

Article

Not peer-reviewed version

---

# Regularized Kaczmarz Solvers for Robust Inverse Laplace Transforms

---

Marta González-Lázaro , [Eduardo Viciano](#) , Víctor Valdivieso , [Ignacio Fernández](#) \* ,  
[Francisco Manuel Arrabal-Campos](#) \*

Posted Date: 21 May 2025

doi: 10.20944/preprints202505.1660.v1

Keywords: inverse laplace transform; kaczmarz method; wasserstein regularizati






Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

# Regularized Kaczmarz Solvers for Robust Inverse Laplace Transforms

Marta González-Lázaro <sup>2</sup>, Eduardo Viciano <sup>1</sup> , Víctor Valdivieso <sup>1</sup>, Ignacio Fernández<sup>2\*</sup>  and Franciso Manuel Arrabal-Campos <sup>1\*</sup> 

<sup>1</sup> Department of Engineering, Research Centre CIAIMBITAL, Escuela Superior de Ingeniería, University of Almería, Spain

<sup>2</sup> Department of Chemistry and Physics, Research Centre CIAIMBITAL, University of Almería, Spain

\* Correspondence: fmarrabal@ual.es, ifernan@ual.es

**Abstract:** Inverse Laplace transforms (ILTs) are fundamental to a wide range of scientific and engineering applications—from diffusion NMR spectroscopy to medical imaging—yet their numerical inversion remains severely ill-posed, particularly in the presence of noise or sparse data. In this work, we introduce a novel family of Kaczmarz-based ILT solvers that embed advanced regularization directly into the iterative projection framework. We propose three algorithmic variants—Tikhonov-Kaczmarz, Total Variation (TV)-Kaczmarz, and Wasserstein-Kaczmarz—each incorporating a distinct penalty to stabilize solutions and mitigate noise amplification. The Wasserstein-Kaczmarz method, in particular, leverages optimal transport theory to impose geometric priors, yielding enhanced robustness for multi-modal or highly overlapping distributions. We benchmark these methods against established ILT solvers—including CONTIN, Maximum Entropy (MaxEnt), TRAI<sub>n</sub>, ITAMeD, and PALMA—using synthetic single- and multi-modal diffusion distributions contaminated with 1 % controlled noise. Quantitative evaluation via mean squared error (MSE), Wasserstein distance, total variation, peak signal-to-noise ratio (PSNR), and runtime demonstrates that Wasserstein-Kaczmarz attains an optimal balance of speed (0.53 s per inversion) and accuracy ( $MSE = 4.7 \times 10^{-8}$ ), while TRAI<sub>n</sub> achieves the highest fidelity ( $MSE = 1.5 \times 10^{-8}$ ) at a modest computational cost. These results elucidate the inherent trade-offs between computational efficiency and reconstruction precision and establish regularized Kaczmarz solvers as versatile, high-performance tools for ill-posed inverse problems. To promote reproducibility and further development, all code is freely available at <https://github.com/fmarrabal/kaczmarz-ilt-solver>.

**Keywords:** inverse laplace transform; kaczmarz method; wasserstein regularization

## 1. Introduction

Inverse Laplace transforms (ILTs) are basic math transforms in many scientific and engineering fields. They facilitate the manipulation and the solution of differential equations in diverse areas such as heat conduction [1,2], electrical circuit analysis [3], mechanical vibration studies [4], and control systems design [5]. In applications ranging from magnetic resonance imaging (MRI) [6] to industrial process control, the accuracy of ILT calculations is critical to ensuring reliable and robust results.

Despite their broad utility, obtaining accurate and efficient numerical ILT solutions remains challenging. Many traditional approaches struggle with precision and computational speed, especially for complex or high-frequency functions. These difficulties are further compounded when the underlying problems are ill-posed—a common occurrence in real-world scenarios where sparse or noisy data can induce instability [7]. Such ill-posedness is frequently encountered in:

- **Image Reconstruction:** Recovering detailed images from incomplete or noisy measurements (e.g., in medical or astronomical imaging). [8]
- **Inverse Heat Conduction:** Estimating temperature distributions or heat fluxes from limited thermal data. [9]

- **Seismic Inversion:** Inferring subsurface properties from seismic signals. [10]
- **Machine Learning:** Fitting models in the presence of overfitting or high-dimensional instability. [11]

To mitigate these challenges, regularization techniques have been extensively developed and applied [12–16]. Landmark advances in computational regularization methods—such as those discussed in [17–19]—have provided stable frameworks for tackling ill-posed inverse problems. Yet, many of these methods require delicate parameter tuning and may still suffer from convergence issues in complex settings.

Potential strategies to address these challenges are based on the *Kaczmarz method* that has emerged as a highly promising approach. [20] The original Kaczmarz algorithm was introduced in 1937 by Kaczmarz as an iterative method to solve consistent systems of linear equations by sequentially projecting the current iterate onto the hyperplanes defined by each equation. Although simple and easy to implement, its performance is highly sensitive to the ordering of the equations. In its classical (cyclic) form the convergence can be slow when successive rows are “nearly parallel” or when the system is large and ill-posed. The Randomised Kaczmarz (RK) algorithm, proposed by Strohmer and Vershynin, [21] was a major breakthrough. In their formulation, each row is randomly selected with a probability that increases proportionally to the square of its norm. This choice, which often leads to exponential convergence on average, helps mitigate problems caused by cyclic misordering. More recently, Gower and Richtárik [22] developed a unifying framework for randomised iterative methods in linear systems that not only recovers the RK method as a special case, but also shows that, if a positive definite ‘geometry’ matrix and sampling matrix are appropriately chosen, variants including coordinate descent and block methods can be derived. Their analysis provides a unique convergence theorem from which many known complexity results are derived and opens the door to novel algorithmic variants and preconditioning strategies. In the work of He, Dong and Li, [23] a new probability distribution is proposed for the RK algorithm, in which the probability of selecting a given row in the current iteration becomes proportional to the square of the sine of the angle between that row and the row chosen in the previous iteration. This angle-based selection is based on the observation that when consecutive hyperplanes are nearly aligned, the benefit of an additional projection diminishes. The method aims to avoid redundant projections and improve the speed of convergence by ‘rewarding’ the selection of rows that are more ‘orthogonal’ (or, in other words, form a greater angle) relative to the previously used row. In the research area, the authors propose acceleration strategies that, instead of projecting onto a single hyperplane, project onto the intersection of two hyperplanes. These dual-projection or intersection-based schemes have the potential to reduce the error even faster at each iteration. The convergence analysis presented in their study shows that these accelerated variants enjoy better convergence rates than the classical RK method with uniform or norm-based probabilities.

This method has gained popularity for its simplicity, speed, robustness, and convergence properties. These attributes have led to its successful application in computed tomography, [24] signal processing, [25] and data reconciliation [26]. Its inherent adaptability and flexibility render it an attractive candidate for exploration of its applicability to the ILT problem.

Nevertheless, the use of the Kaczmarz method for ILT remains largely unexplored, leaving its potential for solving ILTs an open question in numerical analysis. The primary objective of this work is to adapt the Kaczmarz method specifically for ILT, thereby extending its advantages to the domain of ill-posed inverse problems. By incorporating regularization directly into the iterative scheme (whether via Tikhonov, Total Variation, or more advanced approaches), our goal is to enhance both the computational efficiency and the noise resilience of ILT solvers. Several established approaches underscore the current efforts to resolve ILT challenges:

- **MaxEnt (Maximum Entropy):** Utilizes non-parametric solutions that integrate prior knowledge, ensuring stability and robustness. [27–29]

- **TRAIIn (Trust-Region Algorithm):** Dynamically adjusts the trust-region to reconstruct ILT with superlinear convergence. [30,31]
- **ITAMeD (Iterative Thresholding Algorithm for Multiexponential Decays):** Combines  $\ell_1$ -based sparsity with fast iterative methods (e.g., FISTA) to resolve discrete ILT components.[32]
- **PALMA:** Merges  $\ell_1$  regularization and maximum entropy priors to automatically balance smoothness and sparsity.[33]

While each of these methods offers distinct advantages, they often require careful parameter tuning and may not guarantee robust convergence under challenging conditions. The Kaczmarz method, in contrast, provides an intuitive row-by-row projection strategy and can integrate seamlessly with a variety of regularization frameworks. Its iterative refinement process is similar to the Gram-Schmidt orthogonalisation principle, reinforcing geometric insights that often translate into stable, noise-tolerant performance. By adapting the Kaczmarz method to ILT, we aim to exploit these strengths and develop an approach that is both computationally efficient and robust to the complexities of ill-posed Laplace inversion.

As a proof of concept, our investigation targets *diffusion NMR* (whether PFGSE—Pulsed Field Gradient Spin Echo or DOSY—Diffusion-Ordered Spectroscopy). In diffusion NMR, the measured signal attenuation due to progressively stronger magnetic field gradients results in a multi-exponential decay curve. This decay is mathematically equivalent to a Laplace transformation of the underlying distribution of diffusion coefficients. Traditional ILT approaches applied in this context are vulnerable to instability and inaccuracy. Nonetheless, the utilisation of algebraic reconstruction for multi-exponential decays has previously been established in its most fundamental form. This has been achieved through the application of ILT reconstruction to molecular weight estimation in globular proteins and exopolysaccharides.[34,35]

By adapting the Kaczmarz method for ILT and integrating robust regularization, our approach aims to recover complex diffusion profiles, accurately decompose overlapping components, and handle diverse sample behaviors (e.g., polydisperse systems). The enhanced numerical stability and efficiency of the proposed solver hold significant promise for advancing ILT applications in chemistry, biophysics, materials science, and chemical engineering.

This research has far-reaching implications for real-world applications of ILT. The development of an enhanced, reliable and efficient **Kaczmarz solver** has the potential to substantially enhance the accuracy of problem-solving and system modelling across a broad spectrum of scientific and engineering domains. The present study has implemented several variations of the Kaczmarz algorithm, incorporating various regularization approaches, including **Tikhonov, total variation, and Wasserstein regularization**. The enhanced computational capabilities of such regularized Kaczmarz algorithms have the potential to transform numerous research and industrial domains.

**Structure of This Work.** Section 2 details the methodological adaptations of the Kaczmarz method for the ILT problem, with a focus on discretizing the Fredholm integral formulation and incorporating regularization. Section 3 presents a comparative evaluation of the algorithm against established ILT methods in terms of accuracy, convergence speed, and noise robustness. Finally, Section 4 discusses practical limitations, potential extensions, and the broader implications of the work for solving ill-posed inverse problems.

### 1.1. Objectives and Contributions

This work aims to:

- We adapt the Kaczmarz method to the specific requirements of ILT by discretizing the Fredholm integral formulation and integrating advanced regularization techniques.
- We incorporate regularization methods (such as Tikhonov, Total Variation, and Wasserstein) directly into the Kaczmarz iterative process to effectively address the ill-posedness inherent in ILT problems.

- We conduct an empirical evaluation of the proposed algorithm's accuracy, convergence speed, and robustness against noise, benchmarking it against existing ILT methodologies.
- We discuss the broader implications, limitations, and future directions for extending Kaczmarz-based solvers to other classes of inverse problems.

## 2. Regularization Optimization for Ill-Posed Problems

Ill-posed problems are a class of mathematical problems that fail to meet one or more of the criteria for well-posedness established by Hadamard [7]. According to Hadamard, a problem is well-posed if it satisfies the following three conditions:

1. **Existence:** A solution to the problem exists.
2. **Uniqueness:** The solution is unique.
3. **Stability:** The solution's behavior changes continuously with changes in the initial data.

If any of these criteria are not satisfied, the problem is termed ill-posed. Ill-posed problems commonly arise in inverse problems, integral equations, and partial differential equations, particularly in scenarios involving physical systems where data is noisy, incomplete, or ambiguous. Such problems are inherently unstable, meaning that small perturbations in the input data can lead to disproportionately large deviations in the solution.

These challenges necessitate additional techniques to stabilize the problem and obtain meaningful solutions. In the absence of stabilization, solutions to ill-posed problems may be non-unique, physically implausible, or overly sensitive to noise in the data.

