A TEMPORAL MULTISCALE APPROACH FOR TISSUE QUANTIFICATION USING MR FINGERPRINTING ACQUISITIONS

Samuel Cortinhas*, Mohammad Golbabae† and Matthias J. Ehrhardtx

*Department of Mathematical Sciences, †Department of Computer Science, xInstitute for Mathematical Innovation, University of Bath, UK

ABSTRACT

Quantitative MRI (qMRI) is becoming increasingly important for research and clinical applications, however, state-of-the-art reconstruction methods for qMRI are computationally prohibitive. We propose a temporal multiscale approach to reduce computation times in qMRI. Instead of computing exact gradients of the qMRI likelihood, we propose a novel approximation relying on the temporal smoothness of the data. These gradients are then used in a coarse-to-fine (C2F) approach, for example using coordinate descent. The C2F approach was also found to improve the accuracy of solutions, compared to similar methods where no multiscaling was used.

Index Terms— quantitative MRI, MR fingerprinting, nonlinear inverse problems, coordinate descent.

1. INTRODUCTION

Quantitative MRI improves on standard (qualitative) MRI by providing a map of tissue parameters, e.g. proton density ($\rho$), longitudinal ($T_1$) and transverse ($T_2$) relaxation times, and off-resonance frequency ($\omega$). These parameters are related to the physics of nuclear magnetic resonance via the Bloch equations [1]. Quantifying this information has the potential benefits of being more sensitive to early indicators of disease, among others [2].

Magnetic Resonance Fingerprinting (MRF) [3], is a state-of-the-art technology to make multi-parametric qMRI acquisitions fast. The proposed numerical approaches for MRF [3, 4, 5] are mostly based on template-matching signals to a presimulated dictionary of possible solutions discretised over a domain. However, limitations of this approach include memory problems and low accuracy.

Dictionary-free reconstruction using a Bloch response model [6, 7] requires solving a nonlinear fitting problem. For this, computing gradients requires evaluations of the physical forward mapping (i.e. solving the Bloch equations) and their derivatives across a potentially very long temporal sequence, which can severely slow down computations. In this paper, we introduce a multiscale approach across the temporal domain to reduce computation times in qMRI. Furthermore, we present a C2F approach using a variant of coordinate descent.

1.1. MRF Signal model

We consider an Inversion Recovery balance Steady-State Free Precession (IR-bSSFP) excitation pulse sequence [8] of length $L$, with constant repetition time $T_R$ (e.g. 10 ms). The magnetisation $m_l$ after the $l$-th excitation pulse, measured at the middle of the $T_R$ time interval, satisfies the recursive formula

$$m_l(u) = A_l m_{l-1}(u) + b$$

where $u = (T_1, T_2, \omega)$, $A_l = E V_{\omega} R_{\alpha_l} V_{\omega}^T$ and $b = (1 - e^2)m_c$ [6, 8]. In particular, $m_c = (0, 0, 1)^T$, $m_0 = -m_c$, $e_1 = \exp(-T_R/T_1)$, $e_2 = \exp(-T_R/T_2)$, $E = \text{diag}(e_1, e_1, e_2)$. Finally, $(\alpha_l)_{l=1}^L$ denotes the flip angle sequence, $\phi = 2\pi \omega T_R$ and $R_{\alpha_l}, V_{\omega}$ are 3D rotation matrices (see [6]).

We follow the modelling of Dong et al. [6], which consists of the nonlinear equation $Q_l(\rho, u) = y_i$, where $y_i$ denotes the measured $k$-space data and the nonlinear forward operator is defined by

$$Q_l(\rho, u) = P_l F (\rho T_{x,y} m_l(u)),$$

where $m_l$ is defined via (1), $T_{x,y}$ is the transverse projection onto the complex plane, i.e. $T_{x,y} m_l = [m_l]_1 + i[m_l]_2$, $F$ is the Fourier transform and $P_l$ is a temporally-varying $k$-space subsampling operator.

