RESEARCH ARTICLE

High-resolution in vivo MR-STAT using a matrix-free and parallelized reconstruction algorithm

Oscar van der Heide ⓘ | Alessandro Sbrizzi | Peter R. Luijten | Cornelis A.T. van den Berg

Center for Image Sciences, University Medical Center Utrecht, Utrecht, the Netherlands

Correspondence
Oscar van der Heide, University Medical Center, Heidelberglaan 100, 3508 GA Utrecht, The Netherlands.
Email: o.vanderheide@umcutrecht.nl

Funding information
Dutch Technology Foundation, Grant/Award Number: 14125

Abstract

MR-STAT is a recently proposed framework that allows the reconstruction of multiple quantitative parameter maps from a single short scan by performing spatial localisation and parameter estimation on the time-domain data simultaneously, without relying on the fast Fourier transform (FFT). To do this at high resolution, specialized algorithms are required to solve the underlying large-scale nonlinear optimisation problem. We propose a matrix-free and parallelized inexact Gauss–Newton based reconstruction algorithm for this purpose. The proposed algorithm is implemented on a high-performance computing cluster and is demonstrated to be able to generate high-resolution ($1\,\mathrm{mm} \times 1\,\mathrm{mm}$ in-plane resolution) quantitative parameter maps in simulation, phantom, and in vivo brain experiments. Reconstructed T_1 and T_2 values for the gel phantoms are in agreement with results from gold standard measurements and, for the in vivo experiments, the quantitative values show good agreement with literature values. In all experiments, short pulse sequences with robust Cartesian sampling are used, for which MR fingerprinting reconstructions are shown to fail.

KEYWORDS

large-scale nonlinear optimization, MR fingerprinting, MR-STAT, parallel computing, quantitative MRI

1 | INTRODUCTION

Conventional magnetic resonance imaging (MRI) methods rely on the Fourier transform relationship between the signal and the local magnetization value for spatial encoding. Tissue differentiation is possible in the resulting qualitative images because different tissue types have distinct MR-related biophysical properties like T_1 and T_2 relaxation times. Quantitative magnetic resonance imaging (qMRI) methods aim to estimate MR-related biophysical properties like T_1 and T_2 relaxation times. Quantitative images could provide additional diagnostic value and are more suited for the purpose of multicenter studies and computer-aided diagnosis.[1,2] The most straightforward and robust choices for T_1 and T_2 mapping sequences, that is, single-echo (inversion recovery) spin-echo sequences, have prohibitively long scan times. Over time, a multitude of alternative pulse sequences have been developed that reduce acquisition times to clinically acceptable levels.[3-6] In recent years, acquisition times have been reduced even further with advanced reconstruction techniques that include more information about the underlying physical processes in the reconstructions,[7] add a priori knowledge in the form of sparsity or low-rank constraints,[8] and/or allow estimation of multiple parameter maps simultaneously.[9,10] A prime example is magnetic resonance fingerprinting (MRF).[11] In MRF, a transient-state pulse sequence with quasi-random

---

Abbreviations: bSSFP, balanced steady-state free precession; CSF, cerebrospinal fluid; FFT, fast Fourier transform; HFEN, high-frequency error norms; L-BFGS, limited-memory Broyden–Fletcher–Goldfard–Shanno; MAPE, mean absolute percentage error; MRF, magnetic resonance fingerprinting; MR-STAT, magnetic resonance spin tomography in the time domain; NRMSE, normalized root-mean-square errors; nuFFT, non-uniform fast Fourier transform; qMRI, quantitative magnetic resonance imaging; SVD, singular value decomposition; TE, echo time; TR, repetition time; VARPRO, variable projection method.

---

components is used and many highly undersampled *k*-spaces are acquired in a single, short acquisition. A fast Fourier transform (FFT) is applied to each *k*-space to generate many snapshot images. Then, on a voxel-per-voxel basis, the measured *fingerprints* are matched to a precomputed Bloch-equation based dictionary to obtain the quantitative parameters. Through this novel combination of transient-state sequences with a pattern recognition step, MRF has been able to reduce qMRI acquisition times drastically.

Magnetic Resonance Spin TomogrAphy in Time-domain (MR-STAT)[12] is a recently proposed qMRI framework that, similarly to MRF, aims to estimate multiple parameter maps from a single short scan simultaneously. However, instead of performing FFTs in a separate step for spatial localisation of signal, parameter maps are fitted directly to the measured time-domain signal using a Bloch-equation based volumetric signal model. That is, a single large-scale nonlinear optimisation problem is solved numerically, where the spatial localisation and parameter estimation are performed *simultaneously*. In addition, instead of using a dictionary matching procedure, in MR-STAT gradient-based iterative methods are used to solve the optimisation problem. Compared with MRF, the MR-STAT approach results in different trade-offs made in the reconstruction. Since the FFT is no longer used explicitly to transform back and forth between image space and frequency space, the spatial gradient encoding is entangled directly into the MR-STAT signal model. With this approach, data from different readouts of a transient-state pulse sequence can be combined naturally into a single reconstruction process. There is no reliance on dictionary compression[13] or compressed sensing[14] techniques to suppress aliasing artefacts. As will be demonstrated, MR-STAT allows for the reconstruction of high-quality parameter maps from very short scans even when using standard and experimentally reliable Cartesian sampling strategies.

Solving the nonlinear optimisation problem that results from using the volumetric signal model in MR-STAT does introduce new computational challenges. As will be discussed, the computational and memory requirements scale quadratically with the resolution and parallelizing the computations is nontrivial, because the FFT is not used to decouple the unknowns spatially. In Sbrizzi et al.,[12] to alleviate the computational challenges at high resolution, a 1D FFT along the readout direction was still employed to decouple the problem in one direction in space, resulting in many smaller and independent 1D subproblems to be solved. This hybrid approach benefits only partly from the abovementioned advantages of using a volumetric signal model, for example, dynamical behaviour during readouts cannot be taken into account. Furthermore, it can only be used with Cartesian sampling strategies. Thirdly, if the technique is applied to 3D acquisitions, each of the resulting 2D subproblems will itself be a large-scale problem. Therefore, to unlock the full potential of MR-STAT, a specialized reconstruction algorithm is required that:

1. does not require storage of large model matrices (i.e., is *matrix-free*);
2. is suitable for a parallel computing implementation to reduce computation times; and
3. is extensible to non-Cartesian sampling strategies.

In the current work, we present a reconstruction algorithm for MR-STAT based on an inexact Gauss–Newton method (see Steihaug[15] and Algorithm 7.2 in Nocedal and Wright[16]) that satisfies the above requirements. For partial derivative computations that are needed in the reconstruction, we propose to use exact algorithmic differentiation. With the new reconstruction algorithm, we demonstrate the potential of the MR-STAT framework through simulation studies, phantom experiments, and by reconstructing high-resolution in vivo brain maps. Although in principle the reconstruction algorithm can be used with non-Cartesian sampling, in all experiments we will use Cartesian sampling patterns. The reason is that Cartesian sequences—which are used in the vast majority of clinical exams—are challenging to work with in the context of conventional MRF,[17] whereas, with MR-STAT, parameter maps can be reconstructed successfully even when using very short acquisitions of the order of seconds per slice.

2 | THEORY

In this section we first review the MR-STAT framework as presented by Sbrizzi et al.[12] We then discuss the computational challenges resulting from the large-scale reconstruction problem and propose techniques to deal with these challenges.

2.1 | MR-STAT framework

The time evolution of a single-spin isochromat $\mathbf{m} = (m_x, m_y, m_z)$ with spatial coordinates $\mathbf{r} = (x, y, z)$ and tissue properties $\theta = (T_1, T_2, \dots)$ is governed by the Bloch equations.

Let $m = m_x + im_y$ be the transverse component of the magnetization in the rotating frame. The demodulated time-domain signal s is equal to the volume integral of the transverse magnetisation of all spins within the field of view V, weighted by their effective proton spin densities ρ. For the purpose of this work, ρ also includes the amplitude of the receive coil sensitivity and the transceive phase, thus ρ is a complex quantity, that is, $\rho = \rho_x + i\rho_y$. In short:

$$s(t) = \int_V \rho(\mathbf{r}) m(\mathbf{r}, \theta(\mathbf{r}), t) \, d\mathbf{r}. \tag{1}$$

After discretization of the field of view V into N_v voxels, each having volume Δ_V, Equation (1) becomes

$$s(t) = \sum_{j=1}^{N_v} \rho_j m_j(\theta_j, t) \Delta_V. \tag{2}$$

Here, m_j is the magnetization in voxel j, which can be computed by numerical integration of the Bloch equations.

Let N_t be the total number of signal samples and let $t_1, \ldots, t_{N_t}$ denote the sampling times. Define the magnetization vector $\mathbf{m}_j$ in voxel j as

$$\mathbf{m}_j := \left(m_j(\theta_j, t_1), \ldots, m_j(\theta_j, t_{N_t}) \right) \in \mathbb{C}_t^N \tag{3}$$

and the signal vector $\mathbf{s} \in \mathbb{C}^{N_t}$ as

$$\mathbf{s} = \sum_{j=1}^{N_v} \rho_j \mathbf{m}_j. \tag{4}$$

Note that if we introduce the *magnetization matrix* $\mathbf{M} \in \mathbb{C}^{N_t \times N_v}$,

$$M_{i,j} := m_j(\theta_j, t_i), \tag{5}$$

and proton density vector $\rho \in \mathbb{C}^{N_v}$,

$$\rho = \left(\rho_1, \ldots, \rho_{N_v} \right), \tag{6}$$

then $\mathbf{s}$ can be written as

$$\mathbf{s} = \mathbf{M}\rho. \tag{7}$$

Let N_p denote the number of distinct parameters per voxel (including real and imaginary parts of the proton density). Then s depends on $N := N_v \times N_p$ different parameters. All parameters are concatenated into a single vector $\alpha \in \mathbb{R}^N$ in such a way that indices $\{ j + kN_v | k = 0 \ldots, N_p - 1 \}$ denote the parameters associated with voxel j.

Now, given a vector of measured time-domain samples $\mathbf{d} \in \mathbb{C}^{N_s}$, define the residual vector $\mathbf{r} \in \mathbb{C}^{N_t}$ as

$$\mathbf{r}(\alpha) = \mathbf{d} - \mathbf{s}(\alpha), \tag{8}$$

and define the nonlinear least-squares objective function $f : \mathbb{R}^N \to \mathbb{R}$ as

$$f(\alpha) = \frac{1}{2} \|\mathbf{r}(\alpha)\|_2^2. \tag{9}$$

The parameter maps α^* are obtained by solving numerically

$$\alpha^* = \operatorname{argmin}_\alpha f(\alpha), \tag{10}$$

subject to the physical constraints represented by the Bloch equations and realistically attainable intervals for the parameters.

