Reconstructing the Magnetic Field in an Arbitrary Domain via Data-driven Bayesian Methods and Numerical Simulations

Georgios E. Pavlou[a,*], Vasiliki Pavlidou[b,c], Vagelis Harmandaris[a,d,e]

[a]*Institute of Applied and Computational Mathematics, Foundation for Research and Technology- Hellas, Heraklion, GR-71110, Greece*
[b]*Institute of Astrophysics, Foundation for Research and Technology- Hellas, Vasilika Vouton, Heraklion, GR-71110, Greece*
[c]*Department of Physics, University of Crete, Heraklion, GR-70013, Greece*
[d]*Department of Mathematics and Applied Mathematics, University of Crete, Heraklion, GR-71409, Greece*
[e]*Computation-Based Science and Technology Research Center, The Cyprus Institute, Nicosia, 2121, Cyprus*

Abstract

Inverse problems are prevalent in numerous scientific and engineering disciplines, where the objective is to determine unknown parameters within a physical system using indirect measurements or observations. The inherent challenge lies in deducing the most probable parameter values that align with the collected data. This study introduces an algorithm for reconstructing parameters by addressing an inverse problem formulated through differential equations underpinned by uncertain boundary conditions or variant parameters. We adopt a Bayesian approach for parameter inference, delineating the establishment of prior, likelihood, and posterior distributions, and the subsequent resolution of the maximum a posteriori problem via numerical optimization techniques. The proposed algorithm is applied to the task of magnetic field reconstruction within a conical domain, demonstrating precise recovery of the true parameter values.

Keywords: Inverse problems, Reconstruction, Bayesian Inference, Finite Element Method

1. Introduction

Inverse problems are encountered across a vast spectrum of scientific and engineering disciplines, from astrophysics to medical imaging, and their reach extends to geophysics, non-destructive testing, among others. Inverse problems are framed by the challenge of deducing unknown system properties or parameters from indirect, often noisy, and uncertain, experimental or observational data [1, 2]. Moreover, inverse problems are sometimes connected to reconstruction problems, where one wants to calculate a physical quantity in a whole domain, where it is only partially known from experiments. Therefore, the intrinsic ill-posedness of inverse problems complicates the quest for accurate and robust solutions and introduces complexities in numerical reconstruction efforts, where the goal is to compute physical values throughout a domain based on partial data [3, 4].

The principal objectives of this work are to develop a versatile, data-driven Bayesian algorithm for solving inverse problems with applications in reconstructing physical quantities (that that be calculated from a forward problem usually as a solution of a differential equation) from sparse data. By integrating Bayesian inference with clustering techniques, we can robustly infer boundary conditions and provide detailed reconstructions of the observed physical quantity in the given domain. Our results are supported by figures that elucidate the performance of our algorithm in various scenarios regarding the reconstruction of the Magnetic Field (MF) in a specific, arbitrarily defined, domain.

Statistical methods for addressing inverse problems have proliferated, with frequentist and Bayesian approaches providing distinctive strategies for parameter estimation and uncertainty analysis. More particularly, Bayesian methods [5] offer a structured probability-based framework that synthesizes prior information with observed data to articulate the posterior distribution of a parameter, which inherently captures the uncertainty of parameter inference and helps navigate the multidimensional parameter space. The increasing applications of inverse problems amplify the call for novel methods. Challenges include managing high-dimensional parameters, computational burdens, and the need for sophisticated algorithms designed to process large amounts of data efficiently [6].

Reconstruction techniques are a fundamental aspect of computational science, aimed at creating comprehensive models from incomplete or indirect data. These techniques are essential in transforming observational measurements into detailed representations of physical systems. Methods that resolve inverse problems are critically

*Corresponding author: e-mail: gepavlou@iacm.forth.gr

important, as they provide a means to infer the full spectrum of system parameters, some of which may not be directly observable. Our objective is to accurately reconstruct the spatial distribution of physical quantities. For example, in geophysics [7] and astrophysics [8], a wide range of methodologies are applied to reconstruct the MF. Techniques such as boundary element methods [9], Taylor series expansions [10], polynomial reconstructions [11], the pulsed wire method [12], full vector tomography [13], and approaches derived from Information Field Theory [14] have advanced MF reconstruction. Moreover, recent advancements in physics-informed machine learning have shown great promise in solving inverse problems by leveraging physical laws as prior knowledge within the learning process. These methods combine the strengths of data-driven approaches with the robustness of physics-based models, leading to improved interpretability and generalization of solutions [15, 16]. Particularly noteworthy is their application in multi-scale modeling of molecular systems, where they facilitate the reintroduction of atomic detail in coarse-grained models. For instance, a recent work utilizes a physics-informed deep learning framework to accurately re-introduce atomic detail in coarse-grained configurations of multiple poly(lactic acid) stereoisomers, showcasing the potent utility of these methods in complex molecular reconstructions [17].

Despite these developments, there remains a need for a more versatile and generalized methodology that can be broadly applied across different domains. To this end, sophisticated approaches that incorporate Bayesian inference algorithms stand out, as they facilitate the assimilation of prior knowledge with observational data, yielding a robust framework for comprehensive reconstructions, regardless of whether we are interested in the MF or any other physical quantity.

Building on this need, our paper presents a new data-driven Bayesian algorithm that is especially suited for inverse problems. This algorithm, which focuses on the particular challenge of identifying boundary condition parameters from diverse data sources, be it experimental, observational, or synthetic, incorporates a probabilistic model that incorporates existing parameter knowledge and constructs a likelihood function to assess the model's consistency with observed data. By utilizing Bayes' theorem, we orchestrate an amalgamation of prior knowledge and observational data, culminating in a well-founded probabilistic estimation of the parameters in question. The reconstruction aspect is not merely a byproduct, but a targeted objective of our algorithm, aiming to provide a detailed and accurate portrayal of physical values across any given domain, rather than simply Inferring internal parameters of partial differential equations.

The utility of our algorithm extends beyond magnetic fields and serves as a robust tool for reconstruction in a range of scientific disciplines. For example, in medical imaging, particularly magnetic resonance imaging (MRI), reconstruction techniques often rely on solving inverse problems to create detailed images from raw data [18]. In electrical engineering, inverse problems are integral to signal processing tasks, such as signal deconvolution, where a signal is reconstructed from incomplete frequency data [19]. Geophysics uses inverse problems in seismic imaging to deduce the internal structure of the Earth from surface seismic wave measurements [7]. In astrophysics, inverse problems are crucial for tasks such as inferring the mass distribution of galaxy clusters from gravitational lensing data [20]. Lastly, quantum tomography in quantum mechanics involves reconstructing quantum states or processes from measurement data, a process that is typically framed as an inverse problem [21]. Overall, our proposed methodology is designed not only to tackle the nuanced complexities of inverse problems, but to enable the comprehensive reconstruction of physical quantities from sparse data, be it the Galactic MF, for example.

The remainder of the paper is organized as follows. In Section 2, we present the mathematical formulation of the inverse problem and describe our algorithm in detail, while also providing information on its various aspects and practical usage. In Section 3, we present how the forward problem is solved in our application. More specifically, we show how one can use the Finite Element Method to calculate the MF in a given domain. In Section 4 we apply our algorithm to the problem of reconstructing the MF in a domain. We demonstrate its effectiveness in synthetic data, showcasing its ability to accurately recover the true parameter values. We discuss the cases of multiple priors where clustering of the data is necessary. Finally, Section 5 offers concluding remarks and discusses future perspectives. Furthermore, in the Appendix, we provide additional details on the formulation of the problem and the optimization algorithm.

2. Mathematical Formulation and Presentation of the Algorithm

2.1. Mathematical framework for the inverse problem

Consider a physical system with known parameters $\mathbf{x}$ (which can, for example, be the points on the grid of the physical domain Ω or other known quantities relevant to the problem) and unknown parameters $\boldsymbol{\theta}$ that can be measured by some indirect observation or experiment that produces the data set $\mathbf{y}$. The relationship between the parameters $\mathbf{x}$, $\boldsymbol{\theta}$ and the observed data $\mathbf{y}$ can be described by a forward model $f(\mathbf{x};\boldsymbol{\theta})$, which maps the parameters to the expected measurements. In many cases, the forward model involves solving a partial differential equation subject to certain boundary or initial conditions [5], where the unknown parameters, in our case, can refer to any

subdomain of Ω (for example, they may help us determine the boundary conditions) or to parameters that affect the initial conditions. Mathematically, we can formulate the forward problem as [22]:

$$\mathbf{y} = f(\mathbf{x}; \boldsymbol{\theta}) + \boldsymbol{\epsilon} \tag{1}$$

where $\boldsymbol{\epsilon}$ is a noise term that accounts for measurement errors or uncertainties.

Suppose that from the observed data $\mathbf{y}$, we want to infer information about the statistical properties of the parameters $\boldsymbol{\theta}$ that determine the boundary conditions. This defines an inverse problem. To be more concrete, consider a set of n_y, possibly sparse, measurements $\mathbf{y}$ randomly distributed in the known physical domain Ω. In many real-world scenarios, the physical system and the observed data involve numerous variables and measurements that are of high dimension, while the domain geometry could also be large. Our algorithm is designed to handle high-dimensional problems, where, for example, known parameters $\mathbf{x}$, $\mathbf{y}$, and unknown parameters $\boldsymbol{\theta}$ can potentially have large dimensions. Furthermore, the data $\mathbf{y}$ can be sparsely distributed in a large domain Ω. Finally, the parameters $\boldsymbol{\theta}$ can appear anywhere in the theoretical model; for example, they may appear in the differential equations that define the forward model, in the boundary conditions, etc. Therefore, the main challenge is to calculate the statistical properties of the parameters $\boldsymbol{\theta}$ using Bayesian inference.

In the context of this work, we assume that the physical system is described by a set of differential equations and that the unknown parameters $\boldsymbol{\theta}$ (treated as random variables) appear in the boundary conditions of the system. We focus on inferring the statistical properties of $\boldsymbol{\theta}$, rather than the boundary conditions themselves. To improve the robustness and precision of our estimates, we repeat the optimization procedure multiple times with different initial guesses for $\boldsymbol{\theta}$, and average the results. This approach helps to account for potential local optima or non-convexity of the posterior function, and allows us to explore the uncertainty and variability of the estimated parameters. Compared to previous work, which focused mainly on inverse problems with uncertain initial conditions or internal parameters of partial differential equations [5, 23, 24, 25, 26], our approach addresses the relatively less explored area of inverse problems with uncertain boundary conditions. Therefore, our approach is distinct from previous work on inverse problems, as we are specifically interested in the statistical properties of the parameters that affect the boundary conditions.

2.2. Bayesian inverse problems

Bayesian inference is a statistical framework that allows us to update our prior beliefs about the unknown parameters $\boldsymbol{\theta} = (\theta_1, ..., \theta_{n_\theta})$, where n_θ is the number of parameters, based on observed data $\mathbf{y}$ [5, 27]. In this context, we can use Bayesian inference to infer the statistical properties of the unknown parameters that determine the boundary conditions of the physical system. We start by specifying a prior probability distribution for the parameters $\boldsymbol{\theta}$, denoted as $p(\boldsymbol{\theta})$. The choice of a prior distribution can incorporate our prior knowledge, beliefs, or assumptions about the values of the unknown parameters. For instance, we might choose a uniform if we have no prior information about the parameters or a normal distribution if we believe that they are centered around some central value. If we have more than one parameter, the total prior is the product of each one:

$$p(\boldsymbol{\theta}) = \prod_{i=1}^{n_\theta} p(\theta_i) \tag{2}$$

Given the observed data $\mathbf{y}$, the posterior distribution over the parameters $\boldsymbol{\theta}$ is given by the Bayes' theorem:

$$p(\boldsymbol{\theta}|\mathbf{y}) = \frac{p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{y})} \tag{3}$$

where $p(\mathbf{y}|\boldsymbol{\theta})$ is the likelihood function that describes the probability of observing the data $\mathbf{y}$ given the parameters $\boldsymbol{\theta}$, and $p(\mathbf{y})$ is a normalization constant. The likelihood function is typically obtained from the forward model $f(\mathbf{x}; \boldsymbol{\theta})$ and the noise term ϵ as follows:

$$p(\mathbf{y}|\boldsymbol{\theta}) = \prod_{i=1}^{n_y} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - f(\mathbf{x}; \boldsymbol{\theta}))^2}{2\sigma^2}\right) \tag{4}$$

where n_m is the number of measurements, σ is the standard deviation of the noise term, and y_i is the i-th measurement in the data set.

