International Journal on Magnetic Particle Imaging
Vol 10, No 1, Suppl 1, Article ID 2403028, 4 Pages

Infinite Science Publishing

Proceedings Article

RegularizedLeastSquares.jl: Modality Agnostic Julia Package for Solving Regularized Least Squares Problems

Niklas Hackelberg[a,b,*] · Mirco Grosser[a,b] · Artyom Tsanda[a,b] · Fabian Mohn[a,b] · Konrad Scheffler[a,b] · Martin Möddel[a,b] · Tobias Knopp[a,b]

[a]Section for Biomedical Imaging, University Medical Center Hamburg-Eppendorf, Hamburg, Germany
[b]Institute for Biomedical Imaging, Hamburg University of Technology, Hamburg, Germany
[*]Corresponding author, email: niklas.hackelberg@tuhh.de

Abstract

Image reconstruction in Magnetic Particle Imaging (MPI) is an ill-posed linear inverse problem. A standard method for solving such a problem is the regularized least squares approach, which uses, a regularization function to reduce the impact of measurement noise in the reconstructed image by leveraging prior knowledge. Various optimization algorithms, including the Kazcmarz method or the Alternating Direction Method of Multipliers (ADMM), and regularization functions, such as l_2 or Fused Lasso priors have been employed. Therefore, the creation and implementation of cutting-edge image reconstruction techniques necessitate a robust and adaptable optimization framework. In this work, we present the open-source Julia package RegularizedLeastSquares.jl, which provides a large selection of common optimization algorithms and allows flexible inclusion of regularization functions. These features enable the package to achieve state-of-the-art image reconstruction in MPI.

I. Introduction

Image reconstruction for medical imaging, such as Magnetic Particle Imaging (MPI) or Magnetic Resonance Imaging (MRI), involves solving an ill-posed inverse problem. A common approach to solve such a problem, is to consider the regularized least squares problem

$$\underset{\boldsymbol{x}}{\operatorname{argmin}} \; \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2 + r(\boldsymbol{x}), \quad (1)$$

where $\boldsymbol{A}$ is the imaging operator, $\boldsymbol{x}$ is the solution, $\boldsymbol{b}$ is the measured data and r is a regularization function.

The first term is the data fidelity cost and it ensures that the model $\boldsymbol{A}\boldsymbol{x}$ fits to the measurements $\boldsymbol{b}$. Here the choice of a least-squares term is popular, which can be motivated by the Gauss-Markov theorem. The second term ensures that $\boldsymbol{x}$ matches the prior knowledge, such as an expectation on the smoothness of the result, expressed by the regularization function.

In the context of a system-matrix based MPI reconstruction $\boldsymbol{A} \in \mathbb{C}^{M\times N}$ and $\boldsymbol{b} \in \mathbb{C}^{M}$ are Fourier coefficients of the system matrix and the measurement voltage, respectively. And lastly $\boldsymbol{x} \in \mathbb{R}_+^N$ is the particle concentration. The Kaczmarz algorithm is often used to solve problem (1), as it offers fast convergence due to the rows of the system matrix being nearly orthogonal [1]. However, several others algorithms such as the Conjugate Gradient Normal Residual (CGNR) [2], Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) [3] or Alternating Direction Method of Multipliers (ADMM) [4] have also been used in MPI. These methods have been combined with a variety of regularization functions, including an l_2 prior for Tikhonov regularization [2], an l_1 and total variation

(TV) regularization prior (Fused Lasso) [4, 5], an l_1-prior in the wavelet domain [3] and a machine-learning based plug-and-play prior (PP) [6]. Finally, MPI reconstruction methods often apply a variety of additional modifications, such as projections of $\boldsymbol{x}$ into a target domain, weightings of the system matrix and scaling of the regularization parameters. As a result MPI reconstruction packages require a very flexible optimization framework capable of handling this large range of possible problem formulations.

In this work we present the open source Julia package RegularizedLeastSquares.jl, which provides functionality to solve problem (1). The package places a focus on providing flexibility and extensibility, making it particularly suitable for algorithmic research in MPI. More precisely, it provides a variety of different optimization algorithms and regularization terms that can be combined to compose custom regularization functions. Currently, the package is used as the reconstruction layer of the Julia image reconstruction packages MPIReco.jl [7] and MRIReco.jl [8].

II. Methods and Materials

RegularizedLeastSquares.jl consists of one type hierarchy for solvers and one for regularization terms. Solvers are constructed with a given image operator, an (optional) list of regularization terms, and solver specific arguments, such as the iteration number for iterative solvers. While solvers and regularization terms are decoupled concepts, not every solver can accept every combination of terms. It is possible to list applicable solvers for a given combination of image operator, data and terms.

Regularization Terms are the building blocks used to construct a regularization function. They can be separated into two groups, the first of which contains the core regularization terms. These terms implement a proximal map [9] either as a term with a regularization parameter λ, such as the l_1 or l_2 prior, or as a projection, such as a mapping to $\mathbb{R}_+^N$. The second group of regularization terms allows the nesting of terms to adapt the inputs to the proximal maps. An example of such a term would be the transformation of $\boldsymbol{x}$ to the Wavelet domain or a normalization of λ based on the energy of the system matrix rows or the measurement data.

