The Journal of Open Source Software

# PyQMRI: An accelerated Python based Quantitative MRI toolbox

**Oliver Maier**[*1], **Stefan M Spann**[1], **Markus Bödenler**[1, 2], **and Rudolf Stollberger**[1, 3]

**1** Institute of Medical Engineering, Graz University of Technology, Graz, Austria **2** Institute of eHealth, University of Applied Sciences FH JOANNEUM, Graz, Austria **3** BioTechMed Graz, Austria

**DOI:** 10.21105/joss.02727

**Software**

- Review
- Repository
- Archive

**Editor:** Kevin M. Moerman

**Reviewers:**

- @grlee77
- @agahkarakuzu
- @DARSakthi

**Submitted:** 02 October 2020
**Published:** 12 December 2020

## Summary

Various medical examinations are seeing a shift to a more patient centric and personalized view, based on quantitative instead of qualitative observations and comparisons. This trend has also affected medical imaging, and particularly quantitative MRI (qMRI) gained importance in recent years. qMRI aims to identify the underlying biophysical and tissue parameters that determine contrast in an MR imaging experiment. In addition to contrast information, qMRI permits insights into diseases by providing biophysical, microstructural, and functional information in absolute quantitative values. For quantification, biophysical models are used, which describe the relationship between image intensity and physical properties of the tissue for certain scanning sequences and sequence parameters. By performing several measurements with different sequence parameters (e.g. flip angle, repetition time, echo time) the related inverse problem of identifying the tissue parameters sought can be solved.

Quantitative MR typically suffers from increased measurement time due to repeated imaging experiments. Therefore, methods to reduce scanning time by means of optimal scanning protocols and subsampled data acquisition have been extensively studied. However, these approaches are typically associated with a reduced SNR, and can suffer from subsampling artifacts. To address both aspects, it has been shown that the inclusion of a biophysical model in the reconstruction process leads to much faster data acquisition, while simultaneously improving image quality. The inverse problem associated with this special reconstruction approach requires dedicated numerical solution strategies (Block et al., 2009; Doneva et al., 2010; Donoho, 2006; Lustig et al., 2007; Maier, Schoormans, et al., 2019; Roeloffs et al., 2016; Sumpf, 2012), commonly known as model-based reconstruction. Model-based reconstruction is based on variational modeling, and combines parallel imaging and compressed sensing to achieve acceleration factors as high as 10x the speed of fully sampled acquisitions. The method directly solves for the unknown parameter maps from raw k-space data. The repeated transition from k-space to image-space, combined with the involved non-linear iterative reconstruction techniques to identify the unknown parameters, often leads to prolonged reconstruction times. This effect gets even more demanding if 3D image volumes are of interest.

Recently, the upsurge of computationally powerful GPUs has led to a variety of GPU based implementations to speed up computation time of highly parallelizeable operations (e.g., the Fourier transformation in MRI (Knoll et al., 2014)). As model-based approaches possibly deal with hundred Gigabytes of data (e.g. diffusion tensor imaging), available memory on current GPUs (e.g. 12 GB) can be a limiting factor. Thus, most reconstruction and fitting algorithms are applied in a slice-by-slice fashion to the volumetric data by taking a Fourier transformation along a fully sampled acquisition direction, effectively yielding a set of 2D problems. Hence,

[*] Corresponding author.

the additional information in form of the third dimension of volumetric data is neglected, leading to a loss in performance.

To utilize full 3D information in advanced reconstruction and fitting algorithms on memory limited GPUs, special solutions strategies are necessary to leverage the speed advantage, e.g., hiding memory latency of repeated transfers to/from the GPU to host memory. This can be achieved using asynchronous execution strategies. However, correct synchronization of critical operations can be error prone. To this end, we propose `PyQMRI`, a simple to use Python toolbox for quantitative MRI.