The general form of an ill-posed problem can be expressed mathematically as:

$$Kx = f, \quad K : X \rightarrow Y \quad (1)$$

where  $K$  is a compact operator mapping elements from the vector space  $X$  to  $Y$ ,  $x \in X$  represents the unknown solution, and  $f \in Y$  is the data. Compact operators, by their nature, amplify noise, exacerbating the instability of the problem.

In practical scenarios, the observed data  $f$  is often corrupted by noise, which is represented as:

$$f_\delta = f + \delta, \quad \|f - f_\delta\| < \delta \quad (2)$$

where  $\delta$  denotes the noise level. The presence of noise makes it difficult to directly solve Equation 1 because small errors in  $f$  can lead to large errors in  $x$ . This inherent instability underscores the need for regularization techniques to constrain the solution space and ensure stability.

Given the perturbed data  $f_\delta$ , the solution set can be expressed as:

$$Q_\delta := \{x \in X : \|Kx - f_\delta\| \leq \delta\} \quad (3)$$

This formulation defines a set of possible solutions that satisfy the noise-constrained condition. However, the dependence of the solution  $x_\delta$  on the noisy data  $f_\delta$  introduces significant challenges. The inequality  $\|Kx - f_\delta\| \leq \delta$  does not inherently ensure either stability or uniqueness of the solution. Small variations in  $f_\delta$  can produce large variations in  $x_\delta$ , leading to instability. Moreover, multiple values of  $x$  may satisfy the inequality, making the solution non-unique.

These issues highlight the core difficulties of solving ill-posed problems directly. Without further constraints or regularization, the resulting solutions are often impractical, unreliable, or physically meaningless. Regularization methods address these deficiencies by imposing additional structure or prior knowledge about the solution, which will be discussed in subsequent sections.

To address the instability inherent in ill-posed problems, regularization methods impose additional constraints or introduce prior knowledge about the solution. This process helps ensure that the solution is stable, unique, and physically meaningful. Regularization techniques are discussed in detail in the following sections.

To address these challenges, additional information is introduced in the form of regularization. Regularization imposes constraints or preferences on the solution, effectively narrowing the solution space to ensure stability and feasibility. This additional information can reflect physical, mathematical, or empirical considerations. For instance, in many applications, solutions are expected to be smooth or adhere to certain positivity constraints.

One widely accepted approach to regularization is Tikhonov regularization [19]. Tikhonov regularization stabilizes the solution by introducing a stabilizing functional, modifying the optimization problem to:

$$\min_{x \in X} \|Kx - f_\delta\|^2 + \lambda \|Lx\|^2 \quad (4)$$

where  $L$  is a linear operator that incorporates prior knowledge about the solution (e.g., smoothness or sparsity), and  $\lambda > 0$  is the regularization parameter that balances the trade-off between fidelity to the data and the stabilizing functional.

The choice of  $L$  and  $\lambda$  significantly impacts the solution. Common choices for  $L$  include the identity operator (standard Tikhonov regularization) or higher-order derivatives to enforce smoothness. Various methods exist to determine the optimal  $\lambda$ , such as the L-curve criterion [36], cross-validation, or generalized cross-validation.[37]

The operator  $K$  in Equation 1 is often a continuous compact operator. For many problems,  $K$  is expressed as a Fredholm integral equation of the first kind:

$$Kx := \int_a^b k(t,s)x(s)ds = f(t), \quad K : X \rightarrow Y, \quad t \in (c,d) \quad (5)$$

Here,  $k(t,s)$  is the kernel function, which defines the relationship between the unknown function  $x(s)$  and the data  $f(t)$ . The kernel function  $k(t,s)$  itself lies in a Hilbert space, providing a structured framework for its analysis.

A **Hilbert space** is a complete vector space equipped with an inner product that induces a norm. The inner product, denoted  $\langle u, v \rangle$ , is a bilinear form that satisfies the following properties for all  $u, v, w \in H$  and  $\alpha \in \mathbb{R}$  or  $\mathbb{C}$ :

1. **Linearity:**  $\langle \alpha u + v, w \rangle = \alpha \langle u, w \rangle + \langle v, w \rangle$ .
2. **Symmetry:**  $\langle u, v \rangle = \overline{\langle v, u \rangle}$ .
3. **Positive definiteness:**  $\langle u, u \rangle \geq 0$ , and  $\langle u, u \rangle = 0$  if and only if  $u = 0$ .

A norm is induced by the inner product as  $\|u\| = \sqrt{\langle u, u \rangle}$ . Completeness means that every Cauchy sequence in the Hilbert space converges to an element within the space. Examples of Hilbert spaces include  $\ell^2$  (the space of square-summable sequences) and  $L^2(a,b)$  (the space of square-integrable functions on  $[a,b]$ ).

Compact operators such as  $K$  are characterized by their ability to map bounded sets in  $X$  to relatively compact sets in  $Y$ . This property exacerbates the ill-posedness of the problem, as small perturbations in  $f(t)$  can lead to disproportionately large changes in  $x(s)$ . Compactness implies that the image under  $K$  of any bounded sequence in  $X$  has a subsequence that converges in  $Y$ . The integral form of  $K$  highlights its dependence on the kernel function  $k(t,s)$ , which encapsulates the physics or underlying principles of the problem at hand.

Discretization of the Fredholm integral operator is commonly performed to solve the problem numerically. This process leads to the matrix formulation:

$$\mathbf{Y} = \mathbf{K}\mathbf{x} \quad (6)$$

where  $\mathbf{Y}$  represents the discrete approximation of  $f$ ,  $\mathbf{x}$  is the discrete solution, and  $\mathbf{K}$  is the discretized operator derived from  $k(t,s)$ . However, discretization introduces numerical challenges such as instability due to the ill-conditioning of  $\mathbf{K}$ . Ill-conditioning means that small errors in  $\mathbf{f}$  can result in

large errors in  $\mathbf{x}$ , making the system sensitive to perturbations. These challenges necessitate the use of regularization techniques to mitigate the effects of noise and stabilize the solution.

The Fredholm integral equation of the first kind provides a mathematical framework for many inverse problems. The Hilbert space setting and compact operator properties play a central role in the analysis and solution of such equations, while regularization techniques are crucial for addressing their inherent instability.

Herein, we delve into several widely used algorithms developed to solve such problems. These include CONTIN, [38] Maximum Entropy, [29] ITAMeD,[32] TRAIIn,[31] and PALMA, [33] among others. Each algorithm employs unique approaches to tackle the challenges posed by ill-posedness, noise, and instability in the context of Fredholm integral equations. We will explore their mathematical foundations, implementation strategies, and applications across various scientific fields.

### 2.1. The CONTIN Algorithm: A Constrained Regularization Approach

The CONTIN algorithm, developed by S.W. Provencher, is a powerful and flexible method for solving noisy linear operator equations, including Fredholm integral equations of the first kind. It is designed to handle the inherent ill-posedness of such problems by incorporating constraints and regularization techniques. CONTIN addresses equations of the form:

$$f(t) = \int_a^b k(t,s)x(s)ds + \epsilon_t, \quad t = 1, \dots, N, \quad (7)$$

where  $f(t)$  are the observed data points,  $k(t,s)$  represents the kernel function,  $x(s)$  is the unknown solution, and  $\epsilon_k$  accounts for noise. This formulation encompasses a wide range of experimental contexts, such as photon correlation spectroscopy and Laplace transforms. To solve Equation 7, CONTIN discretizes it into a system of linear algebraic equations:

$$\mathbf{Y} = \mathbf{K}\mathbf{x} + \boldsymbol{\epsilon}, \quad (8)$$

where  $\mathbf{Y}$  is the vector of observed data,  $\mathbf{K}$  is the discretized kernel matrix,  $\mathbf{x}$  is the solution vector, and  $\boldsymbol{\epsilon}$  represents noise. The ill-conditioning of  $\mathbf{K}$  necessitates regularization to ensure stable and meaningful solutions.

CONTIN minimizes the following functional to determine the optimal solution:

$$V(\alpha) = \|\mathbf{W}(\mathbf{Y} - \mathbf{K}\mathbf{x})\|^2 + \alpha^2 \|\mathbf{R}\mathbf{x}\|^2, \quad (9)$$

where  $\mathbf{W}$  is a weighting matrix based on the covariance of the noise,  $\mathbf{R}$  represents the regularization operator, and  $\alpha$  is the regularization parameter. The second term penalizes non-physical or oscillatory solutions, promoting stability and smoothness.

The functional  $V(\alpha)$  comprises two main components:

1. **Data Fidelity Term:**  $\|\mathbf{W}(\mathbf{Y} - \mathbf{K}\mathbf{x})\|^2$ , which quantifies the deviation of the model solution from the observed data, weighted by the noise covariance matrix  $\mathbf{W}$ .
2. **Regularization Term:**  $\alpha^2 \|\mathbf{R}\mathbf{x}\|^2$ , where  $\mathbf{R}$  encodes prior knowledge about the solution, such as smoothness or sparsity. The parameter  $\alpha$  balances these two terms.

The minimization of  $V(\alpha)$  is performed using numerical optimization techniques. The gradient of  $V(\alpha)$  with respect to  $\mathbf{x}$  is given by:

$$\nabla V(\alpha) = -2\mathbf{K}^T \mathbf{W}^T (\mathbf{Y} - \mathbf{K}\mathbf{x}) + 2\alpha^2 \mathbf{R}^T \mathbf{R}\mathbf{x}. \quad (10)$$

Iterative solvers, such as conjugate gradient or quasi-Newton methods, are employed to find  $\mathbf{x}$  that minimizes  $V(\alpha)$  under the imposed constraints.

The regularization parameter  $\alpha$  plays a critical role in balancing the data fidelity and smoothness of the solution. CONTIN iteratively adjusts  $\alpha$  using the following algorithm:

**Algorithm 1:** CONTIN Algorithm for Adaptive Regularization**Input:** System matrix  $\mathbf{K} \in \mathbb{R}^{p \times n}$ ,Data vector  $\mathbf{Y} \in \mathbb{R}^p$ ,Weighting matrix  $\mathbf{W}$ ,Regularization operator  $\mathbf{R}$ ,Initial regularization parameter  $\alpha > 0$ ,Convergence tolerance  $\text{tol}$ ,Maximum iterations  $\text{maxIter}$ .**Output:** Regularized solution  $\mathbf{x} \in \mathbb{R}^n$  and optimized parameter  $\alpha$ .**Initialization:**1. Solve the regularization problem for an initial guess of  $\alpha$  to obtain  $\mathbf{x}^{(0)}$ .**for**  $k = 0, 1, 2, \dots, \text{maxIter} - 1$  **do****(a) Evaluate Objective and Gradient:**

Define the objective function

$$V(\alpha) = \|\mathbf{W}(\mathbf{Y} - \mathbf{K}\mathbf{x}^{(k)})\|^2 + \alpha^2 \|\mathbf{R}\mathbf{x}^{(k)}\|^2.$$

Compute its gradient with respect to  $\alpha$ :

$$\nabla V(\alpha) = -2\mathbf{K}^T \mathbf{W}^T (\mathbf{Y} - \mathbf{K}\mathbf{x}^{(k)}) + 2\alpha^2 \mathbf{R}^T \mathbf{R}\mathbf{x}^{(k)}.$$

**(b) Compute Performance Metrics:**

Calculate the residual sum of squares (RSS):

$$\text{RSS} = \|\mathbf{W}(\mathbf{Y} - \mathbf{K}\mathbf{x}^{(k)})\|^2,$$

and compute the regularization term (if applicable).