The posterior distribution $p(\boldsymbol{\theta}|\mathbf{y})$ provides a measure of the uncertainty or confidence we have about the values of the unknown parameters $\boldsymbol{\theta}$ after observing the data $\mathbf{y}$. We can compute various statistics of interest from the posterior distribution, such as mean, variance, or credible intervals. In the context of inverse problems, the maximum likelihood estimate (MLE) is an alternative approach where the goal is to find the parameter values that maximize the likelihood function. MLE focuses solely on the likelihood, disregarding any prior information about the

parameters. However, in our Bayesian framework, we employ the Maximum a Posteriori (MAP) method, which extends the MLE concept by incorporating prior information. MAP maximizes posterior probability, offering a more comprehensive inference method that takes into account both the data (through the likelihood) and our prior knowledge (through the prior distribution). Specifically, our objective is to find the value of $\boldsymbol{\theta}$ that maximizes the posterior function:

$$\boldsymbol{\theta}_{\mathrm{MAP}} = \arg\max_{\boldsymbol{\theta}}(p\,(\boldsymbol{\theta}|\mathbf{y})). \tag{5}$$

In practice, it is usually more efficient to maximize the logarithm of the posterior function as:

$$\boldsymbol{\theta}_{\mathrm{MAP}} = \arg\max_{\boldsymbol{\theta}}(\ln p\,(\boldsymbol{\theta}|\mathbf{y})) = \arg\min_{\boldsymbol{\theta}}(-\ln p\,(\boldsymbol{\theta}|\mathbf{y})). \tag{6}$$

and $\boldsymbol{\theta}$ is calculated with proper numerical optimization techniques (more details will be given in the next Section) without having to evaluate the normalization constant $p(\mathbf{y})$.

We should note that alternatively to maximum a posteriori methods, Bayesian inference can be performed using Markov chain Monte Carlo (MCMC) methods [28]. MCMC algorithms draw samples from the posterior distribution $p(\boldsymbol{\theta}|\mathbf{y})$ without having to evaluate the normalization constant $p(\mathbf{y})$. The samples can be used to estimate the statistics of interest or to perform model selection and comparison.

In summary, Bayesian inference provides a powerful framework for inferring the statistical properties of the unknown parameters that determine the boundary conditions of a physical system. By incorporating prior knowledge, beliefs, or assumptions about the values of the parameters, we can obtain more accurate and robust estimates and quantify our uncertainty about the inferred values.

2.3. Presentation of the Algorithm for solving the Inverse Problem

The proposed algorithm aims to infer the statistical properties of unknown parameters $\boldsymbol{\theta}$ in a physical system described by differential equations. It is particularly adept at handling inverse problems with uncertain boundary conditions. Here, we present the algorithm and then discuss its main characteristics and stages (see also the flow diagram in Figure 1):

Algorithm 1 Data Driven Reconstruction

Require: (Input) Data set $\mathbf{y}$, forward model $f(\mathbf{x};\boldsymbol{\theta})$, noise level σ, optional clustering algorithm $C(\mathbf{y})$
Ensure: (Output) Estimated value of $\boldsymbol{\theta}$ that maximizes the posterior function, Reconstructed physical quantity
Step 1: *Optional Clustering*
if clustering is used **then**
(a) Apply clustering algorithm $C(\mathbf{y})$ to partition $\mathbf{y}$ into clusters
(b) For each cluster, select a prior for $\boldsymbol{\theta}$ based on prior knowledge
end if
Step 2: Choose a likelihood function reflecting the probability of observing $\mathbf{y}$ given $\boldsymbol{\theta}$
Step 3: Compute the posterior distribution $p(\boldsymbol{\theta}|\mathbf{y})$ via Bayes' theorem
Step 4: Define and solve the MAP problem to estimate $\boldsymbol{\theta}$ using numerical optimization, and repeat a set number of times to refine $\boldsymbol{\theta}$ estimate and assess statistical error
Step 5: Solve the forward problem with the estimated $\boldsymbol{\theta}$ to complete reconstruction of the physical quantity in the whole domain

The algorithm comprises several stages, each designed to systematically address different aspects of the inverse problem. Initially, an optional clustering step groups data points based on their proximity and characteristics, facilitating differentiated analysis for each cluster. This step is particularly useful for sparsely distributed data. The subsequent steps involve choosing a likelihood function based on the probability of observing the data given the parameters, computing the posterior distribution using Bayes' theorem, and solving a MAP problem through numerical optimization. This process is iterated to refine the estimates of $\boldsymbol{\theta}$ and assess statistical errors. Finally, the algorithm solves the forward problem with the estimated parameters, completing the reconstruction of the physical quantity across the entire domain.

In the following, we provide a few comments on the above reconstruction algorithm.

- We assume that the physical system is described by a set of differential equations, which involve the unknown parameters $\boldsymbol{\theta}$. For the applications considered here, the latter appear in the boundary conditions of the system. Therefore, our method can be applied to inverse problems with uncertain boundary conditions, which has received less attention in previous research. In terms of physical domain, the proposed algorithm can be applied to any geometry, regardless of its complexity. The algorithm operates in a high-dimensional parameter space and can accommodate data that are sparsely distributed across extensive domains.

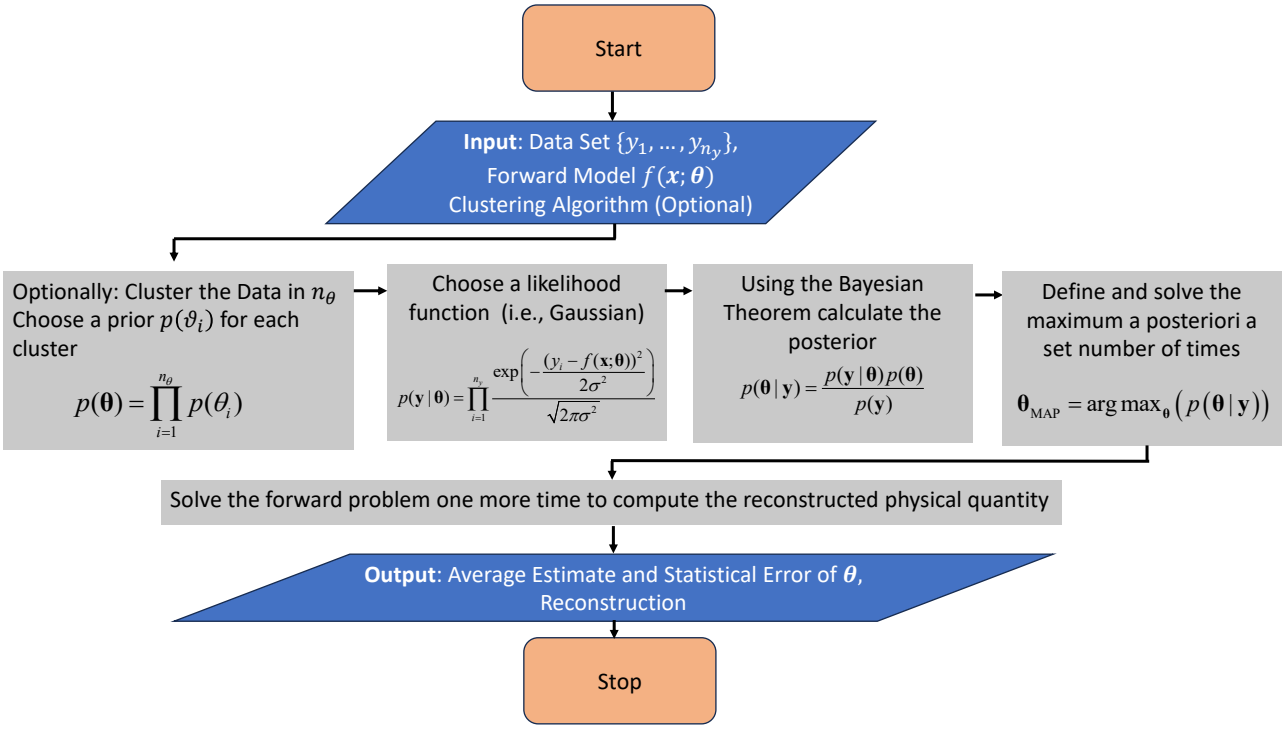


Figure 1: Flowchart of our algorithm

- If data $\mathbf{y}$ are sparsely distributed, clustering can be applied to group the data points, leading to differentiated analysis for each group. The selection of clustering algorithms should be considered judiciously to avoid bias and errors in estimating boundary conditions. For clustering in our examples, we use *k-means* [29, 30] and the Silhouette method [31] to find the number of clusters. More information is presented in Appendix A.

- Depending on the physical system under consideration, different forward models, likelihood functions, and prior distributions may be appropriate. Although normal priors are assumed in this paper, alternative distributions, such as exponential or gamma, may be more suitable for other problems.

- The optimization algorithm chosen to maximize the posterior function should align with the complexity and accuracy requirements of the problem at hand. A popular choice is to use stochastic optimization methods [32], such as dual annealing [33] or genetic algorithms [34]. These methods are known for their ability to handle complex problems and provide robust solutions. For more information, see Appendix B.

- We consider Gaussian noise for our likelihood function and a constant noise term σ. However, the noise level can also be treated as a random parameter within the Bayesian framework, allowing us to estimate the noise in conjunction with the model parameters.

- Our algorithm is constructed to operate within a complete Bayesian framework. It can accommodate clustering if necessary and, despite the increased computational demand, is designed to solve the forward model directly.

- Surrogate models [35] can be used to reduce computational costs but can affect precision depending on the quality of the approximation.

Overall, the proposed algorithm is versatile and can be applied to various domains, adapting to different conditions and data distributions. It provides a comprehensive approach for exploring the statistical parameters of unknown quantities within a full Bayesian context. Furthermore, one of the biggest advantages of our approach, which we will focus on with the examples we present, is that it is applicable if we have sparse data in a large domain, even if they are irregularly distributed in it or noisy. The application of the algorithm will be presented in Section 4 where synthetic sparse data, which represent actual physical measurements, will be produced.

3. The Forward Problem

3.1. The Finite Element Method in the unconstrained problem

3.1.1. Mathematical Details

Here we present the forward problem that governs the MF reconstruction. We compute the MF (denoted as $\mathbf{B}$) numerically in a proper 3D geometry given specific boundary conditions with the Finite Element Method (FEM) [36, 37], by dividing our system into smaller and simpler parts, the finite elements. To implement our partial differential equation (PDE) problem in numerical code, we use the popular *FEniCSx* Python package [38, 39, 40]. The MF is considered time independent and there are no sources, so the system of Maxwell equations reduces to the solution of a Laplace equation, also imposing the constraint $\nabla \cdot \mathbf{B} = 0$. We approach the forward problem in two variants: an unconstrained scenario where the Poisson equation is solved directly, and a constrained scenario where the solution is subject to the divergence-free condition of the magnetic field.

In the first scenario, we consider the Poisson PDE with a source term $\boldsymbol{\rho}(\mathbf{x})$ [1] and accompanied with boundary conditions (BC) of the Dirichlet type:

$$\begin{aligned} -\nabla^2 \mathbf{B}(\mathbf{x}) &= \boldsymbol{\rho}(\mathbf{x}), \qquad \mathbf{x} \in \boldsymbol{\Omega} \\ \mathbf{B}(\mathbf{x}) &= \mathbf{f}(\mathbf{x}), \qquad \mathbf{x} \in \partial\boldsymbol{\Omega}. \end{aligned} \tag{7}$$

We use the Cartesian coordinate system with $\mathbf{x} = (x, y, z)$ as the coordinate vector. Our computational domain Ω or mesh or geometry of our problem will be a proper 3D object like a cone and $\partial\Omega$ is its' boundary.

This problem is equivalent to the functional minimization of the following action J:

$$J(\mathbf{B}) = \frac{1}{2} \int_\Omega d\mathbf{x} \, |\nabla \mathbf{B}|^2 - \int_\Omega d\mathbf{x} \boldsymbol{\rho} \cdot \mathbf{B} \tag{8}$$

where, $\nabla\mathbf{B}$ is the gradient of a vector [2].