2. PROPOSED MULTISCALE APPROACH

2.1. Multiscale Bloch mapping

Instead of computing the full Bloch response $(m_l)_{l=1}^L$, we seek an efficient method to approximate a subset of this sequence, i.e. $(m_l)_{l \in S}$, with $S \subseteq \{1, 2, ..., L\}$. For convenience, we introduce the notation

$$S_N(\delta) = \{\delta, \delta + N, \delta + 2N, ..., \delta + (\lfloor L/N \rfloor - 1)N\},$$

to denote a uniform grid with increment size $N$ and offset $\delta \in \{1, 2, ..., N\}$.

Note that the dependency of $A_l$ on $l$ (see (1)) only appears as a result of the flip angle sequence $(\alpha_l)_{l=1}^L$. With this in mind, we make the simplifying assumption that $\alpha_l = \bar{\alpha}_i$, $i \in \{l + 1, ..., l + N\}$, with $\bar{\alpha}$ chosen to be the mean of the $\alpha$ over this interval (see fig. 1 for an example). The iteration matrices $A_l$ then lose their dependency on $l$ over each interval, i.e. $A_l = A_i$, $i \in \{l + 1, ..., l + N\}$, and (1) can be simplified as

$$m_{l+N} = A^N m_l + \sum_{k=0}^{N-1} A^k b. \quad (3)$$

For an efficient implementation of (3) we rely on the spectral decomposition of $A$. Its eigenvalues $\lambda_1, \lambda_2, \lambda_3$ and corresponding eigenvectors $v_1, v_2, v_3$ are easily found by observing the similarity transform $A = V_{\omega} (E R_{\bar{\alpha}}) V_{\omega}^{-1}$. We then define $D = \text{diag}(\lambda_1, \lambda_2, \lambda_3)$ and $Z = [v_1, v_2, v_3]$ to write $A = ZDZ^{-1}$. In particular, for $(G_i)_{i=1}^3$ denoting the $3 \times 3$ elementary matrices with a 1 in the $i$-th diagonal slot and 0’s elsewhere, we have

$$A^k = \sum_{i=1}^3 \lambda_i^k U_i \quad (4)$$

where $U_i$ are the eigenvectors of $A$ corresponding to $\lambda_i$. For the update formula (3) we have

$$m_{l+N} = \sum_{i=1}^3 \lambda_i^N U_i$$

arXiv:2010.03386v1 [math.OC] 7 Oct 2020
where $U_i = ZG_i Z^{-1}$ for $i = 1, 2, 3$ and $k \in \mathbb{N}$. Putting (4) into (3) leads to the formula
\[ m_{i+N} = \left( \sum_{i=1}^{3} \lambda_i^N U_i \right) m_i + \left( \sum_{i=1}^{3} \frac{1 - \lambda_i^N}{1 - \lambda_i} U_i \right) b. \]

We denote this multiscale Bloch mapping by $m_S = (m_{iS})_{i \in S}$, which approximates the true Bloch mapping $m = (m_i)_{i=1}^3$.

### 2.2. Multiscale Bloch mapping derivative

From (3), it follows that
\[ m'_{i+N} = (\partial T_1 m_{i+N}, \partial T_2 m_{i+N}, \partial \omega m_{i+N}) \]
using the notation that $\partial \chi m = \partial m / \partial \chi$ for $\chi$ denoting one of the variables ($T_1, T_2, \omega$). In particular,
\[ \partial \chi m_{i+N} = \partial \chi A^N m_i + A^N \partial \chi m_i + \sum_{k=0}^{N-1} \partial \chi A^k b + A^k \partial \chi b. \]

Using the representation of $A$ in (4), we have
\[ \partial \chi A^k = \sum_{i=1}^{3} k \lambda_i^{k-i} \partial \chi \lambda_i U_i + \lambda_i^k \partial \chi U_i, \]
where
\[ \partial \chi U_i = (\partial \chi ZG_i - U_i \partial \chi Z) Z^{-1}. \]

