

Fast Parallel MR Image Reconstruction via B1-based, Adaptive Restart, Iterative Soft Thresholding Algorithms (BARISTA)

Matthew J. Muckley*, *Student Member, IEEE*, Douglas C. Noll, *Senior Member, IEEE*,
and Jeffrey A. Fessler, *Fellow, IEEE*

Abstract—Sparsity-promoting regularization is useful for combining compressed sensing assumptions with parallel MRI for reducing scan time while preserving image quality. Variable splitting algorithms are the current state-of-the-art algorithms for SENSE-type MR image reconstruction with sparsity-promoting regularization. These methods are very general and have been observed to work with almost any regularizer; however, the tuning of associated convergence parameters is a commonly-cited hindrance in their adoption. Conversely, majorize-minimize algorithms based on a single Lipschitz constant have been observed to be slow in shift-variant applications such as SENSE-type MR image reconstruction since the associated Lipschitz constants are loose bounds for the shift-variant behavior. This paper bridges the gap between the Lipschitz constant and the shift-variant aspects of SENSE-type MR imaging by introducing majorizing matrices in the range of the regularizer matrix. The proposed majorize-minimize methods (called BARISTA) converge faster than state-of-the-art variable splitting algorithms when combined with momentum acceleration and adaptive momentum restarting. Furthermore, the tuning parameters associated with the proposed methods are unitless convergence tolerances that are easier to choose than the constraint penalty parameters required by variable splitting algorithms.

Index Terms—MR Image Reconstruction, Compressed Sensing, FISTA, Majorize-Minimize, Parallel MRI

I. INTRODUCTION

MAGNETIC resonance imaging (MRI) is an imaging modality where improving the resolution requires increased acquisition time. As a result, the cost of MRI also increases with higher resolution since the cost is directly proportional to the scan time. In addition to reducing the cost of high-resolution MRI, scanning time reductions can also help accommodate pediatric and elderly patients that have difficulty remaining motionless during long scans. Such scan time reductions are facilitated by combining undersampling and

advanced signal processing methods to remove the associated aliasing artifacts. SENSitivity Encoding (SENSE) is an MRI technique that undoes aliasing effects caused by undersampling by exploiting variations in the sensitivity profiles (i.e., B1 maps) of multiple coils placed around the patient [1]. When the image can reasonably be assumed to be sparse in some transform domain, compressed sensing techniques can be applied to facilitate further accelerations [2].

Image estimation that leverages SENSE MRI and compressed sensing assumptions can be mathematically formulated as an ℓ_1 -regularized optimization problem [3]. Since the ℓ_1 term is nondifferentiable, these problems are difficult to minimize using standard gradient-based methods. Some methods convert such problems into a different form where fast minimization techniques can be applied. One such class includes variable splitting algorithms, where one forms a constrained optimization problem and then proceeds within the augmented Lagrangian formalism to find the solution to the original ℓ_1 -regularized problem [4]–[7]. A difficulty with applying these methods is the tuning of a constraint penalty parameter that heavily affects convergence speed. Sufficient conditions for optimally choosing these parameters are unknown, so current practice is to resort to heuristics for setting these parameters [7]. Since the optimal parameter can change from problem to problem (i.e., patient to patient), robust performance of these methods can be difficult to ascertain.

Alternatives to variable splitting methods are majorize-minimize methods such as fast iterative soft thresholding (FISTA) [8]. FISTA methods converge at a rate that depends on the *Lipschitz constant*, a constant that upper bounds the eigenvalues of the Hessian of the data fit term. This constant is on the order of the maximum of the sum of squared absolute values of the sensitivity coils. As such, the Lipschitz constant can be very loose for low signal regions that occur at the center of the object in SENSE MRI. As a result, majorize-minimize methods such as FISTA have performed poorly relative to their variable splitting counterparts in MRI applications [7].

We address the looseness of the Lipschitz bound by formulating tighter bounds that vary spatially based on the sensitivity coil profiles. The approach requires finding a diagonal majorizer in the range of the regularizing matrix. In this paper we show that for several regularizers of interest (including orthogonal wavelets, anisotropic total variation, and undecimated Haar wavelets), such diagonal upper bounds are simple to compute and give large accelerations relative to FISTA

Copyright (c) 2010 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

This work was supported by the University of Michigan MCubed program and NIH grants 1P01 CA87634 and R01 NS 058576.

*M. J. Muckley is with the Department of Biomedical Engineering, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: mmuckley@umich.edu)

D. C. Noll is with the Department of Biomedical Engineering, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: dnoll@umich.edu)

J. A. Fessler is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: fessler@umich.edu)

Manuscript received July 25, 2014; revised October 3, 2014, Accepted October 5, 2014. Asterisk indicates corresponding author.

with the Lipschitz constant. When combined with adaptive momentum restarting [9], these methods outperform variable splitting methods in all of these cases. The proposed methods also use parameters in the form of convergence tolerances, but in numerical experiments we found that once a reasonable choice was made for these parameters, further optimization was not necessary.

II. PROBLEM FORMULATION AND GENERAL APPROACH

From compressed sensing theory, one can recover a sparse signal by minimizing a convex cost function with ℓ_1 regularization [3]. Let C denote the number of sensitivity coils, D denote the number of data points, and N denote the number of pixels to be estimated. The ℓ_1 -minimization procedure for parallel MR image reconstruction can be mathematically formulated as

$$\begin{aligned} \hat{\mathbf{x}} &= \underset{\mathbf{x} \in \mathcal{M}}{\operatorname{argmin}} \{f(\mathbf{x}) + \beta R(\mathbf{x})\}, \\ f(\mathbf{x}) &= \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2, R(\mathbf{x}) = \|\mathbf{R}\mathbf{x}\|_1, \mathbf{A} = \mathbf{F}\mathbf{S}, \end{aligned} \quad (1)$$

where $\mathbf{F} \in \mathbb{C}^{D \times CN}$ is a block-diagonal matrix with each block having the same down-sampled DFT operator and $\mathbf{S} \in \mathbb{C}^{CN \times N}$ is a block-column matrix with diagonal blocks. We include a masking set, \mathcal{M} , that constrains elements outside the mask to be zero. We call $f(\mathbf{x})$ the data fit term and $R(\mathbf{x})$ the regularizer. Weighted quadratic data fit terms used for noise correlations between coils can be converted to this unweighted form [7]. The parameter, β , must be selected by the user to balance trade-offs between the data fit term and the regularizer (Monte Carlo techniques have been developed for estimating these parameters that perform well under mean-squared error metrics [10].) Defining $\mathbf{A} = \mathbf{F}\mathbf{S}$, we note that \mathbf{S} gives \mathbf{A} a highly shift-variant nature, a property that we will consider in our algorithm design. \mathbf{R} is a sparsifying transform. If \mathbf{R} is left-invertible (i.e., $\mathbf{NR} = \mathbf{I}$ for some matrix, \mathbf{N}), we say that (1) is a *synthesis* reconstruction problem since we can define $\mathbf{u} = \mathbf{R}\mathbf{x}$ and rewrite (1) as an optimization problem over \mathbf{u} . We assume that $\mathbf{R} \in \mathbb{C}^{M \times N}$. If \mathbf{R} is not left-invertible, then we call (1) an *analysis* reconstruction problem and assume that $\mathbf{R} \in \mathbb{R}^{M \times N}$. This restriction of \mathbf{R} to be real-valued includes important classes of analysis regularizers such as total variation [2] and undecimated Haar wavelets [7]. Each of these regularization forms necessitates different algorithm considerations.

Although solving (1) allows one to obtain high-quality estimates of \mathbf{x} with less data, (1) is typically difficult to minimize. Most methods instead minimize a different problem related to (1). The related problem should be easy to minimize relative to (1), but still offer information relevant to the solution of (1). Two procedures for defining and minimizing related problems are *majorize-minimize* procedures and *variable splitting* procedures. For completeness we note that ‘‘corner rounding’’ has also been proposed for dealing with the nondifferentiability of the ℓ_1 regularizer [2], but this has been found to yield algorithms slower than those of the variable splitting class [7]. Our method is of the majorize-minimize class, but is different from previous majorize-minimize methods in that it carefully

considers any coupling of the structures of \mathbf{A} and \mathbf{R} . We outline the general approach in the following section.

A. Separable Quadratic Surrogates

Majorize-minimize methods work by forming a surrogate cost function (i.e., a majorizer, $\phi_k(\mathbf{x})$) and then minimizing the surrogate each iteration to find the minimizer of the original cost function. Any quadratic of the form $\frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$ can be majorized with a *separable quadratic surrogate* (SQS), a procedure that we briefly review [11], [12]. If a surrogate, $\phi_k(\mathbf{x})$, satisfies the following two conditions, then decreasing the surrogate will decrease the original cost function [13]:

$$f(\mathbf{x}^{(k)}) = \phi_k(\mathbf{x}^{(k)}), \quad (2)$$

$$f(\mathbf{x}) \leq \phi_k(\mathbf{x}). \quad (3)$$

We allow the surrogate to be indexed by k since it can vary with iteration. We form such a surrogate for SENSE MRI by first rewriting $f(\mathbf{x})$ around a current estimate, $\mathbf{x}^{(k)}$, as

$$\begin{aligned} f(\mathbf{x}) &= f(\mathbf{x}^{(k)}) + \Re \left\{ (\mathbf{A}^H (\mathbf{A}\mathbf{x}^{(k)} - \mathbf{y}))^H (\mathbf{x} - \mathbf{x}^{(k)}) \right\} \\ &\quad + \frac{1}{2} (\mathbf{x} - \mathbf{x}^{(k)})^H \mathbf{A}^H \mathbf{A} (\mathbf{x} - \mathbf{x}^{(k)}), \end{aligned} \quad (4)$$

where $\Re \{ \cdot \}$ returns the real part of its argument and \mathbf{A}^H is the Hermitian transpose of \mathbf{A} . If we have $\mathbf{A}^H \mathbf{A} \preceq \mathbf{D}_f \in \mathbb{R}^{N \times N}$ for some diagonal matrix, \mathbf{D}_f (where $\mathbf{M} \succeq \mathbf{0}$ implies that \mathbf{M} is positive semidefinite), we can write

$$\begin{aligned} f(\mathbf{x}) &\leq \phi_k(\mathbf{x}) = f(\mathbf{x}^{(k)}) \\ &\quad + \Re \left\{ (\mathbf{A}^H (\mathbf{A}\mathbf{x}^{(k)} - \mathbf{y}))^H (\mathbf{x} - \mathbf{x}^{(k)}) \right\} \\ &\quad + \frac{1}{2} \left\| \mathbf{x} - \mathbf{x}^{(k)} \right\|_{\mathbf{D}_f}^2 \\ &= \psi(\mathbf{x}, \mathbf{x}^{(k)}) + \zeta, \\ \psi(\mathbf{x}, \mathbf{x}^{(k)}) &:= \frac{1}{2} \left\| \mathbf{x} - (\mathbf{x}^{(k)} - \mathbf{D}_f^{-1} \mathbf{A}^H (\mathbf{A}\mathbf{x}^{(k)} - \mathbf{y})) \right\|_{\mathbf{D}_f}^2, \end{aligned} \quad (5)$$

