





# MRI-NUFFT: Doing non-Cartesian MRI has never been easier

Pierre-Antoine Comby<sup>1,2</sup> , Guillaume Daval-Fr  rot<sup>3</sup> , Caini Pan<sup>1,2</sup>, Asma Tanabene<sup>1,2,5</sup>, L  na Oudjman<sup>1,2</sup>, Matteo Cencini<sup>4</sup>, Philippe Ciuciu<sup>1,2</sup> , and Chaithya GR<sup>1,2</sup> 

1 MIND, Inria, France 2 Universit   Paris-Saclay / CEA, France 3 Chipiron, France 4 INFN, Pisa Division, Italy 5 Siemens Healthineers, Princeton, United States of America  Corresponding author

DOI: [10.21105/joss.07743](https://doi.org/10.21105/joss.07743)

## Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Kevin M. Moerman](#)  

## Reviewers:

- [@mathieboudreau](#)
- [@bwheelz36](#)
- [@cncastillo](#)

Submitted: 01 November 2024

Published: 15 April 2025

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

## Summary

MRI-NUFFT is a python package that provides a universal interface to various Non-Uniform Fast Fourier Transform libraries running on CPU or GPU (gpuNUFFT, finufft, cufinufft, pyNUFFT), adding compatibility with standard array library (NumPy, CuPy, PyTorch, TensorFlow, etc.) On top of these libraries it extends the existing NUFFT operations to provide a physical model of the MRI acquisition process (e.g. multi-coil acquisition and static-field inhomogeneities). It also provides a wide variety of customizable implementations of non-Cartesian sampling trajectories, as well as density compensation methods. Finally, it proposes optimized auto-differentiation with respect to the data and sampling locations for machine learning. With MRI-NUFFT one can experiment with non-Cartesian sampling in MRI, get access to the latest advances in the field and state-of-the-art sampling patterns.

## Statement of need

MRI is an non-invasive biomedical imaging technique, where raw data is sampled in the spatial frequency domain (k-space) and final images are obtained by applying an inverse (fast) Fourier transform on this data. Traditionally, the data is sampled on a Cartesian grid (often partially by skipping lines to accelerate the acquisition) and reconstructed using FFT-based algorithms. However, the Cartesian approach is not always the best choice for data collection, and non-Cartesian sampling schemes have been proposed to improve image quality, reduce acquisition time or enable new imaging modalities. But the reconstruction of non-Cartesian data is more challenging and requires the use of non-uniform fast Fourier transform (NUFFT) algorithms. Several NUFFT libraries have been developed in the past few years, but they are not always easy to use or don't account for the specificities of MRI data acquisition (e.g. multi-coil acquisition, static-field inhomogeneities, density compensation, etc.). Also their performances can vary a lot depending on the use cases (2D vs 3D data, number of coils, etc.).

Moreover, non-Cartesian acquisitions are still an active research field, with new sampling patterns being proposed regularly. With MRI-NUFFT one can easily experiment with these new patterns and compare them to existing ones. Furthermore, there has been a growing interest in using deep learning to jointly learn MRI acquisition and reconstruction, which requires to compute the gradients of the reconstruction with respect to the raw data and/or the sampling locations.