**(c) Adjust  $\alpha$ :**Using a numerical optimization method (e.g., conjugate gradient or quasi-Newton), update  $\alpha$  by minimizing the trade-off criterion

$$C(\alpha) = \|\mathbf{W}(\mathbf{Y} - \mathbf{K}\mathbf{x}^{(k)})\|^2 - \text{trace}(\mathbf{H}),$$

where  $\mathbf{H}$  is the influence matrix.**(d) Statistical Evaluation:**Compute Fisher's  $F$ -statistic:

$$F = \frac{\text{RSS}_{\alpha_1} - \text{RSS}_{\alpha_2}}{\text{RSS}_{\alpha_2} / (N - p)},$$

where  $\text{RSS}_{\alpha_1}$  and  $\text{RSS}_{\alpha_2}$  are the RSS values for two successive  $\alpha$  values,  $N$  is the number of data points, and  $p$  is the number of parameters.**(e) Update the Solution:**Using the updated  $\alpha$ , solve the regularization problem (via an iterative solver) to obtain  $\mathbf{x}^{(k+1)}$ .**(f) Convergence Check:**

$$\text{if } \frac{\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|}{\|\mathbf{x}^{(k)}\| + 10^{-15}} < \text{tol}, \quad \text{terminate.}$$

**return**  $\mathbf{x}^{(k+1)}$  and the final  $\alpha$ .



This algorithm allows for the incorporation of equality and inequality constraints, which reflect prior knowledge about the solution. For instance, non-negativity constraints can be imposed to ensure that the solution  $s(\lambda)$  remains physically meaningful:

$$\mathbf{D}\mathbf{x} \geq \mathbf{d}, \quad (11)$$

$$\mathbf{E}\mathbf{x} = \mathbf{e}, \quad (12)$$

where  $\mathbf{D}$  and  $\mathbf{E}$  are user-defined matrices, and  $\mathbf{d}$  and  $\mathbf{e}$  are vectors specifying the constraints.

## 2.2. The Maximum Entropy Algorithm: A Regularization Method for Ill-Posed Problems

The Maximum Entropy (MaxEnt) algorithm is a widely used method for solving ill-posed problems. MaxEnt relies on the principle of entropy maximization to find the most probable solution consistent with the observed data, incorporating prior knowledge while avoiding overfitting to noise. We assume an objective of the form,

$$\chi^2(x) - \alpha S(x),$$

or equivalently

$$J(x) = \frac{1}{2} \sum_{j=1}^p \frac{(Y_j - \sum_{i=1}^n K_{ji} x_i)^2}{\sigma_j^2} - \alpha \sum_{i=1}^n x_i \ln x_i,$$

where  $\alpha > 0$  is a Lagrange multiplier (or regularization parameter) controlling the balance between data fidelity  $\chi^2$  and entropy  $S(x)$ .

For the entropy term

$$S(x) = - \sum_i x_i \ln(x_i),$$

its gradient with respect to  $x_i$  is  $-1 - \ln(x_i)$ . For the  $\chi^2$  term, we carefully accumulate contributions from the residual  $\mathbf{Y} - \mathbf{K}\mathbf{x}$ , scaled by  $\sigma_j^{-2}$ .

After exponentiating the negative gradient, solutions are often re-scaled to respect constraints such as  $\sum_i x_i = F$  (e.g.,  $F = 1$  for a probability distribution). This step is crucial for maintaining physically or probabilistically meaningful solutions.

To solve this ill-posed problem, MaxEnt maximizes the entropy functional  $S$ :

$$S = - \sum_{i=1}^n x_i \log\left(\frac{x_i}{F}\right), \quad F = \sum_{i=1}^n x_i, \quad (13)$$

where  $x_i$  is the discretized value of  $x(s)$ , and  $F$  ensures normalization. The entropy maximization is subject to constraints that enforce consistency with the observed data. The optimization problem is formulated as:

$$Q = S - \lambda \chi^2, \quad (14)$$

where  $\lambda$  is a Lagrange multiplier, and  $\chi^2$  quantifies the fit to the data:

$$\chi^2 = \sum_{i=1}^p \frac{(Y_j - \sum_{i=1}^n \mathbf{K}_{ij} x_i)^2}{\sigma_j^2}, \quad (15)$$

where  $Y_i$  is the observed data,  $\mathbf{K}_{ij}$  represents the integral kernel,  $x_j$  is the solution vector, and  $\sigma_i$  is the standard deviation of the noise. The optimization is performed iteratively:

**Algorithm 2: MaxEnt Algorithm**

**Input:** System matrix  $\mathbf{K} \in \mathbb{R}^{p \times n}$ , data vector  $\mathbf{Y} \in \mathbb{R}^p$ , uncertainties  $\{\sigma_j\}_{j=1}^p$ , regularization parameter  $\alpha > 0$ , convergence tolerance  $\text{tol}$ , maximum iterations  $\text{maxIter}$ .

**Output:** MaxEnt solution  $x \in \mathbb{R}^n$ .

**Initialization:**

1. Set  $x_i^{(0)} \leftarrow \frac{F}{n}$  for  $i = 1, \dots, n$ , where  $F$  is the desired sum (e.g.  $F = 1$ ).
2. Precompute  $\mathbf{K}_j$ , row norms or other needed quantities.

**for**  $k = 0, 1, 2, \dots, \text{maxIter} - 1$  **do**

**(a) Compute the residuals:**

$$r_j = \mathbf{Y}_j - \sum_{i=1}^n \mathbf{K}_{ji} x_i^{(k)}, \quad j = 1, \dots, p.$$

**(b) Compute gradient of the objective:**

- $\chi^2$ -part:

$$(\nabla \chi^2)_i = - \sum_{j=1}^p \frac{\mathbf{K}_{ji}}{\sigma_j^2} r_j, \quad i = 1, \dots, n.$$

- Entropy part  $S(x) = - \sum_i x_i \ln(x_i)$ :

$$(\nabla S)_i = -[\ln(x_i^{(k)}) + 1], \quad i = 1, \dots, n.$$

- Combined gradient of  $\chi^2 - \alpha S$ :

$$\nabla_i = (\nabla \chi^2)_i - \alpha (\nabla S)_i.$$

**(c) Update rule (multiplicative):**

$$x_i^{(k+1)} = x_i^{(k)} \exp[-\eta \nabla_i], \quad i = 1, \dots, n,$$

where  $\eta$  is a step size (often chosen as  $\eta = 1$ , or tuned adaptively).

**(d) Normalize if needed:**

$$x_i^{(k+1)} \leftarrow \frac{x_i^{(k+1)}}{\sum_{j=1}^n x_j^{(k+1)}} \times F, \quad i = 1, \dots, n.$$

**(e) Check convergence:**

$$\text{if } \frac{\|x^{(k+1)} - x^{(k)}\|}{\|x^{(k)}\| + 10^{-15}} < \text{tol}, \quad \text{terminate.}$$

**return**  $x^{(\text{maxIter})}$  if no earlier termination.

### 2.3. Trust-Region Algorithm: A Regularization Approach for Inversion Problems

The Trust-Region Algorithm for the Inversion (TRAI) is a robust and iterative regularization technique. Traditional iterative methods often struggle to balance convergence with robustness, resulting in solutions that may either converge too slowly or become unstable in the presence of measurement errors.

To address these issues, this algorithm has been developed as a powerful iterative regularization method. TRAIIn minimizes the standard least-squares objective

$$\Phi(x) = \|\mathbf{K}x - \mathbf{Y}\|_2^2,$$

while imposing two key constraints: non-negativity of the solution ( $x \geq 0$ ) and a bound on the change between consecutive iterations ( $\|x^{(k+1)} - x^{(k)}\|_2 \leq r^{(k)}$ ), where  $r^{(k)}$  denotes the trust-region radius at iteration  $k$ . These constraints ensure that each iteration remains within a reliably stable region, effectively controlling the step size and preventing divergence.

At the heart of TRAIIn lies the solution of a trust-region subproblem at each iteration. A trial step  $\mathbf{z}$  is computed by approximately minimizing

$$\min_{\mathbf{z}} \Phi(x^{(k)} + \mathbf{z}) \quad \text{subject to} \quad \|\mathbf{z}\|_2 \leq r^{(k)},$$

using the truncated conjugate gradient method. The quality of this trial step is then quantified by the ratio

$$\rho = \frac{\Delta\Phi_{\text{actual}}}{\Delta\Phi_{\text{predicted}}},$$

which compares the actual reduction in the objective to that predicted by the model. Based on the value of  $\rho$ , the trust-region radius is adjusted adaptively:

- If  $\rho > \mu_2$ , the radius is increased ( $r^{(k+1)} = \nu_2 r^{(k)}$ ).
- If  $\mu_1 \leq \rho \leq \mu_2$ , the radius remains unchanged.
- If  $\rho < \mu_1$ , the radius is decreased ( $r^{(k+1)} = \nu_1 r^{(k)}$ ).

Typical parameter values are  $\mu_1 = 0.25$ ,  $\mu_2 = 0.75$ ,  $\nu_1 = 0.5$ , and  $\nu_2 = 2$ . The iterative process is terminated when the residual norm satisfies

$$\|\mathbf{K}x^{(k)} - \mathbf{Y}\|_2 \leq \tau\epsilon,$$

with  $\epsilon$  representing the noise level and  $\tau$  a constant slightly greater than 1 (e.g.,  $\tau = 1.02$ ).

The following algorithm provide a detailed exposition of TRAIIn, describing the adaptive strategies that guarantee both stability and convergence even in the presence of significant noise.

**Algorithm 3: Trust-Region Algorithm for the Inversion (TRAIIn)****Input:** System matrix  $\mathbf{K} \in \mathbb{R}^{p \times n}$ ,Data vector  $\mathbf{Y} \in \mathbb{R}^p$ ,Noise level  $\epsilon$ ,Initial guess  $x^{(0)} \in \mathbb{R}^n$  (with  $x^{(0)} \geq 0$ ),Initial trust-region radius  $r^{(0)} > 0$ ,Parameters  $\mu_1, \mu_2, \nu_1, \nu_2$  (typical values:  $\mu_1 = 0.25, \mu_2 = 0.75, \nu_1 = 0.5, \nu_2 = 2$ ),Termination tolerance  $\tau$  (e.g.,  $\tau = 1.02$ ),Maximum iterations  $\text{maxIter}$ .**Output:** Reconstructed solution  $x \in \mathbb{R}^n$ .**Initialization:**1. Set  $x^{(0)}$  (ensuring  $x^{(0)} \geq 0$ ).2. Set  $r^{(0)}$ .**for**  $k = 0, 1, 2, \dots, \text{maxIter} - 1$  **do****(a) Solve the Trust-Region Subproblem:**Compute a trial step  $\mathbf{z}$  by approximately solving

$$\min_{\mathbf{z}} \Phi(x^{(k)} + \mathbf{z}) = \|\mathbf{K}(x^{(k)} + \mathbf{z}) - \mathbf{Y}\|_2^2,$$

subject to  $\|\mathbf{z}\|_2 \leq r^{(k)}$  (e.g., using the truncated conjugate gradient method).**(b) Update the Solution:**

Set

$$x^{(k+1)} = x^{(k)} + \mathbf{z},$$

and enforce  $x^{(k+1)} \geq 0$  if needed.**(c) Evaluate the Reduction Ratio:**Compute the predicted reduction  $\Delta\Phi_{\text{predicted}}$  and the actual reduction  $\Delta\Phi_{\text{actual}}$  in the objective function

$$\Phi(x) = \|\mathbf{K}x - \mathbf{Y}\|_2^2.$$

Then calculate the ratio

$$\rho = \frac{\Delta\Phi_{\text{actual}}}{\Delta\Phi_{\text{predicted}}}.$$

**(d) Update the Trust-Region Radius:**

- If  $\rho > \mu_2$ , set  $r^{(k+1)} = \nu_2 r^{(k)}$ .
- If  $\mu_1 \leq \rho \leq \mu_2$ , set  $r^{(k+1)} = r^{(k)}$ .
- If  $\rho < \mu_1$ , set  $r^{(k+1)} = \nu_1 r^{(k)}$ .