where ζ is a constant that arises from completing the square and is independent of \mathbf{x} . Decreasing $\psi(\mathbf{x}, \mathbf{x}^{(k)})$ causes $\phi_k(\mathbf{x})$ to decrease by the same amount. Standard majorize-minimize procedures use $\mathbf{D}_f = L\mathbf{I}$, where L is the maximum eigenvalue of $\mathbf{A}^H \mathbf{A}$. We instead use a more general \mathbf{D}_f that is a tighter bound for $\mathbf{A}^H \mathbf{A}$. In the case where $\mathbf{A} = \mathbf{F}\mathbf{S}$, we have $\mathbf{A}^H \mathbf{A} = \mathbf{S}^H \mathbf{F}^H \mathbf{F} \mathbf{S}$. In general $\mathbf{F}^H \mathbf{F} \preceq F\mathbf{I}$, where F is the maximum eigenvalue of $\mathbf{F}^H \mathbf{F}$. In the case of Cartesian MRI with unitary DFT matrices, $F = 1$. One can estimate F offline in the non-Cartesian case via power iteration since it does not depend on the object. Noting this, we have

$$\mathbf{D}_f := \mathbf{F}\mathbf{S}^H \mathbf{S} \succeq \mathbf{A}^H \mathbf{A}, \quad (6)$$

where $\mathbf{S}^H \mathbf{S}$ is a diagonal matrix with the sum of the squared absolute values of the sensitivity coils along its diagonal. We could use \mathbf{D}_f to upper bound any SENSE-type quadratic data fit term with a separable quadratic surrogate. We will use this property in the following sections. Furthermore, \mathbf{D}_f is easy to compute once one has determined the coil sensitivities, and with the recent development of fast algorithms for SENSE map estimation it is quickly available in online settings [14].

B. Proposed Minimization Algorithm

We note through the majorization conditions that solving the following problem will decrease the cost function in (1):

$$\mathbf{x}^{(k+1)} = \underset{\mathbf{x} \in \mathcal{M}}{\operatorname{argmin}} \left\{ \eta(\mathbf{x}, \mathbf{x}^{(k)}) := \psi(\mathbf{x}, \mathbf{x}^{(k)}) + \beta R(\mathbf{x}) \right\}. \quad (7)$$

The minimization problem in (7) is where synthesis and analysis regularizers differ. In the synthesis case, $\eta(\mathbf{x}, \mathbf{x}^{(k)})$ is either fully separable or it can be converted into a fully separable form. When $\eta(\mathbf{x}, \mathbf{x}^{(k)})$ is fully separable, closed-form solutions exist via shrinkage functions. In the analysis case, closed form solutions do not exist and we have to run a few steps of an iterative algorithm to decrease $\eta(\mathbf{x}, \mathbf{x}^{(k)})$. We discuss the synthesis case in detail in Section III and the analysis case in Section IV.

Iteratively applying (7) qualifies as a majorize-minimize procedure, and as such it can be accelerated with momentum techniques [8]. Momentum accelerations can be enhanced with adaptive momentum restarting [9]. This gives a general algorithm, which we call B1-based, Adaptive Restart, Iterative Soft Thresholding Algorithm, or BARISTA, since it has step sizes that depend on the sensitivity or B1 maps. Fig. 1 shows the overall algorithm. Variants of this general form are shown

Fig. 1. BARISTA: B1-based, Adaptive Restart, Iterative Soft Thresholding Algorithm

```

1: initialize  $k = 0, \mathbf{z}^{(0)} = \mathbf{x}^{(0)}, \mathbf{D}_f, \alpha$ 
2: while  $k < K$  do
3:    $\tau^{(k+1)} = (1 + \sqrt{1 + 4(\tau^{(k)})^2})/2$ 
4:    $\mathbf{x}^{(k+1)} = \underset{\mathbf{x} \in \mathcal{M}}{\operatorname{argmin}} \eta(\mathbf{x}, \mathbf{z}^{(k)})$ 
5:    $\kappa = \left\| \underset{\mathbf{x} \in \mathcal{M}}{\mathbf{z}^{(k)} - \mathbf{x}^{(k+1)}} \right\|_2 \left\| \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \right\|_2$ 
6:   if  $\Re \left\{ \left\langle \mathbf{z}^{(k)} - \mathbf{x}^{(k+1)}, \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \right\rangle \right\} > \alpha \kappa$  then
7:      $\mathbf{z}^{(k+1)} = \mathbf{x}^{(k+1)}$ 
8:      $\tau^{(k+1)} = 1$ 
9:   else
10:     $\mathbf{z}^{(k+1)} = \mathbf{x}^{(k+1)} + \frac{\tau^{(k)} - 1}{\tau^{(k+1)}} (\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})$ 
11:   end if
12:    $k = k + 1$ 
13: end while
14:  $\hat{\mathbf{x}} = \mathbf{x}^{(K)}$ 

```

in Fig. 2 for the synthesis case and Fig. 3 for the analysis case. The tracking of the momentum is provided by the $\tau^{(k)}$ parameter and an auxiliary variable, $\mathbf{z}^{(k)}$. If the algorithm takes a step in a certain direction, then $\mathbf{z}^{(k+1)}$ takes a larger step in the same direction where the size is determined by $\tau^{(k)}$. The restart is shown with the “if” statement at the end of an algorithm step. If the cosine of the angle between $\mathbf{x}^{(k+1)} - \mathbf{z}^{(k)}$ and $\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$ is greater than α , then the momentum is wiped away. This helps prevent the generalized gradient and the momentum term from taking the algorithm in opposite directions. In our numerical experiments we found that good values for α are negative and lie near 0; we used $\alpha = -\cos(4\pi/9)$. As stated previously, one challenge is in the minimization of $\eta(\mathbf{x}, \mathbf{z}^{(k)})$. Another associated challenge is the design of matrices similar to \mathbf{D}_f , but in the range of

the regularizer for both synthesis and analysis regularization. The following sections discuss these topics.

III. SYNTHESIS REGULARIZATION

A. Synthesis Algorithm Formulation

We use the term *synthesis regularization* when \mathbf{R} is left-invertible, which allows rewriting the minimization problem in the basis of the regularizer. For notational simplicity in this section, we discuss \mathbf{R} that forms a *tight frame*, i.e., $\mathbf{R}^H \mathbf{R} = \mathbf{I}$. Orthogonal wavelet transforms for SENSE MRI and DFT/DCT regularizers for dynamic MRI are examples of unitary matrices that might be used in synthesis problems in MRI. Defining $\mathbf{u} = \mathbf{R}\mathbf{x}$, we can rewrite (1) as

$$\hat{\mathbf{u}} = \underset{\mathbf{u} \in \mathcal{M}_{\text{synth}}}{\operatorname{argmin}} \left\{ \frac{1}{2} \left\| \mathbf{y} - \mathbf{A}\mathbf{R}^H \mathbf{u} \right\|_2^2 + \beta \left\| \mathbf{u} \right\|_1 \right\}, \quad (8)$$

$$\hat{\mathbf{x}} = \mathbf{R}^H \hat{\mathbf{u}}$$

where $\mathcal{M}_{\text{synth}}$ is a synthesis mask that restricts a subset of the synthesis coefficients to be zero. It is less natural to use a mask for the synthesis approach than for analysis, so if masking is desired we recommend using the algorithm outlined in Section IV. Now, if we find a diagonal matrix, \mathbf{D}_R , such that

$$\mathbf{D}_R \succeq \mathbf{R}\mathbf{D}_f\mathbf{R}^H \quad (9)$$

and defining $\mathbf{B} = \mathbf{A}\mathbf{R}^H$, the surrogate in (7) is

$$\begin{aligned} \eta(\mathbf{u}, \mathbf{u}^{(k)}) &= \frac{1}{2} \left\| \mathbf{u} - (\mathbf{u}^{(k)} - \mathbf{D}_R^{-1} \mathbf{B}^H (\mathbf{B}\mathbf{u}^{(k)} - \mathbf{y})) \right\|_{\mathbf{D}_R}^2 + \beta \left\| \mathbf{u} \right\|_1, \end{aligned} \quad (10)$$

the constrained minimum of which is a shrinkage solution:

$$\begin{aligned} \mathbf{u}^{(k+1)} &= \underset{\mathbf{u} \in \mathcal{M}_{\text{synth}}}{\operatorname{argmin}} \eta(\mathbf{u}, \mathbf{u}^{(k)}) \\ &= P_{\mathcal{M}_{\text{synth}}}(\operatorname{shrink}(\mathbf{b}^{(k)}, \beta \mathbf{d}_R^{-1})), \end{aligned} \quad (11)$$

where $\mathbf{b}^{(k)} = \mathbf{u}^{(k)} - \mathbf{D}_R^{-1} \mathbf{B}^H (\mathbf{B}\mathbf{u}^{(k)} - \mathbf{y})$. \mathbf{d}_R^{-1} is a vector composed of the diagonal elements of \mathbf{D}_R^{-1} , and the shrinkage function is defined as $\operatorname{shrink}(\mathbf{b}, \beta) = \operatorname{diag}\{\operatorname{sign}(b_i)\}(|\mathbf{b}| - \beta)_+$. In this case, $|\cdot|$ denotes the absolute value function, $(\cdot)_+$ sets values less than zero to zero, $\operatorname{diag}\{\cdot\}$ takes the input elements and arranges them as a diagonal matrix, and $\operatorname{sign}(\cdot)$ returns the complex sign of its argument. The $P_{\mathcal{M}_{\text{synth}}}$ operator projects its argument on to the set, $\mathcal{M}_{\text{synth}}$, which in this case corresponds to setting all elements outside the mask to zero. Fig. 2 shows the synthesis version of BARISTA. Our goal now is to design \mathbf{D}_R , which will allow us to take larger step sizes in step 5 of Fig. 2 and apply more aggressive shrinkage in step 6.

B. Diagonal Upper Bounds in Unitary Bases

The challenge in designing \mathbf{D}_R is that it must be constructed in the basis of the regularizer, while \mathbf{D}_f is in the basis of the image. For this purpose we will use Theorem 1, which gives a means of constructing \mathbf{D}_R . Theorem 1 can be applied for any \mathbf{R} , although it is most useful for unitary regularizing matrices that have rows with compact support. We will use it here since orthogonal wavelets fall into this class.