2.2 | Computational challenges

Note that Equation (10) is a nonlinear optimization problem that requires iterative algorithms to be solved. At each iteration, the signal $\mathbf{s} = \mathbf{M}\rho$ needs to be computed and that requires the Bloch equations to be integrated for each voxel. In addition, the gradient of f (i.e., the vector of partial derivatives of f with respect to each of the parameters) needs to be computed. From the least-squares structure of the problem, it follows that the gradient can be expressed as

$$\mathbf{g} = \nabla f = \Re(\mathbf{J}^H \mathbf{r}), \tag{11}$$

where $\mathbf{J} \in \mathbb{C}^{N_t \times N_v}$ is the Jacobian matrix defined as

$$\mathbf{J}(\alpha) := \left[\frac{\partial \mathbf{r}}{\partial \alpha_1} \ \cdots \ \frac{\partial \mathbf{r}}{\partial \alpha_N} \right], \tag{12}$$

$\mathbf{J}^H$ is the Hermitian transpose of $\mathbf{J}$, and $\Re$ is the real-part operator.

A gradient-descent type algorithm could be used to minimize Equation (10), but it may result in poor convergence (see chapter 3 of Nocedal and Wright[16]). Second-order methods (i.e., Newton methods) typically lead to better convergence. At each iteration, these methods require the inversion of a linear system involving (an approximation to) the Hessian matrix $\mathbf{H} \in \mathbb{R}^{N \times N}$, which includes curvature information and is defined as

$$\mathbf{H}(\alpha) := \left[\frac{\partial^2 f}{\partial \alpha_i \partial \alpha_j} \right]_{i,j=1}^{N_p}. \tag{13}$$

A second-order MR-STAT reconstruction algorithm would follow the steps outlined in Algorithm 1.

Algorithm 1 Minimize $f(\alpha) = \frac{1}{2}\|\mathbf{d} - \mathbf{s}(\alpha)\|_2^2$

Require: Initial guess α

  while not converged **do**

    1. Compute residual: $\mathbf{r} = \mathbf{d} - \mathbf{s} = \mathbf{d} - \mathbf{M}\rho$

    2. Compute gradient: $\mathbf{g} = \Re\left(\mathbf{J}^H \mathbf{r}\right)$

    3. Solve linear system: $\mathbf{Hp} = -\mathbf{g}$

    4. Update parameters: $\alpha = \alpha + \mathbf{p}$

  end while

Using Algorithm 1 for MR-STAT poses several practical challenges, due to the large scale of the problem. First of all, to estimate N parameters, the number of sample points N_t will in general be of the order of $N = N_p \times N_v$ as well. Assuming $N_t \approx N$, it follows that $\mathbf{M}$ will be of size $N_t \times N_v \approx (N_p N_v) \times N_v$ (complex entries) and $\mathbf{J}$ will be of size $2N_t \times N \approx (N_p N_v) \times (N_p N_v)$ (complex entries). Since $\mathbf{H}$ will be of size $(N_p N_v) \times (N_p N_v)$ as well, it follows that all three matrices scale with N_v^2. In Table 1, the required computer memory to store matrices of these sizes is reported for various values of N_v for the case $N_p = 4$. It can be seen that, even for 2D acquisitions, it will be infeasible to store these matrices in memory for clinically relevant resolutions.

Secondly, the actual time needed to compute the entries of $\mathbf{M}$, $\mathbf{J}$, and $\mathbf{H}$ scales with N_v^2 as well. When using a regular desktop computer, the reconstruction times quickly become too long to make MR-STAT useful in clinical practice.

Fortunately, as will be detailed in the next section, 1 only requires matrix–vector products with the matrices $\mathbf{M}$, $\mathbf{J}$, and (approximations to) $\mathbf{H}$. These matrix–vector products can be computed without having to store the full matrices in memory. Moreover, the computation of the matrix–vector products can be distributed efficiently among multiple computing cores on a high-performance computing cluster, reducing the MR-STAT computation times to acceptable levels for offline reconstructions.

2.3 | Solution strategies

2.3.1 | Computing the time-domain signal s

In the first step of Algorithm 1, we need to compute $\mathbf{r} = \mathbf{d} - \mathbf{s}$ for the current estimate of the parameters α. Recall that

$$\mathbf{s} = \mathbf{M}\rho = \sum_{j=1}^{N_v} \rho_j \mathbf{m}_j. \tag{14}$$

Since the time evolution of the magnetization in each voxel is assumed to be independent of other voxels, the $\mathbf{m}_j$ can be computed independently from each other. In particular, storage of the matrix $\mathbf{M}$ is not required for computing $\mathbf{s}$; see Algorithm 2.

Algorithm 2 Compute $\mathbf{s}(\alpha) = \mathbf{M}\rho$ (matrix-free, serial)

  Initialize $\mathbf{s}$ = zeros(N_t,1)

  for $j \leftarrow 1$ to N_v **do**

    1. Integrate Bloch equations in time to obtain $\mathbf{m}_j = \left[m_j(t_1), \ldots, m_j(t_{N_t})\right]^T$

    2. Set $\mathbf{s} = \mathbf{s} + \rho_j \mathbf{m}_j$

  end for

  Return $\mathbf{s}$

Note that Algorithm 2 only requires the allocation of two vectors of length N_t, which is feasible on modern computing architectures for both 2D and 3D acquisitions. The computation of $\mathbf{s}$ can then be parallelized using N_c computing cores by following the procedure outlined in Algorithm 3 (see also Stöcker et al.[18] and Liu et al.[19]).

Algorithm 3 Compute $\mathbf{s}(\alpha) = \mathbf{M}\rho$ (matrix-free, parallel)

Require: Master process p_m, slave processes p_i for $i \in [1, \ldots, N_c]$.

  1. p_m distributes α: each p_i receives the parameters associated with voxels $[(i-1) * N_v + 1, \ldots, i * N_v]$.

  2. Each p_i uses Algorithm 2 to compute a "local" version of the signal $\tilde{\mathbf{s}}_i$.

  3. Each p_i communicates $\tilde{\mathbf{s}}_i$ back to the p_m.

  4. On p_m the signal $\mathbf{s}$ is computed as $\mathbf{s} = \sum_{i=1}^{N_c} \tilde{\mathbf{s}}_i$.

TABLE 1 On-disk sizes of MR-STAT matrices for $N_p = 4$ and $N_t = N = 4 \times N_v$ for an increasing number of voxels N_v. The memory sizes are computed as $2 \times N_v \times N_t \times 8$ bytes ($\mathbf{M}$), $2 \times N \times N_t \times 8$ bytes ($\mathbf{J}$), and $N^2 \times 8$ bytes ($\mathbf{H}$), respectively. The factors of 2 come from the real and imaginary components and the factor of 8 represents the bytes necessary to store 64-bit floating point numbers

Image size	Voxels (N_v)	M	J	H
64×64	4.096	1 GB	4 GB	2 GB
128×128	16.384	16 GB	64 GB	32 GB
256×256	65.536	256 GB	1.024 GB	512 GB
512×512	262.144	4.096 GB	16.384 GB	8.192 GB

The communication requirements for this parallelized algorithm can be summarized as follows:

- to distribute the parameters, the master process sends N/N_c parameters to each of the N_c slaves;
- to receive the local signals from the slaves, each slave sends a vector of length $N_t \approx N$ to the master process.

2.3.2 | Computing the gradient g

To compute $\mathbf{g} = \nabla f$ for the current estimate of the parameters $\boldsymbol{\alpha}$, recall that

$$\mathbf{g} = \Re(\mathbf{J}^H \mathbf{r}). \tag{15}$$

Since $\mathbf{J}$ is defined as

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial \mathbf{r}}{\partial \alpha_1} & \cdots & \dfrac{\partial \mathbf{r}}{\partial \alpha_N} \end{bmatrix}, \tag{16}$$

it follows that

$$\mathbf{g} = \begin{bmatrix} \Re\left(\left\langle \dfrac{\partial \mathbf{r}}{\partial \alpha_1}, \mathbf{r} \right\rangle\right) \\ \vdots \\ \Re\left(\left\langle \dfrac{\partial \mathbf{r}}{\partial \alpha_N}, \mathbf{r} \right\rangle\right) \end{bmatrix}. \tag{17}$$

To compute the $\partial \mathbf{r}/\partial \alpha_i$, again note that the magnetization in different voxels is assumed to evolve independently. Hence, if α_i is a parameter associated with voxel j (i.e., $j = i \bmod N_v$), it follows that

$$\frac{\partial \mathbf{r}}{\partial \alpha_i} = -\frac{\partial\left(\sum_{r=1}^{N_v} \rho_r \mathbf{m}_r\right)}{\partial \alpha_i} = -\frac{\partial\left(\rho_j \mathbf{m}_j\right)}{\partial \alpha_i}. \tag{18}$$

Using Algorithm 4 only requires storage of one vector of length N for the output and—in principle—one complex vector of length N_t to store the intermediate partial derivative vector. In practice, we will compute the N_p partial derivatives for each voxel simultaneously, so that N_p complex vectors of length N_t are stored simultaneously. For a high-resolution 2D scan, this only requires limited memory (in the order of tens of megabytes).

Algorithm 4 Compute $\mathbf{g}(\boldsymbol{\alpha}) = \Re(\mathbf{J}^H \mathbf{r})$ (matrix-free, serial)

Initialize $\mathbf{g}$ = zeros(N,1)

for $j \leftarrow 1$ to N_v **do**

    for $k \leftarrow 1$ to N_p **do**

        1. Set $i = j + (k-1)N_v$

        2. Compute partial derivative $\dfrac{\partial \mathbf{r}}{\partial \alpha_i} = -\dfrac{\partial(\rho_j \mathbf{m}_j)}{\partial \alpha_i}$

        3. Set $\mathbf{g}[i] = \Re\left(\left\langle \dfrac{\partial \mathbf{r}}{\partial \alpha_i}, \mathbf{r} \right\rangle\right)$

    end for

end for

Return $\mathbf{g}$

Algorithm 5 Compute $\mathbf{g}(\boldsymbol{\alpha}) = \Re(\mathbf{J}^H \mathbf{r})$ (matrix-free, parallel)

Require: Master process p_M, slave processes p_i for $i \in [1, \ldots, N_c]$.

    1. p_m distributes $\boldsymbol{\alpha}$: each p_i receives the parameters associated with voxels $[(i-1) * N_v + 1, \ldots, i * N_v]$.

    2. p_m distributes $\mathbf{r}$ to each p_i.