## Features

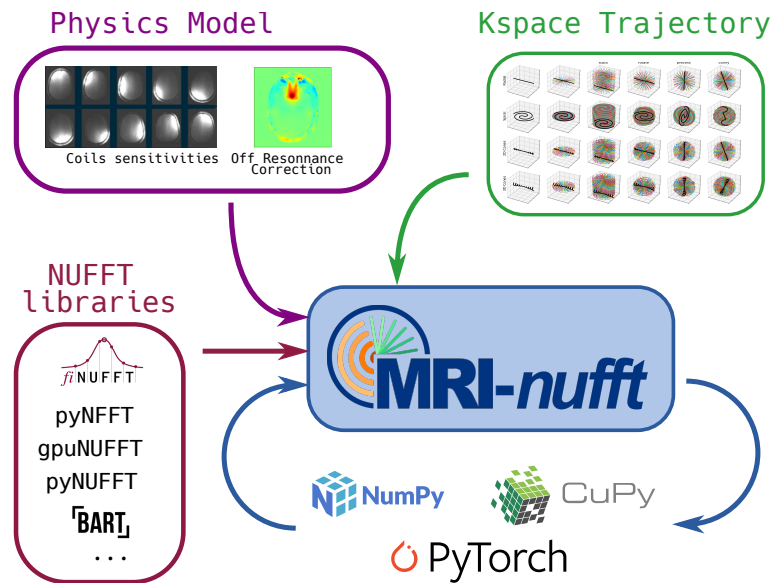


Figure 1: MRI-NUFFT as an interface for non-Cartesian MRI

## NUFFT Library compatibility

MRI-NUFFT is compatible with the following NUFFT libraries: `finufft` [Alex H. Barnett et al. (2019)] [A. H. Barnett, 2020], `cufinufft` [Shih et al., 2021], `gpuNFFT` [Knoll et al., 2014], `TorchKbNufft` [Muckley et al., 2020], `pyNFFT`, `SigPy` [Ong Frank & Lustig Michael, 2019] and `BART` [Uecker et al., 2015]. Using our [benchmark](#) we can also determine which NUFFT implementation provides the best performances both in term of computation time and memory footprint. At the time of writing, `cufinufft` and `gpuNFFT` provide the best performances by leveraging CUDA acceleration. MRI-NUFFT supports as well standard array libraries (NumPy, CuPy, PyTorch, TensorFlow, etc.) and optimizes data copies, relying on the array-API standard. It also provides several enhancements on top of these backends, notably an optimized 2.5D NUFFT (for stacks of 2D non uniform trajectories, commonly used in MRI), and a data-consistency term for iterative reconstruction ( $\mathcal{F}_\Omega^*(\mathcal{F}_\Omega x - y)$ ).

## Extended Fourier Model

MRI-NUFFT provides a physical model of the MRI acquisition process, including multi-coil acquisition and static-field inhomogeneities. This model is compatible with the NUFFT libraries, and can be used to simulate the acquisition of MRI data, or to reconstruct data from a given set of measurements. Namely we provide a linear operator that encapsulates the forward and adjoint NUFFT operators, the coil sensitivity maps and (optionally) the static field inhomogeneities. The forward model is described by the following equation:

$$y(\nu_i) = \sum_{j=1}^N x(\mathbf{u}_j) e^{-2\pi i \mathbf{u}_j \cdot \nu_i} + n_i, \quad i = 1, \dots, M$$

where:  $x(\mathbf{u})$  is the spatially varying image contrast acquired;  $y_1, \dots, y_M$  are the sampled points at frequency locations;  $\Omega = \{\nu_1, \dots, \nu_M \in [-1/2, 1/2]^d\}$ ;  $\mathbf{u}_j$  are the  $N$  spatial locations of image voxels, and  $n_i$  is a zero-mean complex-valued Gaussian noise, modeling the thermal noise of the scanner.

This can also be formulated using the operator notation  $\mathbf{y} = \mathcal{F}_\Omega(x) + \mathbf{n}$

As the sampling locations  $\Omega$  are non-uniform and the image locations  $\mathbf{u}_j$  are uniform,  $\mathcal{F}_\Omega$  is a NUDFT operator, and the equation above describe a Type 2 NUDFT. Similarly the adjoint operator is a Type 1 NUDFT:

**Table 1:** Correspondence Table between NUFFT and MRI acquisition model.

NUFFT Type	Operation	MRI Transform	Operator
Type 1	Adjoint	K-space $\rightarrow$ image	$\mathcal{F}_\Omega^*$
Type 2	Forward	Image $\rightarrow$ k-space	$\mathcal{F}_\Omega$

### Parallel Imaging Model

In MRI the acquired signal can be received by multiple antennas ("coils"). Each coil possesses a specific sensitivity profile (i.e. each sees the object differently due to its physical layout).

$$\tilde{\mathbf{y}} = \begin{bmatrix} \mathcal{F}_\Omega S_1 \\ \vdots \\ \mathcal{F}_\Omega S_L \end{bmatrix} \mathbf{x} + \mathbf{n}_\ell = \mathcal{F}_\Omega S \otimes \mathbf{x} + \tilde{\mathbf{n}}$$

where  $S_1, \dots, S_L$  are the sensitivity maps of each coil. Such maps can be acquired separately by sampling the k-space low frequencies, or estimated from the data.

### Off-resonance correction model

The constant magnetic field  $B_0$  applied in an MRI machine (typically 1.5, 3 or 7 Teslas) is inherently disturbed by metal implants or even simply by difference in magnetic susceptibilities of tissues (such at air-tissue interfaces close to the nose and ear canals). Those field perturbations introduce a spatially varying phase shift in the acquired frequencies (noted  $\Delta\omega_0$ ), causing the physical model to deviate from the ideal Fourier model. Fortunately, this inhomogeneity map can be acquired separately or estimated then integrated as:

$$y(t_i) = \int_{\mathbb{R}^d} x(\mathbf{u}) e^{-2i\pi\mathbf{u} \cdot \boldsymbol{\nu}_i + \Delta\omega(\mathbf{u})t_i} d\mathbf{u}$$

where  $t_i$  is the time at which the frequency  $\nu_i$  is acquired. With these mixed-domain field perturbations, the Fourier model does not hold anymore and the FFT algorithm can no longer be used. The main solution (Sutton et al., 2003) is to interpolate the mixed-domain exponential term by splitting it into single-domain weights  $b_{m,\ell}$  and  $c_{\ell,n}$ , where  $L \ll M, N$  regular Fourier transforms are performed to approximate the non-Fourier transform.

$$x(\mathbf{u}_n) = \sum_{\ell=1}^L c_{\ell,n} \sum_m^M y(t_m) b_{m,\ell} e^{2i\pi\mathbf{u} \cdot \boldsymbol{\nu}_i}$$

The coefficients  $B = (b_{m,\ell}) \in \mathbb{C}^{M \times L}$  and  $C = (c_{\ell,n}) \in \mathbb{C}^{L \times N}$  can be estimated within MRI-NUFFT.