Fig. 2. BARISTA for synthesis

```

1: initialize  $k = 0, \mathbf{u}^{(0)} = \mathbf{R}\mathbf{x}^{(0)}, \mathbf{z}^{(0)} = \mathbf{u}^{(0)}, \mathbf{D}_f, \alpha$ 
2: calculate  $\mathbf{D}_R \succeq \mathbf{R}\mathbf{D}_f\mathbf{R}^H$  according to Theorem 1
3: while  $k < K$  do
4:    $\tau^{(k+1)} = (1 + \sqrt{1 + 4(\tau^{(k)})^2})/2$ 
5:    $\mathbf{b}^{(k)} = \mathbf{z}^{(k)} - \mathbf{D}_R^{-1}\mathbf{B}^H(\mathbf{B}\mathbf{z}^{(k)} - \mathbf{y})$ 
6:    $\mathbf{u}^{(k+1)} = P_{\mathcal{M}_{\text{synth}}}(\text{shrink}(\mathbf{b}^{(k)}, \beta\mathbf{d}_R^{-1}))$ 
7:    $\kappa = \|\mathbf{z}^{(k)} - \mathbf{u}^{(k+1)}\|_2 \|\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\|_2$ 
8:   if  $\Re\{\langle \mathbf{z}^{(k)} - \mathbf{u}^{(k+1)}, \mathbf{u}^{(k+1)} - \mathbf{u}^{(k)} \rangle\} > \alpha\kappa$  then
9:      $\mathbf{z}^{(k+1)} = \mathbf{u}^{(k+1)}$ 
10:     $\tau^{(k+1)} = 1$ 
11:   else
12:      $\mathbf{z}^{(k+1)} = \mathbf{u}^{(k+1)} + \frac{\tau^{(k)} - 1}{\tau^{(k+1)}}(\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)})$ 
13:   end if
14:    $k = k + 1$ 
15: end while
16:  $\hat{\mathbf{x}} = \mathbf{R}^H \mathbf{u}^{(K)}$ 

```

Theorem 1. Let $\mathbf{R} \in \mathbb{C}^{M \times N}$ be any matrix and let $\mathbf{D}_f \in \mathbb{R}^{N \times N}$ be diagonal with diagonal elements $d_{n,f}$. Let \mathbf{r}_m be the m th row of \mathbf{R} and let $\mathcal{S}_m \subset \{1, \dots, N\}$ be the support set for \mathbf{r}_m . Define $t_m = \max_{n \in \mathcal{S}_m} (d_{n,f})$ and $d_{m,R} = \sum_{l=1}^M \min(t_m, t_l) |\langle \mathbf{r}_m, \mathbf{r}_l \rangle|$. Let \mathbf{D}_R be a diagonal matrix with diagonal elements $d_{m,R}$, then $\mathbf{D}_R \succeq \mathbf{R}\mathbf{D}_f\mathbf{R}^H$.

The Appendix shows a proof of Theorem 1. Theorem 1 states that any matrix of the form $\mathbf{R}\mathbf{D}_f\mathbf{R}^H$ can be upper bounded with a diagonal matrix by taking maximums over patches of \mathbf{D}_f and scaling those maximums by sums of inner products. These inner product sums increase as \mathbf{R} becomes less unitary, but in our synthesis case we assume unitary \mathbf{R} so $d_{m,R} = t_m$. We have found that this is an effective majorizing matrix for unitary regularizing matrices, and we used Theorem 1 to design \mathbf{D}_R for orthogonal Haar and Daubechies D4 wavelets in our numerical experiments where we ran the algorithm in Fig. 2.

IV. ANALYSIS REGULARIZATION

A. Analysis Algorithm Formulation

In the analysis setting \mathbf{R} is not left-invertible and we can no longer define $\mathbf{u} = \mathbf{R}\mathbf{x}$ and rewrite (1) as an optimization problem over \mathbf{u} . As such, we must leave (1) in its original form. The forms of \mathbf{R} of this type that are of interest for SENSE MRI include anisotropic total variation regularizers [2] and undecimated wavelets where the approximation coefficients are unregularized [7], [15]. However, we can still form a quadratic surrogate for the data fit term. This gives the *analysis denoising problem*. Since we do not have a closed-form solution of this problem, we run a few iterations of a denoising procedure. Fig. 3 shows the overall analysis algorithm, while the denoising procedure is shown in Fig. 4.

We must decide on a stopping criterion for the iterative algorithm used for the denoising step. Previous methods have used a fixed iteration count for this step [7], [16], but we instead use an $\epsilon^{(k)}$ criterion. When large steps are being taken in the outer iterations, the denoising step only needs to provide

Fig. 3. BARISTA for analysis

```

1: initialize  $k = 0, \mathbf{z}^{(0)} = \mathbf{x}^{(0)}, \mathbf{D}_f, \alpha, \epsilon^{(0)}$ 
2: calculate  $\mathbf{D}_R \succeq \mathbf{R}\mathbf{D}_f^{-1}\mathbf{R}^T$  as outlined in Section IV-C
3: while  $k < K$  do
4:    $\tau^{(k+1)} = (1 + \sqrt{1 + 4(\tau^{(k)})^2})/2$ 
5:   Run Fig. 4 algorithm to  $\epsilon^{(k)}$  convergence to get  $\mathbf{x}^{(k+1)}$ 
6:    $\kappa = \|\mathbf{z}^{(k)} - \mathbf{x}^{(k+1)}\|_2 \|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|_2$ 
7:   if  $\Re\{\langle \mathbf{z}^{(k)} - \mathbf{x}^{(k+1)}, \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \rangle\} > \alpha\kappa$  then
8:      $\mathbf{z}^{(k+1)} = \mathbf{x}^{(k+1)}$ 
9:      $\tau^{(k+1)} = 1$ 
10:   else
11:      $\mathbf{z}^{(k+1)} = \mathbf{x}^{(k+1)} + \frac{\tau^{(k)} - 1}{\tau^{(k+1)}}(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})$ 
12:   end if
13:    $\epsilon^{(k+1)} = \max(\min(\epsilon_{\text{diff}} \frac{\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|_2}{\|\mathbf{x}^{(k)}\|_2}, \epsilon^{(k)}), \epsilon_{\text{min}})$ 
14:    $k = k + 1$ 
15: end while
16:  $\hat{\mathbf{x}} = \mathbf{x}^{(K)}$ 

```

an approximate solution to progress the algorithm, whereas very accurate solutions are beneficial for later iterations where the outer steps are small.

Fig. 3 shows a strategy for choosing $\epsilon^{(k)}$ that was effective in our numerical experiments. $\epsilon^{(k)}$ is chosen to be ϵ_{diff} times the norm-difference of the previous iteration, restricted between the upper and lower bounds of $\epsilon^{(k-1)}$ and ϵ_{min} . We choose ϵ_{diff} to balance the cost of solving the denoising problem and progressing the outer iterations. In all experiments we used $\epsilon_{\text{diff}} = 10^{-1}$. We choose ϵ_{min} based on the precision level of the machine that runs the algorithm; its primary purpose is to prevent the algorithm from stalling as a result of numerical precision. For double precision, we observed that $\epsilon_{\text{min}} = 10^{-12}$ gave agreeable convergence in later iterations. We set $\epsilon^{(0)} = 10^{-1}$. We decrease $\epsilon^{(k)}$ monotonically so that the denoising subproblem is solved more accurately as the algorithm progresses toward the solution.

B. Analysis Denoising Subroutine

We now discuss the so-called *analysis denoising problem* that needs to be solved in step 6 of the algorithm in Fig. 3, which is formulated as follows:

$$\begin{aligned}
 \mathbf{x}^{(k+1)} &= \underset{\mathbf{x} \in \mathcal{M}}{\text{argmin}} \eta(\mathbf{x}, \mathbf{x}^{(k)}), \\
 \eta(\mathbf{x}, \mathbf{x}^{(k)}) &= \frac{1}{2} \left\| \mathbf{b}^{(k)} - \mathbf{x} \right\|_{\mathbf{D}_f}^2 + \beta \|\mathbf{R}\mathbf{x}\|_1, \quad (12) \\
 \mathbf{b}^{(k)} &= \mathbf{x}^{(k)} - \mathbf{D}_f^{-1} \mathbf{A}^H (\mathbf{A}\mathbf{x}^{(k)} - \mathbf{y}).
 \end{aligned}$$

This is equivalent to solving step 4 of the algorithm in Fig. 1. There are many potential approaches to solving this step, including nonlinear CG [2] and split Bregman schemes [4]. As mentioned in the previous section, our goal is to minimize the cost function in (12) to some pre-specified numerical precision. As a result, whatever procedure is chosen should perform well under all numerical precision environments, a property not satisfied by nonlinear CG due to its corner-rounding parameter or split Bregman due to its constraint penalty parameter. Instead, we choose to extend the results in [16] to general ℓ_1

regularizers, adapt it to complex numbers, and reintroduce our diagonal majorizing matrix in the range of the regularizer. This approach meets the numerical precision requirements and gave agreeable convergence speed in numerical experiments. Our derivation requires real-valued \mathbf{R} , which includes interesting classes of anisotropic total variation and undecimated Haar wavelet regularizers. Fig. 4 shows the algorithm that arises from extending the results in [16]. The Appendix derives this algorithm.

Fig. 4. Analysis denoising algorithm

```

1: initialize  $j = 0, \mathbf{q}^{(0)}, \mathbf{v}^{(0)} = \mathbf{v}^{(0)}, \mathbf{D}_f, \mathbf{D}_R, \epsilon, \alpha$ 
2: repeat
3:    $\tau^{(j+1)} = (1 + \sqrt{1 + 4(\tau^{(j)})^2})/2$ 
4:    $\mathbf{x}^{(k,j+1)} = P_{\mathcal{M}}(\mathbf{b}^{(k)} - \beta \mathbf{D}_f^{-1} \mathbf{R}^T \mathbf{v}^{(j)})$ 
5:    $\mathbf{q}^{(j+1)} = P_{\mathcal{P}^M}(\mathbf{v}^{(j)} - \beta^{-1} \mathbf{D}_R^{-1} \mathbf{R} \mathbf{x}^{(k,j+1)})$ 
6:    $\kappa = \|\mathbf{v}^{(k)} - \mathbf{q}^{(k+1)}\|_2 \|\mathbf{q}^{(k+1)} - \mathbf{q}^{(k)}\|_2$ 
7:   if  $\Re\{\langle \mathbf{v}^{(k)} - \mathbf{q}^{(k+1)}, \mathbf{q}^{(k+1)} - \mathbf{q}^{(k)} \rangle\} > \alpha \kappa$  then
8:      $\mathbf{v}^{(k+1)} = \mathbf{q}^{(k+1)}$ 
9:      $\tau^{(k+1)} = 1$ 
10:  else
11:     $\mathbf{v}^{(k+1)} = \mathbf{q}^{(k+1)} + \frac{\tau^{(k)} - 1}{\tau^{(k+1)}} (\mathbf{q}^{(k+1)} - \mathbf{q}^{(k)})$ 
12:  end if
13:   $j = j + 1$ 
14: until  $\frac{\|\mathbf{x}^{(k,j)} - \mathbf{x}^{(k,j-1)}\|_2}{\|\mathbf{x}^{(k,j-1)}\|_2} \leq \epsilon$ 
15:  $\mathbf{x}^{(k+1)} = P_{\mathcal{M}}(\mathbf{b}^{(k)} - \beta \mathbf{D}_f^{-1} \mathbf{R}^T \mathbf{q}^{(j)})$ 