    3. Each p_i uses Algorithm 4 to compute a "local" gradient $\tilde{\mathbf{g}}_i$.

    4. Each p_i communicates $\tilde{\mathbf{g}}_i$ back to the p_m.

    5. On p_m the gradient $\mathbf{s}$ is computed by vertical concatenation of the $\tilde{\mathbf{g}}_i$.

Utilizing Algorithm 4, the computation of $\mathbf{g}$ can be performed in parallel, as outlined in Algorithm 5. The parallelization schemes for both the signal and gradient computations are visualized in Figure 1.

Communication requirements for the parallel gradient computation can be summarized as follows:

- to distribute the parameters, the master process sends N/N_c parameters to each of the N_c slaves;
- to distribute the residual vector, the master process sends a vector of length N_t to each slave;
- to receive the local gradients from the slaves, each slave sends a vector of length N/N_c to the master process.

Note that, for both Algorithms 3 and 5, the communication requirements scale linearly with the number of parameters N for a fixed number of cores N_c. Since $N = N_v \times N_p$, it follows that the communication requirements scale linearly with N_v as well. As discussed in the previous section, the computational requirements scale quadratically with N_v. Therefore we hypothesize that, as long as $N_c \ll N_v$, the communication overhead is

negligible compared with reduction in computation times achieved by dividing the computation load over N_c computing cores. That is, we expect the total computation time to decrease linearly with the number of cores available under this assumption. This hypothesis is confirmed in Figure 2 in the results section.

2.3.3 | Incorporating curvature information

Given the ability to compute the gradient **g** using the matrix-free, parallelized algorithm from the previous subsection, in principle the so-called limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS[20]) method can be applied to obtain the update step **p** at each iteration. The L-BFGS method approximates the inverse of the Hessian matrix using a limited number of gradient vectors from previous iterations. However, in practice the L-BFGS method was observed to result in poor convergence.

Alternatively, since we are dealing with a least-squares problem, a Gauss–Newton method might be used, in which the Hessian matrix **H** in Algorithm 1 is approximated by $\Re(\mathbf{J}^H\mathbf{J})$ and

$$\Re\left(\mathbf{J}^H\mathbf{J}\right)\mathbf{p} = -\mathbf{g} \tag{19}$$

is solved to obtain update steps **p**. Note that the matrix $\Re(\mathbf{J}^H\mathbf{J})$ is of the same size as the Hessian matrix itself and thus, in principle, cannot be stored in computer memory. If, however, we use iterative techniques (e.g., a conjugate gradient method) to solve the linear system $\Re(\mathbf{J}^H\mathbf{J})\mathbf{p} = -\Re(\mathbf{g})$, we only need matrix–vector products with $\Re(\mathbf{J}^H\mathbf{J})$. In the previous subsection we outlined how matrix–vector products of the form $\mathbf{J}^H\mathbf{v}$ may be computed in a matrix-free, parallelized fashion. Similar techniques can be applied to matrix–vector products of the form $\mathbf{J}\mathbf{v}$. Hence matrix–vector products of the form $\Re(\mathbf{J}^H\mathbf{J})\mathbf{v}$ can be computed in a matrix-free, parallelized fashion by first computing $\mathbf{y} = \mathbf{J}\mathbf{v}$ and subsequently computing $\Re(\mathbf{J}^H\mathbf{y})$. With this technique, the linear system in Equation (19) can be solved numerically even for large-scale problems. In practice, it will not be necessary to solve Equation (19) to high precision and the number of iterations in this *inner loop* can be limited, resulting in an *inexact* Gauss–Newton method (see Steihaug[15] and Algorithm 7.2 in Nocedal and Wright[16]) as outlined in Algorithm 6.

Algorithm 6 (Inexact Gauss–Newton MR-STAT) Minimize $f(\boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{d} - \mathbf{s}(\boldsymbol{\alpha})\|_2^2$

Require: Initial guess $\boldsymbol{\alpha}_0$, initial trust radius Δ_0

  while not converged **do**
    Compute **s** (matrix-free, parallel)
    Set $\mathbf{r} = \mathbf{d} - \mathbf{s}$
    Compute $\mathbf{g} = \mathbf{J}^H\mathbf{r}$ (matrix-free, parallel)
    Solve with CG iterations (inner GN loop):
      $\Re\left(\mathbf{J}^H\mathbf{J}\right)\mathbf{p} = -\mathbf{g}$ (matrix-free, parallel)
    Set $\boldsymbol{\alpha} = \boldsymbol{\alpha} + \mathbf{p}$
  end while
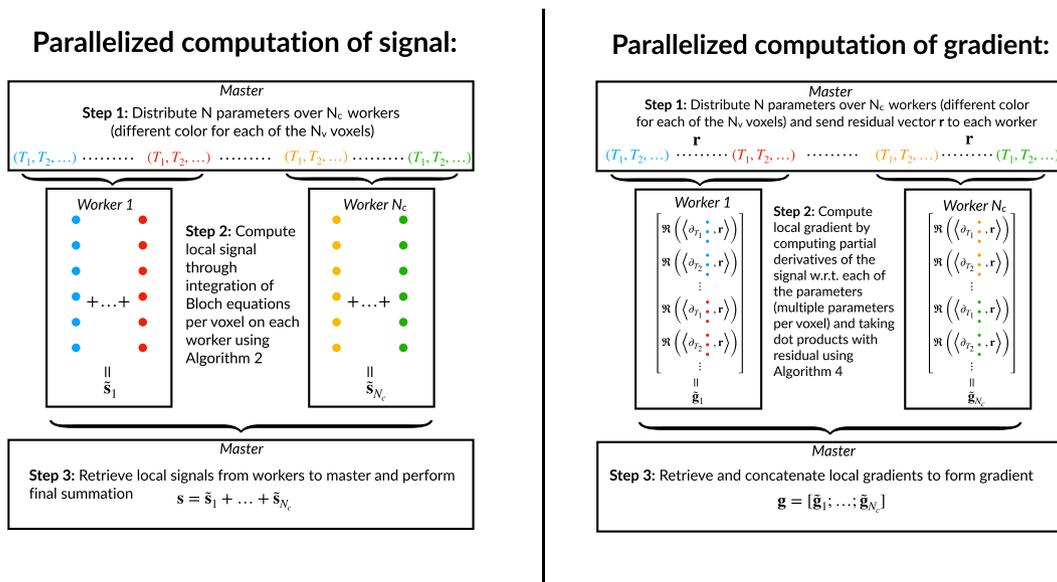
FIGURE 1 Visualization of the (matrix-free) algorithms to compute the signal [left] and the gradient [right] in a parallelized fashion
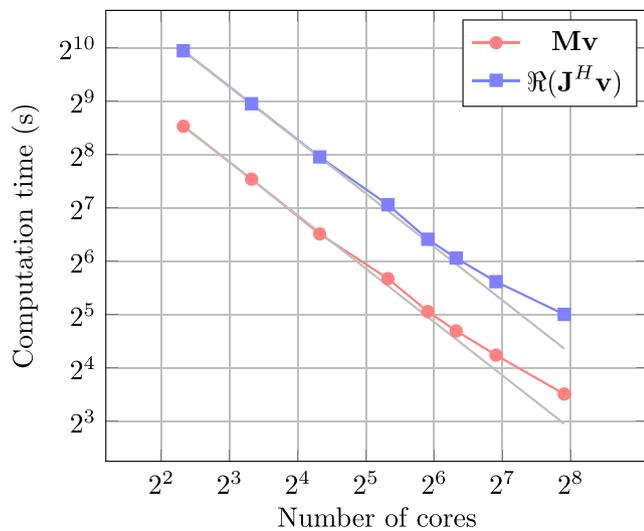
FIGURE 2 Time needed to compute matrix–vector products of the form **Mv** and $\mathbf{J}^H\mathbf{v}$ for different numbers of cores N_c used on a high-performance computing cluster

3 | METHODS

The matrix-free, parallelized MR-STAT reconstruction algorithms was tested on both simulated and experimentally acquired data.

3.1 | Pulse sequence

In all test cases, a transient-state 2D balanced gradient-echo pulse sequence similar to the pulse sequence in Sbrizzi et al.[12] was used. Throughout the whole sequence, the TR was fixed and TE was set to TR/2. A linear, Cartesian sampling strategy was employed, together with time-varying flip angles that change according to a smoothly varying pattern. We refer to Supporting Information S1 for more details on the sampling trajectory and flip-angle pattern. The phase of the RF pulse alternated between 0 and 180°. Changing the flip angles prevents the spins from reaching a steady state and, by following a smoothly varying pattern, the spin-echo behaviour of bSSFP sequences[21] is preserved to a large extent. This spin-echo-like behaviour is needed for proper T_2 estimation and at the same time it also effectively eliminates sensitivity to ΔB_0 within a certain passband of off-resonances.[22] An added benefit of the smoothly changing flip-angle train is the improved convexity of the minimization landscape.[23]

Each RF pulse has a Gaussian envelope, and at the start of the pulse sequence a non-selective inversion pulse is played out for enhanced T_1 encoding. The pulse sequence was implemented on a 1.5T clinical MR system (Ingenia, Philips Healthcare, Best, the Netherlands).

3.2 | Reconstructions

All reconstruction code was written in the open-source Julia programming language.[24] To compute the MR signal for a given set of parameters, an optimized Bloch-equation solver was implemented, which also takes into account the slice profile.[25] To compute exact partial derivatives, algorithmic differentiation in forward mode[26] was implemented. We refer to Supporting Information S2 for more details.

The inexact Gauss–Newton method was implemented using a trust-region framework (following Steihaug[15] and Algorithm 7.2 in Nocedal and Wright[16]). In order to facilitate bound constraints on the parameters, reflection at feasible boundaries was incorporated.[27] For the L-BFGS method, an implementation from the Optim.jl package[28] was used. The reconstruction algorithm was implemented on a high-performance computing cluster, which consists of multiple Intel Xeon Gold 6148 nodes with 40 cores each, on which the CentOS Linux 7 (Core) operating system is installed.