# Statement of need

`PyQMRI` aims at reducing the required reconstruction time by means of a highly parallelized `PyOpenCL` (Klöckner et al., 2012) implementation of a state-of-the-art model-based reconstruction and fitting algorithm, while maintaining the easy-to-use properties of a Python package. In addition to processing small data (e.g. 2D slices) completely on the GPU, an efficient double-buffering based solution strategy is implemented. Double-buffering allows overlapping computation and memory transfer from/to the GPU, thus hiding the associated memory latency. By overlapping the transferred blocks it is possible to pass on 3D information utilizing finite differences based regularization strategies (Maier, Schloegl, et al., 2019). Figure 1 shows a schematic of the employed double-buffering scheme. To make sure that this asynchronous execution strategy yields the expected results, unit-testing is employed.

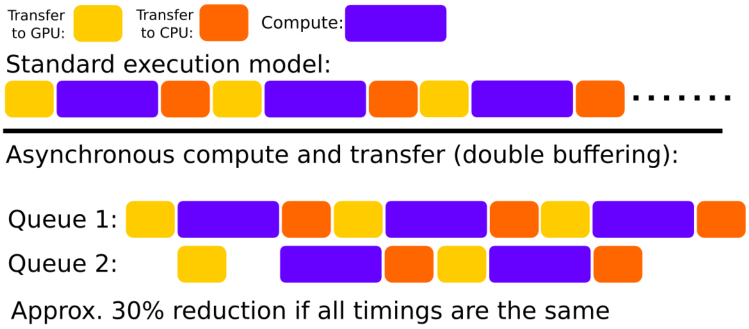
Transfer to GPU:
Transfer to CPU:
Compute:
Standard execution model:
Asynchronous compute and transfer (double buffering):
Queue 1:
Queue 2:
Approx. 30% reduction if all timings are the same

**Figure 1:** Simple double-buffering scheme using two separate command queues and overlaping transfer/compute operations.

Currently, 3D acquisitions with at least one fully sampled dimension can be reconstructed on the GPU, including stack-of-X acquisitions or 3D Cartesian based imaging. Of course 2D data can be reconstructed as well. The combination of reconstruction and non-linear fitting is based on an iteratively regularized Gauss-Newton (IRGN) approach combined with a primal-dual inner loop. Regularization strategies include total variation (TV) (Rudin et al., 1992) and total generalized variation (TGV) (Bredies et al., 2010; Knoll et al., 2011) using finite differences gradient operations. In addition to the combined reconstruction and fitting algorithm from k-space data, `PyQMRI` can also be used to speed-up non-linear parameter fitting of complex or real valued image data. The main advantage of fitting the complex (k-space) data is that the assumed Gaussian noise characteristics for the commonly used $L_2$ data fidelity term are valid. This is especially important for problems suffering from poor SNR, e.g. Diffusion Tensor Imaging, where the wrong noise assumption can lead to significant errors in the quantification process (Jones & Basser, 2004).

`PyQMRI` comes with several pre-implemented quantiative models. In addition, new models can be introduced via a simple text file, utilizing the power of `SymPy` to generate numerical models as well as their partial derivatives in Python. Fitting can be initiated via a command line interface (CLI) or by importing the package into a Python script. Due to `PyQMRI`'s OpenCL backend, no vendor specific hardware restrictions are present. However, current limitations of the `gpyfft` package, used to wrap the `clfft`, constrain the use to GPU devices only. A switch to other `clfft` wrappers might solve this limitation in future releases, but `gpyfft` is the only one that currently supports fast non-power-of-two transformations up to 13.

`PyQMRI` and its predecessors have been succesfully used in several scientific publications. Examples include $T_1$ quantification from subsampled radial FLASH and inversion-recovery Look-Locker data (Maier, Schoormans, et al., 2019), diffusion tensor imaging (Maier, Spann, Bogensperger, et al., 2020), and ongoing work on aterial spin labeling (Maier, Spann, Pinter, Gattringer, Pirpamer, et al., 2020; Maier, Spann, Pinter, Gattringer, Hinteregger, et al., 2020), as well as low-field $T_1$ mapping at multiple fields using fast field-cycling MRI.