**(e) Convergence Check:**

$$\text{if } \|\mathbf{K}x^{(k+1)} - \mathbf{Y}\|_2 \leq \tau \epsilon, \quad \text{terminate and return } x^{(k+1)}.$$

**return**  $x^{(k+1)}$ .

#### 2.4. Iterative Thresholding Algorithm for Inversion Problems

The Iterative Thresholding Algorithm for Inversion Problems is a robust and efficient regularization method. By employing sparsity-promoting regularization and leveraging the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA). This provides high-resolution reconstruction of the In-

verse Laplace Transform. This task is notoriously ill-posed, as it is sensitive to noise and requires regularization. The method formulates the problem as an  $\ell_1$ -regularized least-squares minimization:

$$\min_x \frac{1}{2} \|\mathbf{K}x - \mathbf{Y}\|_2^2 + \tau \|x\|_1, \quad (16)$$

where:

- $\mathbf{K}$  is the kernel impact operator mapping elements from the vector space  $X$  to  $Y$ ,
- $x$  is the discretized unknown solution,
- $\mathbf{Y}$  is the discrete approximation of  $f$ ,
- $\|x\|_1$  promotes sparsity in  $x$ ,
- $\tau$  is the regularization parameter.

FISTA is employed to efficiently solve the optimization problem. The algorithm iteratively updates the solution  $x$  as follows:

---

**Algorithm 4:** Iterative Update Algorithm with Proximal Operator

---

**Initialization:** Set the initial values:

$$x^{(0)} = 0, \quad y^{(1)} = x^{(0)}, \quad t^{(1)} = 1.$$

**while** *Convergence criterion is not met* **do**

**Step 1: Update A** Compute the proximal update:

$$x^{(k)} \leftarrow \text{prox}_{\tau/L} \left( y^{(k)} - \frac{1}{L} \nabla f(y^{(k)}) \right),$$

where  $f(x) = \frac{1}{2} \|\mathbf{K}x - \mathbf{Y}\|_2^2$  and  $\text{prox}_{\tau/L}$  is given by:

$$\text{prox}_{\tau/L}(z) = \text{sgn}(z) \cdot \max(|z| - \tau/L, 0).$$

**Step 2: Update t**

$$t^{(k+1)} \leftarrow \frac{1 + \sqrt{1 + 4(t^{(k)})^2}}{2}.$$

**Step 3: Update y**

$$y^{(k+1)} \leftarrow x^{(k)} + \frac{t^{(k)} - 1}{t^{(k+1)}} (x^{(k)} - x^{(k-1)}).$$

**Convergence Criterion:** Terminate iterations when:

$$\|x^{(k)} - x^{(k-1)}\|_2 < \epsilon,$$

where  $\epsilon$  is a user-defined tolerance.

---

### 2.5. PALMA Algorithm: Hybrid Regularization for Inversion Problems

The PALMA algorithm, standing for "Proximal Algorithm for  $\ell_1$  combined with MAXEnt prior," is a hybrid regularization method designed to address the challenges of Inversion Problems. This algorithm integrates the principles of sparsity and entropy maximization to provide robust and efficient solutions for the inverse Laplace transform problem. The PALMA algorithm solves the following constrained optimization problem:

$$\min_{x \in \mathbb{R}^N} \Psi(x) \quad \text{subject to} \quad \|\mathbf{K}x - \mathbf{Y}\|_2 \leq \eta, \quad (17)$$

where:

- $\Psi(x)$  is the hybrid regularization function combining entropy and sparsity priors,
- $\eta$  is the tolerance related to the experimental noise level.

The hybrid regularization function is defined as:

$$\Psi(x) = \lambda \text{ent}(x, a) + (1 - \lambda) \|x\|_1, \quad (18)$$

where:

- $\text{ent}(\mathbf{X}, a)$  is the negative Shannon entropy with a flat prior  $a$ :

$$\text{ent}(x, a) = \sum_{i=1}^N \frac{x_i}{a} \log\left(\frac{x_i}{a}\right), \quad (19)$$

- $\|x\|_1 = \sum_{i=1}^N |x_i|$  is the  $\ell_1$  norm promoting sparsity,
- $\lambda \in [0, 1]$  balances the trade-off between entropy and sparsity.

The PALMA algorithm employs a proximal optimization approach to solve the hybrid regularization problem. The key steps are included in the algorithm as follows:

---

#### Algorithm 5: Proximal and Projection Algorithm

---

**Initialization:** Set the initial variables:

$$\begin{aligned} \mathbf{V}^{(0,1)} &\in \mathbb{R}^N, \\ \mathbf{V}^{(0,2)} &\in \mathbb{R}^M, \\ \mathbf{x}^{(0)} &\leftarrow \left(\mathbf{I}_N + \mathbf{K}^T \mathbf{K}\right)^{-1} \left(\mathbf{V}^{(0,1)} + \mathbf{K}^T \mathbf{V}^{(0,2)}\right), \end{aligned}$$

where  $\mathbf{I}_N$  is the identity matrix.

**while** *Convergence criterion is not met* **do**

**Step 1: Compute Proximal and Projection Updates**

$$\begin{aligned} \mathbf{Z}^{(k,1)} &\leftarrow \text{prox}_{\Psi}(\mathbf{V}^{(k,1)}), \\ \mathbf{Z}^{(k,2)} &\leftarrow \text{proj}_{\|\cdot - \mathbf{I}\|_2 \leq \eta}(\mathbf{V}^{(k,2)}), \end{aligned}$$

**Step 2: Compute Update for U**

$$\mathbf{U}^{(k)} \leftarrow \left(\mathbf{I}_N + \mathbf{K}^T \mathbf{K}\right)^{-1} \left(\mathbf{Z}^{(k,1)} + \mathbf{K}^T \mathbf{Z}^{(k,2)}\right).$$

**Step 3: Update  $x$**

$$x^{(k+1)} \leftarrow x^{(k)} + \gamma \left(\mathbf{U}^{(k)} - x^{(k)}\right),$$

where  $\gamma \in (0, 2)$  is a relaxation parameter.

**Stopping Criterion:** Terminate iterations when:

$$\|x^{(k+1)} - x^{(k)}\|_2 < \epsilon,$$

where  $\epsilon$  is a user-defined tolerance.

---

The PALMA algorithm requires careful selection of parameters. The entropy prior  $a$  is typically chosen as the area under the expected spectrum, providing a meaningful baseline for the algorithm. The noise tolerance  $\eta$  is estimated based on the noise level in the experimental data, ensuring the algorithm's robustness against measurement inaccuracies. Lastly, the weight  $\lambda$  balances the influence of entropy and sparsity in the regularization process, with suggested values ranging between 0.01 and 0.5 to achieve an optimal trade-off between these competing factors.

### 3. Kaczmarz Method for Solving Linear Systems

The Kaczmarz method is an iterative algorithm for solving systems of linear equations of the form:

$$\mathbf{K}x = \mathbf{Y},$$

where  $\mathbf{K} \in \mathbb{R}^{m \times n}$  is the system matrix,  $x \in \mathbb{R}^n$  is the unknown solution vector, and  $\mathbf{Y} \in \mathbb{R}^m$  is the right-hand side vector. First introduced by Stefan Kaczmarz in 1937, this algorithm has found applications in various fields such as image reconstruction, signal processing, and numerical optimization.

The core idea of the Kaczmarz method is to iteratively project the current estimate of the solution onto the hyperplane defined by one row of the Hilbert Space. This projection-based approach guarantees convergence to a solution, even for inconsistent systems, by minimizing the residual error in the least-squares sense. This concept of projection is closely related to the Gram-Schmidt orthogonalization process. While the Gram-Schmidt process orthogonalises a set of vectors, the Kaczmarz method applies successive orthogonal projections to iteratively converge to the solution. This geometric insight underscores the method's elegance and its potential for stable, noise-tolerant performance.

This Gram-Schmidt orthogonalization, a classical method in linear algebra, systematically constructs an orthonormal basis for a vector space by projecting vectors onto subspaces spanned by previously orthogonalized vectors. Drawing parallels to Gram-Schmidt orthogonalization helps elucidate the geometric and algebraic principles underlying the Kaczmarz method.

#### 3.1. Gram-Schmidt Orthogonalization

The Gram-Schmidt process is a method for orthogonalizing a set of linearly independent vectors in a finite-dimensional vector space. Given a set of  $m$  linearly independent vectors  $v_1, v_2, \dots, v_m \in \mathbb{R}^n$ , the algorithm constructs an orthonormal basis  $u_1, u_2, \dots, u_m \in \mathbb{R}^n$ .

---

#### Algorithm 6: Gram-Schmidt Orthogonalization

---

**Input:** Linearly independent vectors  $v_1, v_2, \dots, v_m \in \mathbb{R}^n$ .

**Output:** Orthonormal basis vectors  $u_1, u_2, \dots, u_m \in \mathbb{R}^n$ .

**begin**

    Compute the first orthonormal vector:

$$u_1 = \frac{v_1}{\|v_1\|}.$$

**for**  $i = 2, 3, \dots, m$  **do**

        Subtract the projections onto previously computed basis vectors:

$$u_i = v_i - \sum_{j=1}^{i-1} \text{Proj}_{u_j}(v_i),$$

        where the projection  $\text{Proj}_{u_j}(v_i)$  is defined as:

$$\text{Proj}_{u_j}(v_i) = \frac{\langle u_j, v_i \rangle}{\|u_j\|^2} u_j.$$

        Normalize  $u_i$  to ensure orthonormality:

$$u_i = \frac{u_i}{\|u_i\|}.$$

**return**  $\{u_1, u_2, \dots, u_m\}$ .

---

The resulting set  $\{u_1, u_2, \dots, u_m\}$  forms an orthonormal basis for the subspace spanned by the original vectors  $\{v_1, v_2, \dots, v_m\}$ .

Each step in the Gram-Schmidt process involves projecting a vector  $v_i$  onto the subspace spanned by the previously orthogonalized vectors. The residual vector, obtained after subtracting these projections, lies orthogonal to the subspace, ensuring orthogonality in the resulting basis.

### 3.2. Kaczmarz Algorithm as an Orthogonalization Process

The Kaczmarz method can be interpreted as a sequential orthogonalization process where the current estimate of the solution vector is projected onto the hyperplanes defined by the rows of  $A$ . Each row  $a_i$  of  $A$  represents a hyperplane in  $\mathbb{R}^n$ , defined by the equation:

$$\langle \mathbf{K}_i, x \rangle = \mathbf{Y}_i.$$

The Kaczmarz method can be described step-by-step as follows:

---

#### Algorithm 7: Kaczmarz Method for Solving Linear Systems

---

**Input:** System matrix  $\mathbf{K} \in \mathbb{R}^{m \times n}$ , right-hand side  $\mathbf{Y} \in \mathbb{R}^m$ , initial guess  $x_0 \in \mathbb{R}^n$ , maximum iterations `max_iter`, tolerance `tol`.

**Output:** Approximate solution  $x \in \mathbb{R}^n$ .

Initialize  $x_0$ ;

**for**  $k = 1, 2, \dots, \text{max\_iter}$  **do**

**for**  $i = 1, 2, \dots, m$  **do**

        Compute the distance and direction:

$$d_i = \frac{\mathbf{Y}_i}{\|\mathbf{K}_i\|^2} \mathbf{K}_i$$

        Compute the projection:

$$\text{Proj}_{\mathbf{K}_i}(x_k) = \frac{\langle \mathbf{K}_i, x_k \rangle}{\|\mathbf{K}_i\|^2} \mathbf{K}_i$$

        Update the solution in the orthogonal direction:

$$x_{k+1} = x_k + d_i - \text{Proj}_{\mathbf{K}_i}(x_k)$$

**if**  $\|x_{k+1} - x_k\| < \text{tol}$  **then**

**break;**

**return**  $x_{k+1}$ ;

---

The algorithm successively reduces  $d_i - \text{Proj}_{\mathbf{K}_i}(x_k)$  by projecting the current estimate onto the hyperplanes defined by the rows of  $A$ . After multiple iterations, these projections converge the estimate to a solution  $x^*$  that minimizes the residual error in the least-squares sense:

$$x^* = \arg \min_x \|\mathbf{K}x - \mathbf{Y}\|^2.$$

In the Kaczmarz method, each update step, represented by  $\text{Proj}_{\mathbf{K}_i}(x_k) = \frac{\langle \mathbf{K}_i, x_k \rangle}{\|\mathbf{K}_i\|^2} \mathbf{K}_i$ , is analogous to the projection operation in Gram-Schmidt orthogonalization. However, rather than orthogonalizing vectors to construct an orthonormal basis, the method iteratively adjusts the solution vector  $x_k$  to satisfy the equation of each hyperplane. By combining these projections iteratively, the Kaczmarz method ensures convergence to the optimal solution that minimizes the residual error in the least-squares sense.