For all experiments, T_1, T_2, and ρ (complex) maps are reconstructed. For the data obtained with clinical MR systems, we also reconstruct $|B_1^+|$ to take transmit field inhomogeneities into account. The off-resonance ΔB_0 was set to zero and thus it was not reconstructed, because of the flat spectral response of the balanced sequence within the passband. The nonlinear parameters were initialized as follows: $T_1 = 1000$ ms, $T_2 = 100$ ms, $|B_1^+| = 1$ a.u. and $\Delta B_0 = 0$ Hz. In previous work,[12] the variable projection method (VARPRO[29]) was utilized to separate the linear parameters (i.e., proton density) from the nonlinear parameters. The VARPRO method in principle requires computation (through SVD or QR decomposition) and storage of an orthogonal basis for the matrix **M**. For the matrix sizes in the current work, that would be computationally infeasible and it is nontrivial to extend the VARPRO technique to matrix-free methods. Therefore, in the current work we treat the proton densities as nonlinear parameters. We only make use of the linearity to provide an initial guess for the proton densities. That is, given the initial guess for the nonlinear parameters, the (complex) proton density was initialized as the least-squares solution to the linear system $\mathbf{M}(\alpha_0)\rho = \mathbf{d}$ obtained using a linear solver (LSQR). Based on the resulting initial guess for the proton density, a mask was drawn to exclude regions with no significant proton density from subsequent simulations.

In all reconstructions, logarithmic scaling is applied to both T_1 and T_2 parameters. The variable substitution brings both variables into a similar range and thus improves convergence of the algorithm.

TABLE 2 Acquisition parameters for in vivo MR-STAT brain scans

Acquisition parameter	Subjects 1 and 2	Subject 3
Field strength	1.5T	1.5T
In-plane resolution	1 mm × 1 mm	1 mm × 1 mm
Field of view	224 mm × 224 mm	224 mm × 224 mm
Slice thickness	5 mm	3 mm
TR	7.6 ms	7.9 ms
TE	3.8 ms	3.95 ms
Readout bandwidth	85.6 kHz	85.6 kHz
Pulse duration	0.76 ms	0.81 ms
Scan time	13.6 s	14.15 s

Numerical Brain Phantom Reconstruction
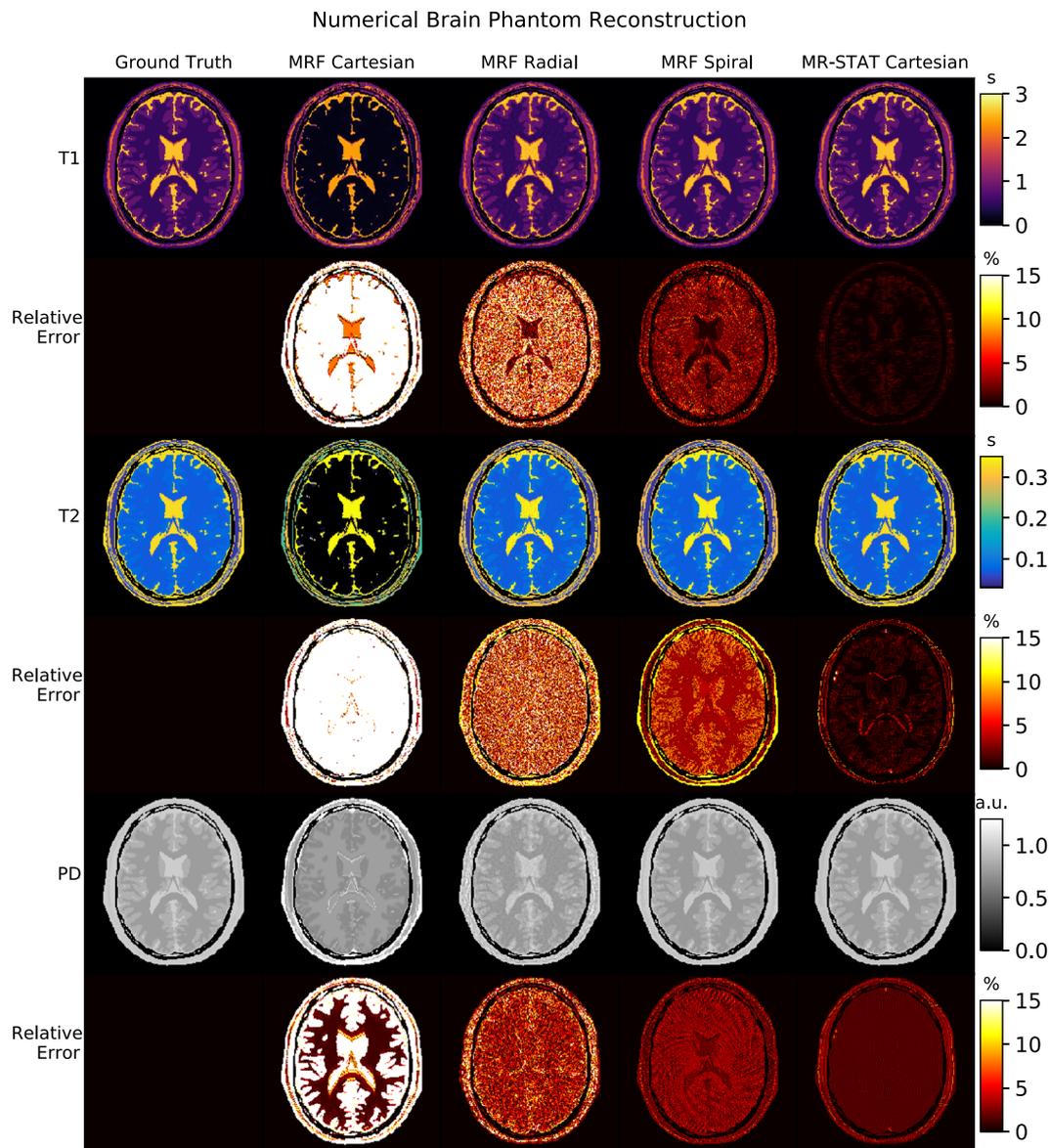
FIGURE 3 First column: ground-truth T_1, T_2, and proton density maps for the numerical brain phantom. Second, third and fourth columns: reconstructed parameter maps and relative error maps for MRF with linear Cartesian, golden angle radial, and golden angle spiral trajectories, respectively. Fifth column: reconstructed parameter maps and relative error maps for MR-STAT using a linear, Cartesian sampling trajectory. The MRF spiral and MR-STAT reconstructions both show excellent agreement with the ground-truth values. The radial MRF reconstructions show residual aliasing artefacts and the Cartesian MRF reconstruction is heavily biased

The reconstruction code will be made available online at https://gitlab.com/mrutrecht/mrstat after acceptance of the manuscript for publication.

3.3 | Numerical brain simulation

Signal from a numerical brain phantom[30] with a field of view of 192 mm × 192 mm and voxel size of 1 mm × 1 mm was simulated using the transient-state pulse sequence. A total number of 8 × 192 = 1536 readouts were simulated (each phase encoding line was acquired eight times,

Parameter	Metric	Units	MRF Cartesian	MRF radial	MRF spiral	MR-STAT Cartesian
T_1	NRMSE	[a.u.]	0.2302	0.0432	0.0110	0.0025
	MAPE	[%]	65.6	8.5	2.5	0.4
	HFEN	[a.u.]	18.1	15.6	15.7	15.7
T_2	NRMSE	[a.u.]	0.2486	0.0756	0.0492	0.0048
	MAPE	[%]	56.1	8.1	4.9	0.9
	HFEN	[a.u.]	3.2	2.5	2.4	2.5
ρ	NRMSE	[a.u.]	1.0979	0.2626	0.1129	0.0830
	MAPE	[%]	15.8	4.8	2.7	1.8
	HFEN	[a.u.]	8.5	5.6	5.5	5.4

TABLE 3 Three different error metrics (NRMSE, HFEN, MAPE) computed for the MRF (Cartesian, radial, and spiral) and MR-STAT (Cartesian) reconstructions on the numerical brain phantom. No noise was added to the data in these reconstructions. The MR-STAT reconstructions result in the lowest errors, because the reconstructions do not suffer from undersampling artefacts and there are no discretization errors due to a finite dictionary
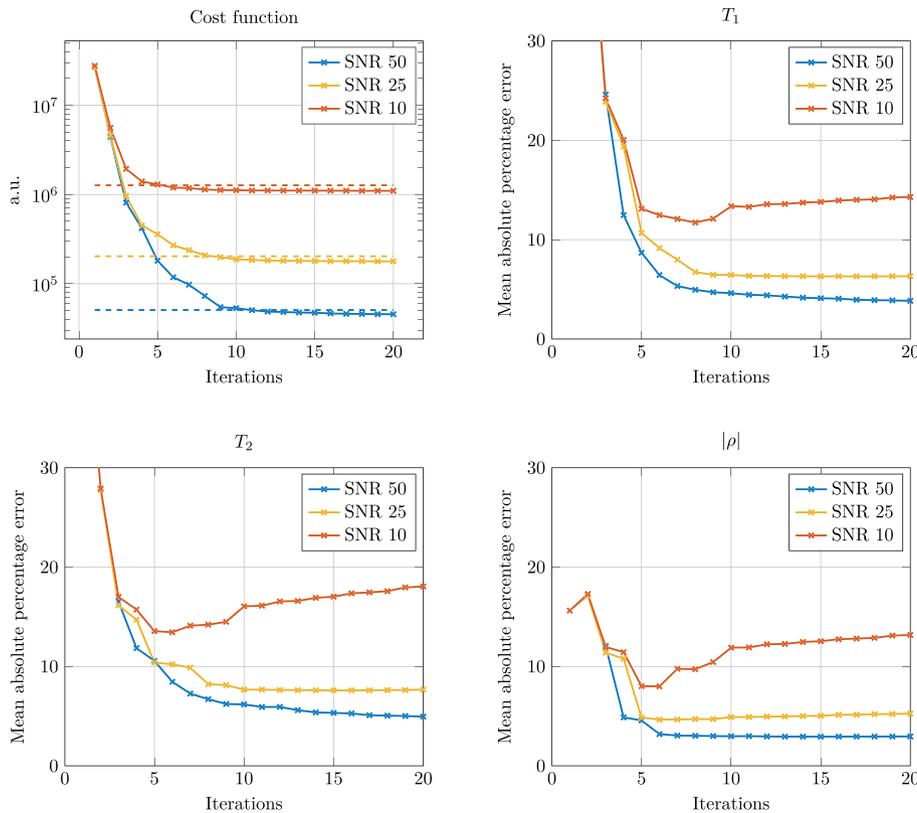
FIGURE 4 Top left: convergence curves for the inexact Gauss–Newton MR-STAT method applied to data generated from the numerical brain phantom with different noise levels (SNR 50, 25, 10). In all cases, the value of the cost function converges to the value expected based on the noise level. Top right and bottom row: mean absolute percentage errors for T_1, T_2, and proton density (magnitude) maps per iteration of the inexact Gauss–Newton method for different noise levels

but note that for each readout line the flip angle and thus state of the spins is different) with a TR of 7.88 ms and a TE of 3.94 ms. The total sequence duration was 12.1 s.

Reconstructions were performed using 64 cores. The number of outer and inner iterations for the inexact Gauss–Newton method was limited to 40 and 20, respectively.