The FEM is applied to convert the PDE for our physical system into a variational problem [36, 37]. To achieve this, we multiply the PDE with a test function $\upsilon^k(\mathbf{x})$, which vanishes at the boundary, and integrate over the domain Ω:

$$-\int_\Omega d\mathbf{x} \left(\nabla^2 B^k(\mathbf{x})\right) \upsilon^k(\mathbf{x}) = \int_\Omega d\mathbf{x} \rho^k(\mathbf{x}) \upsilon^k(\mathbf{x}), \quad \mathbf{x} \in \Omega, \quad k = x, y, z. \tag{10}$$

In FEM terminology, $\boldsymbol{\upsilon}(\mathbf{x}) \equiv (\upsilon^x(\mathbf{x}), \upsilon^y(\mathbf{x}), \upsilon^z(\mathbf{x}))$ is called a test function and lies in a vector function space $\hat{V}$, while the unknown function $\mathbf{B}(\mathbf{x})$ is called a trial function and lies in a possible different vector function space V. Integration by parts on the first-hand side of this equation gives us the following equation (the surface term that is also produced vanishes because $\boldsymbol{\upsilon}(\boldsymbol{x})$ vanishes in the boundary):

$$\int_\Omega d\mathbf{x} \nabla B^k(\mathbf{x}) \cdot \nabla \upsilon^k(\mathbf{x}) = \int_\Omega d\mathbf{x} \rho^k(\mathbf{x}) \upsilon^k(\mathbf{x}), \quad \mathbf{x} \in \Omega, \quad k = x, y, z. \tag{11}$$

This is the variational or weak form of our original problem (see Eq. (7)). The trial and test spaces in this problem are the following:

$$V = \left\{ H^1(\Omega) : \boldsymbol{\upsilon}(\mathbf{x}) = \mathbf{f}(\mathbf{x}) \text{ for } \mathbf{x} \in \partial\Omega \right\} \tag{12}$$

and

$$\hat{V} = \left\{ H^1(\Omega) : \boldsymbol{\upsilon}(\mathbf{x}) = 0 \text{ for } \mathbf{x} \in \partial\Omega \right\}. \tag{13}$$

In the above equations $H^1(\Omega)$ is the Sobolev space containing functions $\boldsymbol{\upsilon}$ so both $\boldsymbol{\upsilon}$ and $\nabla\boldsymbol{\upsilon}$ are square integrable. To solve the problem of Eq. (11) we have to transform it from a continuous variational problem to a discrete one by introducing finite-dimensional test and trial spaces $V_h \subset V$ and $\hat{V}_h \subset \hat{V}$, respectively, and the choice of the said spaces follows directly from the finite elements we want to use in our solution. Assume that we have a basis $\left\{\phi_j\right\}_{j=1}^N$

[1] Even though , in our application, we have no sources we keep the discussion here quite general. Note also that we assume that the MF is time independent

[2] For a general vector, say $\mathbf{u}$, $\nabla\mathbf{u}$ is its' gradient i.e. a matrix / tensor ($\boldsymbol{e_i e_j} \frac{\partial u_j}{\partial x_i}$ in Cartesian notation) or simply the Jacobean. So:

$$|\nabla \mathbf{u}|^2 = \mathrm{tr}\left((\nabla \mathbf{u})^2\right) = \sum_{j=1}^{3} \frac{\partial u_j}{\partial x_j} \frac{\partial u_j}{\partial x_j} = \left(\frac{\partial u_x}{\partial x}\right)^2 + \left(\frac{\partial u_y}{\partial y}\right)^2 + \left(\frac{\partial u_z}{\partial z}\right)^2 \tag{9}$$

for V_h and a basis $\left\{\hat{\phi}_j\right\}_{j=1}^{N}$ for $\hat{V}_h$ where N denotes the dimension of the spaces mentioned above, which is equal to the number of nodes in the FEM scheme. Although the analytical form of the basis depends on the actual finite element, we use the fact that the value of ϕ_j is one in the j node and zero in the others. If we denote the coordinates of each node as $\mathbf{p}_i$, then:

$$\phi_j\left(\mathbf{p}_i\right) = \delta_{ji} = \begin{cases} 1, j = i \\ 0, j \neq i \end{cases} \tag{14}$$

where δ_{ji} is the Kronecker delta.

We now make an ansatz for discretizing $\mathbf{B}$, denoted as $\mathbf{B}_h \in V_h$:

$$\mathbf{B}_h = \left(\sum_{j=1}^{N} U_j^x \phi_j, \sum_{j=1}^{N} U_j^y \phi_j \sum_{j=1}^{N} U_j^z \phi_j\right), \;\; U_j^{x,y,z} = B_h^{x,y,z}\left(\mathbf{p}_j\right), \;\; \mathbf{p}_j \in \text{nodes} \tag{15}$$

where for each component we split the sum in the inner and boundary nodes:

$$\sum_{j=1}^{N} U_j^k \phi_j = \sum_{j=1}^{N_{\text{in}}} U_j^k \phi_j + \sum_{j=1}^{N_{\text{bn}}} U_j^k \phi_j, k = x, y, z \tag{16}$$

where $\left(U_j^x, U_j^y, U_j^z\right) \in \mathbb{R}^{3N}$ is the vector of degrees of freedom to be computed and we split the sum in the inner and boundary nodes, with N_{in} being the number of the former and N_{bn} the number of the latter ($N_{\text{in}} + N_{\text{bn}} = N$). However, from the Dirichlet boundary conditions we can immediately find the unknowns U_j^k that correspond to the boundary nodes:

$$B_h^k\left(\mathbf{p}_j\right) = f^k\left(\mathbf{p}_j\right) = U_j^k, \mathbf{p}_j \in \text{boundary nodes} \;\; k = x, y, z.. \tag{17}$$

Finally, if we choose

$$\boldsymbol{\upsilon} = \left(\phi_i, \phi_i, \phi_i\right), \quad i = 1, N_{\text{in}} \tag{18}$$

from Eq. (11) we can assemble the following system of algebraic equations:

$$\sum_{j=1}^{N_{\text{in}}} U_j^k \int_{\Omega} d\mathbf{x} \nabla\phi_j \cdot \nabla\phi_i = \int_{\Omega} d\mathbf{x} \rho^k(\mathbf{x}) \phi_i - \sum_{j=1}^{N_{\text{bn}}} f_j^k \int_{\Omega} d\mathbf{x} \nabla\phi_j \cdot \nabla\phi_i, \;\; i = 1, N_{\text{in}}, \;\; k = x, y, z. \tag{19}$$

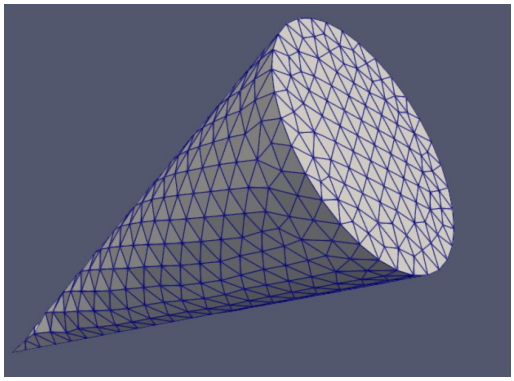
The resulting algebraic system includes the stiffness matrix derived from the inner products of basis function gradients:

$$A_{ij} = \int_{\Omega} d\mathbf{x} \nabla\phi_i \cdot \nabla\phi_j. \tag{20}$$

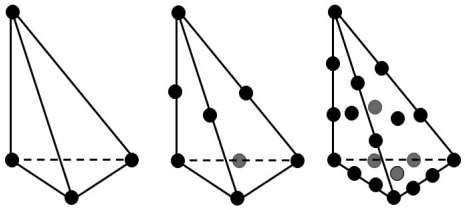
The solution of the above system provides the coefficients U_j^k, thus obtaining the solution $\mathbf{B}_h$ throughout the domain. There are many ways to solve Eq. 19 numerically. In our case, we opted to use a direct solver (LU-factorization) [41].

3.1.2. The Geometry and the Finite Element Method Basis

For the geometric representation of our cone domain, we use a mesh divided into tetrahedral elements (see Fig. 2a): we divide the cone into triangular pyramids [42]. The solution will be calculated as points in the pyramids (nodes). We will consider 3 cases for the elements: $\mathbb{P}_1$, where the nodes are in the vertices, $\mathbb{P}_2$, where the nodes are on the vertices and in the center of each side of the triangles, and $\mathbb{P}_3$, where on the vertices, one on each third or each side and one in the center of each triangle (see Fig. 2b). These elements are also called Lagrange elements [42, 43, 44]. In our case, vector-valued (or generally tensor-valued) Lagrange elements are constructed by using a Lagrange element for each component.



(a) An example of our cone geometry



(b) The finite elements that we use. From left to right: $\mathbb{P}_1$, $\mathbb{P}_2$ and $\mathbb{P}_3$ triangle pyramids.

Figure 2: The physical-computational domain of our problem (left figure) and the Finite Elements used (rigth figure)

For the basis that we use in our calculation, see Refs. [42, 45]. For example, the basis functions for a $\mathbb{P}_1$ tetrahedron with nodes at the points $(0,0,0)$, $(1,0,0)$, $(0,1,0)$, $(0,0,1)$ are as follows:

$$\left\{ \begin{array}{c} \phi_0 = 1 - x - y - z \\ \phi_1 = x \\ \phi_2 = y \\ \phi_3 = z \end{array} \right\} \tag{21}$$

This basis respects the property of Eq. (14). Before substituting it into Eq. (15) or (19), we have to apply a coordinate transformation to account for the position and orientation of the tetrahedron in the physical mesh.

In cases of systems involving multiple PDEs, such as a vector and a scalar function, a mixed function space is employed, with the popular Taylor-Hood element $\mathbb{P}_k - \mathbb{P}_{k-1}$ being a common choice for such scenarios. This element allows for a distinct approximation of vector and scalar fields, ensuring accuracy and compliance with the constraints of the governing equations.

3.1.3. Example

For the boundary conditions $\mathbf{f}(\mathbf{x}) \equiv \mathbf{B}_{\mathrm{BC}}(\mathbf{x})$ we, generally, consider two cases, a purely analytical one with boundary conditions the respect the equation $\nabla \cdot \mathbf{B} = 0$, for example:

$$\mathbf{B}_{\mathrm{BC}}(\mathbf{x}) = (10x + y - z, \quad x - 15y + z, \quad x - y + 5z) \text{ for } \mathbf{x} = (x, y, z) \in \partial\Omega. \tag{22}$$

We solve the problem in a cone geometry with its vertex at the origin $(0,0,0)$, height of 1 and base radius of 0.25 and by using the finite element $\mathbb{P}_1$. The results are shown in Figures 3a,b. This finite element was also used to check the accuracy of our solver in cases that can be solved analytically. The results shown in Figures 3a and 3b illustrate the magnitude and vector direction of the MF, respectively. Furthermore, Figures 3c and 3d provide information on the divergence of the MF and the relationship between the absolute value of the divergence and the magnitude of the MF. The color gradients in these figures correspond to different magnitudes of the MF, with the divergence nearing $\nabla \cdot \mathbf{B} \sim 10^{-3}$, indicating a low divergence, as expected from the boundary condition that enforces $\nabla \cdot \mathbf{B} = 0$. However, as we will see in the next example, this value can be reduced further if we consider a constrained calculation where the constraint $\nabla \cdot \mathbf{B} = 0$ is taken into account in the FEM scheme.

Another case of interest is if one or more boundary conditions are generated from one of more normal distributions $\mathcal{N}(m, s)$ where m is the mean value and s is the standard deviation of each distribution. However, this calculation will be presented in Section 4, when we discuss the generation of the synthetic sparse data.

3.2. Constrained Minimization

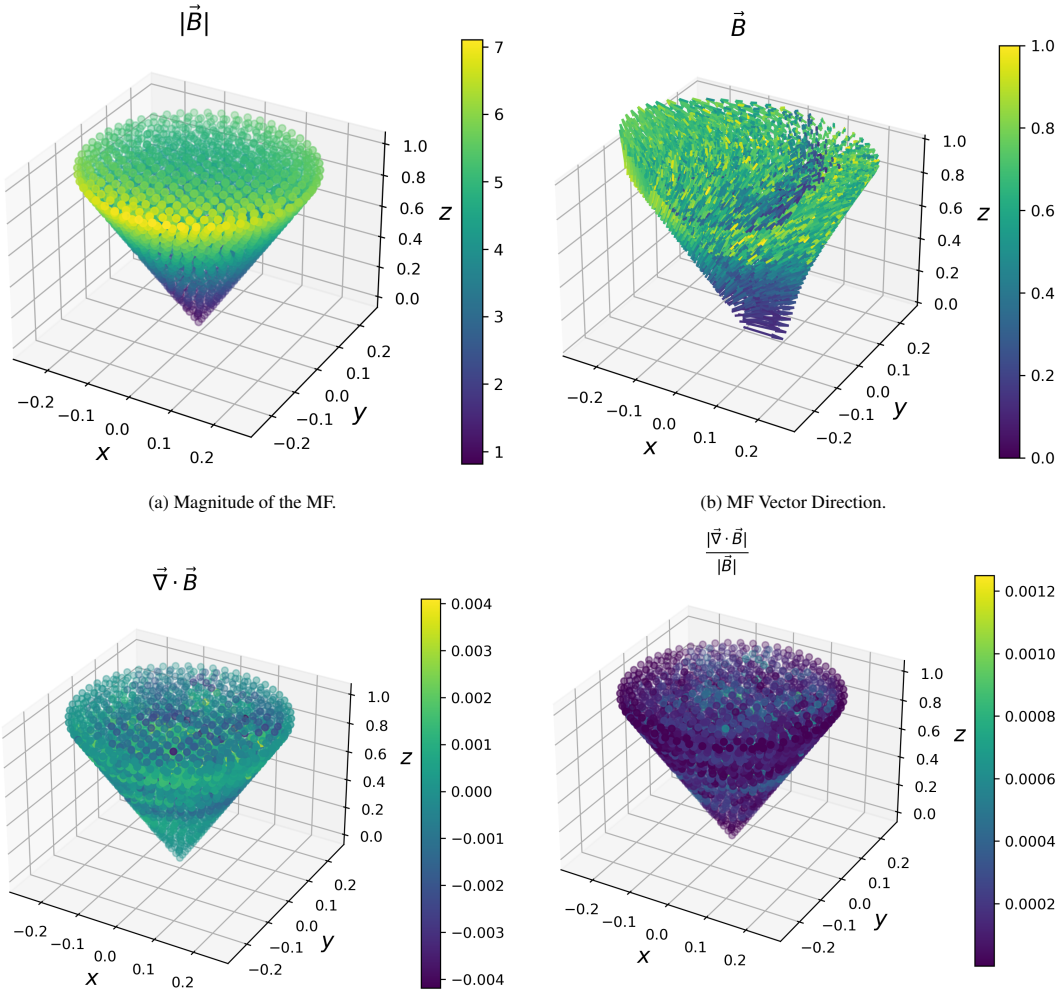
3.2.1. Formulation of the Forward Constrained Problem

For the solution to represent an actual Magnetic Field we have to ensure that our solution obeys Maxwell's equation $\nabla \cdot \mathbf{B} = 0$. This means that we have to solve the Poisson equation while ensuring that the aforementioned condition holds. The above problem is equivalent to the functional minimization of the following action J:

$$J(\mathbf{B}, \lambda) = \frac{1}{2} \int_\Omega d\mathbf{x} \, |\nabla \mathbf{B}|^2 - \int_\Omega d\mathbf{x} \boldsymbol{\rho} \cdot \mathbf{B} + \int_\Omega d\mathbf{x} \lambda \nabla \cdot \mathbf{B} \tag{23}$$

where λ is a Lagrange multiplier. Minimizing with respect to $\mathbf{B}$ gives

$$-\nabla^2 \mathbf{B}(\mathbf{x}) = \nabla\lambda + \boldsymbol{\rho}(\mathbf{x}), \quad \mathbf{x} \in \Omega \tag{24}$$



(a) Magnitude of the MF.