### 2.3. Objective functions

Analogous to (2), we define $Q_{i,S}(\rho, u) = P_{i} F_{i}(\rho T_{x,y} m_{i,S}(u))$ for the multiscale version of the forward operator. To reconstruct the tissue parameters $x = (\rho, u)$, we minimise the multiscale objective function
\[ F_S(x) = \frac{1}{2|S|} \sum_{i \in S} \|Q_{i,S}(x) - y_i\|^2, \]
where $\| \cdot \|$ denotes the usual complex norm. This approximates the true objective function $F$, given by $F = F_{S(1)}$.

The corresponding gradient of (5) is calculated as
\[ \nabla F_S(x) = \frac{1}{|S|} \sum_{i \in S} (Q_i^* S(x))^* [Q_{i,S}(x) - y_i]. \]

3. ALGORITHMS

In this chapter, we discuss algorithms to minimise our objective functions.

#### 3.1. Projected coordinate descent with backtracking

We will need the following backtracking condition
\[ F_S(x^{k+1}) \leq F_S(x^k) + \langle \partial F_S(x^k), \Delta_{i+1} \rangle + \frac{\| \Delta_{i+1} \|^2}{2\tau_i^k} \]
and a projection $\pi_i(x) = \arg \min \{ \| y - x \| : y \in S \}$ (e.g. $\theta = 1$) when applied to tissue parameter maps that take non-negative values (e.g. $\rho, T_1, T_2$) and $\pi_i(x) = x$ otherwise (e.g. $\omega$). Note that $\partial_i F_S(x^k)$ corresponds to the $i$-th component of $\nabla F_S(x^k)$, $(\cdot, \cdot)$ denotes the standard Euclidean inner product and $\Delta_{i+1} = x_{i+1}^k - x_i^k$.  

**Algorithm 1:** Projected coordinate descent with backtracking (PCDB)

```plaintext
input: $x^0 \in \mathbb{R}^{n \times m \times f}$, $\tau^0 \in \mathbb{R}^f$, $N, K \in \mathbb{R}$
for $k = 0$ to $K - 1$ do
    $x_k^{k+1} = x_k^k$;
    $S = S_N (\text{randi}(N))$;
    for $i = 1$ to $f$ do
        for $c = 1$ to $C$ do
            $x_{i,k+1}^k = \pi_i ( x_{i,k}^k - \tau_{i,k} \partial_i F_S(x_{i,k}^k))$;
            if (6) is satisfied then
                $x_{i,k}^k = x_{i,k+1}^k$;
                $\tau_{i,k+1}^k = \eta \tau_{i,k}^k$;
                Break inner loop;
            else
                $\tau_{i,k}^k = \eta \tau_{i,k}^k$;
                if $c = C$ then
                    $x_{i,k+1}^k = x_{i,k}^k$;
        output: $(x^K, \tau^K)$
```

The inputs $x^0$, $\tau^0$, $N$ and $K$ correspond to the initial guess, the initial step size, the grid increment and the number of iterations, respectively. Backtracking involves updating the step size component-wise at each iteration, at most $C$ times (e.g. $C = 50$), via $\eta$ and $\eta$ (e.g. $\eta = 1.2$, $\eta = 0.75$) to ensure monotonic descent of the objective function. Note that randi($N$) represents choosing an integer uniformly at random from the set $\{1, ..., N\}$. 

3.2. Coarse-to-fine

We let \( N \) be a vector of natural numbers of length \( J \), with entries that are strictly decreasing, e.g. \( N = (25, 10, 5, 1) \). The entries in \( N \) specify the increments of uniform grids that are then used in algorithm 1. Moreover, we let \( K \) be another vector of natural numbers of length \( J \), whose entries correspond to the number of iterations we use on each grid, e.g. \( K = (250, 200, 100, 950) \).

