated application

As mentioned earlier, it is only necessary to compute S and $[AA]$, and to perform the SVD and matrix inversion once for a dataset with a constant mask, even if there are multiple time-steps, deformation vector components, etc.

4.1.5 Performance and vectorization

The dimension dependence of the benchmark timings presented in Table 3 reflects the increase in the number of points from 200 to 200^3 (in our specific case where $L_x = L_y = L_z = 200$) and in the number of Fourier base vectors from 23 ($2N_x + 1$, with $N_x = 11$) to 23^3 . The latter aspect quickly dominates algorithmic performance, since the complexity of SVD is $\mathcal{O}(n^3)$, when n represents the matrix dimension. Indeed, comparison of the 2D and 3D benchmark timings reveals an exponent of 2.5, which is slightly

lower than 3.0 (likely because of the non-negligible contribution of better scaling algorithm components and of computational overhead). A closer examination of timings revealed that the SVD operation (implemented using JAX NumPy's `linalg.svd`) in the 3D benchmark accounts for over 92 % (133 s) of the computation time. The remaining 8 % (11 s) is spent precomputing the exponential functions and calculating the various matrix and vector elements of the linear system. Without vectorization, the duration of this part increases to a dominant 843 s, which represents a 75-fold slowdown. The significantly reduced run-time of the deformation data processing, compared to the 3D benchmark—despite the increased number of points—is due to the reduced number of considered Fourier modes ($N_i = 4$). When varying the number of Fourier modes in the 3D benchmark from $(2 \cdot 11 + 1)^3 = 12167$ to $(2 \cdot 10 + 1)^3 = 9261$ and $(2 \cdot 8 + 1)^3 = 4913$, a scaling exponent of 3.5 ($R^2 = 99\%$) is found for the SVD part and a



scaling exponent of 1.0 for the rest (mostly matrix/vector element computations), in close agreement with the theoretical complexities of 3 and 1 (see Figure 10). After compensating the SVD and matrix assembly timings of the benchmarks for their expected scaling as a function of the number of points and Fourier modes—assuming unsuccessful vectorization—we obtain strongly negative regression slopes (-1.3 as a function of the number of Fourier modes, -0.8 as a function of the number of spatial points). These values are a testimony to the gain offered by vectorization.

4.2 Choice of N and padding

The choice of N_x , N_y , and N_z is application-specific. Increasing these values reduces smoothing, increases fitting fidelity (higher spatial frequencies are captured), and permits more accurate extrapolation, (albeit generally over shorter distances; interpolation is typically more robust). On the other hand, using fewer modes enhances robustness to noise through low-pass filtering and reduces computational cost. In other words, cut-off selection involves a trade-off between robustness, noise suppression, and computational efficiency versus achievable resolution and fidelity, and is often guided by knowledge about relevant physiological and/or anatomical length scales, e.g., in clinical applications.

As expected, the periodicity of the Fourier basis results in fitted functions that reflect that periodicity. When performing extrapolation, it is therefore important to insert sufficient padding to avoid overly constraining the extrapolation by forcing it to fit a periodic boundary condition. The padding in turn increases the length of the spatial Fourier basis oscillations relative to the extent of the data, such that large padding can necessitate an increase of

N_i . In the case of the deformation data processed here, a grid extent increase by 10 % was found to yield satisfactory results.

4.3 Robustness to different missing data patterns

In practical applications, data incompleteness may arise in various forms—ranging from random dropout, to spatially structured missingness, or large contiguous gaps. The developed method is capable of handling all of these, as long as the characteristic length-scale of relevant features is not smaller than the gap size. If critical high-frequency content is absent—due to large contiguous gaps—, accurate reconstruction becomes fundamentally unattainable. This limitation is not unique to our method and can only be overcome if additional prior knowledge is available. Therefore, the size and spatial distribution of the missing regions, as well as the expected spectral content of the underlying data, must be considered before applying this method. The maximum tolerable block size before significant degradation occurs is application-specific and depends on both the frequency characteristics of the data and the desired level of reconstruction detail or smoothness.

4.4 Limitations

While the proposed method is robust and efficient for structured grids with arbitrary masking, the following main limitations should be noted. First, the method assumes that the underlying distribution can be well represented by a truncated Fourier basis. This acts as an inherent low-pass filter, which is beneficial for denoising but may limit the ability to reconstruct high-frequency features or sharp discontinuities when they are genuinely present in the data. Second, the optional SVD-based regularization step, while effective in stabilizing ill-conditioned systems, becomes computationally demanding as the number of Fourier modes increases, and may present a bottleneck for large-scale applications. It is also important to note that the method is designed specifically for structured (voxel-based) data, where acceleration strategies such as vectorization and precomputation are most effective. Generalizing it to unstructured grids would require a reformulation of the matrix construction and solution process and is not within the current scope of this work.