For comparison purposes, we also performed MRF reconstructions on signal from the numerical brain phantom using the Cartesian trajectory, as well as signal from radial and spiral trajectories, for which MRF is known to work well. In all three cases, the same flip-angle train, TE, and TR were used. For the radial case, k_{max} was extended by a factor of $\sqrt{2}$ and each readout the spoke was rotated by the golden angle. For the spiral acquisition, a variable-density spiral was generated[31,32] that would require 24 interleaves to sample the inner region of k-space fully and 48 interleaves for the outer region of k-space. The spiral was rotated by the golden angle each readout. Data point $\mathbf{d}^{MRF}$ was then simulated by applying a forward operator, consisting of the (non-uniform) FFT[33] and an undersampling operator, on fingerprints simulated using the numerical brain phantom. To perform the MRF reconstructions, a dictionary consisting of 100 T_1 values from 0.1–5.0 s in increments of 4% and 100 T_2 values from 0.01–2.0 s in increments of 5.5% was generated. The T_1 and T_2 values of the phantom were not used in generating the dictionary. The dictionary was compressed in the time direction to rank 5[34] using the SVD. For all trajectories, (linear) low-rank forward operators $\mathbf{E}^{MRF}$ were formed that consisted of the expansion of low-rank coefficients to the full time series, a nuFFT operator, and a sampling operator compression.[13] Low-rank snapshot images $\mathbf{x}^{MRF}$ were reconstructed from the undersampled data $\mathbf{d}^{MRF}$ by solving the linear system

$$\mathbf{x}^{MRF} = \text{argmin}_\mathbf{x} \|\mathbf{d}^{MRF} - \mathbf{E}^{MRF}\mathbf{x}\|_2^2 \tag{20}$$

with LSQR (similar to e.g. Zhao et al.[35] and low-rank inversion in Assländer et al.[13]). Finally, dictionary matching with the compressed dictionary was performed on $\mathbf{x}^{MRF}$ to obtain the parameter estimates.
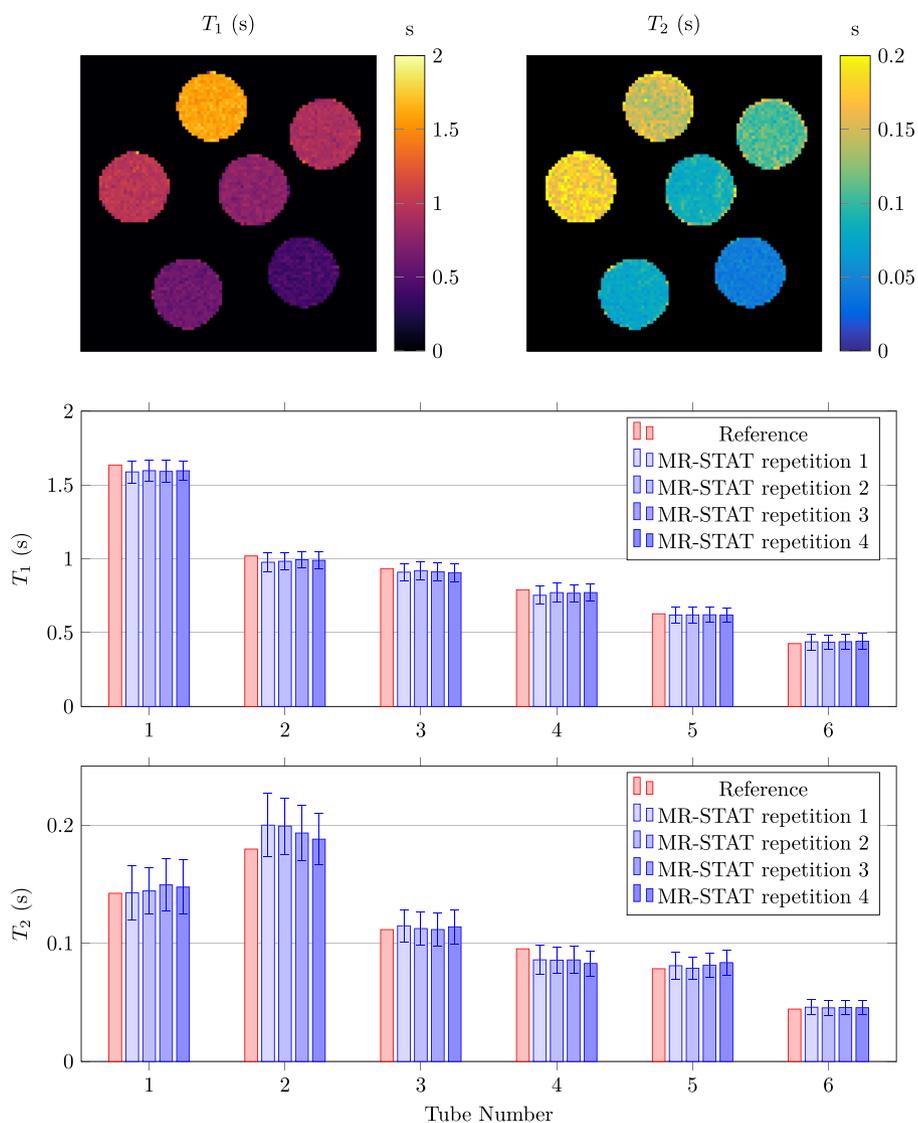
FIGURE 5 Top row: T_1 and T_2 maps reconstructed with MR-STAT from gel-phantom data. Middle and bottom rows: comparison of mean T_1 and T_2 values obtained with MR-STAT and gold standard methods for each of the six gel phantom tubes. For MR-STAT, the acquisition has been repeated four times

To study the effect of noise on the MR-STAT reconstruction algorithm further, additional reconstructions were performed, where complex Gaussian noise was added to the simulated signal such that $\|signal\|_2/\|noise\|_2 = 50, 25$, and 10.

3.4 | Gel phantom experiment

Signal from a 2D transverse slice of six gadolinium-doped gel phantoms (TO5, Eurospin II test system, Scotland) was collected on the 1.5T MR system using the manufacturer's 13-channel receive headcoil. In total, $8 \times 96 = 768$ readout lines were acquired with a spatial resolution of 1 mm $\times$ 1 mm $\times$ 5 mm and a field of view of 96 mm $\times$ 96 mm. The TR and TE were 7.4 ms and 3.7 ms, respectively, resulting in a total acquisition time of 5.7 s. For reproducibility purposes, the MR-STAT scan was repeated four times, with full relaxation in between the different scans.

Parameters that describe the pulse sequence were exported from the scanner and subsequently loaded into Matlab.[36] The measured signals from different receive channels were compressed into a single signal by applying principal component analysis and choosing the principle mode.[37] Reconstructions of the parameter maps were performed using the inexact Gauss–Newton method on the computing cluster using 32 cores. The number of inner iterations was limited to 15, whereas the number of outer iterations was limited to ten.

To assess correctness of the T_1 and T_2 maps reconstructed with MR-STAT, data were also acquired using gold standard methods in the form of an inversion-recovery single spin-echo protocol with inversion times of [50, 100, 150, 350, 550, 850, 1250] ms for T_1 mapping and a single-echo spin-echo protocol with echo times of [8, 28, 48, 88, 138, 188] ms for T_2 mapping.

3.5 | In-vivo experiments

Using the 1.5T clinical MR system, we also acquired signal from 2D transverse slices of the brain in three healthy volunteers. Each volunteer gave written informed consent. For each acquisition, a total of $8 \times 192 = 1536$ readout lines were acquired, with acquisition parameters as reported
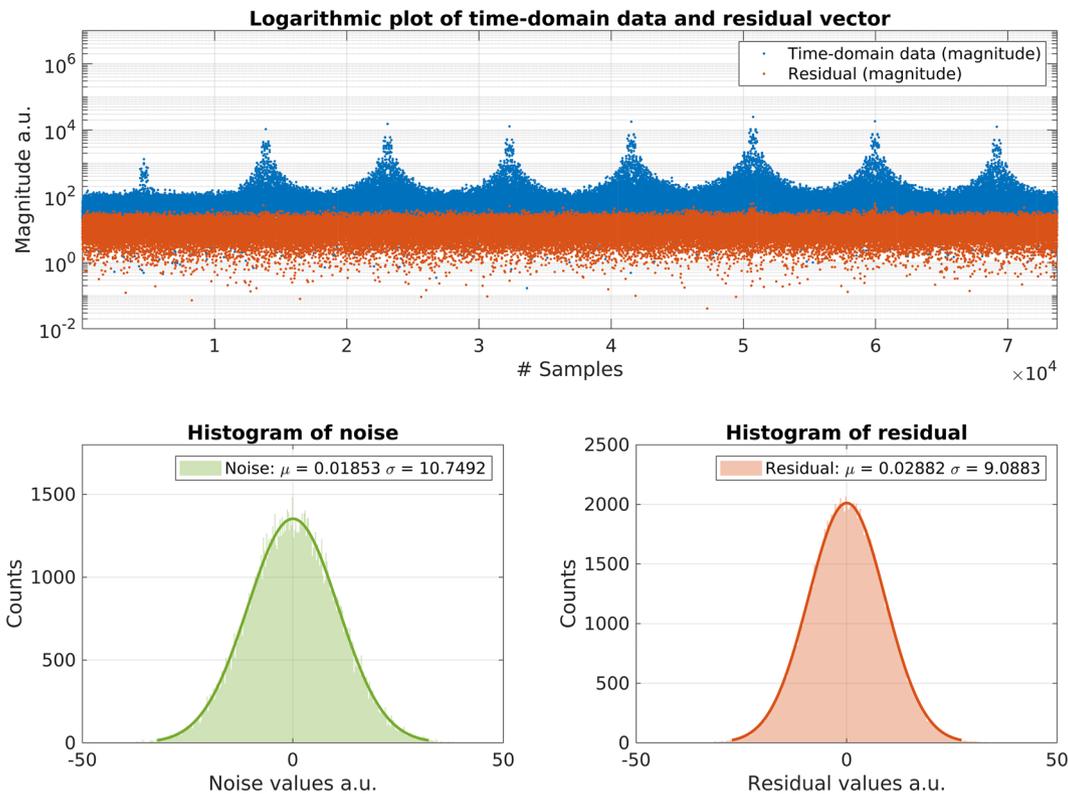
FIGURE 6 Top row: logarithmic plot of the magnitude of the measured time-domain data obtained from the gel phantoms and the magnitude of the residual vector entries after the fifth iteration of the inexact Gauss–Newton method. Bottom left: histogram of noise values (real and imaginary values concatenated). The noise was measured using the receive channels right before the actual acquisition and was subjected to the same preprocessing steps as the data used in the reconstruction (e.g., compression to a single channel using SVD). Bottom right: histogram of the residual vector entries (real and imaginary values concatenated) after the fifth iteration of the inexact Gauss–Newton method

in Table 2. After masking approximately, 25 000 voxels remain for which quantitative parameters are estimated. The MR-STAT reconstructions were performed with 64 cores using the same reconstruction settings as for the gel phantom experiment.