# Related Work

The increased importance of qMRI is reflected by a multitude of open-source toolboxes, each focusing on a subset or combination of qMRI applications. Most tools show a strong focus on neurological applications (mostly brain) but are usually not limited to this application area. hMRI (Tabelow et al., 2019) is Matlab based and builds upon SPM (Friston, 2007). It extends the spatial registration and statistical inference capabilities of SPM by relaxometry and quantification of the magnetisation transfer effect. mrQ (Mezer et al., 2013) offers relaxometry combined with the ability to quantify the macromolecular tissue volume, apparent volume of interacting water protons, and the water-surface interaction rate and is completely written in Matlab. Another Matlab based project is qMRlab (Karakuzu et al., 2020) which offers a multitude of quantification algorithms including relaxometry, diffusion imaging, quantitative susceptibility mapping, field mapping, and quantitative magnetization transfer. It further offers routines for visualization, simulation, and protocol optimization of quantitative MRI examinations. Another Matlab based software is qmap (Hurley et al., 2011) which offers a collection of tools for quantitative MRI.

qMRI is also present in the Python community with PyMRT (Metere & Möller, 2017) offering tools for image analysis and relaxometry. Another powerful Pyhton package, with a focus on neuroimaing, is DIPY (Garyfallidis et al., 2014), offering a multitude of ways to evaluated diffusion and perfusion MRI data. Other software packages focus on fast execution and fitting, like the QUIT (Wood, 2018) toolbox, which is entirely written in C++ to speed up the computations. All of the above mentioned qMRI toolboxes have in common that they usually require image data for the fitting process and, thus, are not suitable for accelerated acquired data or require dedicated reconstruction algorithms prior to fitting.

A recent extensions to BART (Uecker et al., 2015) allows for $T_1$ quantification from 2D radially acquired inversion recovery Look-Locker data. The approach utilizes a model-based reconstruction algorithm to estimate $T_1$ directly from k-space (Wang et al., 2018, 2019). Even though this approach can handle undersampled data and incorporates the whole MRI acquisition pipeline, it is currently limited to this single quantification model.

To the best of the authors knowledge `PyQMRI` is the only available Python toolbox that offers real 3D regularization in an iterative solver for model-based qMRI problems and for arbitrary large volumetric data, while simultaneously utilizing the computation power of recent GPUs. Further, the ability to use symbolic equations to generate new models seems to be unique as other tools require modifications of the code to include new quantification models.

## Algorithms

PyQMRI deals with the following general problem structure:

$$\min_{u,v} \quad \frac{1}{2}\|A(u)-d\|_2^2 + \gamma(\alpha_0\|\nabla u - v\|_{1,2,F} + \alpha_1\|\mathcal{E}v\|_{1,2,F})$$

which includes a non-linear forward operator ($A$), mapping the parameters $u$ to (complex) data space $d$, and a non-smooth regularization functional due to the $L^1$-norms of the T(G)V functional (Bredies et al., 2010; Knoll et al., 2011). Setting $\alpha_1 = 0$ and $v = 0$ the problem becomes simple TV regularization (Rudin et al., 1992). The gradient $\nabla$ and symmetrized gradient $\mathcal{E}$ operators are implemented using finite differences. To further improve the quality of the reconstructed parameter maps PyQMRI uses a Frobenius norm to join spatial information from all maps in the T(G)V functionals (Bredies, 2014; Knoll et al., 2017; Maier, Schoormans, et al., 2019). Box constraints, limiting each unknown parameter in $u$ to a physiological meaningful range, can be set in conjunction with real or complex value constraints.

Following the Gauss-Newton approach a sequence $k$ of linearized sub-problems of the form

$$\min_{u,v} \quad \frac{1}{2}\|\mathrm{D}A|_{u=u^k}u - \tilde{d}^k\|_2^2 + \gamma_k(\alpha_0\|\nabla u - v\|_{1,2,F} + \alpha_1\|\|\mathcal{E}v\|_{1,2,F}) + \frac{\delta_k}{2}\|u-u^k\|_{M_k}^2$$

needs to be solved to find a solution of the overall problem. The matrix $\mathrm{D}A|_{u=u^k} = \frac{\partial A}{\partial u}(u^k)$ resembles the Jacobian of the system. The subproblems can be recast into a saddle-point structure by application of the Fenchel duality