(b) MF Vector Direction.

(c) Divergence of the MF.

(d) The ratio of the absolute value of the divergence of the MF to its Magnitude.

Figure 3: Calculation of MF for the unconstrained case ($\nabla^2\mathbf{B}(\mathbf{x}) = 0$) with analytical BC calculated from Eq. (22) in a 3D cone domain. We plot the magnitude of the MF (Fig. a), its vector (Fig. b), its divergence (Fig. c), and the ratio of the magnitude of the divergence to the magnitude of the MF (Fig. d). We can see that $\nabla \cdot \mathbf{B} \sim 10^{-3}$.

while minimization with respect to λ gives back the constrain

$$\nabla \cdot \mathbf{B}(\mathbf{x}) = 0. \tag{25}$$

The rigorous proof is shown in Appendix C.

This problem is similar to the simple steady-state Navier-Stokes problem [46]:

$$\begin{aligned} -\nabla^2\mathbf{u} - \nabla P &= \boldsymbol{\rho} \\ \nabla \cdot \mathbf{u} &= 0 \end{aligned} \tag{26}$$

accompanied, of course, with proper boundary conditions. In this equation $\mathbf{u}$ is the velocity and pressure P plays the role of the Lagrange multiplier.

3.2.2. Finite Element Method applied in the constrained minimization problem

From the results of the previous Section we can summarize the PDE problem as follows:

$$\begin{aligned} -\nabla^2\mathbf{B}(\mathbf{x}) &= \nabla\lambda(\mathbf{x}) + \boldsymbol{\rho}(\mathbf{x}), \quad \mathbf{x} \in \boldsymbol{\Omega} \\ \nabla \cdot \mathbf{B} &= 0, \qquad \mathbf{x} \in \boldsymbol{\Omega} \\ \mathbf{B}(\mathbf{x}) &= \mathbf{f}(\mathbf{x}), \quad \mathbf{x} \in \partial\boldsymbol{\Omega}. \end{aligned} \tag{27}$$

The boundary condition for the Lagrange multiplier λ can be found from the first PDE of this system up to a constant. To solve the problem with FEM, we need to find the variational form of the above system. To define the

test and trial functions, we use a mixed functional space. Consider the vector function spaces (12) and (13) and the function space:

$$Q = \left\{ q \in L^2(\Omega) \,\middle|\, \int_\Omega d\mathbf{x} q = 0 \right\} \tag{28}$$

where L^2 is the space of square-integrable functions. Our solution we have that $(\mathbf{B}, \lambda) \in W$. The space W is a mixed (product) function space $W = V \times Q$, such that $\mathbf{B} \in V$ and $\lambda \in Q$. In a similar manner for the trial functions we have $(\boldsymbol{\upsilon}, q) \in \hat{W} = \hat{V} \times Q$. We multiply the first PDE of the system shown in Eq. (27) with the test function $\boldsymbol{\upsilon}$ and the second one with the test function q. Similarly to Section 3.1.1 we have:

$$\begin{aligned} \int_\Omega d\mathbf{x} \nabla B^k(\mathbf{x}) \cdot \nabla \upsilon^k(\mathbf{x}) + \int_\Omega d\mathbf{x} \lambda(\mathbf{x}) \partial_k \upsilon^k(\mathbf{x}) = \int_\Omega d\mathbf{x} \rho^k(\mathbf{x}) \upsilon^k(\mathbf{x}), \quad k = x, y, z \\ \int_\Omega d\mathbf{x} \nabla \cdot \mathbf{B}(\mathbf{x}) q(\mathbf{x}) = 0, \quad \mathbf{x} \in \Omega. \end{aligned} \tag{29}$$

The ansatz for the Magnetic Field $\mathbf{B}_h \in V_h$ is the same as that for the unconstrained case and is given in Eq. (15). In a similar manner, for the Lagrange multiplier we consider a basis $\left\{\varphi_j\right\}_{j=1}^M$ for $Q_k \subset Q$ and the relevant expansion is:

$$\lambda = \sum_{j=1}^M L_j \varphi_j \tag{30}$$

If we substitute this equation in the system (29) and set $\upsilon^k = \phi_i,\ i = 1, N$ and $q = \varphi_i,\ i = 1, M$ we find:

$$\begin{aligned} \sum_{j=1}^N A_{ij} U_j^k + \sum_{j=1}^M C_{ij} L_j = \int_\Omega d\mathbf{x} \rho^k(\mathbf{x}) \phi_i, \quad i = 1, N, \quad k = x, y, z \\ \sum_{k=x,y,z} \sum_{j=1}^N C_{ji}^k U_j^k = 0, \quad i = 1, M. \end{aligned} \tag{31}$$

where the $N \times N$ stiffness matrix $\mathbf{A} = \left\{A_{ij}\right\},\ i, j = 1, N$ is given by Eq. (20) and we have also introduced the following $N \times M$ matrix:

$$\mathbf{C} = \begin{Bmatrix} C_{ij}^x \\ C_{ij}^y \\ C_{ij}^z \end{Bmatrix}, \quad i = 1, ..., N, \quad j = 1, ..., M \tag{32}$$

where

$$C_{ij}^k = \int_\Omega d\mathbf{x} \partial_k \phi_i \varphi_j. \tag{33}$$

So, if we assemble the system in matrix form, we have:

$$\begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{0} & \mathbf{C^x} \\ \mathbf{0} & \mathbf{A} & \mathbf{0} & \mathbf{C^y} \\ \mathbf{0} & \mathbf{0} & \mathbf{A} & \mathbf{C^z} \\ (\mathbf{C^x})^\perp & (\mathbf{C^y})^\perp & (\mathbf{C^z})^\perp & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{U^x} \\ \mathbf{U^y} \\ \mathbf{U^z} \\ \mathbf{L} \end{bmatrix} = \begin{bmatrix} \left\{\rho_i^x \phi_i\right\} \\ \left\{\rho_i^y \phi_i\right\} \\ \left\{\rho_i^z \phi_i\right\} \\ \mathbf{0} \end{bmatrix} \tag{34}$$

where $\mathbf{C}^k = \left\{C_{ij}^k\right\}, \mathbf{U}^k = \left\{U_j^k\right\},\ k = x, y, z,\ i = 1, ..., N,\ j = 1, ..., M$.

For the finite elements, we opted to use the Taylor-Hood ones that we mentioned in Section 3.1.2, namely the $\mathbb{P}_3 - \mathbb{P}_2$ one. So, in the expansions of each component of $\mathbf{B}(\mathbf{x})$ we use the $\mathbb{P}_3$ tetraedron, and in the Lagrange multiplier expansion we use the $\mathbb{P}_2$ tetraedron. This finite element is a popular choice in the solution of the Navier-Stokes problem [47] however, the construction of finite elements that are divergence-free is a subject of active research [48].

The system (34) is a linear system of saddle-point type [49]. To solve it, we use a preconditioner that generally transforms a linear system into another system with more favorable properties for linear solvers. For finite element

problems a usual choice would be a block diagonal matrix defined as $\mathbf{P} = \text{diag}\{\mathbf{A}, \mathbf{A}, \mathbf{A}, \mathbf{M}\}$ where $\mathbf{M}$ is a mass matrix with elements:

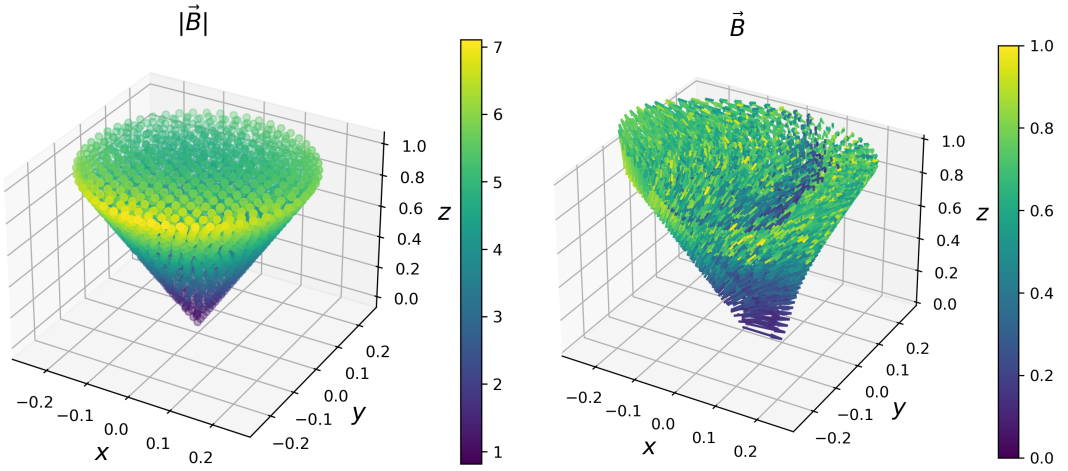
$$M_{ij} = \int_{\Omega} d\mathbf{x}\phi_i\varphi_j \tag{35}$$

After applying the above transformation the new system is solved with the minimal residual method [50].

3.2.3. Example

In this example, we revisit the calculations from Section 3.1.3, this time incorporating the divergence-free constraint $\nabla \cdot \mathbf{B} = 0$ as a fundamental aspect of our model from the outset, using the Taylor-Hood element $\mathbb{P}_3 - \mathbb{P}_2$. The results, illustrated in Figures 4, demonstrate the impact of this constraint on the calculation.

Comparing the constrained results with those of the unconstrained case (Figures 3), we observe a marked improvement in adherence to the divergence-free condition. Figures 4c and 4d show a significant reduction in divergence values across the domain, confirming the effectiveness of the constraint in preserving the physical precision of our model. This example underscores the importance of incorporating such constraints into computational models to ensure that the results are not only mathematically robust but also physically meaningful.



(a) Magnitude of the MF.

(b) MF Vector Direction.

(c) Divergence of the MF.

(d) The ratio of the absolute value of the divergence of the MF to its Magnitude.

Figure 4: Calculation of MF for the constrained case, (where we impose the constrain $\nabla \cdot \mathbf{B} = 0$) with analytical BC calculated from Equation 22 in a 3D cone domain. We plot the magnitude of the MF (Fig. a), its vector (Fig. b), its divergence (Fig. c) and the ratio of the magnitude of the divergence to the magnitude of the MF (Fig. d). We can see that $\nabla \cdot \mathbf{B} \sim 10^{-5}$ which is an improvement over what we can see in Fig 3.

4. Reconstructing the Magnetic Field in a Specific Domain

4.1. The Reconstruction process

In this section, we present an application of our proposed algorithm. Using synthetic data and the solution of the forward problem, we reconstruct the MF in an arbitrarily given domain. To simulate the sparse direct observational data of the MF, we start by creating synthetic data from a simulation using the forward problem in the cone domain Ω for the following boundary conditions. As a first example, we consider the following problem:

$$\mathbf{B}_{\mathrm{BC}}(\mathbf{x}) = \Big(\mathcal{N}(m, s), \quad 2y - 5z, \quad 10y - 2z\Big) \ \text{ for } \mathbf{x} = (x, y, z) \in \partial\Omega. \tag{36}$$

In the following, since in the actual physical problem we may not know the true boundary conditions but rather we only have access to uncertain measurements, we choose to use BCs as above, that is, consisting of one or more normal distributions for the x-component. So, as shown in Eq. (36), our first application will be with a normal distribution with mean $m = 10$ and standard deviation $s = 0.5$. We will also consider the case where the cone is divided into planes and we have a different normal distribution generating the BC in each one. Although we focus on normal distributions for the x-component in this case, our approach is capable of handling BCs derived from any probability density function (PDF), catering to the diverse and complex nature of actual measurement methods.

Having solved the forward problem and obtained a set of data, to simulate irregularly distributed sparse data, we randomly remove a percentage of the points from the aforementioned set of data. The remaining points represent the sparse direct observational data of the x component of the MF that we will use in the application of our algorithm. It is important to note that for simplicity, we only used normal distributions for the x component of the boundary conditions in this simulation. In a more realistic scenario, the above scenario may be applied to all components of the MF or, alternatively, separately to the strength and direction of the field.