To demonstrate the effect of accelerated acquisitions, we also performed reconstructions using time-domain data corresponding to the first 896 TR and the first 448 TR from one of the subjects. The corresponding acquisition times were 6.8 and 3.4 s, respectively.

One of the in vivo brain datasets was also used to test the effectiveness of the parallelization scheme. Individual matrix–vector products of the form $\mathbf{Mv}$ and $\mathbf{J}^H\mathbf{v}$ were computed and timed for 5, 10, 20, 40, 60, 80, 100, 120, and 240 cores, respectively.

4 | RESULTS

4.1 | Parallelization

In Figure 2, the time required to compute matrix–vector products of the form $\mathbf{Mv}$ and $\mathbf{J}^H\mathbf{v}$ for the one of the in vivo datasets is shown for an increasing number of computing cores N_c. Initially we observe a linear decrease in computation times; however, this linear decrease flattens beyond approximately 64 cores. This effect can be explained by the increase in communication overhead when using more cores and increased competition between cores for shared resources like memory bandwidth and cache memory. Although the linear decrease flattens beyond 64 cores, a decrease in computation times is still observed even when going towards 240 cores. Because, for MR-STAT, reconstruction times are dominated by the computation of these matrix–vector products, the reconstruction times can thus be reduced effectively by the proposed parallel implementation.

4.2 | Numerical brain phantom

The T_1, T_2, and proton density maps reconstructed using (Cartesian) MR-STAT and Cartesian, radial, and spiral MRF are shown in Figure 3, as well as the corresponding absolute relative error maps. It can be seen that the parameter maps reconstructed with either MR-STAT or spiral MRF are in excellent agreement with the ground truth. The radial MRF reconstructions show stronger residual streaking artefacts, but, in general, the estimated parameter values are close to the ground truth. For the Cartesian case, the MRF reconstruction is unable to cope with the high undersampling (factor 192), resulting in severely biased parameter maps.
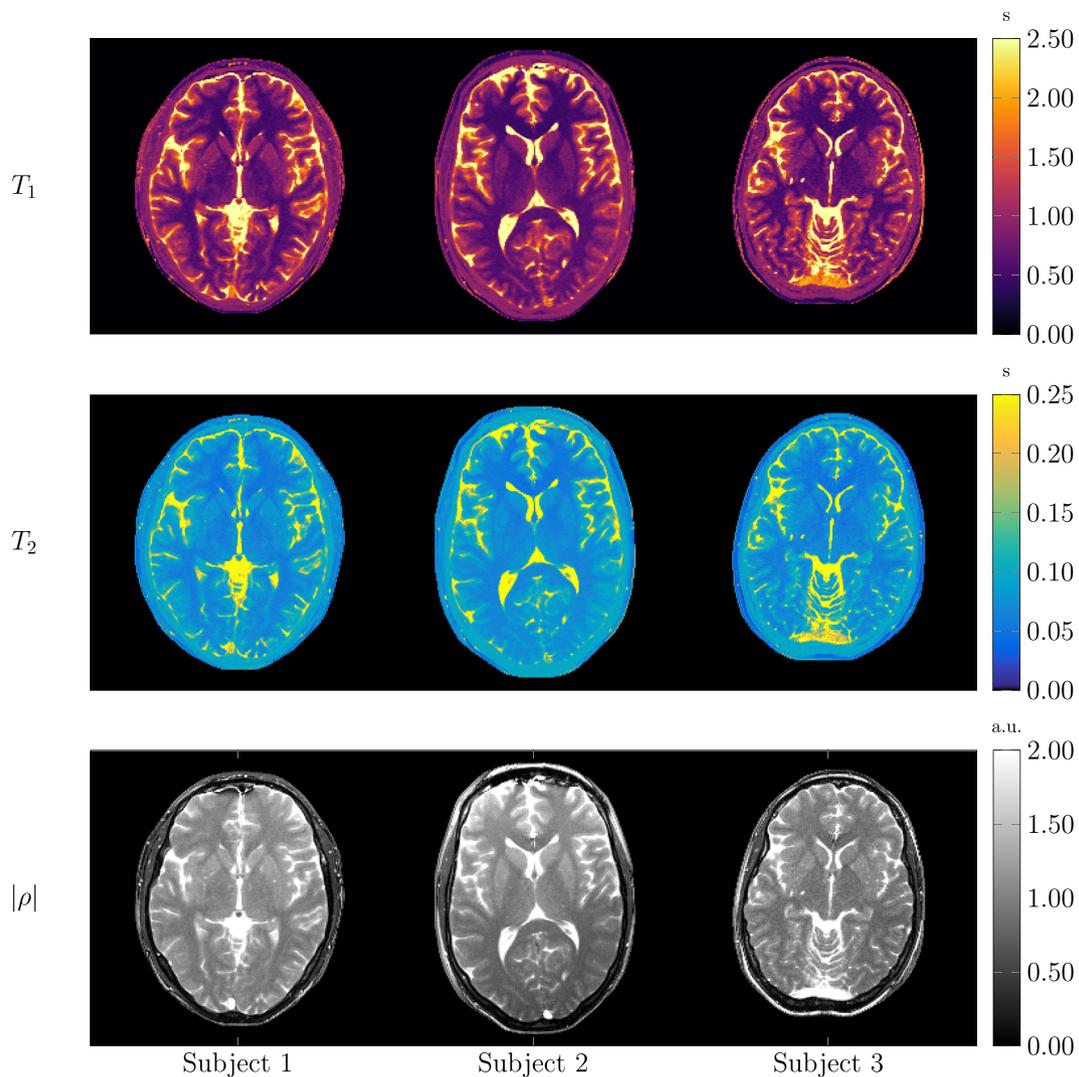
FIGURE 7 T_1, T_2, and proton density (magnitude) maps reconstructed with MR-STAT from in vivo brain data obtained at 1.5T (Philips, Ingenia) from multiple healthy volunteers. The in-plane resolution was $1 \times 1\,mm^2$ for all three subjects. For subjects 1 and 2, the acquisition time was 13.6 s and the slice thickness was 5 mm. For subject 3, the acquisition time was 14.15 s and the slice thickness was 3 mm

TABLE 4 Mean T_1 and T_2 values and standard deviation in white and gray matter regions for each of the three in vivo brain scans

Tissue type	Subject	T_1	T_2
Frontal white matter	1	505 ± 48 ms	53.3 ± 4.0 ms
	2	542 ± 48 ms	57.4 ± 3.8 ms
	3	519 ± 54 ms	56.1 ± 4.3 ms
Putamen (gray matter)	1	874 ± 64 ms	74.8 ± 4.4 ms
	2	956 ± 66 ms	80.2 ± 4.5 ms
	3	895 ± 107 ms	78.4 ± 7.0 ms

To quantify the quality of the reconstructions, normalized root-mean-square errors (NRMSE), high-frequency error norms (HFEN[38], with standard deviation of 1.5 pixels) and mean absolute percentage errors (MAPE) were computed and are reported in Table 3. It can be seen that the MR-STAT reconstruction results in the lowest NRSME and MAPE for all three parameters. The HFEN for the radial and spiral MRF and Cartesian MR-STAT reconstructions are similar.

In Figure 4, convergence curves for MR-STAT with the inexact Gauss–Newton method for different SNR levels (50, 25, and 10) are shown, as well as mean absolute percentage errors per iteration for T_1, T_2, and proton density. For the higher SNR cases, the error values stabilize and the method converges after relatively few, for example, ten, outer iterations. It can be seen that, for the lowest SNR case, overfitting occurs after around six iterations. Based on these observations, the number of outer iterations for the in vivo case was chosen to be ten.

4.3 | Gel phantoms

In Figure 5, reconstructed T_1 and T_2 maps for the gel phantoms are shown and the mean T_1 and T_2 values per tube are compared with the gold standard measurements. It can be seen that the mean values are in excellent agreement. The mean values reported for the different repetitions
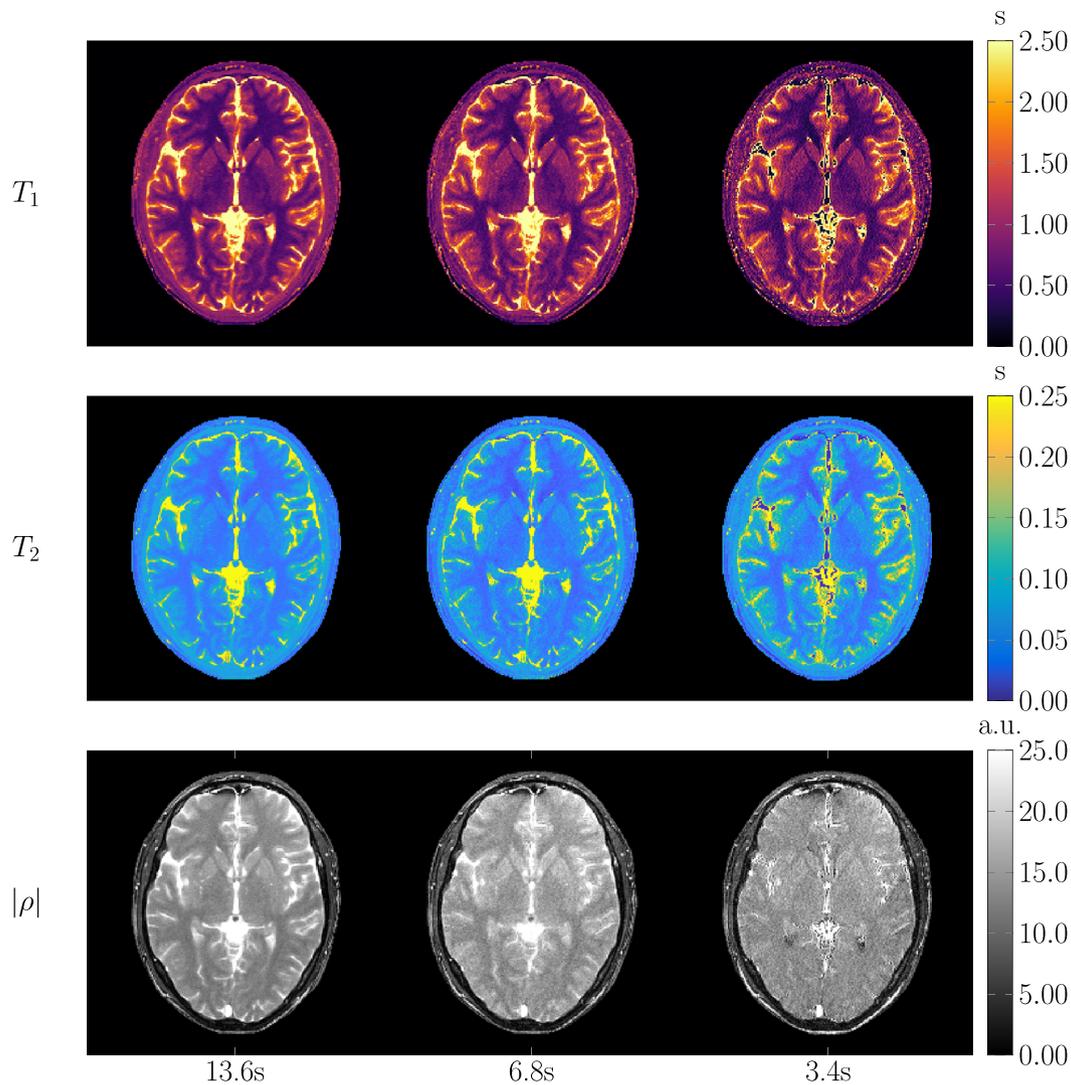
FIGURE 8 In vivo T_1, T_2, and proton density (magnitude) maps at 1 mm × 1 mm in-plane resolution reconstructed with MR-STAT based on acquisitions of, respectively, 13.6, 6.8, and 3.4 s on a 1.5T MR system (Philips, Ingenia)