Creating a regularization function r_i for an l_1 prior in the wavelet domain looks as follows:

```
# Prepare regularization terms
core = L1Regularization(0.8)
wop = WaveletOp(Float32, shape = (32,32))
reg = TransformedRegularization(core, wop)
```

First a (core) l_1 prior with $\lambda = 0.8$ is created. It is then nested inside a regularization term that transforms the proximal map input into the wavelet domain.

Solvers are the optimization algorithms used to solve problem (1) for a given set of regularization terms. The solvers are organized based on their algorithm type, such as row-action algorithms like Kaczmarz, proximal gradient algorithms like FISTA or primal dual algorithms like ADMM. Solvers are not limited to working with dense matrices and instead can work on any matrix-like data type implementing certain interface functions. This allows using matrix-free imaging operators, which can save memory and allow for efficient computations of matrix-vector products.

Solvers are created and used like this:

```
# Create solver
solver = Kaczmarz(A, reg = reg, ...)
x = solve(solver, b)
```

More detailed examples and a list of available solvers, regularization terms and their parameters can be found in the documentation[1].

III. Results

To showcase the versatility of RegularizedLeastSquares.jl, we reconstructed two phantoms with different regularization terms and solver combinations considering the common MPI least-squares approach

$$\underset{\boldsymbol{x}\in\mathbb{R}_+^N}{\operatorname{argmin}} \left\|\boldsymbol{A}^{\text{red}}\boldsymbol{x}-\boldsymbol{b}^{\text{red}}\right\|_{\boldsymbol{W}}^2 + r_i(\boldsymbol{x}), \tag{2}$$

where $\boldsymbol{W}$ is a symmetric, positive definite weighting matrix and $\|\boldsymbol{x}\|_{\boldsymbol{W}} := \|\boldsymbol{W}^{\frac{1}{2}}\boldsymbol{x}\|_2$ denotes the weighted Euclidean norm. The measurements were performed with the preclinical MPI system (25/20 FF) from Bruker. The first phantom is a spiral with two windings, while the second phantom contains three dots arranged at the corners of a triangle. Both phantoms are filled with the tracer perimag. The reconstructions are based on a system-matrix approach with a 2D 32×32 system matrix. Both $\boldsymbol{A}$ and $\boldsymbol{b}$ were frequency filtered according to an SNR threshold of 1.5 yielding $\boldsymbol{A}^{\text{red}}$ and $\boldsymbol{b}^{\text{red}}$ with $K < M$ frequency components. Additionally, $\boldsymbol{A}^{\text{red}}$ and $\boldsymbol{b}^{\text{red}}$ were multiplied with a weighting matrix $\boldsymbol{W}$. The matrix was chosen to realize diagonal whitening [10] based on the standard deviation of frequency components of background measurements taken during the system-matrix calibration.

Each column in Fig. 1 shows a different combination of solver and regularization terms. The following regularization functions were considered:

[1] https://github.com/JuliaImageRecon/RegularizedLeastSquares.jl/

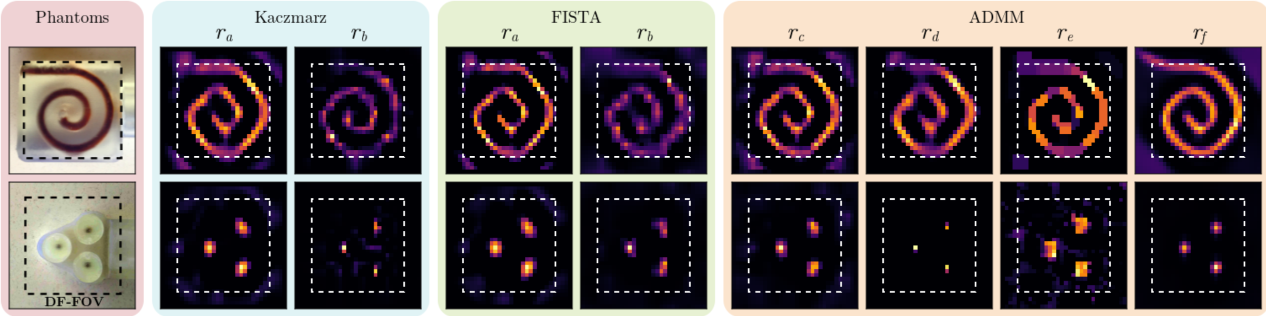

Figure 1: Image reconstruction performed with RegularizedLeastSquares.jl. A spiral and a three dots phantom were measured at the preclinical MPI system from Bruker and reconstructed with a variety of regularization terms and optimization algorithms. The field of view (FOV) covered by the drive field (DF) of the scanner is marked with dashed boxes.