We can now proceed with the application of the inverse problem algorithm, as was introduced in Section 2.3. The unknown parameters θ_i are the mean values of the x-component of the MF at the boundary of the cone while the input is the simulated synthetic sparse data for the x-component of the MF, say $y_i, \ \ i = 1, ..., n_y$, the forward problem, $f(\mathbf{x}; \boldsymbol{\theta})$, which was analyzed in Section 3 (in this application we consider the unconstrained mode), the noise term that characterizes the measurement error (we choose $\sigma = 1$) and a clustering algorithm.

In order to maintain a clear focus on the main contributions of the article while providing full transparency of our method, the detailed steps of the example application of our algorithm are presented in Appendix D. This includes a step-by-step account of the clustering approach, the definition and minimization of the cost function (which is the weighted sum of the differences between the data set values minus the result of the forward problem), the optimization process, and the calculation of residuals, culminating in the reconstruction of the magnetic field within the entire domain of interest. These supplementary steps exemplify the robustness and applicability of our algorithm in various scenarios and ensure that interested readers can replicate or build on our work. Here, we only focus on the general key points: We have one prior for each possible cluster and we consider each to be a normal one; therefore, the total prior is their product. We also consider Gaussian likelihood probability. The likelihood and posterior functions are defined from Eqs. (4) and (3), respectively, while the maximum a posteriori problem is defined from equation (6).

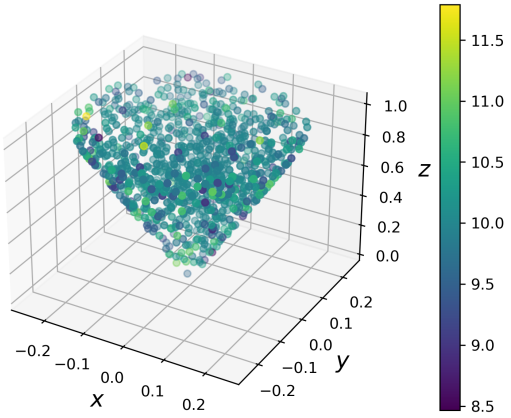
Our proposed inverse problem algorithm should then recover the mean values of these distributions. So, for example, if the synthetic data were created using Eq. (36) we should find, for θ, a value close to 10. A solution of the forward model gives the MF in the whole cone domain, thus completing the reconstruction calculation.

We move on different test cases demonstrating the problem of reconstructing the magnetic field within a given domain while also elaborating the computational efficiency of our algorithm.

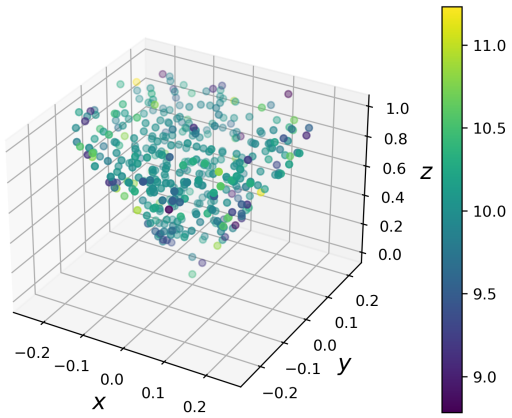
4.2. Test case 1: Single prior data

In our first example, we focus on boundary conditions (BCs) for the x-component of the MF generated by a single distribution, as detailed in Eq. (36), and employing an unconstrained forward model (refer to Section 3.1.1). The domain of interest is a cone with its apex at the origin $(0, 0, 0)$, the height of 1 and the base radius of 0.25. After trying various finite elements, we chose the $\mathbb{P}_1$ ones, since it gives accurate results in this problem. When solving the forward problem within this geometry, we simulate sparse data by randomly omitting a percentage of the approximately 3000 data points within the domain, as shown in Figure 5.

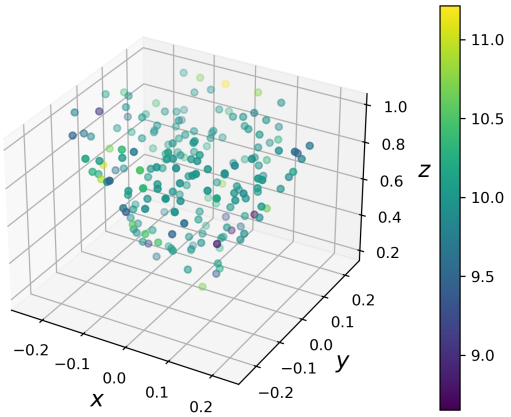
The synthetic data visualized in Figure 5 illustrates various degrees of data sparsity, ranging from 75% to 99% of random data removal. These subfigures effectively demonstrate the progression of data reduction and its implications for the reconstruction process. In each case, our aim was to reconstruct the x-component of the MF from these increasingly sparse datasets, and we applied our algorithm four times, each for each data set shown in the aforementioned subfigures.



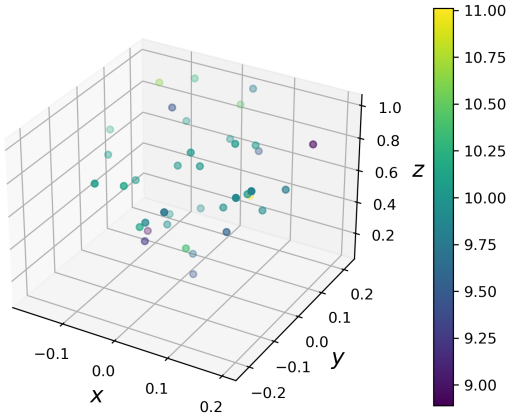
(a) 75% of the data removed.



(b) 90% of the data removed.

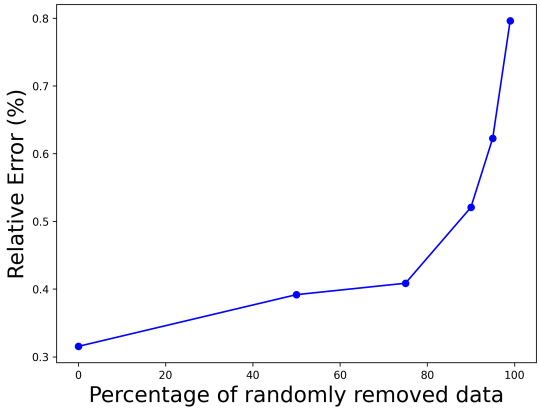


(c) 95% of the data removed.

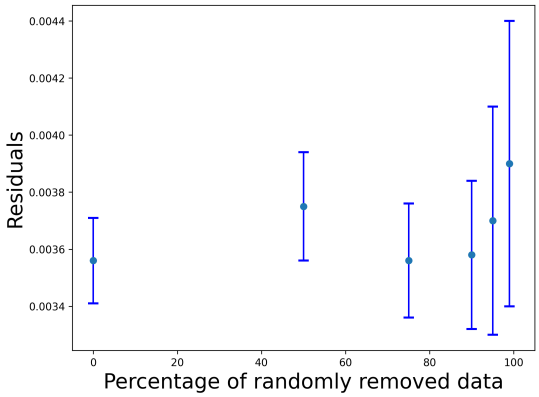


(d) 99% of the data removed.

Figure 5: Synthetic data of the x-component of the MF to be used in the data driven reconstruction calculation as calculated form the unconstrained case forward model with the BCs from Equation 36. In addition, in each subfigure, a different percentage of the data has been randomly removed to simulate a case of "reconstruction from sparse data".



(a) Mean Value and statistical error of the x-component of the MF in the boundary as a function of the sparsity of the data.



(b) The residuals and their statistical error of the calculation of the left plot as a function of the sparsity of the data.

Figure 6: Mean Value of the x-component of the MF in the boundary and the corresponding residuals as a function of the percentage of the data that was randomly removed from the cone domain. The data used in this plots are presented in Table E.1 of Appendix E.

Fig. 6 delves into the quantitative analysis of our reconstruction process. Fig. 6a presents the mean value and associated statistical error of the reconstructed x component at the boundary, influenced by the level of data sparsity. A notable trend is observed where the standard deviation error escalates in tandem with the increase in data removal, underscoring the challenges of reconstruction from sparse data. Similarly, Fig. 6b charts the residuals from the reconstruction, offering a clear visualization of the error's statistical spread, again emphasizing

the impact of data sparsity.

These results confirm the robustness of our algorithm in recovering the initial value of $\theta = 10$ (the results for each sparsity case are summarized in Table E.1 of Appendix E.), although it is evident that data sparsity introduces greater uncertainty, as reflected by the widening error margins. The completion of the reconstruction is achieved by resolving the forward problem with the estimated θ, culminating in a comprehensive reconstruction despite the initial data scarcity (for an example, see Fig. F.10a in Appendix F).

As we reflect on the computational aspects of our proposed algorithm, it is evident that its efficiency is intricately linked to several factors. These include the volume encompassed by the cone domain, the granularity of the solution to the forward problem dictated by the number of finite elements within the cone, and the choice of optimization technique employed for the minimization of the negative logarithm of the posterior distribution. In the execution of the results presented earlier, the algorithm's computational demand was such that each iteration of Step 4 required approximately 10 minutes on a single core of a contemporary PC. The structure of the algorithm lends itself to parallel processing, allowing each iteration, as stipulated in Step 5, to be independently executed on separate threads, thus enhancing computational throughput. The choice of optimization method is also a critical determinant of performance. Our observations reveal that the use of dual annealing significantly reduces the run time to around 10 minutes, as opposed to the 30 minutes necessitated by differential evolution, while maintaining a comparable level of accuracy in the results. This indicates a clear preference for dual annealing in terms of computational efficiency in this specific context. The above computations were performed on a workstation with an AMD Ryzen Threadripper PRO 5995WX Processor, reaching 2.70 GHz to 4.50 GHz, and equipped with 256GB of DDR4 3200MHz memory. The program's computational intensity lies primarily with the CPU rather than memory, enabling efficient execution across platforms with robust multi-threading capabilities. On a consumer-grade Intel i7 10th generation laptop, the computational times were marginally longer.

Furthermore, we probe the scalability of our algorithm by expanding the cone's volume twofold, keeping all other parameters constant. Each iteration of Step 4 then extended to 30 minutes, suggesting a commendable scalability of our approach even as the geometry expands. This characteristic is particularly advantageous, underscoring the algorithm's potential for adaptation to larger and more complex domains without a prohibitive increase in computational requirements.

4.3. Test case 2: Multiple prior data

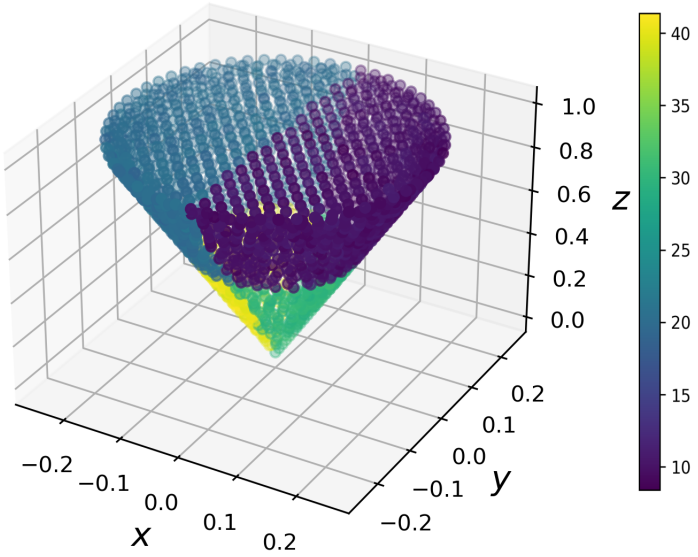
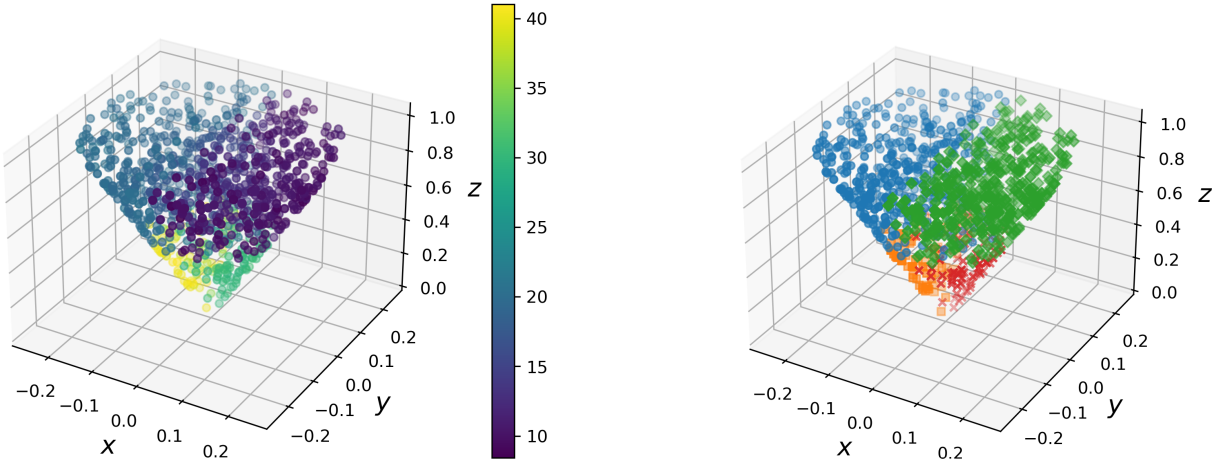


Figure 7: The BCs for the x-component in the case of four planes to be used in the construction of synthetic data of the x-component of the MF to be used in the data driven reconstruction calculation. They were generated from four normal distributions with different mean values and standard deviations.