Algorithm 2: Coarse-to-fine (C2F)

\[
\begin{align*}
\text{input: } & x \in \mathbb{R}^{n \times n \times l}, \tau \in \mathbb{R}^{l}, N, K \in \mathbb{R}^J \\
& \text{for } j = 1 \text{ to } J \text{ do} \\
& \quad \{ (x, \tau) = \text{PCDB}(x, \tau, N(j), K(j)) \}; \\
\text{output: } & x
\end{align*}
\]

4. NUMERICAL RESULTS

We now present two experiments comparing the C2F algorithm with an analogous approach where no multiscaling is used, i.e. algorithm 1 with \( N = 1 \) (FINE). One experiment uses a constant initial guess, whereas the other uses an initial guess generated by BLIP with a coarse dictionary simulated from the discretisation \( T_1 = \{0.5, 1, \ldots, 6\} \) (s), \( T_2 = \{0.05, 0.1, \ldots, 0.6\} \) (s) and \( \omega = \{-50, -40, \ldots, 50\} \) (Hz). Our tests retrospectively simulate MRF data from the ground truth quantitative parameter maps of a healthy volunteer’s brain, provided in [3]. Images have a spatial resolution of 128x128 pixels. We consider a randomised Cartesian subsampling operator [3], in particular, we use an \( n \)-multishot Echo Planar Imaging (EPI) scheme with random offsets on each frame. A subsampling rate of 1/8 is used throughout the tests presented.

Fig. 2 displays the true objective function \( F \) evaluated at each iterate \( x^k \) in the two experiments. Note that the objective graphs corresponding to the C2F algorithm are scaled (in the iteration axis) by \( 1/N \), where \( N \) is the increment of the uniform grid, with vertical partitions indicating positions of refinement. This gives a measure of the relative computational cost between the two methods and is chosen so that the total number of gradients calculated in both algorithms is the same. Fig. 3 then displays the reconstructed tissue parameter maps (odd columns) with the corresponding error maps (even columns).

We observe that C2F successfully minimises the true objective function at a faster rate than FINE in both experiments. This is also reflected in tables 1 and 2, which give the Peak Signal-to-Noise Ratio (PSNR) and the Mean Absolute Percentage Error (MAPE) of final reconstructions parameter-wise. We see that C2F results in consistently lower percentage errors and generally higher PSNR values, with the latter indicating higher image quality. Note that due to the cyclic nature of \( \omega \), pixels that converge to the true value \( \pm \) the period \( (1/T_R = 100 \text{ Hz}) \) are considered accurate since they produce the same Bloch responses (see (1)). MAPE is not given for \( \omega \) because of the \( 0 \)-values in the ground truth. Whilst the accuracy of BLIP is tied to its dictionary size (and is inaccurate here because of this), one can still use BLIP to generate the initial guess for C2F, as we have done, to get superior accuracy without sacrificing memory.

Table 1. PSNR (dB) of reconstructed tissue parameter maps.

| Method   | \( \rho \) | \( T_1 \) | \( T_2 \) | \( \omega \) |
|----------|----------|----------|----------|----------|
| BLIP     | 28.69    | 31.08    | 32.00    | 36.01    |
| FINE     | 32.25    | 21.33    | 28.59    | 20.44    |
| C2F      | 37.63    | 22.86    | 28.32    | 21.87    |
| BLIP+FINE| 50.80    | 35.56    | 33.02    | 64.35    |
| BLIP+C2F | 50.97    | 35.69    | 33.24    | 64.01    |

Table 2. MAPE (%) of reconstructed tissue parameter maps.

| Method   | \( \rho \) | \( T_1 \) | \( T_2 \) | \( \omega \) |
|----------|----------|----------|----------|----------|
| BLIP     | 14.97    | 45.62    | 23.73    |
| FINE     | 11.87    | 162.79   | 30.01    |
| C2F      | 9.67     | 122.33   | 15.29    |
| BLIP+FINE| 2.72     | 2.69     | 3.85     |
| BLIP+C2F | 2.56     | 2.51     | 3.54     |