$$
\begin{aligned}
r_a(\boldsymbol{x}) &= \lambda\,\|\boldsymbol{x}\|_2^2 \\
r_b(\boldsymbol{x}) &= \lambda\,\|\Phi\boldsymbol{x}\|_1 \\
r_c(\boldsymbol{x}) &= \lambda_1\,\|\Phi\boldsymbol{x}\|_1 + \lambda_2\,\|\boldsymbol{x}\|_2^2 \\
r_d(\boldsymbol{x}) &= \lambda_1\,\|\boldsymbol{x}\|_1 + \lambda_2\,\|\boldsymbol{x}\|_2^2 \\
r_e(\boldsymbol{x}) &= \lambda_1\,\|\boldsymbol{x}\|_1 + \lambda_2\,\|\boldsymbol{x}\|_{\mathrm{TV}} \\
r_f(\boldsymbol{x}) &= f_{\mathrm{pp}}(\boldsymbol{x})
\end{aligned}
$$

Here, Φ denotes the wavelet transform and f_{pp} is a learned plug-and-play regularization term that is usually indirectly defined by its proximal mapping function. In this work, we use the denoiser implementation (i.e. the exact network weights) provided by [6]. Note that the various reconstruction parameters, such as the regularization parameters, the number of iterations, and the solver-specific parameters, have been tuned by hand without the intent to reconstruct optimally. Focus of these imaging experiments is to show the flexibility of the package and the effects of the regularization functions. A comparison of regularization functions, solvers, and their optimized parameters in terms of image quality can be found in [11].

One can see the impacts of the different priors on the resulting images. The sparsifying effect of the l_1 term is particularly evident in the dot phantoms, where some reconstructions show only two to three pixels per dot and essentially no signal outside the dots. Similarly, the dot phantoms also clearly show the blurring effect of the l_2 term. Especially for the spiral phantoms, the l_2 term resulted in an artifact at the border of the image, which could be removed by additional priors. While the chosen phantoms are no good use cases for the TV prior, the resulting flattening along parts of the spiral can be observed.

IV. Discussion and Conclusion

In this work we have given an overview of the Julia package RegularizedLeastSquares.jl, which is capable of optimizing regularized least squares problems using a variety of state-of-the-art reconstruction algorithms for MPI and other imaging modalities. The solvers are implemented in an efficient manner and perform optimization steps such as preallocation. In the future we plan to extend the framework with GPU support and further machine-learning integration. By allowing for a flexible construction and combination of optimization algorithms and regularization functions, a user can implement, experiment with, and compare different MPI reconstruction algorithms.

Author's statement

Conflict of interest: Authors state no conflict of interest.

References

[1] J. Rahmer, J. Weizenecker, B. Gleich, and J. Borgert. Signal encoding in magnetic particle imaging: Properties of the system function. *BMC Medical Imaging*, 9(1):1–21, 2009, doi:10.1186/1471-2342-9-4.

[2] T. Knopp, J. Rahmer, T. F. Sattel, S. Biederer, J. Weizenecker, B. Gleich, J. Borgert, and T. M. Buzug. Weighted iterative reconstruction for magnetic particle imaging. *Physics in Medicine & Biology*, 55(6):1577, 2010, doi:10.1088/0031-9155/55/6/003.

[3] F. Lieb and T. Knopp. A wavelet-based sparse row-action method for image reconstruction in magnetic particle imaging. *Medical Physics*, 48(7):3893–3903, 2021.

[4] S. Ilbey, C. B. Top, A. Güngör, T. Çukur, E. U. Saritas, and H. E. Güven. Comparison of System-Matrix-Based and Projection-Based Reconstructions for Field Free Line Magnetic Particle Imaging. *International Journal on Magnetic Particle Imaging IJMPI*, 3(1), 2017.

[5] M. Storath, C. Brandt, M. Hofmann, T. Knopp, J. Salamon, A. Weber, and A. Weinmann. Edge preserving and noise reducing reconstruction for magnetic particle imaging. *IEEE Transactions on Medical Imaging*, 36(1):74–85, 2016.

[6] B. Askin, A. Güngör, D. Alptekin Soydan, E. U. Saritas, C. B. Top, and T. Cukur, PP-MPI: A Deep Plug-and-Play Prior for Magnetic Particle Imaging Reconstruction, in *International Workshop on Machine Learning for Medical Image Reconstruction*, Springer, 105–114, 2022.

[7] T. Knopp, P. Szwargulski, F. Griese, M. Grosser, M. Boberg, and M. Möddel. MPIReco.jl: Julia package for image reconstruction in MPI. *International Journal on Magnetic Particle Imaging IJMPI*, 5(1-2), 2019.

[8] T. Knopp and M. Grosser. MRIReco.jl: An MRI reconstruction framework written in Julia. *Magnetic Resonance in Medicine*, 86(3):1633–1646, 2021.

[9] N. Parikh, S. Boyd, *et al.* Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

[10] T. Kluth and B. Jin. Enhanced reconstruction in magnetic particle imaging by whitening and randomized SVD approximation. *Physics in Medicine & Biology*, 64(12):125026, 2019.

[11] S. Dittmer, T. Kluth, M. T. R. Henriksen, and P. Maass. Deep image prior for 3D magnetic particle imaging: A quantitative comparison of regularization techniques on Open MPI dataset. *International Journal on Magnetic Particle Imaging IJMPI*, 7(1), 2021, doi:10.18416/IJMPI.2021.2103001.

10.18416/ijmpi.2024.2403028