We proceed considering the case of a problem with sparse data by randomly removing a percentage of the solution to simulate sparse data. The resulting data sets can be seen in the left subfigures of Fig. 8, that is, subfigures 8a, 8c and 8e, which correspond to randomly removing 75%, 90% and 95% of the initial data, respectively. As explained above, in the general case, we do not know how many distributions were responsible for the creation of the data. We want to associate those data in clusters, and we perform cluster analysis to find the number of clusters

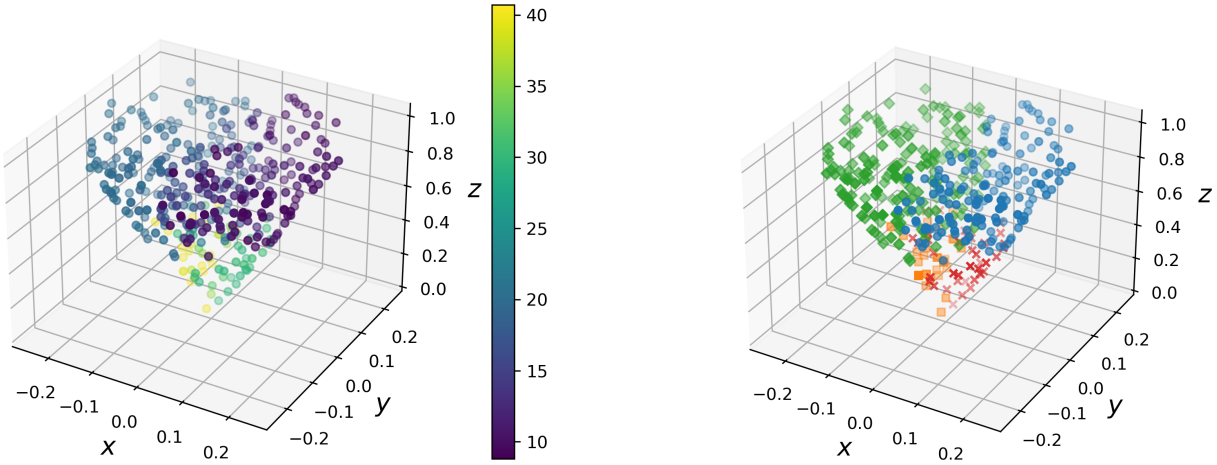
and place each data point in the respective cluster. As mentioned, our procedure is described in Appendix A and accurately predicts the number of clusters, four in this example; see the right subfigures of Fig. 8, namely 8b, 8d, and 8f. We define the regions in the boundary as follows: In the case of four clusters, we have four boundary regions.

Once the regions are identified, we can proceed to solving the inverse problem as discussed above.



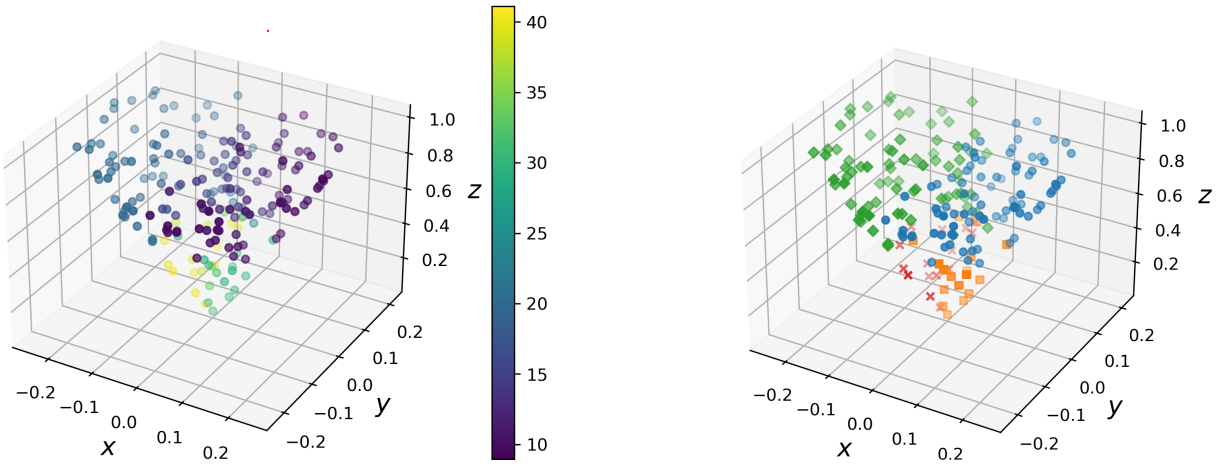
(a) 75% of the data removed.

(b) Clusters corresponding to the left plot.



(c) 90% of the data removed.

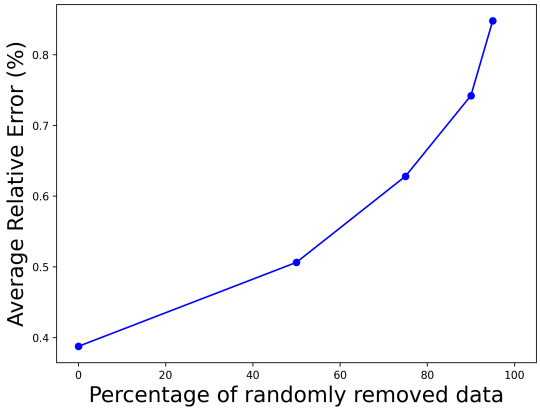
(d) Clusters corresponding to the left plot.



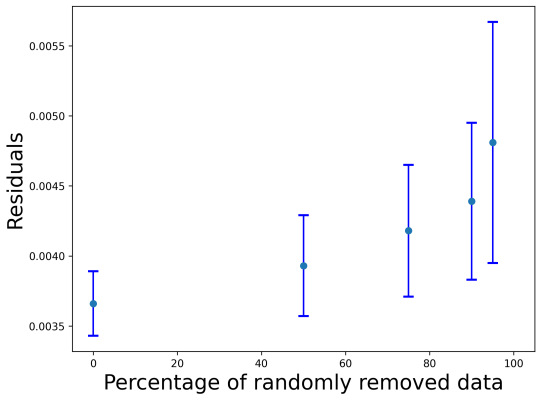
(e) 95% of the data removed.

(f) Clusters corresponding to the left plot.

Figure 8: Left Panels: Synthetic data of the x-component of the MF to be used in the data-driven reconstruction calculation as calculated from the unconstrained case forward model with the BCs from four different normal distributions (see main text). In each subfigure, a different percentage of data has been removed randomly to simulate a case of "reconstruction from sparse data". Right Panels: The corresponding clusters used in the first step of the algorithm.



(a) Average of the Relative Error in the calculation of the x-component of the MF as a function of the sparsity of the data.



(b) The residuals and their statistical error of the calculation of the four plane reconstruction as a function of the sparsity of the data.

Figure 9: Mean Value of the x-component of the MF in the boundary in the fourth plane, as it was defined in the main text, and the corresponding residuals as a function of the percentage of the data that was randomly removed from the cone domain. The data used in these graphs are presented in Table E.2 of Appendix E.

So, picking up from the meticulous cluster analysis, we advance to the inferential phase, where the four established priors are assumed to follow normal distributions, with unknown mean values and, in this example, standard deviation fixed at unity. This assumption is the basis for developing the likelihood and posterior distributions, which subsequently inform the maximum a posteriori (MAP) estimation. This MAP problem is once again solved with the help of dual annealing, as explained in Section 4.1, and, as mentioned, the whole process is analyzed in Appendix D. The application of the algorithm is performed three times for the three data sets shown in the right subfigures of Fig. 8. The scheme very accurately returns the values of $\boldsymbol{\theta} = (10\,, 20\,, 30\,, 40)$ for all sparsity cases considered (the results are summarized in Table E.1 of Appendix E).

Figs. 9a and 9b graphically represent the average of errors in each value and the residuals. The graphical exposition reveals an expected trend: as the data becomes sparser, the statistical error in the parameters $\boldsymbol{\theta}$ and the residuals increases, yet the initial $\boldsymbol{\theta}$ values remain robustly retrieved. The completion of the reconstruction is achieved by resolving the forward problem with the estimated $\boldsymbol{\theta}$ (for an example, see Fig. F.10b in Appendix F).

When addressing the computational demands of the algorithm, the narrative remains consistent with our previous discourse. In the same workstation that we used in the previous example, a single run of Step 4, devoid of parallel processing and employing dual annealing, unfolds over approximately 30 minutes. A shift to differential evolution as the optimization strategy sees this duration elongate to roughly 80 minutes. However, the results are comparably precise. This operational tempo is sustained even when the cone's height is doubled; the algorithm's runtime extends to a modest 90 min, a testament to its scalable nature. We have judiciously allocated separate processing threads for each repetition, an approach that underscores the algorithm's parallelizable potential.

In summation, the algorithm we propose stands validated by its ability to reliably ascertain the boundary conditions' statistical attributes under the stringent test of sparse data. Using the values that were successfully recovered, we can calculate the reconstructed field.

5. Conclusions and Perspectives

In this work, we have developed a comprehensive data-driven Bayesian methodology capable of deducing the statistical characteristics of boundary conditions from sparse data. This approach is tailor-made for reconstruction calculations in complex systems. The crux of our method lies in the implementation of a Bayesian framework that uses the MAP estimation to converge on the most probable parameter values given the observed data.

One of the primary strengths of our algorithm is its reliance on Bayesian inference coupled with MAP. This combination offers several advantages: It integrates prior knowledge about the parameters into the estimation process, provides a probabilistic measure of uncertainty, and helps overcome issues associated with ill-posed inverse problems. By focusing on the MAP estimation, our approach effectively balances the influence of the prior distribution with the likelihood of the observed data, leading to more robust and credible parameter recovery, especially in cases with limited or noisy data.

Furthermore, the versatility of our algorithm is evidenced by its compatibility with any PDF. This flexibility allows it to be applied to a broad spectrum of problems where the underlying PDFs of the boundary conditions may not be normally distributed. The ability to work with any PDF enhances the algorithm's applicability across various scientific fields and ensures that it can adapt to the unique distributions inherent to different physical phenomena.

Central to our method is a clustering algorithm that smartly groups sparse data points. This strategic grouping is essential to estimate the number of normal distributions that build the boundary conditions. Each cluster informs a prior distribution, precisely defined by the cluster's mean and standard deviation. Such a set-up enables a sophisticated parameter estimation through the use of stochastic optimization methods, notably dual annealing, which is particularly beneficial in situations where data availability is a luxury.

Delving into the results of our simulation studies, the algorithm's efficacy becomes clear. We successfully applied it to reconstruct the MF within a conical domain using synthetic data with a variable percentage of sparcity. Our results were encouraging, demonstrating the algorithm's ability to recover the mean values of the distributions defining the boundary conditions with high accuracy. These findings underscore the potential of our methodology to accurately infer unknown parameters and reconstruct the physical quantities of interest, even when the available data are sparse or unevenly distributed. In this work, our calculations, while centered on normal distributions for the x-component of MF boundary conditions, represent only a fraction of possible scenarios. The real test lies in expanding this method to include a broader spectrum of boundary conditions and evaluating its efficacy in real-world situations.

Looking forward, we aim to extend the application of our algorithm to an important astrophysical challenge: the reconstruction of the Milky Way's magnetic field [14]. Using sparse data from magnetohydrodynamic simulations and direct observational input [51], we will evaluate the quality of reconstructions afforded by our method. A particularly intriguing aspect of this future work will be the backtracking of ultra-high-energy cosmic rays [**?**] to their extragalactic origins, made possible by the accurate mapping of the galactic MF. In addition to this application a particularly intriguing aspect of a future work will be the exploration of Bayesian Machine Learning methods, as hinted at in recent works like [17].

In summary, the algorithm presented in this paper not only showcases the effective use of Bayesian inference with MAP in sparse data environments but also promises a wide range of applications in disciplines where understanding the nuances of boundary conditions is paramount. The adaptability and efficiency of our method open new avenues for research, offering potential advancements in fields far beyond the scope of this initial study.

Acknowledgments

G.E.P. acknowledges support from the Foundation for Research and Technology - Hellas Synergy Grant Program through the MagMASim project, jointly implemented by the Institute of Applied and Computational Mathematics and the Institute of Astrophysics.

Additionally, the algorithms and code utilized in this study will be made publicly available for the comminity on GitHub to facilitate further research and collaboration. Interested readers can access these resources at `https://github.com/gepavlou`.