of the MR-STAT scans are also in good agreement with each other (i.e., within standard deviations). In general, the standard deviations for the reconstructed T_2 values are higher than for the T_1 values, indicating a much stronger encoding of T_1 information in the signal, which can be explained by the inversion pulse at the start of the sequence.

To reconstruct the parameter maps, only five iterations of the reconstruction algorithm were needed and the total reconstruction time was approximately nine minutes using 32 computing cores. In Figure 6, a logarithmic plot of the measured signal magnitude and the residual vector after the fifth iteration are displayed for one of the MR-STAT repetitions. Histograms of the measured noise and the residual vectors are also shown. It can be seen that the residual vector follows a zero-mean Gaussian distribution with standard deviation similar to the noise, indicating that the model used in MR-STAT is able to describe the measured time-domain signal adequately.

4.4 | High-resolution 2D brain scan

In Figure 7, the reconstructed T_1, T_2, and proton density (magnitude) maps for the in vivo brain scans performed on the three volunteers are shown. The maps show clear contrast between white matter, gray matter, and cerebrospinal fluid (CSF). The maps corresponding to subject 3 appear noisier compared with the maps corresponding to subjects 1 and 2, which can be explained by the differences in slice thickness in the acquisition (3 mm versus 5 mm). Mean T_1 and T_2 values and standard deviations in regions of white and gray matter are reported in Table 4. The mean values are generally in good agreement with values found in the literature for 1.5T experiments,[6,39,40] although we do observe an underestimation compared with some other studies, especially in white matter. We expect the underestimation is related to magnetization transfer, which is known to affect the signal of balanced gradient-echo sequences (in a way that depends on the TR and RF pulse duration used).[41,42] The reconstruction time for each slice was approximately five hours using 64 cores.

In Figure 8, we show T_1, T_2, and proton density (magnitude) for the same 2D brain slice but reconstructed using data corresponding to, respectively, 13.6, 7.8, and 3.4 s acquisitions. It can be seen that the maps corresponding to the 6.8 s acquisition are comparable to the maps

corresponding to the 13.6 s acquisition, except that more noise is present. Depending on the application, it might be more beneficial to repeat such a shorter sequence multiple times for noise averaging, instead of scanning with the longer sequence. An added benefit of a shorter sequence duration is that the Bloch simulations are faster and thus reconstruction times are reduced by approximately the same factor by which the scan time is reduced. For the 3.4 s acquisition, the MR-STAT problem (Equation (10)) is underdetermined in the sense that the number of data points is fewer than the number of unknowns in the problem. As can be seen in the reconstructed maps, this mostly results in biases in the CSF values. Note that in none of the reconstructions were parallel imaging or compressed sensing techniques utilized.

5 | DISCUSSION AND CONCLUSION

MR-STAT is a framework for obtaining multiple quantitative parameter maps by fitting directly to measured time-domain data obtained from one short scan. Rather than relying on the FFT for spatial localisation of signal in a separate step, the spatial localisation and parameter estimation are performed simultaneously by solving a single nonlinear optimization problem iteratively using a signal model that explicitly includes the spatial encoding gradients. The inherent large scale of the problem brings along new challenges in terms of computer memory requirements and computation times, which make it difficult to perform MR-STAT reconstructions at high resolution. To address these issues, we have presented a parallel and matrix-free reconstruction algorithm for MR-STAT and demonstrated that it can be used to generate high-resolution quantitative parameter maps.

All MR-STAT experiments in the current work have been performed with linear, Cartesian sampling strategies. This sampling strategy offers important advantages in the form of robustness to hardware imperfections (e.g., eddy currents, especially for gradient-balanced sequences[43,44]), less susceptibility to ΔB_0 related blurring artefacts,[45] and direct availability on clinical MR systems. Within the conventional MRF framework, it is more challenging to work with Cartesian sampling strategies, as demonstrated using the simulation experiments. Studies that perform Cartesian MRF[46,47] therefore typically acquire multiple readout lines per snapshot, resulting in much longer acquisition times compared with non-Cartesian MRF acquisitions. A formal explanation of why Cartesian acquisitions are less suitable for MRF is reported in Stolk et al.[17] More advanced iterative MRF reconstructions[13,14,48,49] might perform better with Cartesian sampling than the currently used MRF reconstructions (low-rank inversion followed by low-rank dictionary matching), and an in-depth comparison will be the subject of further studies. It should also be noted that neither the MR-STAT framework nor the currently proposed reconstruction algorithm is restricted to Cartesian sampling and further research is also aimed at incorporating non-Cartesian trajectories into MR-STAT.

An additional benefit of the volumetric signal model used in MR-STAT over FFT-based methods is that dynamic behaviour during readouts (e.g., T_2 decay and ΔB_0 induced phase accumulation) is taken into account. This may be especially beneficial for improving reconstructions based on acquisitions with long readouts (e.g., spiral readouts).

MR-STAT reconstructions are performed by solving a nonlinear optimization problem using gradient-based iterative methods. No precomputed dictionary is used. Compared with dictionary-matching approaches, there are no discretization errors and the reconstruction procedure is also flexible with respect to changes in sequence parameters (e.g., no rebuilding of a dictionary required when scan settings change). A downside of using iterative reconstruction algorithms to solve nonlinear optimization problems is the risk of landing in a local minimum. In practice, with the currently used pulse sequence with smoothly changing flip angles and initial guess of the parameters, we have not encountered issues with local minima.[23]

Whereas with MRF the addition of new parameters results in an exponential increase in dictionary size (and thus also an exponential increase in dictionary generation and matching time), with MR-STAT additional parameters can be added at a quadratic increase in computation time. The quadratic increase can be explained as follows. The total number of parameters to be reconstructed N increases linearly with the number of parameters per voxel ($N = N_p N_v$). Since the minimum number of time points N_t that needs to be acquired—and thus simulated—is of the order of N, the computation time per Bloch simulation increases linearly as well. In addition, the number of partial derivative computations that needs to be performed per voxel also increases linearly. That is, the number of both rows and columns of the Jacobian **J** increases linearly, resulting in approximately a quadratic increase in computation time. In this respect, we do note that, although currently ΔB_0 maps are not reconstructed (because the bSSFP sequence used in this work is designed not to be sensitive to ΔB_0 within the passband), ΔB_0 is considered as parameter in all Bloch simulations and partial derivative computations. In addition, for the MR-STAT experiments described in the article, we used pulse sequences such that $N_t \approx 2N$, so that the problem remains overdetermined when an additional parameter is reconstructed. Therefore, assuming a pulse sequence is used that has sufficient ΔB_0 encoding,[10,50] we do not expect to see an increase in computation times when reconstructing ΔB_0 as an additional parameter.

For the phantom experiment, we observed that the noise level was reached for the residuals. However, this was not observed for the in vivo case, as certain effects are still accounted for in the model. Examples include patient motion, blood flow, magnetization transfer, and diffusion effects.

A limitation of the proposed method is that, at this moment, reconstruction times are still long for high-resolution scans, especially when compared with the dictionary matching procedures used in MRF. Even when employing a high-performance computing cluster, reconstruction times are of the order of hours for a single 2D brain slice. Although possible from a memory point of view, 3D reconstructions will take too long for practical purposes with the current reconstruction setup. The main bottleneck in the reconstructions is formed by the partial derivative computations needed to solve Equation (19). Further research is aimed at performing these computations on GPU architectures,[51,52] reducing the

computational effort through algorithmic improvements[53] and through the use of surrogate models.[54] Together with (cloud) computing resources becoming cheaper and more accessible over time, we believe it is possible to accelerate the computations to such an extent that MR-STAT becomes applicable in clinical settings.

Further research is also aimed at reducing the acquisition time and improving the precision and accuracy of MR-STAT parameter maps by incorporating parallel imaging,[55] compressed sensing, and through sequence optimization.

The main aim of the MR-STAT project is to explore possibilities to achieve very short acquisition times beyond what is possible with FFT-based frameworks. Although the MR-STAT framework in principle allows for much flexibility in the data acquisition process (e.g., non-Cartesian acquisitions), in the current work we have opted for Cartesian sampling patterns because of their robustness to hardware imperfections and because they clearly exemplify the benefits of skipping the FFT step (i.e., no introduction of artificial aliasing noise through application of the FFT to undersampled k-spaces). An additional benefit is the direct availability of such sequences on clinical MR systems. In the current work, we used constantly varying flip-angle trains; however, as shown in Supporting Information S3, MR-STAT could even be used with Cartesian bSSFP sequences with a fixed flip angle per k-space that require little to no pulse programming for their implementation.