#### 4. Adaption of the Kaczmarz Method with Regularization for Solving Inverse Laplace Transforms

In this section we consider the following Fredholm integral of the first kind as Laplace Transform and it is defined as:

$$\mathcal{L}\{x(s)\}(t) = \int_a^b e^{-ts} x(s) ds, \quad (20)$$

where  $t \geq 0$  and  $s \in [a, b]$ . We view  $\{x(s)\}$  as the unknown function we wish to recover (i.e., the “inverse Laplace transform” of  $f(t)$ ), and the kernel  $k(t, s) = e^{-ts}$  is the classical Laplace transform kernel. To solve (20) via the Kaczmarz method, we first discretize both the integral domain in  $s$  and the function values in  $t$ . The goal is to recast the integral equation into a linear system of the form

$$\mathbf{K} x = \mathbf{Y},$$

where  $\mathbf{K}$  approximates the continuous operator  $\mathcal{L}\{x(s)\}(t) = \int_a^b e^{-ts} x(s) ds$ ,  $\mathbf{Y}$  corresponds to sampled values of  $f(t)$ , and  $x$  is the discretized representation of  $x(s)$ .

Step 1: Partition the Interval  $[a, b]$ .

Divide the interval  $[a, b]$  into  $n$  subintervals by points  $s_1, s_2, \dots, s_n$ . Often, a uniform spacing is chosen, i.e.  $\Delta s = \frac{b-a}{n}$ , but non-uniform partitions may be used if certain regions of  $[a, b]$  require finer resolution.

Step 2: Approximate the Integral.

Replace the integral by a numerical quadrature:

$$\int_a^b e^{-ts} x(s) ds \approx \sum_{j=1}^n w_j e^{-ts_j} x(s_j),$$

where  $w_j$  is the quadrature weight for the  $j$ -th subinterval. Common choices for  $w_j$  include the rectangle rule ( $w_j = \Delta s$ ) or more accurate schemes such as trapezoidal or Gaussian quadrature.

Step 3: Sample  $\mathcal{L}\{x(s)\}(t)$  at Discrete Points.

Select  $m$  distinct values of  $t$ , say  $t_1, t_2, \dots, t_m$ . For each  $t_i$ , approximate  $\mathcal{L}\{x(s)\}(t_i)$  by the given data or by sampling from the continuous function  $\mathcal{L}\{x(s)\}(t)$ .

Step 4: Assemble the Linear System.

Define the matrix  $\mathbf{K} \in \mathbb{R}^{m \times n}$  by

$$\mathbf{K}_{i,j} = w_j e^{-t_i s_j}, \quad i = 1, \dots, m, \quad j = 1, \dots, n.$$

Next, form the vector  $\mathbf{Y} \in \mathbb{R}^m$  by

$$\mathbf{Y}_i = \mathcal{L}\{x(s)\}(t_i), \quad i = 1, \dots, m.$$

Finally, let  $x \in \mathbb{R}^n$  represent the unknown discrete values  $x_j = x(s_j)$ . Thus, the approximate equation

$$\int_a^b e^{-t_i s} x(s) ds \approx \mathcal{L}\{x(s)\}(t_i)$$

becomes the matrix problem

$$\mathbf{K} x = \mathbf{Y},$$

where  $\mathbf{K} \in \mathbb{R}^{m \times n}$  is the system matrix and  $\mathbf{Y} \in \mathbb{R}^m$  is the right-hand side. By sequentially projecting the current estimate  $x_k$  onto the hyperplanes defined by each row of  $\mathbf{K}$ , the Kaczmarz method con-

verges toward a solution that satisfies  $\mathbf{K}x \approx \mathbf{Y}$ . In the presence of noise or ill-posedness, however, *regularization* may be needed to stabilize and improve the quality of the solution. Below, we consider two common forms of regularization:

- **Tikhonov Regularization**, which penalizes large norms of the solution.
- **Total Variation (TV) Regularization**, which promotes piecewise-smooth solutions by penalizing large discrete gradients.

#### 4.1. Tikhonov Regularization

Tikhonov regularization (sometimes called  $\ell_2$ -regularization) introduces a parameter  $\lambda > 0$  to counteract noise amplification and large solution components. Conceptually, we replace the classical “distance and direction” step

$$d_i = \frac{\mathbf{Y}_i}{\|\mathbf{K}_i\|^2} \mathbf{K}_i$$

by incorporating a term  $\lambda x_k$  that dampens the updates:

$$d_i = \frac{\mathbf{Y}_i - \lambda x_k}{\|\mathbf{K}_i\|^2 + \lambda} \mathbf{K}_i.$$

When  $\lambda$  is large, updates become smaller, driving the solution to remain bounded; when  $\lambda \rightarrow 0$ , we recover the unregularized Kaczmarz step.

#### 4.2. Total Variation Regularization

Total Variation (TV) regularization is well-suited for problems where the solution  $x$  is expected to be piecewise constant or smooth. Instead of penalizing the size of  $x$  directly, we penalize its discrete gradient. In a one-dimensional setting (i.e., each component  $x_i$  for  $i = 1, \dots, n$ ), a simple finite-difference approximation of the TV penalty involves terms of the form

$$\alpha (2x_i - x_{i-1} - x_{i+1}),$$

where  $\alpha > 0$  controls the strength of the regularization. Thus, the “distance and direction” step can be modified to

$$d_i = \frac{\mathbf{Y}_i}{\|\mathbf{K}_i\|^2} \mathbf{K}_i - \alpha [\text{TV-penalty on } x_k].$$

In practice, one updates each component  $x_i$  with information from its neighbors, promoting locally constant or smoothly varying solutions. Boundary conditions (e.g., for  $x_1$  and  $x_n$ ) may need special handling.

#### 4.3. Algorithm: Kaczmarz Method with Regularization

This subsection presents the Kaczmarz method with regularization algorithm to solve the ILTs. In many applications (e.g., inverse problems or image reconstruction), the solution may also be constrained to be non-negative. The pseudocode below illustrates how to incorporate these features into each Kaczmarz iteration, and we subsequently discuss practical tips for parameter selection, boundary conditions, and stopping criteria.

**Algorithm 8:** Kaczmarz Method for Solving Linear Systems (with Optional Regularization)

**Input:** System matrix  $\mathbf{K} \in \mathbb{R}^{m \times n}$ , right-hand side  $\mathbf{Y} \in \mathbb{R}^m$ , initial guess  $x_0 \in \mathbb{R}^n$ , maximum iterations `max_iter`, tolerance `tol`, regularization type (*None*, *Tikhonov*, *TV*), regularization parameter(s)  $\lambda, \alpha, \dots$

**Output:** Approximate solution  $x \in \mathbb{R}^n$ .

Initialize  $x_0$ ;

**for**  $k = 1, 2, \dots, \text{max\_iter}$  **do**

**for**  $i = 1, 2, \dots, m$  **do**

    (a) Compute the distance and direction based on regularization:

$$\text{if Regularization} = \text{None: } d_i = \frac{\mathbf{Y}_i}{\|\mathbf{K}_i\|^2} \mathbf{K}_i$$

$$\text{if Regularization} = \text{Tikhonov: } d_i = \frac{\mathbf{Y}_i - \lambda x_k}{\|\mathbf{K}_i\|^2 + \lambda} \mathbf{K}_i$$

$$\text{if Regularization} = \text{TV: } d_i^j = \frac{\mathbf{Y}_i}{\|\mathbf{K}_i\|^2} (\mathbf{K}_i)_j - \alpha \left( 2x_k^j - x_k^{j-1} - x_k^{j+1} \right)$$

    (b) Compute the projection:

$$\text{Proj}_{\mathbf{K}_i}(x_k) = \frac{\langle \mathbf{K}_i, x_k \rangle}{\|\mathbf{K}_i\|^2} \mathbf{K}_i$$

    (c) Update the solution in the orthogonal direction:

$$x_{k+1} = x_k + d_i - \text{Proj}_{\mathbf{K}_i}(x_k)$$

    (d) (Optional) Impose Non-negativity:

$$x_{k+1} = \max(0, x_{k+1}).$$

**if**  $\|x_{k+1} - x_k\| < \text{tol}$  **then**

**break;**

**return**  $x_{k+1}$ ;

In practice, it is essential to choose the regularization parameters carefully. Larger values of  $\lambda$  or  $\alpha$  impose stronger regularization, leading to more stabilization but also potentially slower convergence and a biased estimate. Boundary conditions matter especially for the TV case, where one must decide how to handle edges (e.g., via forward/backward differences or reflection). If physical considerations dictate non-negativity (such as concentration profiles in inversion problems), projecting each iterate onto the positive orthant provides a simple and effective enforcement of  $x_k \geq 0$ . In terms of stopping criteria, one can monitor  $\|x_{k+1} - x_k\|$ , the residual norm  $\|\mathbf{K}x_{k+1} - \mathbf{Y}\|$ , or both. In summary, the Kaczmarz method furnishes a flexible row-by-row updating scheme; by directly embedding Tikhonov or Total Variation regularization into its “distance and direction” step, it effectively mitigates noise amplification while supporting structurally relevant constraints such as smoothness, sparsity, or non-negativity.

#### 4.4. Discussion and Practical Considerations

- **Choice of Parameters.** Both Tikhonov ( $\lambda$ ) and TV ( $\alpha$ ) regularization parameters significantly affect the trade-off between fitting the data and enforcing smoothness/small norms. These typically require some experimentation or cross-validation strategy to select optimal values.
- **Computational Cost.** Each inner iteration of the Kaczmarz method only accesses one row of  $A$  at a time, which can be advantageous for large problems. However, TV regularization can introduce

additional computation because one must update each component  $x_i$  with information from its neighbors.

- **Convergence Behavior.** The Kaczmarz method often converges quickly in practice, especially if the rows of  $A$  are processed in a randomized or optimal order. Regularization may slow down convergence slightly but typically leads to more stable and physically meaningful solutions in noisy or ill-conditioned scenarios.

By integrating either Tikhonov or TV regularization into the classical Kaczmarz method, we obtain a family of algorithms capable of handling ill-posed and large-scale problems, making them suitable for applications in signal/image processing, tomography, and inverse problems where stability and interpretability are crucial.

## 5. Wasserstein-Kaczmarz Method for Solving Inverse Laplace Transforms

ILTs for numerical solving can become particularly difficult when the parameter  $s$  contains limited information, the signal-to-noise ratio is low, or the sampling in the  $t$ -domain is sparse. Such scenarios exacerbate the ill-posed nature of the inverse problem, rendering standard regularization approaches ( $L_1$  or  $L_2$  norms) insufficient. These traditional techniques often fail to capture the underlying distribution accurately, resulting in oversmoothed or excessively noisy reconstructions under sparse or noisy data conditions.

Wasserstein regularization, inspired by optimal transport theory, is a robust method for aligning the reconstructed  $x(s)$  with a reference distribution  $x_{\text{ref}}(s)$ . This method is particularly effective for preserving the geometric structure of  $x(s)$ , even under high noise conditions. The  $p$ -Wasserstein distance between two distributions  $f$  and  $g$  is defined as:

$$W_p(f, g) = \left( \inf_{\gamma \in \Pi(f, g)} \int |x - y|^p d\gamma(x, y) \right)^{1/p},$$

where  $\Pi(f, g)$  represents all transport plans  $\gamma(x, y)$  with marginals  $f$  and  $g$ .