```

The $P_{\mathcal{P}^M}(\cdot)$ operator in Fig. 4 projects its argument on to the set, \mathcal{P}^M , the ℓ_∞ -unit ball. This set arises from our dual formulation discussed in the Appendix. In this case this means that $P_{\mathcal{P}^M}(\cdot)$ examines each element in its input vector and normalizes any elements with an absolute value greater than 1 to an absolute value of 1, preserving the complex sign. For this inner denoising step we include an ϵ stopping criterion, the choice of which as discussed in Section IV depends on the step sizes of the outer iterations of the algorithm in Fig. 3. Although not noted in Fig. 4, we also included a maximum iteration number to prevent the algorithm from stalling. We did not observe that this was necessary in our numerical experiments, but we wanted to ensure stable convergence in a variety of circumstances. We measure the convergence based on $\mathbf{x}^{(k,j+1)}$, which is calculated from the momentum variable $\mathbf{v}^{(j)}$, although the actual convergence would depend on the dual variable, $\mathbf{q}^{(j)}$. This simplification avoids making extra computations each iteration that would be required to estimate \mathbf{x} from $\mathbf{q}^{(j)}$, and with the adaptive restart we expect $\mathbf{v}^{(j)}$ to be a good approximation for $\mathbf{q}^{(j)}$ near the solution. Lastly, we note that we initialized the analysis denoising algorithm with the last value for \mathbf{q} from the previous run of the algorithm. This warm start greatly helps the convergence speed of the analysis denoising subroutine.

The algorithm in Fig. 4 requires computing a \mathbf{D}_R that satisfies the analysis majorizer condition:

$$\mathbf{D}_R \succeq \mathbf{R} \mathbf{D}_f^{-1} \mathbf{R}^T, \quad (13)$$

where \mathbf{R}^T is the transpose of \mathbf{R} . We discuss \mathbf{D}_R for the cases of anisotropic total variation and undecimated Haar wavelets.

C. Diagonal Majorizers for Analysis Regularizers

One could use Theorem 1 to upper bound any matrix, including $\mathbf{R} \mathbf{D}_f^{-1} \mathbf{R}^T$. However, in practice we have found that bound somewhat loose for the analysis regularizing matrices of anisotropic total variation and undecimated Haar wavelets. We discuss tighter bounds for these two cases. For the case of anisotropic total variation, we choose

$$\mathbf{D}_R = \text{diag}\{\text{abs}(\mathbf{R}) \mathbf{D}_f^{-1} \text{abs}(\mathbf{R}^T) \mathbf{1}\}, \quad (14)$$

where $\text{abs}(\cdot)$ returns a matrix that has entries that are the absolute value of the input matrix. This is guaranteed to be a majorizer as it is a modification of the techniques of De Pierro [17], and we have found it to be very tight for anisotropic total variation. Its calculation is also simple.

For the case of undecimated Haar wavelets, we present a different approach that builds on Theorem 1 via the polarization identity. The idea is to split up a non-orthogonal \mathbf{R} into orthogonal pieces for which Theorem 1 will provide tight diagonal majorizers. Consider \mathbf{R} of the form,

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 \\ \vdots \\ \mathbf{R}_Q \end{bmatrix}, \quad (15)$$

where Q is the number of submatrices of \mathbf{R} . Defining $\mathbf{c}_i = \mathbf{D}_f^{-1/2} \mathbf{R}_i^T \mathbf{w}_i$ for any arbitrary vector $\mathbf{w}^H = [\mathbf{w}_1^H, \dots, \mathbf{w}_Q^H]$ (possibly different sized \mathbf{w}_i), we then have

$$\begin{aligned} \mathbf{w}^H \mathbf{R} \mathbf{D}_f^{-1} \mathbf{R}^T \mathbf{w} &= \sum_{i=1}^Q \|\mathbf{c}_i\|_2^2 + \sum_{i=1}^Q \sum_{j=i+1}^Q 2\Re\{\langle \mathbf{c}_i, \mathbf{c}_j \rangle\} \\ &\leq \sum_{i=1}^Q \|\mathbf{c}_i\|_2^2 + (Q-1) \sum_{i=1}^Q \|\mathbf{c}_i\|_2^2 \\ &= \sum_{i=1}^Q Q \mathbf{w}_i^H \mathbf{R}_i \mathbf{D}_f^{-1} \mathbf{R}_i^T \mathbf{w}_i, \end{aligned} \quad (16)$$

where one proceeds from the first to the second step by applying the polarization identity,

$$2\Re\{\langle \mathbf{c}_i, \mathbf{c}_j \rangle\} \leq \|\mathbf{c}_i\|_2^2 + \|\mathbf{c}_j\|_2^2, \quad (17)$$

and collecting all inner product pairs. Thus, finding a majorizer for each $\mathbf{R}_i \mathbf{D}_f^{-1} \mathbf{R}_i^T$ submatrix will provide a majorizer for $\mathbf{R} \mathbf{D}_f^{-1} \mathbf{R}^T$. Such a structure applies to the 2-level undecimated Haar wavelet case since a 2-level undecimated Haar wavelet can be written as a cascade:

$$\mathbf{R} = \mathbf{R}_B \mathbf{R}_A. \quad (18)$$

In this case the first step of the cascade, \mathbf{R}_A , can be broken up into pieces:

$$\mathbf{R}_A = \begin{bmatrix} \mathbf{R}_{A,1} \\ \mathbf{R}_{A,2} \\ \mathbf{R}_{A,3} \\ \mathbf{R}_{A,4} \end{bmatrix}, \quad (19)$$

where each of the $\mathbf{R}_{A,i}$ is an orthogonal 1-level Haar wavelet transform. We apply the inequality in (16) to majorize $\mathbf{R}_A \mathbf{D}_f^{-1} \mathbf{R}_A^T$ while using Theorem 1 to majorize each

$\mathbf{R}_{A,i}\mathbf{D}_f^{-1}\mathbf{R}_{A,i}^T$ term. We applied the procedure recursively to \mathbf{R}_B since it has a similar structure. In the undecimated Haar wavelet case each of the $\mathbf{R}_{A,1}, \dots, \mathbf{R}_{A,4}$ is a similar operation, so we expect that each of the \mathbf{c}_i will be approximately linearly dependent and this inequality approach will be fairly tight.

V. EXPERIMENTS

A. Experimental Setup

In the interest of reproducible research, MATLAB code for implementing these methods will be uploaded to the Image Reconstruction Toolbox at web.eecs.umich.edu/~fessler/. All experiments were run on a machine with an Intel Xeon E31230 Processor that had four cores with each core running at 3.2 GHz. The machine had 16 GB of memory. All experiments used $\alpha = -\cos(4\pi/9)$.

We compared the convergence speed of BARISTA to state-of-the-art variable splitting methods in several experiments on four data sets. We present *in vivo* brain results in the main paper body and include results for a numerical brain phantom, a breast phantom, and an American College of Radiology phantom in the supplementary material. The variable splitting methods were each of the AL-P1 or split Bregman type [7]. The AL-P2 method in [7] uses condition number heuristics to tune AL penalty parameters, but we found that these condition number heuristics could change between different regularizers. Tuning AL-P2 for each regularizer would have required setting multiple condition number parameters. AL-P1 has only one constraint penalty parameter, μ , and it had comparable speed to AL-P2, so we used AL-P1 with careful manual tuning of μ as a representative of AL-based methods. We also investigated dynamically updating the μ parameter using update rules and parameters proposed by Boyd (Section 3.4.1 of [18]), which helps mitigate tuning difficulties. We initialized such AL methods with dynamic μ updates with one of the manually-tuned μ values. In the plots this method with dynamic μ updates is denoted as “AL, dynamic μ .” We also introduced a diagonal preconditioner for the conjugate gradient (CG) subroutine in step 4 of AL-P1. We used $\mathbf{P} = (\mathbf{S}^H\mathbf{S} + \mu\mathbf{I})^{-1}$ for all wavelet regularizers and $\mathbf{P} = (\mathbf{S}^H\mathbf{S})^{-1}$ for the total variation regularizer. These preconditioners were not mentioned in [7], but we observed that they accelerated AL-P1 on the order of 50% time to reach the same point of convergence. The AL-P1 methods all used 5 preconditioned CG (PCG) iterations for step 4 of AL-P1.

To track convergence, we computed the following normalized residual as a function of iteration:

$$\xi(k) = 20 \log_{10} \left(\frac{\|\mathbf{x}^{(k)} - \mathbf{x}^{(\infty)}\|_2}{\|\mathbf{x}^{(\infty)}\|_2} \right), \quad (20)$$

where $\mathbf{x}^{(\infty)}$ is a “converged” solution obtained by running many thousands of iterations of AL-P1. Note that even though \mathbf{R} is not full column rank in the total variation case, the AL-P1 method is still convergent [19]. In our convergence plot comparisons to AL-based methods we set the lower bound for $\xi(k)$ at -140 dB. We chose to do this for two reasons: 1) our raw MRI data were less precise than single precision

and 2) BARISTA vastly outperformed all other methods in reaching double precision, so these parts of the plots were less interesting. We also stored the time at which the k th estimate was computed and in our figures we plot $\xi(\cdot)$ as a function of elapsed CPU time instead of iteration. We choose to do this since iterations of the proposed majorize-minimize methods and the variable splitting methods have drastically different compute times due to the PCG subroutine and the analysis denoising step in the proposed methods. All methods used identical subroutines for matrix multiplications.

We selected regularization parameter, β , to give visually appealing solutions for each regularizer. In practice the regularization parameter could be estimated via Monte Carlo SURE methods [10].