ACKNOWLEDGMENT

ORCID

Oscar van der Heide (iD) https://orcid.org/0000-0002-8451-525X

REFERENCES

1. Tofts P. *Quantitative MRI of the Brain*. Chichester, UK: John Wiley & Sons Ltd; 2003.
2. Deoni SC. Quantitative relaxometry of the brain. *Top Magn Reson Imaging*. 2010;21(2):101-113.
3. Meiboom S, Gill D. Modified spin-echo method for measuring nuclear relaxation times. *Rev Sci Instrum*. 1958;29(8):688-691.
4. Look DC, Locker DR. Time saving in measurement of NMR and EPR relaxation times. *Rev Sci Instrum*. 1970;41(2):250-251.
5. Deoni SC, Rutt BK, Peters TM. Rapid combined T1 and T2 mapping using gradient recalled acquisition in the steady state. *Magn Reson Med*. 2003;49(3):515-526.
6. Deoni SC, Peters TM, Rutt BK. High-resolution T1 and T2 mapping of the brain in a clinically acceptable time with DESPOT1 and DESPOT2. *Magn Reson Med*. 2005;53(1):237-241.
7. Ben-Eliezer N, Sodickson DK, Block KT. Rapid and accurate T2 mapping from multi-spin-echo data using Bloch-simulation-based reconstruction. *Magn Reson Med*. 2015;73(2):809-817.
8. Zhao B, Lam F, Liang ZP. Model-based MR parameter mapping with sparsity constraints: Parameter estimation and performance bounds. *IEEE Trans Med Imaging*. 2014;33(9):1832-1844.
9. Teixeira RPA, Malik SJ, Hajnal JV. Joint system relaxometry (JSR) and Crámer-Rao lower bound optimization of sequence parameters: A framework for enhanced precision of DESPOT T1 and T2 estimation. *Magn Reson Med*. 2018;79(1):234-245.
10. Shcherbakova Y, van den Berg CA, Moonen CT, Bartels LW. PLANET: An ellipse fitting approach for simultaneous T 1 and T 2 mapping using phase-cycled balanced steady-state free precession. *Magn Reson Med*. 2018;79(2):711-722.
11. Ma D, Gulani V, Seiberlich N, et al. Magnetic resonance fingerprinting. *Nat*; 495:187-192.
12. Sbrizzi A, van der Heide O, Cloos M, et al. Fast quantitative MRI as a nonlinear tomography problem. *Magn Reson Imaging*. 2018;46:56-63.
13. Assländer J, Cloos MA, Knoll F, Sodickson DK, Hennig J, Lattanzi R. Low rank alternating direction method of multipliers reconstruction for MR fingerprinting. *Magn Reson Med*. 2018;79(1):83-96.
14. Davies M, Puy G, Vandergheynst P, Wiaux Y. A compressed sensing framework for magnetic resonance fingerprinting. *SIAM J Imaging Sci*. 2014;7(4):2623-2656.
15. Steihaug T. The conjugate gradient method and trust regions in large scale optimization. *SIAM J Numer Anal*. 2005;20(3):626-637.
16. Nocedal J, Wright SJ (Éds.). Numerical Optimization. *Springer Series in Operations Research and Financial Engineering*. 1999. https://doi.org/10.1007/b98874
17. Stolk CC, Sbrizzi A. Understanding the combined effect of k-space undersampling and transient states excitation in MR fingerprinting reconstructions. *IEEE Trans Med Imaging*. 2019;38(10):2445-2455.
18. Stöcker T, Vahedipour K, Pflugfelder D, Shah NJ. High-performance computing MRI simulations. *Magn Reson Med*. 2010;64(1):186-193.
19. Liu F, Velikina JV, Block WF, Kijowski R, Samsonov AA. Fast realistic MRI simulations based on generalized multi-pool exchange tissue model. *IEEE Trans Med Imaging*. 2017;36(2):527-537.
20. Nocedal J. Updating quasi-Newton matrices with limited storage. *Math Comput*. 2006;35(151):773.
21. Scheffler K, Hennig J. Is TrueFISP a gradient-echo or a spin-echo sequence. *Magn Reson Med*. 2003;49(2):395-397.
22. Assländer J, Glaser SJ, Hennig J. Pseudo steady-state free precession for MR-fingerprinting. *Magn Reson Med*. 2017;77(3):1151-1161.
23. Sbrizzi A, Bruijnen T, van der Heide O, Luijten P, van den Berg CAT. Dictionary-free MR Fingerprinting reconstruction of balanced-GRE sequences. 2017. Available from https://arxiv.org/abs/1711.08905
24. Bezanson J, Edelman A, Karpinski S, Shah VB. Julia: A Fresh Approach to Numerical Computing. *SIAM Rev*. 2017;59(1):65-98.
25. van Valenberg W. *MR Pulse Design Through Optimal Control and Model Order Reduction of the Bloch Equations (Master's thesis)*. Utrecht, Netherlands: Utrecht University; 2015. Available from https://dspace.library.uu.nl/handle/1874/316003
26. Wengert RE. A simple automatic derivative evaluation program. *Commun ACM*. 2002;7(8):463-464.

27. Coleman TF, Li Y. An Interior trust region approach for nonlinear minimization subject to bounds. *SIAM J Optim*. 2005;6(2):418-445.

28. Mogensen PK, Riseth AN. Optim: A mathematical optimization package for Julia. *J Open Source Softw*. 2018;3(24):615.

29. Golub G, Pereyra V. Separable nonlinear least squares: the variable projection method and its applications. *Inverse Probl*. 2003;19(2):R1-R26.

30. Aubert-Broche B, Evans AC, Collins L. A new improved version of the realistic digital brain phantom. *NeuroImage*. 2006;32(1):138-145.

31. Lee JH, Hargreaves BA, Hu BS, Nishimura DG. Fast 3D imaging using variable-density spiral trajectories with applications to limb perfusion. *Magn Reson Med*. 2003;50(6):1276-1285.

32. Jiang Y, Ma D, Seiberlich N, Gulani V, Griswold MA. MR fingerprinting using fast imaging with steady state precession (FISP) with spiral readout. *Magn Reson Med*. 2015;74(6):1621-1631.

33. Fessler J, Sutton B. Nonuniform fast Fourier transforms using min-max interpolation. *IEEE Trans Signal Process*. 2003;51(2):560-574.

34. McGivney DF, Pierre E, Ma D, et al. SVD compression for magnetic resonance fingerprinting in the time domain. *IEEE Trans Med Imaging*. 2014;33(12):2311-2322.

35. Zhao B, Setsompop K, Adalsteinsson E, et al. Improved magnetic resonance fingerprinting reconstruction with low-rank and subspace modeling. *Magn Reson Med*. 2018;79(2):933-942.

36. MATLAB. *Version 8.5.0 (R2015a)*. Natick, Massachusetts: The MathWorks Inc; 2015.

37. Buehrer M, Pruessmann KP, Boesiger P, Kozerke S. Array compression for MRI with large coil arrays. *Magn Reson Med*. 2007;57(6):1131-1139.

38. Ravishankar S, Bresler Y. MR image reconstruction from highly undersampled k-Space data by dictionary learning. *IEEE Trans Med Imaging*. 2011;30(5):1028-1041.

39. Coxon R, Brookes MJ, Francis ST, et al. Water proton T1 measurements in brain tissue at 7, 3, and 1.5T using IR-EPI, IR-TSE, and MPRAGE: results and optimization. *Magn Reson Mater Phys Biol Med*. 2008;21(1–2):121-130.

40. Breger RK, Rimm AA, Fischer ME, Papke RA, Haughton VM. T1 and T2 measurements on a 1.5-T commercial MR imager. *Radiol*. 1989;171(1):273-276.

41. Bieri O, Mamisch TC, Trattnig S, Scheffler K. Steady state free precession magnetization transfer imaging. *Magn Reson Med*. 2008;60(5):1261-1266.

42. Pedro Teixeira RA, Malik SJ, Hajnal JV. Fast quantitative MRI using controlled saturation magnetization transfer. *Magn Reson Med*. 2018;81(2):907-920.

43. Bieri O, Markl M, Scheffler K. Analysis and compensation of eddy currents in balanced SSFP. *Magn Reson Med*. 2005;54(1):129-137.

44. Bruijnen T, Stemkens B, Lagendijk JJ, van den Berg CA, Tijssen RH. Prospective GIRF-based RF phase cycling to prevent eddy current-induced steady-state disruption in balanced SSFP imaging. *Magnet Res Med*. 2019;00:1-13.

45. Bernstein MA, King KF, Zhou XJ. *Handbook of MRI Pulse Sequences*. Amsterdam, NL: Elsevier Academic Press; 2004.

46. Koolstra K, Beenakker JM, Koken P, Webb A, Börnert P. Cartesian MR fingerprinting in the eye at 7T using compressed sensing and matrix completionbased reconstructions. *Magn Reson Med*. 2019;81(4):2551-2565.

47. Buonincontri G, Sawiak SJ. MR fingerprinting with simultaneous B1 estimation. *Magn Reson Med*. 2016;76(4):1127-1135.

48. Zhao B, Setsompop K, Ye H, Cauley SF, Wald LL. Maximum likelihood reconstruction for magnetic resonance fingerprinting. *IEEE Trans Med Imaging*. 2016;35(8):1812-1823.

49. Zhao B, Setsompop K, Adalsteinsson E, et al. Improved magnetic resonance fingerprinting reconstruction with low-rank and subspace modeling. *Magn Reson Med*. 2018;79(2):933-942.

50. Wang CY, Coppo S, Mehta BB, Seiberlich N, Yu X. Magnetic resonance fingerprinting with quadratic RF phase for measurement of T2* simultaneously with δf, T1 , and T2. *Magn Reson Med*. 2019;81(3):1849-1862.

51. Xanthis CG, Venetis IE, Chalkias AV, Aletras AH. MRISIMUL: A GPU-based parallel approach to MRI simulations. *IEEE Trans Med Imaging*. 2014;33(3):607-617.

52. Kose R, Kose K. BlochSolver: A GPU-optimized fast 3D MRI simulator for experimentally compatible pulse sequences. *J Magn Reson*. 2017;281:51-65.

53. van der Heide O, Sbrizzi A, Luijten PR, van den Berg CA. Sparse MR-STAT: Order of magnitude acceleration in reconstruction times. In Proceedings of the 27th Annual Meeting of the International Society for Magnetic Resonance in Medicine; 2019 May 11-16; Montreal, CA. Abstract nr 4538.

54. Yang M, Jiang Y, Ma D, Mehta B, Griswold M. Game of learning Bloch equation simulations for MR fingerprinting. In Proceedings of the 26th Annual Meeting of the International Society for Magnetic Resonance in Medicine; 2018 June 16-21; Paris, FA. Abstract nr 0673.

55. van der Heide O, Eijbersen MA, van den Berg CA, Luijten PR, Sbrizzi A. Enhanced MR-STAT by a multi-coil reconstruction framework. In Proceedings of the 27th Annual Meeting of the International Society for Magnetic Resonance in Medicine; 2019 May 11-16; Montreal, CA. Abstract nr 2414.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of the article.