Appendix A. Clustering

Appendix A.1. k-means

We perform the cluster analysis using the *k-means* method [29, 30]. The general algorithm behind the k-means method is as follows:

1. Choose the number of clusters (that is, 2).
2. Assume two random points anywhere near the data and consider them as the center of two clusters (centroids).
3. Find the distance of every data point from both centroids.
4. Assign every data point to the centroid to which it is nearest, hence making two clusters.
5. Calculate the center of both the formed clusters and shift the centroids there.
6. Go to Step 3 and repeat the process until there is no change in the clusters formed.

Obviously, it is not always clear how many clusters of data there are. The number of clusters that we choose for a given data set cannot be random. We find their number with the Silhouette Method.

Appendix A.2. The Silhouette Method

The Silhouette method [31] is a method to find the optimal number of clusters and to interpret and validate consistency within clusters of data. The Silhouette Method computes silhouette coefficients for each point, which measure how much a point is similar to its own cluster compared to other clusters by providing a visual graphical representation of how well each object has been classified. The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The value of the silhouette ranges

between $(-1, 1)$, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. If most objects have a high value, then the clustering configuration is appropriate. If many points have a low or negative value, then the clustering configuration may have too many or too few clusters.

To find the silhouette coefficient of the i-th point:

1. Compute the average distance of that point with all other points in the same cluster, $a(i)$.
2. Compute the average distance of that point with all points in the closest cluster to its cluster, $b(i)$.
3. The silhouette coefficient is defined as follows:

$$s(i) = \frac{b(i) - a(i)}{\max(b(i), a(i))} \tag{A.1}$$

4. After computing the silhouette coefficient for each point, average it out to get the silhouette score.

We repeat the calculation for a given possible number of clusters, that is, from 2 to 10, for example. The optimal number of clusters is the one that maximizes the average silhouette coefficient across all data points.

Appendix B. Optimization Methods

In this appendix, we provide a brief overview of two optimization methods used in our study: dual annealing and differential evolution. These methods offer powerful tools for global optimization in various contexts.

Dual annealing [33] is a global optimization algorithm based on the concept of simulated annealing. It is specifically designed to find the global minimum of a function within a defined search space. The algorithm begins by generating an initial set of candidate solutions (points) within the specified bounds. It then proceeds to iteratively explore the search space through a combination of local and global search techniques. The key features of dual annealing include simulated annealing, local search, and stochastic perturbation. Simulated annealing employs a temperature schedule that allows the acceptance of worse candidate solutions, allowing the algorithm to escape local minima and explore different regions of the search space. Local search is performed at each iteration to refine the candidate solutions and enhance convergence towards the global minimum. Stochastic perturbation introduces random perturbations to candidate solutions, facilitating effective exploration of the search space. Dual annealing is particularly useful for finding the global minimum of non-convex and multimodal functions in a bounded search space. It excels in scenarios where the objective function exhibits noise or contains numerous local minima.

Differential evolution [34] is a global optimization algorithm inspired by genetic algorithms. The purpose of this method is to discover the global minimum of a function within a bounded search space. The algorithm maintains a population of candidate solutions (vectors) within the specified bounds and iteratively evolves this population through mutation, crossover, and selection operations to generate new candidate solutions for the next generation. The distinctive features of differential evolution include mutation, crossover, and selection. Mutation perturbs the candidate solutions by adding a differential vector, which is computed from the difference between randomly chosen individuals in the population. This mechanism promotes exploration of the search space. The crossover combines information from multiple individuals to generate offspring, and the specific crossover operator determines the exchange of information between individuals. Finally, selection identifies the best individuals from the current generation and the newly generated offspring to form the next generation, thereby enhancing convergence towards the global minimum. Differential evolution is highly effective in finding the global minimum of continuous and multimodal functions within a bounded search space. It is particularly suited for scenarios where the objective function is smooth and exploration of a wide range of solutions is necessary.

Appendix C. Minimization of the Constrained Action

Here we will show how from the minimization of the Constrained Action:

$$J(\mathbf{B}, \lambda) = \frac{1}{2}\int_\Omega d\mathbf{x}\,|\nabla \mathbf{B}|^2 - \int_\Omega d\mathbf{x}\boldsymbol{\rho}\cdot\mathbf{B} + \int_\Omega d\mathbf{x}\lambda\nabla\cdot\mathbf{B} \tag{C.1}$$

where λ is a Lagrange multiplier one can find the differential equations:

$$-\nabla^2\mathbf{B}(\mathbf{x}) = \nabla\lambda + \boldsymbol{\rho}(\mathbf{x}), \quad \mathbf{x}\in\Omega \tag{C.2}$$

and

$$\nabla\cdot\mathbf{B}(\mathbf{x}) = 0. \tag{C.3}$$

Using calculus of variations we have:

$$\begin{aligned} J(\mathbf{B}+\delta\mathbf{B},\lambda+\delta\lambda) &= \frac{1}{2}\int_\Omega d\mathbf{x}\,|\nabla(\mathbf{B}+\delta\mathbf{B})|^2 - \int_\Omega d\mathbf{x}\boldsymbol{\rho}\cdot(\mathbf{B}+\delta\mathbf{B}) \\ &+ \int_\Omega d\mathbf{x}\,(\lambda+\delta\lambda)\,\nabla\cdot(\mathbf{B}+\delta\mathbf{B}) = \\ &= \frac{1}{2}\int_\Omega d\mathbf{x}\left(|\nabla\mathbf{B}|^2 + 2\nabla\mathbf{B}\cdot\nabla\delta\mathbf{B}\right) - \int_\Omega d\mathbf{x}\boldsymbol{\rho}\cdot(\mathbf{B}+\delta\mathbf{B}) \\ &+ \int_\Omega d\mathbf{x}\,(\lambda\nabla\cdot\mathbf{B} + \delta\lambda\nabla\cdot\mathbf{B} + \lambda\nabla\cdot\delta\mathbf{B}) + O\delta^{(2)} \end{aligned} \tag{C.4}$$

From the above equation we can find the variation of the Lagrangian:

$$\begin{aligned} \delta J = 0 = J(\mathbf{B}+\delta\mathbf{B},\lambda+\delta\lambda) - J(\mathbf{B},\lambda) = \\ = \int_\Omega d\mathbf{x}\nabla\mathbf{B}\cdot\nabla\delta\mathbf{B} - \int_\Omega d\mathbf{x}\boldsymbol{\rho}\cdot\delta\mathbf{B} + \int_\Omega d\mathbf{x}\lambda\nabla\cdot\delta\mathbf{B} + \int_\Omega d\mathbf{x}\delta\lambda\nabla\cdot\mathbf{B} = \\ = -\int_\Omega d\mathbf{x}\left(\nabla^2\mathbf{B} + \boldsymbol{\rho} + \nabla\lambda\right)\cdot\delta\mathbf{B} + \int_\Omega d\mathbf{x}\,(\nabla\cdot\mathbf{B})\,\delta\lambda = 0 \end{aligned} \tag{C.5}$$

or

$$-\nabla^2\mathbf{B} = \boldsymbol{\rho} + \nabla\lambda \text{ and } \nabla\cdot\mathbf{B} = 0 \tag{C.6}$$

Appendix D. Details in the implementation of the algorithm for the reconstruction of the Magnetic Field

By following our algorithm we have the following steps: Input: Synthetic Data set $\mathbf{y}$ (sparse or not) simulating measurements or observations of the x-component of the Magnetic Field (MF), the 3D Laplace Partial Differential Equation accompanied with Boundary Conditions of the Dirichlet Type, where those of the x-component depend on the unknown parameters $\boldsymbol{\theta}$ and a clustering algorithm. The algorithm is as follows:

1. In the general case, we do not know the true number of normal distributions that make up the boundary conditions. This is true in our example if the synthetic data are created after dividing the boundary of the cone in planes. To address this, we use a clustering approach to group the sparse data points based on their proximity and value to each other (see the Appendix A). The number of clusters n_c then corresponds to the number of normal distributions that make up the boundary conditions. We associate each cluster with a prior distribution with having each point in the boundary associated with the cluster it is spatially closer to. The unknown parameters $\boldsymbol{\theta}$ are the mean vales of the aforementioned normal distributions. In this work, we assume that the means of the normal distributions in the boundary conditions are independent. We choose as their values the mean of the data of each respective cluster, $\bar{y}_i$, while for the standard deviations we consider them all equal to one. So, the total prior is defined as a product:

$$p(\boldsymbol{\theta}) = \prod_{i=1}^{n_c} p(\theta_i) = \prod_{i=1}^{n_c} \frac{1}{\sqrt{2\pi}} e^{-\frac{(\theta_i-\bar{y}_i)^2}{2}}. \tag{D.1}$$

   with θ our unknown parameters, representing the mean value of the x-component of the MF in each plane.
2. Consider then a initial guess for $\boldsymbol{\theta}$. By solving the forward problem we define the cost function as follows:

$$J(\boldsymbol{\theta}) = \frac{1}{n_y}\sum_{i=1}^{n_y}(y_i - f(\mathbf{x};\boldsymbol{\theta})). \tag{D.2}$$

   The likelihood is considered to be Gaussian:

$$y_i - f(\mathbf{x};\boldsymbol{\theta}) \sim \mathcal{N}(0,1), \quad i = 1,...,n_y \tag{D.3}$$

   or

$$p(\mathbf{y}|\boldsymbol{\theta}) = \prod_{i=1}^{n_y} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(y_i - f(\mathbf{x};\boldsymbol{\theta}))^2}{2}\right). \tag{D.4}$$

3. The logarithm of the posterior is:

$$\ln p(\boldsymbol{\theta}|\mathbf{y}) = -\sum_{j=1}^{n_y}\frac{(y_i - f(\mathbf{x};\boldsymbol{\theta}))^2}{2} - \sum_{j=1}^{n_c}\frac{(\theta_i - \bar{y}_i)^2}{2} + \text{constants}. \tag{D.5}$$

4. The maximum a posteriori problem is defined as shown in the main text. So, to find the maximum a posteriori estimate of the model parameters, we minimize the negative logarithm of the posterior distribution using stochastic optimization methods such as dual annealing.
   After the dual annealing minimization method converges to a global minimum and we find a first approximation for $\boldsymbol{\theta}$ we can also calculate the residuals:

$$R = \frac{1}{n_y} \sum_{i=1}^{n_y} \int_{\Omega} d\mathbf{x} (y_i - f(x; \boldsymbol{\theta}))^2 \tag{D.6}$$

   After that we repeat the calculation and average the results, thus finding $\bar{\boldsymbol{\theta}}$ and $\bar{R}$ and their respective statistical errors. Provided the calculation is successful we should recover the mean value of the parameters that were used to generate the synthetic data i.e. the results of the optimization are the inferred mean values of the normal distributions that make up the boundary conditions.
   Our proposed inverse problem algorithm should then recover the mean values of these distributions.
5. A solution of the forward model gives the MF in the whole cone domain, thus completing the reconstruction calculation.

Therefore, we have as output the parameters $\bar{\boldsymbol{\theta}}$ and the reconstruction of the MF in the whole domain.

Appendix E. Tables with the results that were omitted in the main text

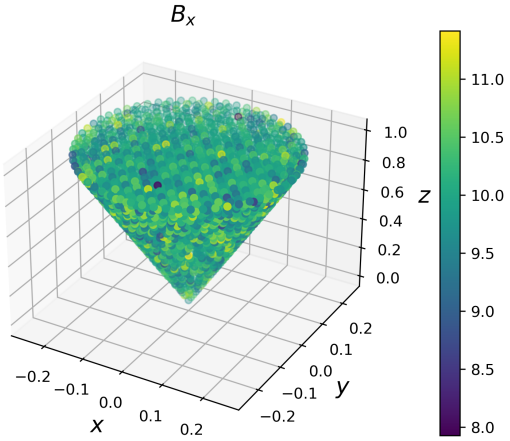
Percentage of data we remove	$\bar{\theta}$	Residuals
0%	10.00342 ± 0.03155	0.00356 ± 0.00015
50%	9.98887 ± 0.03913	0.00375 ± 0.00019
75%	10.01981 ± 0.04094	0.00356 ± 0.00020
90%	9.99900 ± 0.05205	0.00358 ± 0.00026
95%	9.99473 ± 0.06223	0.00370 ± 0.00040
99%	9.98432 ± 0.07948	0.00390 ± 0.00050

Table E.1: Results of the algorithm in the case of one prior. The initial value $\theta = 10$ is successfully recovered.