In our plots we only show BARISTA from the classes of majorize-minimize algorithms as opposed to other methods such as FISTA with $\mathbf{D}_f = \mathbf{L}\mathbf{I}$ since BARISTA was always the fastest majorize-minimize method. Fig. 5 shows an example of the relative convergence speed of majorize-minimize methods in the case of orthogonal Haar wavelet regularization. In this

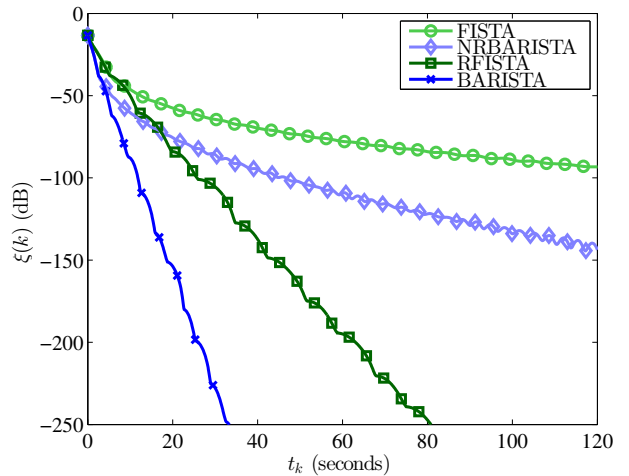


Fig. 5. Comparison of different majorize-minimize methods with orthogonal Haar wavelet regularization. Markers are placed at 50 iteration intervals. FISTA used $\mathbf{D}_f = \mathbf{D}_R = \mathbf{L}\mathbf{I}$ while BARISTA and NRBARISTA (non-restart BARISTA) used the proposed \mathbf{D}_f and \mathbf{D}_R . BARISTA is the fastest method; this was also observed for the other experiments with varying degrees of acceleration. Both restart methods exhibit a stair step pattern, where new “steps” arise when the momentum is restarted.

case, BARISTA was twice as fast as RFISTA (restart FISTA), three times as fast as NRBARISTA, (non-restart BARISTA) and over five times as fast as FISTA in reaching -120 dB. Although RFISTA converges rapidly to double precision, in early iterations it is not competitive with BARISTA or variable splitting methods. In a practical setting, the algorithms may not even be run to convergence, so early-iteration convergence speed is critical for general adoption of the proposed methods. Furthermore, negligible time is required to use the majorizing matrices discussed in this paper, so the factor of two speed-up over RFISTA more or less comes for free. We also observed the factor of two speed-up or greater with the orthogonal Daubechies D4 regularizer and the undecimated Haar wavelet regularizer. Speed-up of BARISTA vs. RFISTA

in the anisotropic total variation case was negligible. We are unsure why this occurred, but it may be that a shift-variant majorizer makes the analysis denoising problem more difficult to solve in the total variation case.

For the *in vivo* experiment, a 3D data set was acquired on a GE 3T scanner with an 8-channel head coil with acquisition parameters $T_R = 25$ ms, $T_E = 5.172$ ms, and voxel size $1 \text{ mm} \times 1.35 \text{ mm} \times 1 \text{ mm}$. The data matrix size was $256 \times 144 \times 128$ uniformly spaced samples. Sensitivity maps were estimated using a quadratic regularized least squares routine [14]. The data were retrospectively undersampled in the Fourier domain using a Poisson disk sampling scheme [20] with a fully sampled center (32-by-32 block), which has been demonstrated to be useful in compressed sensing MRI applications [21]. This sampling pattern simulates one slice of a 3D MRI experiment where the sampling pattern in Fig. 6b is in the phase encode plane [7], [21] (this sampling pattern would be impractical for 2D MRI). Only 20% of the full DFT sampling was used for reconstruction. Fig. 6a shows x

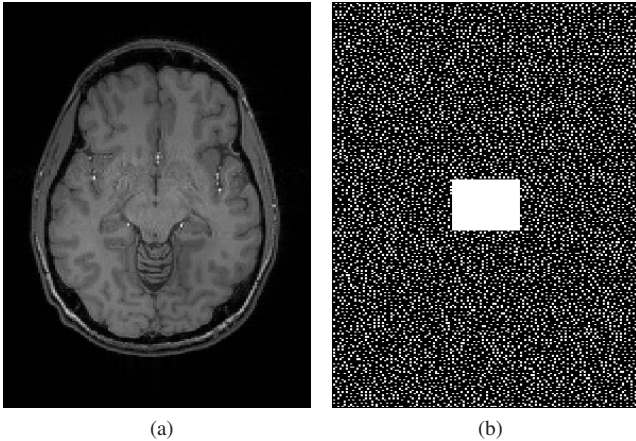


Fig. 6. Images corresponding to the *in vivo* experiments. (a) x estimated from fully sampled data. Some residual noise is present at the center. (b) Sampling pattern for the *in vivo* experiments with a densely sampled 32×32 center.

estimated from fully sampled data, while Fig. 6b shows the Poisson-disc sampling pattern with a densely-sampled center used in all the *in vivo* experiments.

B. Synthesis Regularizer Results

As stated earlier, we performed numerical experiments with orthogonal Haar and Daubechies D4 wavelet regularizers to examine the convergence speed of the proposed method in the synthesis setting. We set the regularization parameter to zero for the approximation coefficients since a sparse model does not fit these coefficients as well as the detail coefficients [15]. Fig. 7a shows an example of the diagonal majorizing elements in the Haar wavelet basis. Fig. 7b shows the majorizer for the Daubechies D4 wavelets. The majorizer for the Daubechies D4 wavelet case is smoother than the Haar case since it requires taking maximums over larger patches.

Fig. 8a and Fig. 8b show the convergence results for Haar and Daubechies D4 wavelets, respectively. BARISTA converges faster than the other methods. The time positions

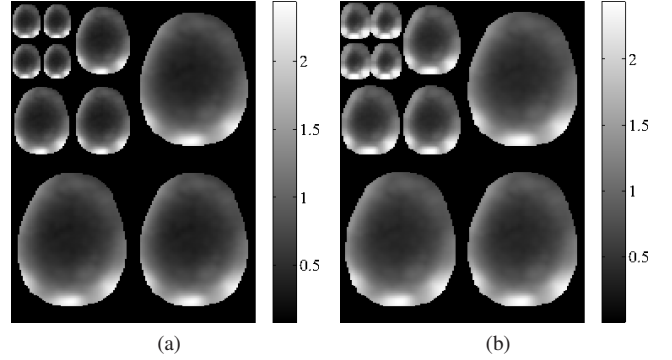


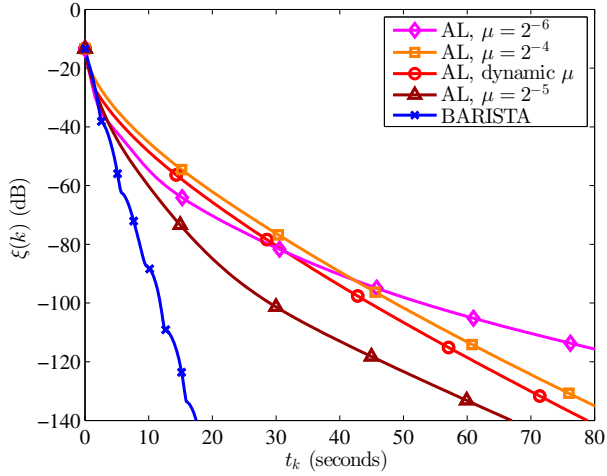
Fig. 7. Examples of diagonal elements of \mathbf{D}_R for synthesis regularizers, rearranged into an image. (a) Elements of the diagonal of \mathbf{D}_R in the Haar wavelet basis. Areas outside the brain have been masked for presentation. (b) Elements of \mathbf{D}_R for the Daubechies D4 basis. Since the rows of a Daubechies D4 matrix have larger support than those of the Haar, the majorizer is smoother. For both cases color bars are shown to give a sense of the variation across the image caused by the sensitivity coils. The Lipschitz constant was 2.98, while the maximum value of the squared absolute values of the sensitivity coils was 3.36. Many of the entries in \mathbf{D}_R are smaller than the Lipschitz constant.

when BARISTA undergoes restart are visible in the stair step pattern in the convergence plots. Several AL parameters are shown to demonstrate the range of speeds of AL-based methods, although we can make no theoretical guarantees on the optimal speed of AL-based methods since we do not know any theoretically optimal way to tune the penalty parameter.

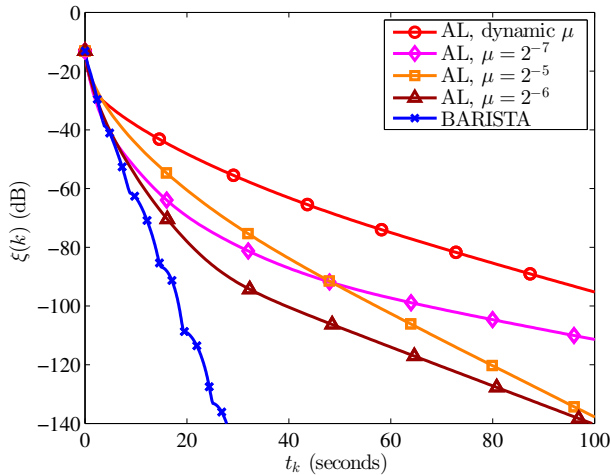
C. Analysis Regularizer Results

We performed numerical experiments with total variation and 2-level undecimated Haar wavelet regularization to examine the convergence speed of the proposed methods in the analysis setting. Our anisotropic total variation implementation took differences in vertical, horizontal, and diagonal directions. We did not regularize the approximation coefficients of the 2-level undecimated Haar wavelet transform [15]. Fig. 9 shows examples of elements from \mathbf{D}_R for the analysis cases. Since for the analysis case we design $\mathbf{D}_R \succeq \mathbf{R}\mathbf{D}_f^{-1}\mathbf{R}^T$, the sensitivity elements are now inverted relative to the synthesis case. Our analysis algorithm formulation required setting the ϵ_{diff} , ϵ_{min} , and $\epsilon^{(0)}$ parameters. We chose $\epsilon_{\text{min}} = 10^{-12}$, $\epsilon_{\text{diff}} = 10^{-1}$, and $\epsilon^{(0)} = 10^{-1}$. We note that although these convergence criteria parameters require some tuning, we were able to use the same convergence criteria for all regularizers in all experiments. Conversely, we had to tune the constraint penalty parameters for the AL-P1 method each time when changing regularizers or data sets.

Fig. 10 shows results for the analysis regularizers. BARISTA matches the other methods in early iterations and outperforms all other methods in later iterations. As previously, the time steps at which the algorithm restarts are shown in the stair step pattern in the convergence plots. We also observed that all algorithms converged slower with the total variation regularizer than the other regularizers. Results with analysis regularizers with an image domain mask were similar and are shown in the supplementary material. Notably, BARISTA



(a) Orthogonal Haar Wavelet Regularization



(b) Orthogonal Daubechies D4 Wavelet Regularization

Fig. 8. Summary of convergence results for two different synthesis regularizers. Markers are placed at 30 iteration intervals for all algorithms. (a) Convergence plot comparing the proposed method to variable splitting methods for orthogonal Haar wavelets. The proposed method with momentum restarting is faster than the other methods. (b) Another convergence plot with orthogonal Daubechies D4 wavelets.

converged about twice as fast when using a mask than without a mask.

VI. DISCUSSION

A. Convergence Speed of BARISTA vs. AL Methods

BARISTA was observed to converge faster than the AL-based methods in both early and late iterations. The early iteration speed of BARISTA is due to its tight approximation of the Hessian of the cost function via the diagonal majorizers developed in this paper and the use of Nesterov momentum acceleration. Nesterov momentum has been added to AL algorithms in some cases [22], although those algorithms require an estimate of the Lipschitz constant, so the diagonal majorizers presented here may be useful for those methods.