### Trajectories generation and expansions

MRI-NUFFT comes with a wide variety of non-Cartesian trajectory generation routines that have been gathered from the literature. It also provides ways to extend existing trajectories and export them to specific formats, for use in other toolkits and on MRI hardware.

## Auto-differentiation for data and sampling pattern

Following the formulation of (Wang & Fessler, 2023), MRI-NUFFT provides automatic differentiation for all NUFFT backends, with respect to both gradients and data (image or k-space). This enables efficient backpropagation through NUFFT operators and supports research on learned sampling model and image reconstruction network.

## MRI-NUFFT utilization

MRI-NUFFT is already used in conjunction with other software such as SNAKE-fMRI (Comby et al., 2024), deepinv (Tachella et al., 2023) and PySAP-MRI (Farrens et al., 2020; Gueddari et al., 2020)

## References

- Barnett, A. H. (2020). *Aliasing error of the  $\exp(\beta\sqrt{1-z^2})$  kernel in the nonuniform fast fourier transform*. <https://doi.org/10.48550/arXiv.2001.09405>
- Barnett, Alex H., Magland, J. F., & Klinteberg, L. af. (2019). *A parallel non-uniform fast fourier transform library based on an "exponential of semicircle" kernel*. <https://doi.org/10.48550/arXiv.1808.06736>
- Comby, P.-A., Vignaud, A., & Ciuciu, P. (2024). *SNAKE-fMRI: A modular fMRI data simulator from the space-time domain to k-space and back*. <https://doi.org/10.48550/arXiv.2404.08282>
- Farrens, S., Grigis, A., El Gueddari, L., Ramzi, Z., G.r., C., Starck, S., Sarthou, B., Cherkaoui, H., Ciuciu, P., & Starck, J.-L. (2020). PySAP: Python Sparse Data Analysis Package for multidisciplinary image processing. *Astronomy and Computing*, 32, 100402. <https://doi.org/10.1016/j.ascom.2020.100402>
- Gueddari, L., Gr, C., Ramzi, Z., Farrens, S., Starck, S., Grigis, A., Starck, J.-L., & Ciuciu, P. (2020). PySAP-MRI: A python package for MR image reconstruction. *ISMRM Workshop on Data Sampling and Image Reconstruction*. <https://inria.hal.science/hal-02399267>
- Knoll, F., Schwarzl, A., Diwok, C., & Sodickson, D. K. (2014). *gpuNUFFT - an open source GPU library for 3D regridding with direct matlab interface*. <https://api.semanticscholar.org/CorpusID:53652346>
- Muckley, M. J., Stern, R., Murrell, T., & Knoll, F. (2020). TorchKbNufft: A high-level, hardware-agnostic non-uniform fast Fourier transform. *ISMRM Workshop on Data Sampling & Image Reconstruction*.
- Ong Frank, & Lustig Michael. (2019). SigPy: A Python Package for High Performance Iterative Reconstruction. *ISMRM 2019*. ISMRM.
- Shih, Y., Wright, G., Andén, J., Blaschke, J., & Barnett, A. H. (2021). *cuFINUFFT: A load-balanced GPU library for general-purpose nonuniform FFTs*. <https://doi.org/10.48550/arXiv.2102.08463>
- Sutton, B. P., Noll, D. C., & Fessler, J. A. (2003). Fast, iterative image reconstruction for MRI in the presence of field inhomogeneities. *IEEE Transactions on Medical Imaging*, 22(2), 178–188. <https://doi.org/10.1109/TMI.2002.808360>
- Tachella, J., Chen, D., Hurault, S., Terris, M., & Wang, A. (2023). *DeepInverse: A deep learning framework for inverse problems in imaging* (latest). <https://doi.org/10.5281/zenodo.7982256>
- Uecker, M., Ong, F., & Tamir, J. (2015). Berkley advanced reconstruction toolbox. *Proc. Intl.*

*Soc. Mag. Reson. Med.* 23. <https://doi.org/10.5281/zenodo.592960>

Wang, G., & Fessler, J. A. (2023). Efficient approximation of Jacobian matrices involving a non-uniform fast Fourier transform (NUFFT). *IEEE Transactions on Computational Imaging*, 9, 43–54. <https://doi.org/10.1109/TCI.2023.3240081>