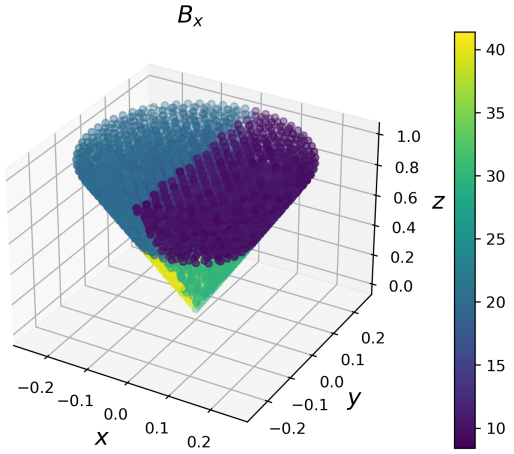
Percentage of data we remove	$\bar{\boldsymbol{\theta}}$	Residuals
0%	10.00537 ± 0.05346	0.00366 ± 0.00023
	19.98861 ± 0.05561	
	29.84011 ± 0.12618	
	40.02537 ± 0.12592	
50%	10.00061 ± 0.07580	0.00393 ± 0.00036
	19.96684 ± 0.05993	
	30.11633 ± 0.16152	
	39.59737 ± 0.17049	
75%	10.00928 ± 0.08977	0.00418 ± 0.00047
	20.01029 ± 0.09527	
	29.41021 ± 0.19511	
	39.70518 ± 0.18892	
90%	10.02409 ± 0.11375	0.00439 ± 0.00056
	19.96101 ± 0.11800	
	29.88206 ± 0.22677	
	39.95013 ± 0.19304	
95%	10.01150 ± 0.12538	0.00481 ± 0.00086
	20.01948 ± 0.11637	
	30.22242 ± 0.27096	
	39.45244 ± 0.26072	

Table E.2: Results of the algorithm in the case of four priors. Initial values are successfully recovered

Appendix F. The Reconstructed Magnetic Field



(a) The Reconstructed Magnetic Field for the case of 1 prior, where we removed 95% of the data.



(b) The Reconstructed Magnetic Field for the case of 4 priors, where we removed 95% of the data.

References

[1] R. C. Aster, B. Borchers, C. H. Thurber, Parameter estimation and inverse problems, Elsevier, 2013. doi:10.1016/c2009-0-61134-x.
URL https://doi.org/10.1016%2Fc2009-0-61134-x

[2] N. Bissantz, H. Holzmann, Statistical inference for inverse problems, Inverse Problems 24 (3) (2008) 034009. doi:10.1088/0266-5611/24/3/034009.
URL https://doi.org/10.1088/0266-5611/24/3/034009

[3] S. Arridge, P. Maass, O. Öktem, C.-B. Schönlieb, Solving inverse problems using data-driven models, Acta Numerica 28 (2019) 1–174. doi:10.1017/s0962492919000059.
URL https://doi.org/10.1017/s0962492919000059

[4] S. I. Kabanikhin, Inverse problems of natural science, Computational Mathematics and Mathematical Physics 60 (6) (2020) 911–914. doi:10.1134/s0965542520060044.
URL https://doi.org/10.1134/s0965542520060044

[5] A. M. Stuart, Inverse problems: A bayesian perspective, Acta Numerica 19 (2010) 451–559. doi:10.1017/s0962492910000061.

[6] T. Cui, Y. Marzouk, K. Willcox, Scalable posterior approximations for large-scale bayesian inverse problems via likelihood-informed parameter and state reduction, Journal of Computational Physics 315 (2016) 363–387. doi:10.1016/j.jcp.2016.03.055.
URL https://doi.org/10.1016/j.jcp.2016.03.055

[7] R. Snieder, J. Trampert, Inverse problems in geophysics, in: CISM International Centre for Mechanical Sciences, Springer Vienna, 1999, pp. 119–190. doi:10.1007/978-3-7091-2486-4_3.
URL https://doi.org/10.1007/978-3-7091-2486-4_3

[8] C. Vogt, T. A. Enßlin, A bayesian view on faraday rotation maps – seeing the magnetic power spectra in galaxy clusters, Astronomy & Astrophysics 434 (1) (2005) 67–76. doi:10.1051/0004-6361:20041839.
URL https://doi.org/10.1051/0004-6361:20041839

[9] M. Liebsch, S. Russenschuck, S. Kurz, BEM-based magnetic field reconstruction by ensemble kálmán filtering, Computational Methods in Applied Mathematics 23 (2) (2022) 405–424. doi:10.1515/cmam-2022-0121.
URL https://doi.org/10.1515/cmam-2022-0121

[10] T. Broeren, K. G. Klein, J. M. TenBarge, I. Dors, O. W. Roberts, D. Verscharen, Magnetic field reconstruction for a realistic multi-point, multi-scale spacecraft observatory, Frontiers in Astronomy and Space Sciences 8 (Sep. 2021). doi:10.3389/fspas.2021.727076.
URL https://doi.org/10.3389/fspas.2021.727076

[11] R. E. Denton, Y.-H. Liu, H. Hasegawa, R. B. Torbert, W. Li, S. Fuselier, J. L. Burch, Polynomial reconstruction of the magnetic field observed by multiple spacecraft with integrated velocity determination, Journal of Geophysical Research: Space Physics 127 (10) (Oct. 2022). doi:10.1029/2022ja030512.
URL https://doi.org/10.1029/2022ja030512

[12] J. E. Baader, S. Casalbuoni, Magnetic field reconstruction using the pulsed wire method: An accuracy analysis, Measurement (Lond.) 193 (110873) (2022) 110873.

[13] C. Donnelly, S. Gliga, V. Scagnoli, M. Holler, J. Raabe, L. J. Heyderman, M. Guizar-Sicairos, Tomographic reconstruction of a three-dimensional magnetization vector field, New Journal of Physics 20 (8) (2018) 083009. doi:10.1088/1367-2630/aad35a.
URL https://doi.org/10.1088/1367-2630/aad35a

[14] A. Tsouros, T. Enßlin, M. Mastorakis, V. Pavlidou, Reconstructing galactic magnetic fields from local measurements for backtracking ultra-high-energy cosmic rays, Astronomy & Astrophysics (Nov. 2023). doi:10.1051/0004-6361/202346423.
URL http://dx.doi.org/10.1051/0004-6361/202346423

[15] M. Raissi, P. Perdikaris, G. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational Physics 378 (2019) 686–707. doi:10.1016/j.jcp.2018.10.045.
URL http://dx.doi.org/10.1016/j.jcp.2018.10.045

[16] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, Nature Reviews Physics 3 (6) (2021) 422–440. doi:10.1038/s42254-021-00314-5.
URL http://dx.doi.org/10.1038/s42254-021-00314-5

[17] E. Christofi, P. Bačová, V. A. Harmandaris, Physics-informed deep learning approach for reintroducing atomic detail in coarse-grained configurations of multiple poly(lactic acid) stereoisomers, Journal of Chemical Information and Modeling (Mar. 2024). doi:10.1021/acs.jcim.3c01870.
URL http://dx.doi.org/10.1021/acs.jcim.3c01870

[18] D. Liang, J. Cheng, Z. Ke, L. Ying, Deep magnetic resonance image reconstruction: Inverse problems meet neural networks, IEEE Signal Processing Magazine 37 (1) (2020) 141–151. doi:10.1109/msp.2019.2950557.
URL http://dx.doi.org/10.1109/MSP.2019.2950557

[19] F. Tobar, A. Robert, J. F. Silva, Gaussian process deconvolution, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 479 (2275) (Jul. 2023). doi:10.1098/rspa.2022.0648.
URL http://dx.doi.org/10.1098/rspa.2022.0648

[20] P. R. Newbury, R. J. Spiteri, Inverting gravitational lenses, SIAM Review 44 (1) (2002) 111–130. doi:10.1137/s0036144500380934.
URL http://dx.doi.org/10.1137/S0036144500380934

[21] N. Cao, J. Xie, A. Zhang, S.-Y. Hou, L. Zhang, B. Zeng, Neural networks for quantum inverse problems, New Journal of Physics 24 (6) (2022) 063002. doi:10.1088/1367-2630/ac706c.
URL http://dx.doi.org/10.1088/1367-2630/ac706c

[22] J. P. Kaipio, E. Somersalo, Statistical and Computational Inverse Problems, Springer New York, 2005. doi:10.1007/b138659.

[23] X. Hu, X. Xu, W. Chen, Numerical method for the inverse heat transfer problem in composite materials with stefan-boltzmann conditions, Advances in Computational Mathematics 33 (4) (2009) 471–489. doi:10.1007/s10444-009-9131-x.

[24] L. Yan, T. Zhou, Adaptive multi-fidelity polynomial chaos approach to bayesian inference in inverse problems, Journal of Computational Physics 381 (2019) 110–128. doi:10.1016/j.jcp.2018.12.025.

[25] G. A. Padmanabha, N. Zabaras, Solving inverse problems using conditional invertible neural networks, Journal of Computational Physics 433 (2021) 110194. doi:10.1016/j.jcp.2021.110194.

[26] D. Z. Huang, T. Schneider, A. M. Stuart, Iterated kalman methodology for inverse problems, Journal of Computational Physics 463 (2022) 111262. doi:10.1016/j.jcp.2022.111262.

[27] J. K. Ghosh, An Introduction to Bayesian Analysis Theory and Methods, Springer New York, 2006.

[28] S. Brooks, Handbook for Markov Chain Monte Carlo, Taylor & Francis, 2011.

[29] L. Kaufman, P. J. Rousseeuw (Eds.), Finding Groups in Data, John Wiley & Sons, Inc., 1990. doi:10.1002/9780470316801.

[30] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer New York, 2009. doi:10.1007/978-0-387-84858-7.

[31] P. J. Rousseeuw, Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, Journal of Computational and Applied Mathematics 20 (1987) 53–65. doi:10.1016/0377-0427(87)90125-7.

[32] S. Maroufpoor, R. Azadnia, O. Bozorg-Haddad, Stochastic optimization, in: Handbook of Probabilistic Models, Elsevier, 2020, pp. 437–448. doi:10.1016/b978-0-12-816514-0.00017-5.

[33] Y. Xiang, D. Sun, W. Fan, X. Gong, Generalized simulated annealing algorithm and its application to the thomson model, Physics Letters A 233 (3) (1997) 216–220. doi:10.1016/s0375-9601(97)00474-x.

[34] J. Lampinen, A constraint handling approach for the differential evolution algorithm, in: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600), IEEE. doi:10.1109/cec.2002.1004459.

[35] M. C. Kennedy, A. O'Hagan, Bayesian calibration of computer models, Journal of the Royal Statistical Society Series B: Statistical Methodology 63 (3) (2001) 425–464. doi:10.1111/1467-9868.00294.

[36] K. H. Huebner, D. L. Dewhirst, D. E. Smith, T. G. Byrom, The Finite Element Method for Engineers, Wiley-Interscience, 2001.

[37] D. L. Logan, A first course in the finite element method, Cengage Learning, 2012.

[38] M. S. Alnaes, A. Logg, K. B. Ølgaard, M. E. Rognes, G. N. Wells, Unified form language: A domain-specific language for weak formulations of partial differential equations, ACM Transactions on Mathematical Software 40 (2014). doi:10.1145/2566630.

[39] M. W. Scroggs, I. A. Baratta, C. N. Richardson, G. N. Wells, Basix: a runtime finite element basis evaluation library, Journal of Open Source Software 7 (73) (2022) 3982. doi:10.21105/joss.03982.

[40] M. W. Scroggs, J. S. Dokken, C. N. Richardson, G. N. Wells, Construction of arbitrary order finite element degree-of-freedom maps on polygonal and polyhedral cell meshes, ACM Transactions on Mathematical SoftwareTo appear (2022). doi:10.1145/3524456.

[41] L. N. Trefethen, D. Bau, Numerical Linear Algebra, SIAM, 1997.

[42] A. Logg, G. N. Wells, J. Hake, DOLFIN: a C++/Python finite element library, in: K. M. A. Logg, G. N. Wells (Eds.), Automated Solution of Differential Equations by the Finite Element Method, Vol. 84 of Lecture Notes in Computational Science and Engineering, Springer, 2012, Ch. 10.

[43] D. N. Arnold, A. Logg, Periodic table of the finite elements, SIAM News 47 (2014).

[44] B. Cockburn, G. Fu, A systematic construction of finite element commuting exact sequences, SIAM Journal on Numerical Analysis 55 (4) (2017) 1650–1688. doi:10.1137/16m1073352.

[45] The DefElement contributors, DefElement: an encyclopedia of finite element definitions, https://defelement.com, [Online; accessed 06-February-2023] (2023).

[46] P. M. Gresho, R. L. Sani, Incompressible Flow and the Finite Element Method, Volume 1, Advection-Diffusion and Isothermal Laminar Flow, Wiley, 2000.

[47] V. John, A. Linke, C. Merdon, M. Neilan, L. G. Rebholz, On the divergence constraint in mixed finite element methods for incompressible flows, SIAM Review 59 (3) (2017) 492–544. doi:10.1137/15m1047696.

[48] N. Sime, J. M. Maljaars, C. R. Wilson, P. E. van Keken, An exactly mass conserving and pointwise divergence free velocity method: Application to compositional buoyancy driven flow problems in geodynamics, Geochemistry, Geophysics, Geosystems 22 (4) (mar 2021). doi:10.1029/2020gc009349.

[49] M. Benzi, G. H. Golub, J. Liesen, Numerical solution of saddle point problems, Acta Numerica 14 (2005) 1–137. doi:10.1017/s0962492904000212.

[50] C. C. Paige, M. A. Saunders, Solution of sparse indefinite systems of linear equations, SIAM Journal on Numerical Analysis 12 (4) (1975) 617–629. doi:10.1137/0712047.

[51] S. e. a. Hutschenreuter, The galactic faraday rotation sky 2020, Astronomy & Astrophysics 657 (2022) A43. doi:10.1051/0004-6361/202140486.
URL http://dx.doi.org/10.1051/0004-6361/202140486