$$\min_u \max_y \ \langle \mathrm{K}u, y\rangle + G(u) - F^*(y),$$

and solved utilizing a well established primal-dual algorithm (Chambolle & Pock, 2011) combined with a line-search (Malitsky & Pock, 2018) to speed-up convergence. Constant terms, stemming from the linearization, are precomputed and fused with the data $d$, yielding $\tilde{d}^k$. The inclusion of the additional $L^2$-norm penalty improves convexity of the subproblem and resembles a Levenberg-Marquardt update for $M_k = diag(\mathrm{D}A|_{u=u^k}^T \mathrm{D}A|_{u=u^k})$. A graphical representation of the involved steps is given in Figure 2. The regularization weights, regularization type (TV/TGV), and the number of outer and inner iterations can be changed using a plain text configuration file. It was shown by (Salzo & Villa, 2012) that the GN approach converges with linear rate to a critical point for non-convex problems with non-differential penalty functions if the initialization is sufficiently close. Thus a meaningful initial guess based on physiological knowledge on the parameters $u$ should be used to initialize the fitting, e.g. mean $T_1$ value of the tissue of interest.

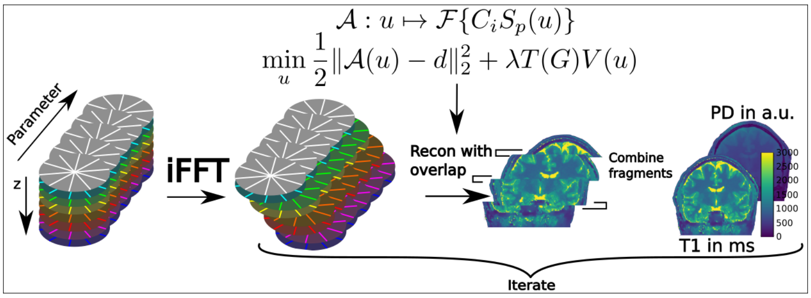
$\mathcal{A}: u \mapsto \mathcal{F}\{C_i S_p(u)\}$
$\min_u \frac{1}{2}\|\mathcal{A}(u)-d\|_2^2 + \lambda T(G)V(u)$
Parameter
z
iFFT
Recon with overlap
Combine fragments
PD in a.u.
3000
2500
2000
1500
1000
500
0
T1 in ms
Iterate

**Figure 2:** Graphical representation of the employed regularized non-linear fitting procedure shown for an exemplary $T_1$ quantification problem. $C_i$ describes complex coil sensitivity information, $\mathcal{F}$ amounts to the sampling process including the Fourier transformation, and $S_p$ equals the non-linear relationship between image intensity and the unknown physical quantities ($T_1$ and Proton Density (PD)).

## Acknowledgements

Oliver Maier acknowledges grant support from the Austrian Academy of Sciences under award DOC-Fellowship 24966.

The authors would like to acknowledge the NVIDIA Corporation Hardware grant support.

## References

Block, K. T., Uecker, M., & Frahm, J. (2009). Model-Based Iterative Reconstruction for Radial Fast Spin-Echo MRI. *IEEE Trans. Med. Imaging*, *28*(11), 1759–1769. https://doi.org/10.1109/TMI.2009.2023119

Bredies, K. (2014). *Recovering Piecewise Smooth Multichannel Images by Minimization of Convex Functionals with Total Generalized Variation Penalty* (pp. 44–77). https://doi.org/10.1007/978-3-642-54774-4_3

Bredies, K., Kunisch, K., & Pock, T. (2010). Total Generalized Variation. *SIAM J. Imaging Sci.*, *3*(3), 492–526. https://doi.org/10.1137/090769521

Chambolle, A., & Pock, T. (2011). A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging. *J. Math. Imaging Vis.*, *40*(1), 120–145. https://doi.org/10.1007/s10851-010-0251-1

Doneva, M., Börnert, P., Eggers, H., Stehning, C., Sénégas, J., & Mertins, A. (2010). Compressed sensing reconstruction for magnetic resonance parameter mapping. *Magn. Reson. Med.*, *64*(4), 1114–1120. https://doi.org/10.1002/mrm.22483