The Wasserstein-2 distance ( $W_2$ ) is typically used. It provides a robust alternative by taking advantage of the geometry of the solution space. By aligning the reconstructed solution with a prior reference  $x_{\text{ref}}$ , it compensates for missing information, preserves physical interpretability, and effectively handles sparse datasets. Moreover, the Wasserstein technique is particularly well-suited for sparse datasets because of its ability to incorporate geometric priors and maintain robustness against noise and missing data. The objective function is defined as:

$$\min_x \|\mathbf{K}x - \mathbf{Y}\|^2 + \lambda W_2^2(x, x_{\text{ref}}),$$

where:

- $W_2^2(x, x_{\text{ref}})$ : The Wasserstein-2 distance aligns the reconstructed  $x$  with the reference distribution  $x_{\text{ref}}$ .
- $x_{\text{ref}}$ : A prior distribution reflecting expected physical properties or sparsity patterns.

In scenarios where the reference distribution  $x_{\text{ref}}$  is unknown, Wasserstein regularization can be adapted to infer or estimate  $x_{\text{ref}}$ . The following approaches can be applied:

$$x_{\text{ref}} = \frac{1}{N},$$

where  $N$  is the number of diffusion coefficients. This ensures no prior bias in the reconstruction. Infer  $x_{\text{ref}}$  dynamically during reconstruction. Starting with  $x_{\text{ref}} = \frac{1}{N}$ , update  $x_{\text{ref}}$  at each iteration based on the reconstructed  $x$ :

$$x_{\text{ref}}^{k+1} = \alpha x^{k+1} + (1 - \alpha) x_{\text{ref}}^k,$$

where  $\alpha$  controls how quickly the reference adapts. This Wasserstein-Kaczmarz Method is presented in the Algorithm 9.

---

**Algorithm 9: Wasserstein-Kaczmarz Method for Solving Inverse Laplace Transforms**


---

**Input:**  $\mathbf{K} \in \mathbb{R}^{m \times n}$ : system matrix,  $\mathbf{Y} \in \mathbb{R}^m$ : right-hand side, maximum iterations  $\text{max\_iter}$ , tolerance  $\text{tol}$ , regularization type Wasserstein, Wasserstein parameter  $\lambda_w$ , relaxation factor  $\gamma$ .

**Output:** Approximate solution  $x \in \mathbb{R}^n$  and number of iterations.

**Initialize:**

1. Set  $x \leftarrow \mathbf{0}$  (zero vector).
2. Define a reference distribution  $x_{\text{ref}} \leftarrow \frac{1}{n} \mathbf{1}$  (uniform).
3. Precompute row norms:  $\|\mathbf{K}_{i,\cdot}\|^2$ , for  $i = 1, \dots, m$ .

**for**  $k = 1, 2, \dots, \text{max\_iter}$  **do**

Store previous solution:  $x_{\text{prev}} \leftarrow x$ .

**for**  $i = 1, 2, \dots, m$  **do**

(a) Compute the distance:

$$d_i = \frac{\mathbf{Y}_i}{\|\mathbf{K}_i\|^2} \mathbf{K}_i$$

(b) Compute the projection:

$$\text{Proj}_{\mathbf{K}_i}(x_k) = \frac{\langle \mathbf{K}_i, x_k \rangle}{\|\mathbf{K}_i\|^2} \mathbf{K}_i$$

(c) Wasserstein-Kaczmarz update:

- (i)  $x_{\text{temp}} = x + \gamma (d_i - \text{Proj}_{\mathbf{K}_i}(x_k))$ ,
- (ii)  $\nabla_{\text{wass}} = \lambda_w (x_{\text{temp}} - x_{\text{ref}})$ ,
- (iii)  $x \leftarrow x_{\text{temp}} - \nabla_{\text{wass}}$ .

(d) Enforce non-negativity:

$$x \leftarrow \max(0, x).$$

(d) Update reference solution:

$$x_{\text{ref}} \leftarrow \alpha x + (1 - \alpha) x_{\text{ref}}, \quad (\text{where } \alpha \text{ is a blending factor, e.g. } 0.5).$$

(e) Check convergence:

$$\text{if } \frac{\|x - x_{\text{prev}}\|}{\|x_{\text{prev}}\| + 10^{-15}} < \text{tol, then return } (x, k).$$

**return**  $(x, \text{max\_iter})$ .

---

This Wasserstein-Kaczmarz algorithm effectively recovers the sought function  $x(s)$  in the presence of noise, sparse data, or limited prior information. The Wasserstein term aligns  $x$  with  $x_{\text{ref}}$  to preserve geometrical or distributional characteristics even under challenging conditions, ensuring a more robust and physically interpretable solution compared to conventional methods.

## 6. Results

In this section, we present the performance results of the evaluated algorithms on synthetic datasets designed to assess their reconstruction accuracy and robustness under noise conditions. The methods compared include Classical Kaczmarz, Tikhonov-Regularized Kaczmarz, TV-Regularized

Kaczmarz, Wasserstein-Kaczmarz, CONTIN, Maximum Entropy (MaxEnt), TRAIIn, ITAMeD, and PALMA.

### 6.1. Synthetic Data Generation and Evaluation Metrics

Synthetic datasets were generated by simulating Laplace-transformed signals from analytically prescribed "ground-truth" distributions – both single- and multi-component – employing Gaussian and exponential functional forms. In order to emulate experimental measurement variability, a controlled noise level of 1% was introduced. In the context of practical diffusion NMR experiments, the absolute value of the diffusion coefficient, denoted by  $D$ , and the reconstruction's noise sensitivity are considered to be essential performance metrics. However, explicit variation of these parameters is not a prerequisite for the mathematical demonstrations presented herein. The performance of each reconstruction method was quantitatively evaluated using the following metrics:

- **Total Computation Time (seconds):** Measures the total runtime required for each algorithm to reach convergence.
- **Mean Squared Error (MSE):** Represents the average squared difference between reconstructed and true distributions, quantifying reconstruction accuracy.

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (x_{\text{recon},i} - x_{\text{true},i})^2.$$

- **Wasserstein Distance:** Measures the distributional difference between reconstructed and true distributions, indicating how closely the overall shapes match.
- **Total Variation (TV):** Quantifies the smoothness of the reconstructed distribution; lower TV indicates smoother solutions, while higher TV indicates sharper, potentially noisier solutions.

$$\text{TV}(x) = \sum_{i=1}^{N-1} |x_{i+1} - x_i|.$$

- **Peak Signal-to-Noise Ratio (PSNR):** Provides a logarithmic measure of the reconstruction quality, with higher values corresponding to better accuracy.

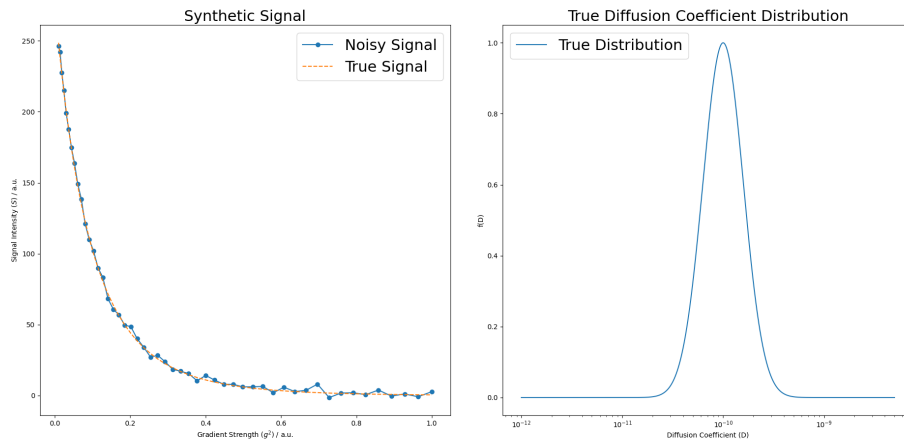
$$\text{PSNR} = 10 \log_{10} \left( \frac{\max(x_{\text{true}})^2}{\text{MSE}} \right).$$

All algorithms were uniformly initialized, and regularization parameters were optimized through the L-curve method.

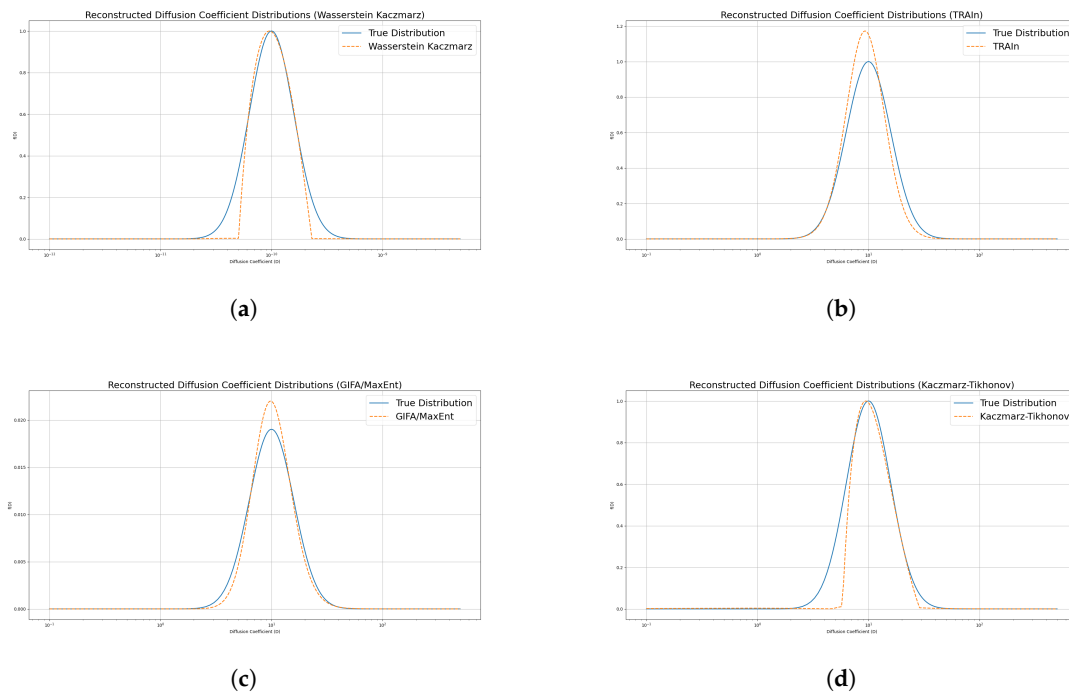
### 6.2. Single-Component Data Reconstruction

Table 1 summarizes the reconstruction results for single-component synthetic data at a 1% noise level, displaying total computation time, MSE, Wasserstein distance, total variation, and PSNR.

Figure 1 illustrates the underlying true distribution along with both the true signal and its noisy counterpart. This visualization provides a qualitative baseline that helps in assessing the effectiveness of the reconstruction algorithms in preserving the original signal structure while mitigating noise.



**Figure 1.** Visual comparison of the true distribution, the true signal, and the noisy signal.



**Figure 2.** Comparison of Reconstruction Algorithms on Single-Component Synthetic Data (1% noise level). Panel (a) shows the Wasserstein-Kaczmarz method, which achieves robust noise mitigation with minimal reconstruction error as quantified by low MSE and Wasserstein distance. Panel (b) displays the TRAI algorithm, noted for its superior accuracy and balanced performance between error reduction and computational efficiency. Panel (c) illustrates the Maximum Entropy (MaxEnt) approach, offering a compromise that preserves signal details while effectively suppressing noise. Panel (d) presents the Tikhonov-Kaczmarz method, which delivers efficient regularization with an optimal trade-off between reconstruction fidelity and processing time.