The late-iteration speed of BARISTA is due to the use of adaptive restart. We are unaware of a means to apply adaptive

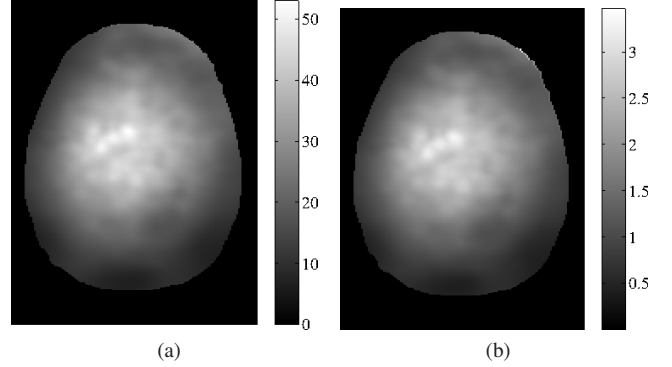


Fig. 9. Examples of diagonal elements of \mathbf{D}_R for analysis regularizers rearranged into an image. (a) A subset of the elements of \mathbf{D}_R for the total variation case with areas outside the brain masked for presentation. Since this matrix must upper bound $\mathbf{R}\mathbf{D}_f^{-1}\mathbf{R}^T$, the sensitivity elements have been inverted. (b) A subset of the elements of \mathbf{D}_R for the undecimated Haar wavelet case.

restart to AL-based methods. We attempted to recover some of the benefits of adaptive restart through the use of dynamic AL parameter updates, but this did not give the same large convergence speed boost as adaptive restart.

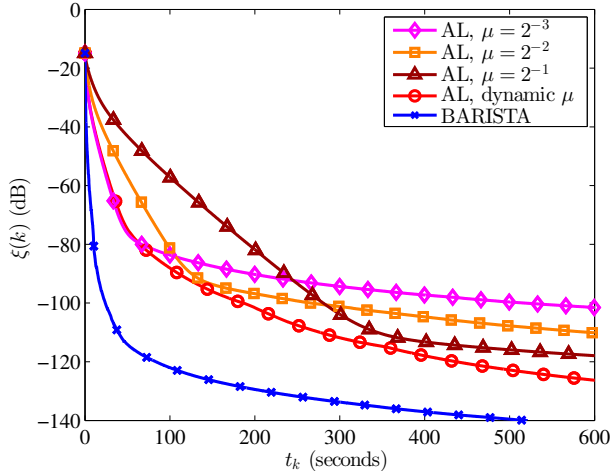
B. Selection of AL Penalty Parameters

We manually optimized the AL penalty parameters for speed. For some cases, such as total variation and 2-level undecimated Haar wavelet regularizers, we observed a trade-off between early and late iteration convergence speed, with smaller parameters favoring early iteration speed and larger parameters favoring late iteration speed. In our tests we chose the small parameters that gave reasonable convergence to -120 dB; however, this behavior suggests that changing the penalty parameter in a dynamic fashion may improve the convergence speed of AL-based methods. The dynamic tuning method from [18] helped in some cases, but not consistently, and we still observed faster convergence with BARISTA in both early and late iterations.

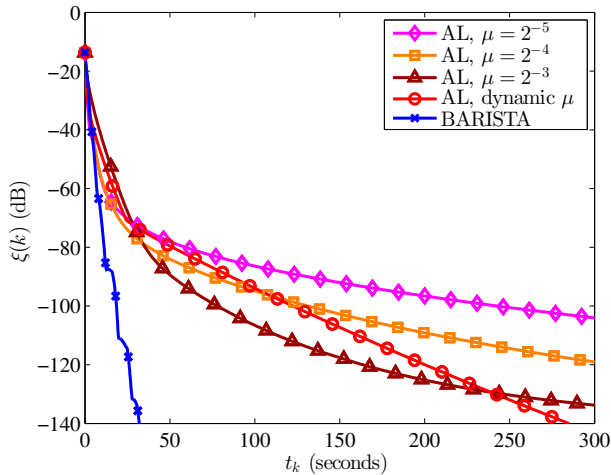
From a theoretical point of view, analysis of AL-based methods considers static penalty parameters [23]. The fact that AL theory considers static penalty parameters is considered one of the primary motivations for using AL methods instead of penalty methods in the first place [7], [23]; adaptively changing the parameter removes this advantage. Conversely, our use of the $\epsilon^{(k)}$ parameter falls within the MFISTA theory provided monotonicity checks are used, and although [16] does not cover the case of adaptive restart we observed stable convergence of BARISTA in all numerical experiments.

C. Surrogate Tightness and Sensitivity Coil Smoothness

In our data set, the sum of squares of absolute values of the sensitivity coils exhibited high variability across the object. As the sensitivity coils become more uniform, our proposed \mathbf{D}_f and \mathbf{D}_R matrices will more closely approximate their Lipschitz counterparts (i.e., $\mathbf{D}_f = \mathbf{L}\mathbf{I}$). In these cases, the advantages of BARISTA will diminish relative to that of RFISTA. However, we typically expect RFISTA to be a



(a) Total Variation Regularization



(b) Undecimated Haar Wavelet Regularization

Fig. 10. Summary of convergence results for two different analysis regularizers. (a) Convergence plot comparing the proposed method to variable splitting methods for total variation regularization. Markers are placed at 100 iteration intervals. The proposed method with momentum restarting is faster than the other methods. (b) Convergence plot comparing BARISTA to variable splitting methods with undecimated Haar wavelet regularization. Markers are placed at 30 iteration intervals. The proposed method with momentum restarting is faster than the other methods, especially in later iterations.

lower bound for the speed of BARISTA. Furthermore, as the sensitivity coils become smoother, the proposed surrogate functions actually become better approximations to the original cost function in (1), so we expect the speed of the proposed methods to be superior with smooth sensitivity coil profiles than our case with large sensitivity coil variability.

D. Tuning the Restart Criterion

The restart criterion in [9] used $\alpha = 0$ in all of their experiments. We found that this choice led to too infrequent restarts because using $\alpha = 0$ allows the momentum and generalized gradient to begin to point in different directions before restarting. Instead, in our experiments we used $\alpha = -\cos(4\pi/9)$. We found that this choice gave very good early-

iteration convergence in 24 numerical experiments with four different regularizers (see supplementary material).

E. Near-monotonicity of BARISTA

In [16] it is stated that when an iterative procedure is applied to minimize the surrogate cost function, one should apply a monotonicity check to ensure stable convergence of the algorithm in the analysis total variation setting. The primary cause of non-monotonicity with FISTA algorithms is when the momentum takes the algorithm in a bad direction near the solution [9]. In our numerical experiments we observed that the combination of the $\epsilon^{(k)}$ parameter and adaptive restart made the monotonicity checks in [16] unnecessary and the proposed method performed as a monotone algorithm. Nonetheless, the monotonicity checks of [16] could be included in a practical setting if monotonicity is still deemed to be an issue.

F. Relations to Proximal Newton Methods

The methods outlined in this paper have some relations to proximal Newton methods (e.g., [24]), which use alternative methods to approximate the Hessian. One issue with such methods is that the memory storage requirements can be undesirably large for medical imaging problems. Low-memory versions of these methods also exist (e.g., L-BFGS, see [24], [25]). BARISTA can also be thought of as having a low-memory approximation to the Hessian due to its diagonal structure, which may be more accurate if the SENSE maps dominate the behavior of the Hessian. Comparisons between our proposed method and these more general proximal Newton methods are an avenue for future investigation. One could even potentially modify BARISTA to use an L-BFGS Hessian approximation update, although proximal Newton methods are often developed for real numbers and may require adaptations for the complex numbers in MRI reconstruction.

VII. CONCLUSION

We have introduced generalizations of the FISTA algorithm, which we call BARISTA, for SENSE-type MR imaging with compressed sensing regularizers that compensate for the shift-variant aspects of the sensitivity coils. The methods gave superior convergence speed relative to state-of-the-art variable splitting methods in numerical experiments. Furthermore, the proposed methods avoid the penalty parameter tuning associated with variable splitting methods, instead relying on unitless convergence tolerance parameters. We have provided heuristics for selecting these parameters and found that the same values worked well across 24 numerical experiments conducted with four different regularizers on four different data sets. We expect that the proposed methods will give fast, high-quality reconstructions across a wide variety of data sets and will aid in the adoption of compressed sensing methods in a clinical setting.

APPENDIX

A. Proof of Theorem 1

To prove Theorem 1 we first define $\mathbf{R}^H = [\mathbf{r}_1, \dots, \mathbf{r}_M]$, where \mathbf{r}_m denotes the m th column of \mathbf{R}^H . We then recognize

that the inner product of two compactly-supported vectors can be computed over either vector's support, i.e.,

$$\langle \mathbf{r}_m, \mathbf{r}_l \rangle = \sum_{n \in \mathcal{S}_m} r_{m,n} r_{l,n}^* = \sum_{n \in \mathcal{S}_l} r_{m,n} r_{l,n}^*. \quad (21)$$

The entries in $\mathbf{V} := \mathbf{R}\mathbf{D}_f\mathbf{R}^H$ are weighted inner products of the form,

$$v_{m,l} = \langle \mathbf{r}_m, \mathbf{r}_l \rangle_{\mathbf{D}_f}, \quad (22)$$

where $v_{m,l}$ is the m th, l th entry of \mathbf{V} . We recall that if $\mathbf{D}_1 \succeq \mathbf{D}_f$, then $\mathbf{R}\mathbf{D}_1\mathbf{R}^H \succeq \mathbf{R}\mathbf{D}_f\mathbf{R}^H$. One such \mathbf{D}_1 is a diagonal matrix where the diagonal entries are defined as

$$d_{1,n} = \begin{cases} t_1, & \text{if } n \in \mathcal{S}_1 \\ d_{n,f}, & \text{otherwise} \end{cases} \quad (23)$$