5 Conclusion

When fitting a Fourier basis to structured data known or defined on a subset of the grid points (mask), a minimization problem must be solved, the Fourier basis loses its unitary property, and determination of the Fourier coefficients becomes non-trivial and computationally expensive. To address this, we developed a library that reduces the number of required operations, minimizes the evaluation of costly non-linear functions, and permits the exploitation of computationally efficient vectorization of vector and matrix operations while reducing the memory footprint. SVD can optionally be employed as part of

the solving step to provide regularization when needed—as is often the case. However, SVD quickly becomes the performance limiting factor, as the number of considered Fourier modes increases. The developed method was verified using Ackley function benchmarks and demonstrated in a deformation image processing application. The library has been made publicly available.

Data availability statement

Implementation of the library is available on GitHub, providing open access to the source code. The repository includes documentation to guide users in installing, configuring, and utilizing the library effectively. Researchers and developers can access the library at <https://github.com/ITISFoundation/MIFT>. Additionally, real deformation data has been published at <https://doi.org/10.5281/zenodo.10590047>, providing data for utilizing the library in real-world applications.

Ethics statement

The studies involving humans were approved by Ethical Review Board of University Medical Center Utrecht. The studies were conducted in accordance with the local legislation and institutional requirements. Written informed consent for participation was not required from the participants or the participants' legal guardians/next of kin in accordance with the national legislation and institutional requirements.

Author contributions

FK: Conceptualization, Formal analysis, Methodology, Software, Validation, Visualization, Writing – original draft. EN: Conceptualization, Funding acquisition, Methodology, Project administration, Supervision, Writing – review & editing. AF:

Methodology, Supervision, Writing – review & editing. VK: Conceptualization, Funding acquisition, Investigation, Project administration, Writing – review & editing. NK: Resources, Supervision, Writing – review & editing.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. This study was financed by the Swiss National Science Foundation through grant 205321_182683.

Acknowledgments

We thank Dr. Jaco J. M. Zwanenburg, associate professor at the Center for Image Sciences, University Medical Center Utrecht, Utrecht, The Netherlands, for generously providing the deformation data used in this research. The availability of this valuable data significantly contributed to the success of our project.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

- Adams, A. L., Viergever, M. A., Luijten, P. R., and Zwanenburg, J. J. (2020). Validating faster dense measurements of cardiac-induced brain tissue expansion as a potential tool for investigating cerebral microvascular pulsations. *NeuroImage* 208:116466. doi: 10.1016/j.neuroimage.2019.116466
- Balédent, O., Gondry-Jouet, C., Meyer, M.-E., De Marco, G., Le Gars, D., Henry-Feugeas, M.-C., et al. (2004). Relationship between cerebrospinal fluid and blood dynamics in healthy volunteers and patients with communicating hydrocephalus. *Invest. Radiol.* 39, 45–55. doi: 10.1097/01.rli.0000100892.87214.49
- Boraschi, A., Spiegelberg, A., Karimi, F., Graf, K., Fallahi, A., Neufeld, E., et al. (2023). The effect of body position change on noninvasively acquired intracranial pulse waves. *Physiol. Meas.* 44:35014. doi: 10.1088/1361-6579/acc3d6
- Enzmann, D. R., and Pelc, N. J. (1992). Brain motion: measurement with phase-contrast MR imaging. *Radiology* 185, 653–660. doi: 10.1148/radiology.185.3.1438741
- Karimi, F., Neufeld, E., Fallahi, A., Boraschi, A., Zwanenburg, J. J., Spiegelberg, A., et al. (2023). Theory for a non-invasive diagnostic biomarker for craniospinal diseases. *NeuroImage Clin.* 37:103280. doi: 10.1016/j.nicl.2022.103280
- Leung, V., Magnussen, J. S., Stoodley, M. A., and Bilston, L. E. (2016). Cerebellar and hindbrain motion in chiari malformation with and without syringomyelia. *J. Neurosurg. Spine* 24, 546–555. doi: 10.3171/2015.8.SPINE15325
- Lin, J.-M. (2018). Python non-uniform fast fourier transform (PYNUFFT): an accelerated non-cartesian mri package on a heterogeneous platform (CPU/GPU). *J. Imaging* 4:51. doi: 10.3390/jimaging4030051
- Perosa, V., Arts, T., Assmann, A., Mattern, H., Speck, O., Oltmer, J., et al. (2022). Pulsatility index in the basal ganglia arteries increases with age in elderly with and without cerebral small vessel disease. *Am. J. Neuroradiol.* 43, 540–546. doi: 10.3174/ajnr.A7450
- Soellinger, M., Rutz, A. K., Kozerke, S., and Boesiger, P. (2009). 3D cine displacement-encoded MRI of pulsatile brain motion. *Magn. Reson. Med.* 61:153–162. doi: 10.1002/mrm.21802
- Spiegelberg, A., Boraschi, A., Karimi, F., Capstick, M., Fallahi, A., Neufeld, E., et al. (2022). Noninvasive monitoring of intracranial pulse waves. *IEEE Trans. Biomed. Eng.* 70, 144–153. doi: 10.1109/TBME.2022.3186748
- Terem, I., Ni, W. W., Goubran, M., Rahimi, M. S., Zaharchuk, G., Yeom, K. W., et al. (2018). Revealing sub-voxel motions of brain tissue using phase-based amplified MRI (AMRI). *Magn. Reson. Med.* 80, 2549–2559. doi: 10.1002/mrm.27236