**Table 1.** Algorithm performance comparison on single-component synthetic data (1% noise level).

Algorithm	MSE	Wasserstein	TV	PSNR (dB)	Time (s)
Classical Kaczmarz	1.6E-7	5.6E-5	3.5E-2	33.57	0.42
Tikhonov-Kaczmarz	2.4E-8	5.2E-5	3.7E-7	33.94	1.26
TV-Kaczmarz	4.1E-7	1.3E-4	2.8E-2	29.5	3.06
Wasserstein-Kaczmarz	4.7E-8	6.1E-5	6.9E-3	21.54	0.53
CONTIN	1.6E-7	9.0E-5	3.2E-2	33.55	2.13
MaxEnt	1.1E-7	5.7E-5	4.2E-2	35.3	1.46
TRAI <sub>n</sub>	1.5E-8	5.4E-8	8.1E-3	29.4	19.36
ITAMeD	2.2E-8	6.2E-5	1.9E-3	26.7	4.35
PALMA	1.2E-6	3.2E-4	3.3E-2	24.94	261.88

As illustrated in Table 1, the performance of various reconstruction algorithms on single-component synthetic data with 1% noise is demonstrated. The TRAI<sub>n</sub> algorithm attains the lowest mean squared error (MSE) of 1.5E-8 and the smallest Wasserstein distance of 5.4E-8, thereby signifying its superior reconstruction accuracy. However, it should be noted that this is accompanied by a higher computational cost (19.36 s). In contrast, the Classical Kaczmarz method, despite its high processing speed (0.42 s), exhibits comparatively elevated errors with an MSE of 1.6E-7 and a greater total variation. The Tikhonov-Kaczmarz and TV-Kaczmarz methods provide a balanced trade-off between accuracy and efficiency, with Tikhonov-Kaczmarz showing a particularly low MSE (2.4E-8) and minimal total variation (3.7E-7) at a moderate runtime (1.26 s). Additionally, MaxEnt delivers a high peak signal-to-noise ratio (35.3 dB), whereas PALMA, despite its extended computational time (261.88 s), exhibits the poorest accuracy, as evidenced by its high MSE and Wasserstein values. The results emphasise the inherent trade-offs between computational efficiency and reconstruction fidelity, guiding the selection of an appropriate algorithm based on specific application needs. The results in Table 1 indicate that the **TRAI<sub>n</sub>** and **Wasserstein-Kaczmarz** methods achieve the highest reconstruction accuracy by recording the lowest MSE and Wasserstein distance values, with competitive PSNR figures. Conversely, while the Classical Kaczmarz method is the fastest in terms of computational time, its performance is compromised by higher errors. Tikhonov and TV regularizations achieve a balance between accuracy and computational cost, thus rendering them viable options when moderate reconstruction quality is deemed acceptable alongside efficiency.

### 6.3. Multi-Component Distribution Reconstruction

The best algorithms evaluated in the previous section were then subjected to further evaluation through the utilisation of multi-modal synthetic distributions, characterised by closely overlapping diffusion peaks. This approach emulates realistic and analytically challenging scenarios. In such conditions, characterised by stringent constraints, **Wasserstein-Kaczmarz** and **TRAI<sub>n</sub>** demonstrated remarkable stability and precision, adeptly disentangling and retrieving individual distribution components in the face of substantial noise contamination. From a pragmatic standpoint, the minimum resolvable separation between adjacent diffusion coefficients ( $\Delta D_{min}$ ) is a pivotal parameter for experimental applicability. Nevertheless, it is not necessary to explicitly investigate  $\Delta D_{min}$  within the purely mathematical framework presented herein.

Table 2 presents a quantitative comparison of the reconstruction algorithms' performance on multi-component synthetic data contaminated with 1% noise. Metrics reported include Mean Squared Error (MSE), Wasserstein distance, Total Variation (TV), Peak Signal-to-Noise Ratio (PSNR), and computational time.

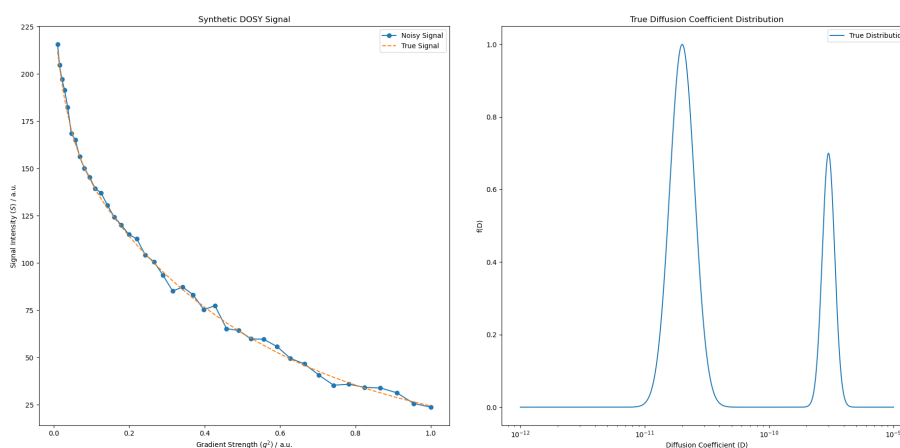
Figure 3 provides a visual representation of the true distribution, alongside the clean and noisy signals. This qualitative visualization underscores the severity of noise interference and highlights the necessity for robust reconstruction methods.

Figure 4 visually demonstrates the algorithms' reconstruction fidelity, particularly highlighting the robustness and accuracy of Wasserstein-Kaczmarz. Table 2 reveals clear differences among the evaluated methods. **TRAI<sub>n</sub>** achieved the lowest MSE and Wasserstein distance, indicating su-



perior accuracy in reconstructing the original distributions despite having a higher Total Variation. **Wasserstein-Kaczmarz** also displayed excellent results with slightly higher MSE but significantly reduced computational time, showcasing its strength as an efficient and accurate alternative. **MaxEnt** provided rapid reconstruction but compromised somewhat on Wasserstein distance accuracy. Lastly, **Tikhonov-Kaczmarz**, although highly effective at noise mitigation, lagged in computational speed. Based on these observations, the algorithms are ranked as follows:

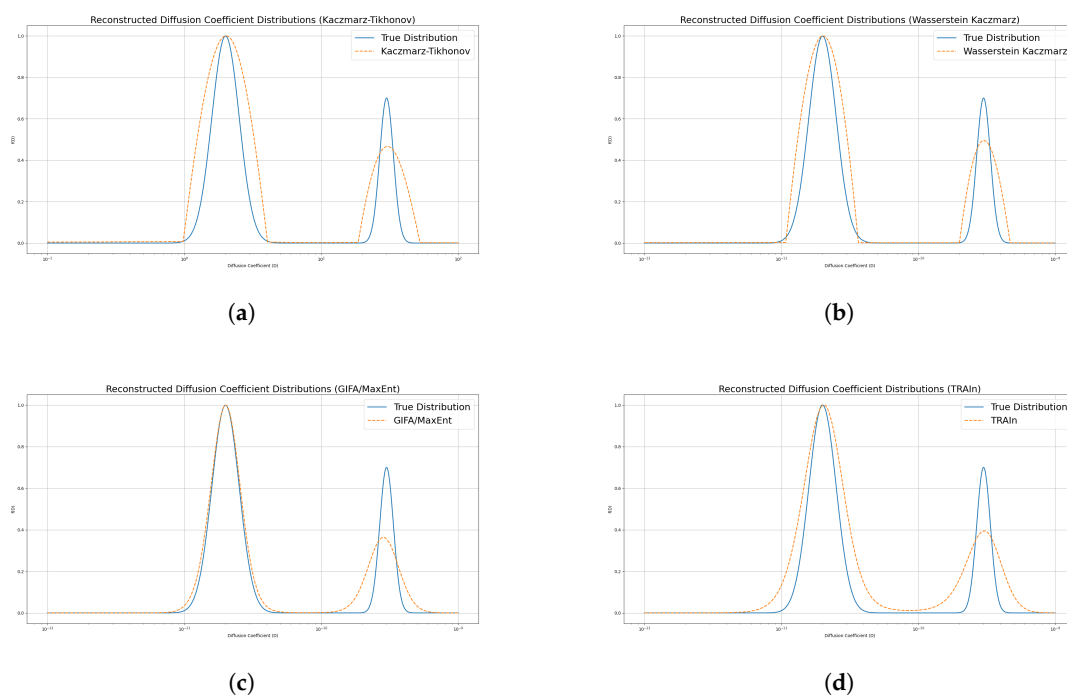
1. **TRAI**n (Best accuracy, balanced efficiency)
2. **Wasserstein-Kaczmarz** (Excellent accuracy and fastest reconstruction)
3. **MaxEnt** (Fast computation, moderate accuracy)
4. **Tikhonov-Kaczmarz** (Good noise reduction, slower computational time)



**Figure 3.** Comparison of True and Noisy Signals with the Underlying True Distribution.

**Table 2.** Algorithm performance comparison on multi-component synthetic data (1% noise level).

Algorithm	MSE	Wasserstein	TV	PSNR (dB)	Time (s)
Tikhonov-Kaczmarz	5.4E-7	3.5E-4	1.2E-2	19.19	108.22
Wasserstein-Kaczmarz	3.6E-7	2.6E-4	1.4E-2	20.98	6.67
MaxEnt	4.1E-7	23.1E-4	1.6E-2	20.42	2.08
TRAI	2.1e-07	2.3E-4	7.9e-02	19.49	22.35



**Figure 4.** Reconstruction of Multi-Component Synthetic Data (1% Noise Level). Panel (a) Tikhonov-Kaczmarz shows effective noise suppression with minimal error; Panel (b) Wasserstein-Kaczmarz demonstrates superior accuracy and computational efficiency; Panel (c) Maximum Entropy (MaxEnt) provides good signal preservation with acceptable noise management; Panel (d) TRAI highlights excellent reconstruction fidelity balanced with moderate processing time.

#### 6.4. Practical Considerations

Optimal parameter tuning was indispensable to each algorithm's efficacy. Regularization parameters (e.g.,  $\lambda$  and  $\alpha$  for Tikhonov and total-variation methods) were selected via L-curve analysis [36,39]. Furthermore, in practical diffusion NMR measurements, both the accurate recovery of absolute diffusion coefficient values and the reconstruction's sensitivity to measurement noise critically affect robustness, while the minimum resolvable separation between adjacent diffusion coefficients ( $\Delta D_{min}$ ) represents a key benchmark for peak discrimination. Incorporation of non-negativity constraints further enhanced the physical plausibility of the reconstructed distributions. However, within the purely mathematical framework presented herein, explicit investigation of noise sensitivity and  $\Delta D_{min}$  was deemed unnecessary.

## 7. Discussion

This study highlights the significant potential of adapting the Kaczmarz method for solving inverse Laplace transform (ILT) problems, particularly when enhanced with robust regularization techniques. The evaluation across multiple synthetic scenarios demonstrates the substantial advantages of these adaptations in terms of accuracy, stability, and computational efficiency compared to traditional methods.

The proposed Wasserstein-Kaczmarz and other algorithms already developed emerged as particularly effective under challenging conditions, such as sparse, noisy, or multi-modal distributions. Wasserstein-Kaczmarz demonstrated superior computational efficiency and strong reconstruction accuracy, making it especially valuable for time-sensitive applications or computationally constrained scenarios. Meanwhile, TRAI provided exceptional accuracy, albeit with a slightly increased computational load, reinforcing its suitability for applications requiring high-fidelity reconstruction.

While established algorithms such as MaxEnt, ITAMeD, and PALMA each possess distinct strengths—ranging from rapid computation to specific robustness features—they also exhibit limi-

tations, particularly in complex or highly noisy conditions. For instance, MaxEnt, despite its rapid execution, showed a compromise in accuracy, particularly evident through increased Wasserstein distances. Similarly, Tikhonov-Kaczmarz, despite its strong performance in noise suppression, was limited by slower computational speeds.

From a practical laboratory standpoint, several additional considerations must guide implementation. First, accurate determination of the absolute diffusion coefficient  $D$  is indispensable for meaningful physical interpretation, necessitating careful calibration and reference measurements. Second, the minimum resolvable separation between adjacent diffusion peaks ( $\Delta D_{min}$ ) must be quantified empirically to ensure reliable discrimination of closely spaced species. Finally, a more exhaustive investigation of signal-to-noise ratio (SNR) effects—in particular, how SNR degradation influences both peak recovery and parameter estimation—will be essential to translate these algorithms from simulation to experiment. Although such detailed experimental studies fall outside the strict mathematical framework presented here, they constitute the next steps for validating and optimizing Kaczmarz-based ILT solvers in real-world analytical contexts.