Donoho, D. L. (2006). Compressed sensing. *IEEE Trans. Inf. Theory*, *52*(4), 1289–1306. https://doi.org/10.1109/TIT.2006.871582

Friston, K. (2007). CHAPTER 2 - Statistical parametric mapping. In K. FRISTON, J. ASHBURNER, S. KIEBEL, T. NICHOLS, & W. PENNY (Eds.), *Stat. Parametr. mapp.* (pp. 10–31). Academic Press. https://doi.org/10.1016/B978-012372560-8/50002-4

Garyfallidis, E., Brett, M., Amirbekian, B., Rokem, A., Walt, S. van der, Descoteaux, M., & Nimmo-Smith, I. (2014). Dipy, a library for the analysis of diffusion MRI data. *Front. Neuroinform.*, *8*. https://doi.org/10.3389/fninf.2014.00008

Hurley, S. A., Mossahebi Puroa, & Samsonov, A. A. (2011). Quantitative MRI analysis package. In *Code repository*. University of Wisconsin - Madison. https://www.medphysics.wisc.edu/%C2%A0samsonov/qmap/doc/index.html

Jones, D. K., & Basser, P. J. (2004). "Squashing peanuts and smashing pumpkins": How noise distorts diffusion-weighted MR data. *Magn. Reson. Med.*, *52*(5), 979–993. https://doi.org/10.1002/mrm.20283

Karakuzu, A., Boudreau, M., Duval, T., Boshkovski, T., Leppert, I. R., Cabana, J.-F., Gagnon, I., Beliveau, P., Pike, G. B., Cohen-Adad, J., & Stikov, N. (2020). qMRLab: Quantitative MRI analysis, under one umbrella. *Journal of Open Source Software*, *5*(53), 2343. https://doi.org/10.21105/joss.02343

Klöckner, A., Pinto, N., Lee, Y., Catanzaro, B., Ivanov, P., & Fasih, A. (2012). PyCUDA and PyOpenCL: A scripting-based approach to GPU run-time code generation. *Parallel Comput.* https://doi.org/10.1016/j.parco.2011.09.001

Knoll, F., Bredies, K., Pock, T., & Stollberger, R. (2011). Second order total generalized variation (TGV) for MRI. *Magn. Reson. Med.*, *65*(2), 480–491. https://doi.org/10.1002/mrm.22595

Knoll, F., Holler, M., Koesters, T., Otazo, R., Bredies, K., & Sodickson, D. K. (2017). Joint MR-PET Reconstruction Using a Multi-Channel Image Regularizer. *IEEE Trans. Med. Imaging*. https://doi.org/10.1109/TMI.2016.2564989

Knoll, F., Schwarzl, A., Diwoky, C., & Sodickson, D. K. (2014). gpuNUFFT - an open-source GPU library for 3D gridding with direct matlab interface. *In ISMRM 23rd Annual Meeting*, 4297.

Lustig, M., Donoho, D., & Pauly, J. M. (2007). Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magn. Reson. Med.* https://doi.org/10.1002/mrm.21391

Maier, O., Schloegl, M., Bredies, K., & Stollberger, R. (2019). 3D Model-Based Parameter Quantification on Resource Constrained Hardware using Double-Buffering. *Proc. 27th Annu. Meet. ISMRM, Montréal, QC, Canada*, 4839.

Maier, O., Schoormans, J., Schloegl, M., Strijkers, G. J., Lesch, A., Benkert, T., Block, T., Coolen, B. F., Bredies, K., & Stollberger, R. (2019). Rapid T1 quantification from high resolution 3D data with model-based reconstruction. *Magn. Reson. Med.* https://doi.org/10.1002/mrm.27502

Maier, O., Spann, S. M., Bogensperger, L., & Stollberger, R. (2020). Fast Simultaneous Multi-Slice Multi-Shell Diffusion Tensor Imaging with Model-based Reconstruction. *Proc. 28th Annu. Meet. ISMRM, First Virtual Meeting*, 4365.