Fig. 2. True objective comparison of FINE with C2F (left), and BLIP+FINE with BLIP+C2F (right). On the left, \( x^0 \) consists of four constant images taking values \((\rho, T_1, T_2, \omega) = (0.42, 2, 0.2, 0)\) (see fig. 3 for units), whereas on the right \( x^0 \) is generated by BLIP with a coarse dictionary (see fig. 3, row 1). In both, the initial step size is \( \tau^0 = (0.1, 1, 0.1, 10^{-4}) \). The remaining inputs are \( N = 1, K = 1000 \) for FINE and BLIP+FINE, \( N = (16, 8, 4, 2, 1), K = (320, 320, 160, 100, 850) \) for C2F, and \( N = (8, 4, 2, 1), K = (80, 400, 300, 740) \) for BLIP+C2F.

5. COMPLIANCE WITH ETHICAL STANDARDS

This is a numerical simulation study for which no ethical approval was required.
Fig. 3. Comparison of FINE with C2F using a constant initial guess (rows 2 and 3) and a good initial guess (rows 4 and 5) generated by BLIP with a coarse dictionary (row 1).

7. ACKNOWLEDGEMENTS

S. Cortinhas acknowledges support from the Institute for Mathematical Innovation and the London Mathematical Society for undergraduate research bursaries. M. J. Ehrhardt acknowledges support from the EPSRC (EP/S026045/1, EP/T026693/1), the Faraday Institution (EP/T007745/1) and the Leverhulme Trust (ECF-2019-478).

8. REFERENCES

[1] F. Bloch, “Nuclear induction,” Physical Review, vol. 70, no. 7-8, pp. 460–474, 1946.

[2] D. Ma, V. Gulani, N. Seiberlich, K. Liu, J. L. Sunshine, J. L. Duerk, and M. A. Griswold, “Magnetic resonance fingerprinting,” Nature, vol. 495, no. 7440, pp. 187–192, 2013.

[3] M. Davies, G. Puy, P. Vandergheynst, and Y. Wiaux, “A compressed sensing framework for magnetic resonance fingerprinting,” SIAM Journal on Imaging Sciences, vol. 7, no. 4, pp. 2623–2656, 2014.

[4] D. F. McGivney, E. Pierre, D. Ma, Y. Jiang, H. Saybasili, V. Gulani, and M. A. Griswold, “Svd compression for magnetic resonance fingerprinting in the time domain,” IEEE Transactions on Medical Imaging, vol. 33, no. 12, pp. 2311–2322, 2014.

[5] J. Assländer, M. A. Cloos, F. Knoll, D. K. Sodickson, J. Henning, and R. Lattanzi, “Low rank alternating direction method of multipliers reconstruction for mr fingerprinting,” Magnetic Resonance in Medicine, vol. 79, no. 1, pp. 83–96, 2018.

[6] G. Dong, M. Hintermüller, and K. Papafitsoros, “Quantitative magnetic resonance imaging: From fingerprinting to integrated physics-based models,” SIAM Journal on Imaging Sciences, vol. 12, no. 2, pp. 927–971, 2019.

[7] A. Sbrizzi, O. van der Heide, M. Cloos, A. van der Toorn, H. Hoogduin, P. R. Luijten, and C. A. T. van den Berg, “Fast quantitative mri as a nonlinear tomography problem,” Magnetic Resonance Imaging, vol. 46, pp. 56–63, 2018.

[8] K. Scheffler, “A pictorial description of steady-states in rapid magnetic resonance imaging,” Concepts in Magnetic Resonance: An Educational Journal, vol. 11, no. 5, pp. 291–304, 1999.

[9] M. Golbabaeae, Z. Chen, Y. Wiaux, and M. E. Davies, “Cover tree compressed sensing for fast mr fingerprint recovery,” in 2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP). IEEE, 2017, pp. 1–6.