Beyond the specific findings of this study, the broader implications suggest significant potential for extending Kaczmarz-based solvers to a wide variety of other inverse problems, including tomography, seismic inversion, and medical imaging. Despite the clear benefits, limitations such as parameter tuning complexity and computational scalability need further exploration. Future research directions should focus on developing adaptive parameter selection methods and investigating hybrid regularization strategies that could further enhance performance and robustness across diverse classes of inverse problems.

## 8. Conclusions

The adapted Kaczmarz method, enhanced with advanced regularization strategies such as Wasserstein and trust-region techniques, represents a significant advancement in addressing the complexities associated with inverse Laplace transform problems. The results underscore the Wasserstein-Kaczmarz algorithm's optimal balance between computational efficiency and reconstruction accuracy, positioning it as a strong candidate for real-world applications demanding rapid and accurate solutions.

The TRAI algorithm also stands out by delivering unmatched accuracy under challenging conditions, demonstrating its value for applications prioritizing precise reconstructions despite higher computational demands.

The methodological advances presented herein not only establish new performance benchmarks for ILT solvers but also pave the way for further investigation of iterative regularization techniques in addressing ill-posed inverse problems across a wide range of scientific and engineering applications. To ensure reproducibility and to support ongoing development, the full implementation of all algorithms described in this study is available at the project's GitHub repository: <https://github.com/fmarrabal/kaczmarz-ilt-solver>.

**Author Contributions:** Conceptualization, M.G.-L., I.F. and F.M.A.-C.; methodology, M.G.-L., E.V. and V.V.; software, M.G.-L. and V.V.; validation, M.G.-L., E.V. and V.V.; formal analysis, M.G.-L.; investigation, M.G.-L., E.V. and V.V.; resources, F.M.A.-C.; data curation, M.G.-L.; writing—original draft preparation, M.G.-L.; writing—review and editing, M.G.-L., E.V., V.V., I.F. and F.M.A.-C.; visualization, M.G.-L.; supervision, I.F. and F.M.A.-C.; project administration, F.M.A.-C.; funding acquisition, I.F. and F.M.A.-C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research has been funded by the State Research Agency of the Spanish Ministry of Science and Innovation (PID2021-126445OB-I00), and by the Gobierno de España MCIN/AEI/10.13039/501100011033 and Unión Europea "Next Generation EU"/PRTR (PDC2021-121248-I00, PLEC2021-007774 and CPP2022-009967).

**Conflicts of Interest:** The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

## References

1. Carslaw, H.S.H.S.; Jaeger, J.C.J.C. *Conduction of heat in solids / by H. S. Carslaw and J. C. Jaeger.*, second edition. ed.; Clarendon Press: Oxford, 1959.
2. Monde, M. Analytical method in inverse heat transfer problem using laplace transform technique. *International Journal of Heat and Mass Transfer* **2000**, *43*, 3965–3975. [https://doi.org/https://doi.org/10.1016/S0017-9310\(00\)00040-5](https://doi.org/https://doi.org/10.1016/S0017-9310(00)00040-5).
3. Desoer, C.A.; Kuh, E.S. *Basic Circuit Theory*; McGraw-Hill, 1969.
4. Inman, D.J. *Engineering vibration / Daniel J. Inman.*, 2nd ed. ed.; Prentice Hall: Upper Saddle River, N.J., 2001.
5. Ogata, K. *Modern Control Engineering*, 4th ed.; Prentice Hall PTR: USA, 2001.
6. Ioannidis, G.S.; Nikiforaki, K.; Kalaitzakis, G.; Karantanas, A.; Marias, K.; Maris, T.G. Inverse Laplace transform and multiexponential fitting analysis of T2 relaxometry data: a phantom study with aqueous and fat containing samples. *European Radiology Experimental* **2020**, *4*, 28. <https://doi.org/10.1186/s41747-020-0154-5>.
7. Hadamard, J. *Lectures on Cauchy's problem in linear partial differential equations*; Yale University / Mrs. Hepsa Ely Silliman memorial lectures, Yale university press: New York, 1923.
8. Bertero, M.; Boccacci, P. *Introduction to Inverse Problems in Imaging*; CRC Press: Boca Raton, 2020. <https://doi.org/10.1201/9780367806941>.
9. Beck, J.V.; Clair, C.R.; Blackwell, B. *Inverse heat conduction*; John Wiley and Sons Inc., New York, NY, 1985.
10. Ha, W.; Shin, C. Laplace-domain full-waveform inversion of seismic data lacking low-frequency information. *GEOPHYSICS* **2012**, *77*, R199–R206. <https://doi.org/10.1190/geo2011-0411.1>.
11. Holt, S.; Qian, Z.; van der Schaar, M. Neural Laplace: Learning diverse classes of differential equations in the Laplace domain. *Proceedings of Machine Learning Research* **2022**, *162*, 8811–8832, [2206.04843].
12. Calvetti, D.; Kaipio, J.P.; Somersalo, E. Inverse problems in the Bayesian framework. *Inverse Problems* **2014**, *30*, 110301. <https://doi.org/10.1088/0266-5611/30/11/110301>.
13. Calvetti, D.; Somersalo, E. Hypermodels in the Bayesian imaging framework. *Inverse Problems* **2008**, *24*, 34013. <https://doi.org/10.1088/0266-5611/24/3/034013>.
14. Calvetti, D.; Somersalo, E. *Bayesian Scientific Computing*; Vol. 215, *Applied Mathematical Sciences*, Springer International Publishing: Cham, 2023. <https://doi.org/10.1007/978-3-031-23824-6>.
15. Bioucas-Dias, J.M.; Figueiredo, M.A.T. A New Twist on Image Denoising and Restoration. *IEEE Transactions on Image Processing* **2007**, *16*, 3343–3355.
16. Jaynes, E.T. *Probability Theory. The Logic of Science*; Cambridge University Press: Cambridge, 2002.
17. Golub, G.H.; Van Loan, C.F. *Matrix computations (3rd ed.)*; Johns Hopkins University Press: USA, 1996.
18. Hansen, P.C. *Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion*; Society for Industrial and Applied Mathematics: USA, 1999.
19. Tikhonov, A.N. On the solution of ill-posed problems and the method of regularization. *Dokl. Akad. Nauk SSSR* **1963**, *151*, 501—504. <https://doi.org/10.1090/S0025-5718-1974-0375817-5>.
20. Kaczmarz, S. Angenäherte Auflösung von Systemen linearer Gleichungen (Approximate Solution for Systems of Linear Equations). *Bulletin International de l'Académie Polonaise des Sciences et des Lettres. Classe des Sciences Mathématiques et Naturelles. Série A, Sciences Mathématiques* **1937**, *35*, 355–357.
21. Strohmer, T.; Vershynin, R. A Randomized Kaczmarz Algorithm with Exponential Convergence. *Journal of Fourier Analysis and Applications* **2009**, *15*, 262–278, [arXiv:math/0702226]. <https://doi.org/10.1007/s00041-008-9030-4>.
22. Gower, R.M.; Richtarik, P. Randomized iterative methods for linear systems. *SIAM Journal on Matrix Analysis and Applications* **2015**, *36*, 1660–1690, [1506.03296]. <https://doi.org/10.1137/15M1025487>.
23. He, S.; Dong, Q.L.; Li, X. The randomized Kaczmarz algorithm with the probability distribution depending on the angle. *Numerical Algorithms* **2023**, *93*, 415–440. <https://doi.org/10.1007/s11075-022-01422-8>.
24. Gordon, R.; Bender, R.; Herman, G.T. Algebraic Reconstruction Techniques (ART) for three-dimensional electron microscopy and X-ray photography. *Journal of Theoretical Biology* **1970**, *29*, 471–481. [https://doi.org/10.1016/0022-5193\(70\)90109-8](https://doi.org/10.1016/0022-5193(70)90109-8).
25. Chi, Y.; Lu, Y.M. Kaczmarz Method for Solving Quadratic Equations. *IEEE Signal Processing Letters* **2016**, *23*, 1183–1187. <https://doi.org/10.1109/LSP.2016.2590468>.
26. Kelly, J.D. Reconciliation of process data using other projection matrices. *Computers and Chemical Engineering* **1999**, *23*, 785–789. [https://doi.org/10.1016/S0098-1354\(99\)00006-X](https://doi.org/10.1016/S0098-1354(99)00006-X).
27. Gamboa, F.; Gassiat, E. Bayesian methods and maximum entropy for ill-posed inverse problems. *The Annals of Statistics* **1997**, *25*, 328–350. <https://doi.org/10.1214/aos/1034276632>.

28. Wright, K.M. Chapter 2 Maximum entropy methods in nmr data processing. In *Data Processing*; 1996; pp. 25–43. [https://doi.org/10.1016/S0922-3487\(96\)80039-6](https://doi.org/10.1016/S0922-3487(96)80039-6).
29. Delsuc, M.A.; Malliavin, T.E. Maximum Entropy Processing of DOSY NMR Spectra. *Analytical Chemistry* **1998**, *70*, 2146–2148. <https://doi.org/10.1021/ac9800715>.
30. Coleman, T.F.; Li, Y. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization* **1996**, *6*, 418–445. <https://doi.org/10.1137/0806023>.
31. Xu, K.; Zhang, S. Trust-Region Algorithm for the Inversion of Molecular Diffusion NMR Data. *Analytical Chemistry* **2014**, *86*, 592–599. <https://doi.org/10.1021/ac402698h>.
32. Urbańczyk, M.; Bernin, D.; Koźmiński, W.; Kazmierczuk, K. Iterative Thresholding Algorithm for Multiexponential Decay Applied to PGSE NMR Data. *Analytical Chemistry* **2013**, *85*, 1828–1833. <https://doi.org/10.1021/ac3032004>.
33. Cherni, A.; Chouzenoux, E.; Delsuc, M.A. PALMA, an improved algorithm for DOSY signal processing. *Analyst* **2017**, *142*, 772–779. <https://doi.org/10.1039/c6an01902a>.
34. Arrabal-Campos, F.M.; Aguilera-Sáez, L.M.; Fernández, I. A diffusion NMR method for the prediction of the weight-average molecular weight of globular proteins in aqueous media of different viscosities. *Analytical Methods* **2019**, *11*, 142–147. <https://doi.org/10.1039/C8AY01817K>.
35. Pessôa, L.C.; Attar, S.B.e.; Sánchez-Zurano, A.; Ciardi, M.; Morillas-España, A.; Ruiz-Martínez, C.; Fernández, I.; Arrabal-Campos, F.M.; Pontes, L.A.; Silva, J.; et al. Exopolysaccharides as bio-based rheology modifiers from microalgae produced on dairy industry waste: Towards a circular bioeconomy approach. *International Journal of Biological Macromolecules* **2024**, p. 135246. <https://doi.org/10.1016/j.ijbiomac.2024.135246>.
36. Hansen, P.C. The L-Curve and its Use in the Numerical Treatment of Inverse Problems. in *Computational Inverse Problems in Electrocardiology*, ed. P. Johnston, *Advances in Computational Bioengineering* **2000**, *4*, 119–142. <https://doi.org/10.1.1.33.6040>.
37. Golub, G.H.; Heath, M.; Wahba, G. Generalized Cross-Validation as a Method for Choosing a Good Ridge Parameter. *Technometrics* **1979**, *21*, 215–223. <https://doi.org/10.1080/00401706.1979.10489751>.
38. Provencher, S.W. CONTIN: A general purpose constrained regularization program for inverting noisy linear algebraic and integral equations. *Computer Physics Communications* **1982**, *27*, 229–242. [https://doi.org/10.1016/0010-4655\(82\)90174-6](https://doi.org/10.1016/0010-4655(82)90174-6).
39. Lloyd, J.J.; Taylor, C.J.; Lawson, R.S.; Shields, R.A. The use of the L-curve method in the inversion of diffusion battery data. *Journal of Aerosol Science* **1997**, *28*, 1251–1264. [https://doi.org/10.1016/S0021-8502\(97\)00018-9](https://doi.org/10.1016/S0021-8502(97)00018-9).

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.