We also note that finding a $\mathbf{D}_R \succeq \mathbf{R}\mathbf{D}_f\mathbf{R}^H$ is equivalent to finding a \mathbf{D}_R such that $\mathbf{w}^H\mathbf{D}_R\mathbf{w} \geq \mathbf{w}^H\mathbf{R}\mathbf{D}_f\mathbf{R}^H\mathbf{w}$ for any vector \mathbf{w} . To accomplish this, we make the partition, $\mathbf{R}^H = [\mathbf{r}_1, \mathbf{R}_1^H]$, where $\mathbf{R}_1^H = [\mathbf{r}_2, \dots, \mathbf{r}_M]$. We also make the partition, $\mathbf{w} = [w_1, \mathbf{w}_2]$. We now have

$$\begin{aligned} \mathbf{w}^H\mathbf{R}\mathbf{D}_1\mathbf{R}^H\mathbf{w} &= t_1|w_1|^2 \|\mathbf{r}_1\|_2^2 \\ &+ t_1 \sum_{m=2}^M 2\Re\{w_1 w_m^* \langle \mathbf{r}_1, \mathbf{r}_m \rangle\} + \mathbf{w}_2^H \mathbf{R}_1 \mathbf{D}_1 \mathbf{R}_1^H \mathbf{w}_2 \\ &\leq t_1|w_1|^2 \|\mathbf{r}_1\|_2^2 \\ &+ t_1 \sum_{m=2}^M (|w_1|^2 + |w_m|^2) |\langle \mathbf{r}_1, \mathbf{r}_m \rangle| + \mathbf{w}_2^H \mathbf{R}_1 \mathbf{D}_1 \mathbf{R}_1^H \mathbf{w}_2, \end{aligned} \quad (24)$$

which comes from applying (21) and (22). This implies that

$$\begin{aligned} \mathbf{R}\mathbf{D}_f\mathbf{R}^H &\preceq \begin{bmatrix} t_1 \sum_{m=1}^M |\langle \mathbf{r}_1, \mathbf{r}_m \rangle| & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_1 \mathbf{D}_1 \mathbf{R}_1^H \end{bmatrix} \\ &+ \begin{bmatrix} 0 & & & \\ t_1 |\langle \mathbf{r}_1, \mathbf{r}_2 \rangle| & & & \\ & \ddots & & \\ & & & t_1 |\langle \mathbf{r}_1, \mathbf{r}_M \rangle| \end{bmatrix}. \end{aligned} \quad (25)$$

Without loss of generality, we can assume that $t_1 \leq t_2 \leq \dots \leq t_M$. If this is not satisfied, then the appropriate permutation can be applied to \mathbf{V} to make it so. The procedure can again be applied to $\mathbf{R}_1 \mathbf{D}_1 \mathbf{R}_1^H$, and then again recursively. Applying this procedure recursively through M gives Theorem 1.

B. Analysis Denoising Derivation

For the extension of the results in [16], we assume $\mathbf{R} \in \mathbb{R}^{M \times N}$, which includes the classes of total variation and undecimated wavelet regularizers that are of interest to us. The difficulty in minimizing (12) is the fact that \mathbf{R} mixes different elements of \mathbf{x} . To decouple the mixing effects, we will introduce dual variables. Let $\gamma \in \mathbb{R}$ and $\nu \in \mathbb{R}$ be two variables and define $\mathcal{P} = \{(\gamma, \nu) \in \mathbb{R}^2 : \gamma^2 + \nu^2 \leq 1\}$. Then for any $c \in \mathbb{C}$ we have

$$|c| = \max_{(\gamma, \nu) \in \mathcal{P}} \{\gamma \Re\{c\} + \nu \Im\{c\}\}, \quad (26)$$

where $\Im\{\cdot\}$ returns the imaginary part of its argument. Noting this, we now have

$$\|\mathbf{R}\mathbf{x}\|_1 = \max_{(\gamma, \nu) \in \mathcal{P}^M} \{\gamma^T \mathbf{R} \Re\{\mathbf{x}\} + \nu^T \mathbf{R} \Im\{\mathbf{x}\}\}, \quad (27)$$

where \mathcal{P}^M is a Cartesian product of M sets of the form of \mathcal{P} . Note that \mathcal{P}^M is the ℓ_∞ -unit ball in \mathbb{C}^M . We also have

$$\begin{aligned} \mathbf{x}^{(k+1)} &= \operatorname{argmin}_{\mathbf{x} \in \mathcal{M}} \max_{(\gamma, \nu) \in \mathcal{P}^M} \theta_k((\gamma, \nu), \mathbf{x}), \\ \theta_k((\gamma, \nu), \mathbf{x}) &= \frac{1}{2} \left\| \mathbf{b}^{(k)} - \mathbf{x} \right\|_{\mathbf{D}_f}^2 + \beta (\gamma^T \mathbf{R} \Re\{\mathbf{x}\} + \nu^T \mathbf{R} \Im\{\mathbf{x}\}). \end{aligned} \quad (28)$$

To simplify notation, we will now drop the ‘‘arg’’ and implicitly take \mathbf{x} from wherever the critical point of the cost function is. Since \mathcal{M} is a convex set, \mathcal{P}^M is a compact, convex set, and (28) is convex in \mathbf{x} and concave in (γ, ν) , we apply Sion's Theorem [26] to exchange the order of maximization and minimization, which gives

$$\max_{(\gamma, \nu) \in \mathcal{P}^M} \min_{\mathbf{x} \in \mathcal{M}} \theta_k((\gamma, \nu), \mathbf{x}). \quad (29)$$

Now we use the fact that since $\mathbf{D}_f \in \mathbb{R}^{N \times N}$ and is diagonal, the weighted 2-norm squared is separable into its real and imaginary parts, i.e.,

$$\begin{aligned} \left\| \mathbf{b}^{(k)} - \mathbf{x} \right\|_{\mathbf{D}_f}^2 &= \left\| \Re\{\mathbf{b}^{(k)}\} - \Re\{\mathbf{x}\} \right\|_{\mathbf{D}_f}^2 + \left\| \Im\{\mathbf{b}^{(k)}\} - \Im\{\mathbf{x}\} \right\|_{\mathbf{D}_f}^2. \end{aligned} \quad (30)$$

Defining $\mathbf{q} = \gamma + i\nu$, the inner minimization in (29) has a solution where $\mathbf{x} = P_{\mathcal{M}}(\mathbf{b}^{(k)} - \beta \mathbf{D}_f^{-1} \mathbf{R}^T \mathbf{q})$. As stated previously, the $P_{\mathcal{M}}(\cdot)$ operator simply sets elements outside the mask to zero. Plugging this back into (29) reveals a new maximization problem:

$$\begin{aligned} &\max_{\mathbf{q} \in \mathcal{P}^M} \left\{ -\frac{1}{2} \left\| \mathbf{b}^{(k)} - \beta \mathbf{D}_f^{-1} \mathbf{R}^T \mathbf{q} \right\|_{\mathbf{D}_f}^2 \right. \\ &+ \left. \frac{1}{2} \left\| P_{\mathcal{M}}(\mathbf{b}^{(k)} - \beta \mathbf{D}_f^{-1} \mathbf{R}^T \mathbf{q}) - (\mathbf{b}^{(k)} - \beta \mathbf{D}_f^{-1} \mathbf{R}^T \mathbf{q}) \right\|_{\mathbf{D}_f}^2 \right\} \\ &= \min_{\mathbf{q} \in \mathcal{P}^M} \left\{ \frac{1}{2} \left\| \mathbf{b}^{(k)} - \beta \mathbf{D}_f^{-1} \mathbf{R}^T \mathbf{q} \right\|_{\mathbf{D}_f}^2 \right. \\ &- \left. \frac{1}{2} \left\| P_{\mathcal{M}}(\mathbf{b}^{(k)} - \beta \mathbf{D}_f^{-1} \mathbf{R}^T \mathbf{q}) - (\mathbf{b}^{(k)} - \beta \mathbf{D}_f^{-1} \mathbf{R}^T \mathbf{q}) \right\|_{\mathbf{D}_f}^2 \right\} \\ &= \min_{\mathbf{q} \in \mathcal{P}^M} \left\{ \frac{1}{2} \left\| P_{\mathcal{M}}(\mathbf{b}^{(k)} - \beta \mathbf{D}_f^{-1} \mathbf{R}^T \mathbf{q}) \right\|_{\mathbf{D}_f}^2 \right\}. \end{aligned} \quad (31)$$

Since the target cost function is now a constrained minimization over a quadratic, we can once again apply the separable quadratic surrogates techniques outlined in Section II. We choose to do this instead of developing other quadratic minimization routines due to the presence of the constraint. Applying this procedure gives the minimization problem over a surrogate:

$$\begin{aligned} \mathbf{q}^{(j+1)} &= \operatorname{argmin}_{\mathbf{q} \in \mathcal{P}^M} \left\{ \frac{1}{2} \left\| \mathbf{q} - (\mathbf{q}^{(j)} - \beta^{-1} \mathbf{D}_R^{-1} \mathbf{R} \mathbf{x}^{(k,j+1)}) \right\|_{\mathbf{D}_R}^2 \right\} \\ \mathbf{x}^{(k,j+1)} &:= P_{\mathcal{M}}(\mathbf{b}^{(k)} - \beta \mathbf{D}_f^{-1} \mathbf{R}^T \mathbf{q}^{(j)}) \end{aligned} \quad (32)$$

for $\mathbf{D}_R \succeq \mathbf{R}\mathbf{D}_f^{-1}\mathbf{R}^T$. This is obtained by recognizing that the $P_{\mathcal{M}}(\cdot)$ operator where \mathcal{M} is a masking set can be formulated as a projection matrix, \mathbf{M} , where $P_{\mathcal{M}}(\cdot) = \mathbf{M}(\cdot)$. The Hessian in (31) arising from the inclusion of this linear projection matrix is $\beta^2\mathbf{R}\mathbf{D}_f^{-1}\mathbf{M}^T\mathbf{D}_f\mathbf{M}\mathbf{D}_f^{-1}\mathbf{R}^T$, which is upper bounded by $\beta^2\mathbf{R}\mathbf{D}_f^{-1}\mathbf{R}^T$. The majorize-minimize algorithm arising from using this surrogate with momentum acceleration and adaptive momentum restart is shown in Fig. 4.

ACKNOWLEDGMENT

The authors thank Hung Nien for the suggestion to incorporate adaptive restart. We would also like to thank Daniel S. Weller for a useful discussion that led to step 2 of (16). Lastly, we would like to thank Gopal Nataraj, Hao Sun, and Paige Castle for their careful proofreading of the manuscript.