Maier, O., Spann, S. M., Pinter, D., Gattringer, T., Hinteregger, N., Enzinger, C., Pfeuffer, J., Bredies, K., & Stollberger, R. (2020). *Non-linear fitting with joint spatial regularization in arterial spin labeling*. http://arxiv.org/abs/2009.05409

Maier, O., Spann, S. M., Pinter, D., Gattringer, T., Pirpamer, L., Enzinger, C., Pfeuffer, J., & Stollberger, R. (2020). Robust Perfusion Parameter Quantification from 3D Single-Shot Multi-Delay ASL measurements. *Proc. 28th Annu. Meet. ISMRM, First Virtual Meeting*, 3290.

Malitsky, Y., & Pock, T. (2018). A First-Order Primal-Dual Algorithm with Linesearch. *SIAM J. Optim.*, *28*(1), 411–432. https://doi.org/10.1137/16M1092015

Metere, R., & Möller, H. E. (2017). PyMRT and DCMPI: Two New Python Packages for MRI Data Analysis. *Proceedings of the International Society of Magnetic Resonance in Medicine*, 3816.

Mezer, A. D., Yeatman, J. N., Stikov, N. F., Kay, K. L., Cho, N.-J. H., Dougherty, R. A., Perry, M. undefined, Parvizi, J. undefined, Hua, L. undefined, Butts-Pauly, K. undefined, & al., et. (2013). Quantifying the local tissue volume and composition in individual brains with magnetic resonance imaging. *Nature Medicine*, *19*(12), 1667–1672. https://doi.org/10.1038/nm.3390

Roeloffs, V., Wang, X., Sumpf, T. J., Untenberger, M., Voit, D., & Frahm, J. (2016). Model-based reconstruction for T1 mapping using single-shot inversion-recovery radial FLASH. *Int. J. Imaging Syst. Technol.*, *26*(4), 254–263. https://doi.org/10.1002/ima.22196

Rudin, L. I., Osher, S., & Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Phys. D Nonlinear Phenom.*, *60*(1-4), 259–268. https://doi.org/10.1016/0167-2789(92)90242-F

Salzo, S., & Villa, S. (2012). Convergence analysis of a proximal Gauss-Newton method. *Comput. Optim. Appl.*, *53*(2), 557–589. https://doi.org/10.1007/s10589-012-9476-9

Sumpf, T. (2012). *Model-based T2 relaxometry using undersampled magnetic resonance imaging* (Vol. 43) [PhD thesis, Mensch und Buch]. https://doi.org/10.24355/dbbs.084-201305311128-0

Tabelow, K., Balteau, E., Ashburner, J., Callaghan, M. F., Draganski, B., Helms, G., Kherif, F., Leutritz, T., Lutti, A., Phillips, C., Reimer, E., Ruthotto, L., Seif, M., Weiskopf, N., Ziegler, G., & Mohammadi, S. (2019). hMRI – A toolbox for quantitative MRI in neuroscience and clinical research. *Neuroimage*, *194*, 191–210. https://doi.org/10.1016/j.neuroimage.2019.01.029

Uecker, M., Ong, F., Tamir, J. I., Bahri, D., Virtue, P., Cheng, J. Y., Zhang, T., & Lustig, M. (2015). Berkeley Advanced Reconstruction Toolbox. *Proceedings of the International Society of Magnetic Resonance in Medicine*, 2486.

Wang, X., Kohler, F., Unterberg-Buchwald, C., Lotz, J., Frahm, J., & Uecker, M. (2019). Model-based myocardial T1 mapping with sparsity constraints using single-shot inversion-recovery radial FLASH cardiovascular magnetic resonance. *Journal of Cardiovascular Magnetic Resonance*, *21*(1). https://doi.org/10.1186/s12968-019-0570-3

Wang, X., Roeloffs, V., Klosowski, J., Tan, Z., Voit, D., Uecker, M., & Frahm, J. (2018). Model-based T1 mapping with sparsity constraints using single-shot inversion-recovery radial FLASH. *Magnetic Resonance in Medicine*, *79*(2), 730–740. https://doi.org/10.1002/mrm.26726

Wood, T. C. (2018). QUIT: QUantitative imaging tools. *Journal of Open Source Software*, *3*(26), 656. https://doi.org/10.21105/joss.00656