REFERENCES

- [1] K. P. Pruessmann, M. Weiger, M. B. Scheidegger, and P. Boesiger, "SENSE: sensitivity encoding for fast MRI," *Mag. Res. Med.*, vol. 42, no. 5, pp. 952–62, Nov. 1999.
- [2] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Mag. Res. Med.*, vol. 58, no. 6, pp. 1182–95, Dec. 2007.
- [3] E. J. Candès, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Comm. Pure Appl. Math.*, vol. 59, no. 8, pp. 1207–23, 2006.
- [4] T. Goldstein and S. Osher, "The split Bregman method for L1-regularized problems," *SIAM J. Imaging Sci.*, vol. 2, no. 2, pp. 323–43, 2009.
- [5] J. Yang, Y. Zhang, and W. Yin, "A fast alternating direction method for TVL1-L2 signal reconstruction from partial Fourier data," *IEEE J. Sel. Top. Sig. Proc.*, vol. 4, no. 2, pp. 288–97, Apr. 2010.
- [6] M. V. Afonso, José. M. Bioucas-Dias, and Mário. A. T. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Trans. Im. Proc.*, vol. 19, no. 9, pp. 2345–56, Sep. 2010.
- [7] S. Ramani and J. A. Fessler, "Parallel MR image reconstruction using augmented Lagrangian methods," *IEEE Trans. Med. Imag.*, vol. 30, no. 3, pp. 694–706, Mar. 2011.
- [8] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, 2009.
- [9] B. O'Donoghue and E. Candès, "Adaptive restart for accelerated gradient schemes," *Found. Computational Math.*, 2014.
- [10] S. Ramani, Z. Liu, J. Rosen, J.-F. Nielsen, and J. A. Fessler, "Regularization parameter selection for nonlinear iterative image restoration and MRI reconstruction using GCV and SURE-based methods," *IEEE Trans. Im. Proc.*, vol. 21, no. 8, pp. 3659–72, Aug. 2012.
- [11] H. Erdoğan and J. A. Fessler, "Ordered subsets algorithms for transmission tomography," *Phys. Med. Biol.*, vol. 44, no. 11, pp. 2835–51, Nov. 1999.
- [12] D. R. Hunter and K. Lange, "A tutorial on MM algorithms," *American Statistician*, vol. 58, no. 1, pp. 30–7, Feb. 2004.
- [13] M. W. Jacobson and J. A. Fessler, "An expanded theoretical treatment of iteration-dependent majorize-minimize algorithms," *IEEE Trans. Im. Proc.*, vol. 16, no. 10, pp. 2411–22, Oct. 2007.
- [14] M. J. Allison, S. Ramani, and J. A. Fessler, "Accelerated regularized estimation of MR coil sensitivities using augmented Lagrangian methods," *IEEE Trans. Med. Imag.*, vol. 32, no. 3, pp. 556–64, Mar. 2013.
- [15] I. W. Selesnick and M. A. T. Figueiredo, "Signal restoration with overcomplete wavelet transforms: comparison of analysis and synthesis priors," in *Proc. SPIE 7446 Wavelets XIII*, 2009, p. 74460D, wavelets XIII.
- [16] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Trans. Im. Proc.*, vol. 18, no. 11, pp. 2419–34, Nov. 2009.
- [17] A. R. De Pierro, "A modified expectation maximization algorithm for penalized likelihood estimation in emission tomography," *IEEE Trans. Med. Imag.*, vol. 14, no. 1, pp. 132–7, Mar. 1995.
- [18] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. & Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.
- [19] H. Nien and J. A. Fessler, "A convergence proof of the split Bregman method for regularized least-squares problems," 2014, arxiv 1402.4371. [Online]. Available: <http://arxiv.org/abs/1402.4371>
- [20] D. Dunbar and G. Humphreys, "A spatial data structure for fast Poisson-disk sample generation," *ACM Trans. on Graphics*, vol. 25, no. 3, pp. 503–8, Jul. 2006, SIGGRAPH.
- [21] M. Lustig, M. Alley, S. Vasanawala, D. L. Donoho, and J. M. Pauly, "L1 SPIR-iT: Autocalibrating parallel imaging compressed sensing," in *Proc. Intl. Soc. Mag. Res. Med.*, 2009, p. 334.
- [22] Y. Ouyang, Y. Chen, G. Lan, and E. Pasillio, "An accelerated linearized direction method of multipliers," 2014, arxiv 1401.6607. [Online]. Available: <http://arxiv.org/abs/1401.6607>
- [23] J. Eckstein and D. P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, no. 1-3, pp. 293–318, Apr. 1992.
- [24] J. Lee, Y. Sun, and M. Saunders, "Proximal Newton-type methods for convex optimization," in *NIPS*, vol. 25, 2012, pp. 827–35. [Online]. Available: <http://papers.nips.cc/paper/4740-proximal-newton-type-methods-for-convex-optimization>
- [25] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization," *ACM Trans. Math. Software*, vol. 23, no. 4, pp. 550–60, Dec. 1997.
- [26] M. Sion, "On general minimax theorems," *Pacific J. Math.*, vol. 8, no. 1, pp. 171–6, 1958.

BARISTA Supplementary Material

Matthew J. Muckley, Douglas C. Noll, and Jeffrey A. Fessler

October 3, 2014

This report shows convergence results from a numerical brain phantom, a real breast phantom, and an American College of Radiology (ACR) phantom to supplement those shown in “Fast Parallel MR Image Reconstruction via B1-Based, Adaptive Restart, Iterative Soft Thresholding Algorithms (BARISTA),” *IEEE Transactions on Medical Imaging*.

1 Experiment with synthetic data

The synthetic data set was generated by downloading an image from the BrainWeb data base [1] and simulating eight receive sensitivity coils using the methods from [2]. The data were sampled using a Poisson-disc sampling pattern with a densely sampled 32-by-32 center similar to Fig. 6b in the main paper body. Sensitivity coil profiles were estimated from noisy, simulated data using a central 32-by-32 block of kspace via the methods in [3]. About 20% of the fully-sampled data were used for reconstruction. Reconstructions were then run using orthogonal Haar wavelets, orthogonal Daubechies D4 wavelets, anisotropic total variation, and 2-level undecimated Haar wavelets as regularizers. Regularization parameters were chosen to give visually appealing images. Fig. 1 shows the true simulation image and reconstructions using orthogonal Haar (Fig. 1c), orthogonal Daubechies D4 (Fig. 1d), total variation (Fig. 1e), and 2-level undecimated Haar wavelet (Fig. 1f) regularizers. Fig. 1a shows the sum of squared absolute values of the sensitivity coils. We show the sum of squares instead of the square root sum of squares since the sum of squares is more relevant to the behavior of $\mathbf{A}^H \mathbf{A}$. Fig. 2 shows convergence results for the four regularizers on this synthetic data set. We plot the normalized residual, $\xi(k)$, defined as

$$\xi(k) = 20 \log_{10} \left(\frac{\|\mathbf{x}^{(k)} - \mathbf{x}^{(\infty)}\|_2}{\|\mathbf{x}^{(\infty)}\|_2} \right), \quad (1)$$

where $\mathbf{x}^{(\infty)}$ is a “converged solution” obtained by running many thousands of iterations of the AL-based methods.

Figures 2a and 2b are similar to those in Figures 2c and 2d in [4]. Figures 2c and 2d show some acceleration of AL methods with optimal tuning in early iterations with anisotropic total variation and 2-level undecimated Haar wavelet regularizers, but BARISTA reaches single precision faster than the AL-based methods. BARISTA used the exact same parameters as in the *in vivo* experiments in [5], whereas extensive time was spent re-tuning the AL-based methods to achieve the results shown. Of particular interest is the AL method with the dynamic μ updates with parameters from [6], which stalls in early iterations. The AL method with dynamic μ updates was initialized with one of the good, manually-tuned parameters, but the μ update heuristics quickly increased the magnitude of μ . This data set required rather small μ parameters compared to the other data sets, which may be due to the fact that the true image and SENSE maps were orders of magnitude different in intensity. Such issues could be mitigated by always ensuring appropriate normalizations of the SENSE maps and data, but the convergence speed results here also show that BARISTA is robust when such normalizations are not performed.

2 Experiment with breast phantom data

We also obtained a SENSE breast phantom imaging data set for numerical experiments that had been used for a previous paper [3]. The breast phantom consisted of two containers plastered with vegetable shortening and filled with “Super Stuff” bolus material (Radiation Products Design Inc. Albertville, MN). Four surface coils were used in the experiment. The data were sampled using a Poisson-disc sampling pattern similar to that in Fig. 6b in [5] with a densely-sampled 32-by-32 center in the orthogonal wavelets case and a 32-by-16 center in the anisotropic total variation and undecimated Haar wavelets case. The sensitivity coil profiles were estimated using the data from the densely-sampled center using the methods described in [3]. About 20% of the fully-sampled data were used for reconstruction. Fig. 3 shows reconstructed images of real breast phantom data for each of the four regularizers of interest (orthogonal Haar, orthogonal Daubechies D4, anisotropic total variation, and 2-level undecimated Haar wavelets). Figure 4 shows the convergence speed results for the breast phantom imaging data set. Figures 4a and 4b are similar to those in Figures 2c and 2d in [4]. BARISTA was the fastest method for all four of the regularizers of interest for this data set.

3 Experiment with ACR phantom data

We also obtained a new SENSE ACR phantom data set for numerical experiments. The phantom was scanned with a GE MR 750 scanner with a 32-channel receive array (Nova Medical Group). The phantom was scanned with a 256-by-256 grid and then retrospectively downsampled with a Poisson-disc sampling pattern similar to that in Fig. 6b in [5] with a densely-sampled 32-by-32 center. The sensitivity coil profiles were estimated using the data from the densely-sampled center using the methods described in [3]. About 20% of the fully-sampled data were used for reconstruction. Since our previous experiments indicated that BARISTA and AL methods were converging to the same solution, in this case we calculated $\mathbf{x}^{(\infty)}$ with BARISTA to minimize the time to run the numerical experiments. The SENSE maps from this experiment and the reconstructed $\mathbf{x}^{(\infty)}$ for each regularizer are shown in Fig. 5.

Fig. 6 shows convergence speed results for the ACR phantom data set. Figures 6a and 6b are similar to those in Figures 2c and 2d in [4]. BARISTA was the fastest method for all four of the regularizers of interest for this data set.

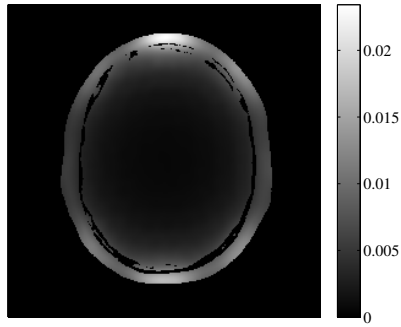
4 Experiments with masking

We also compared convergence speed between BARISTA and the AL-based methods with image-domain masks for total variation and 2-level undecimated Haar wavelet regularizers for each data set. “Converged” images were calculated using BARISTA. All other experimental parameters were the same as run previously. Masks were calculated by first thresholding the $\mathbf{x}^{(\infty)}$ image magnitude at 10% of the maximum magnitude, then performing a 10-by-10 dilation of the mask (20-by-20 dilation for the ACR phantom). Our goal is not to propose a method for doing masked SENSE, but to simply examine algorithm performance with a mask. Examples of image-domain masks for each data set are shown in Fig. 7. Fig. 8 shows convergence results using an image domain mask for each of the four data sets. Figs. 8a and 8b are similar to Fig. 10 in the main paper body. Figs. 8c and 8d are similar to 2c and 2d, respectively. Figs. 8e and 8f are similar to 4c and 4d, respectively. Figs. 8g and 8h are similar to 6c and 6d, respectively. Also of note is that BARISTA converged at least twice as fast when using an image domain mask, so masking appears to be desirable both from an image quality and a convergence speed perspective.

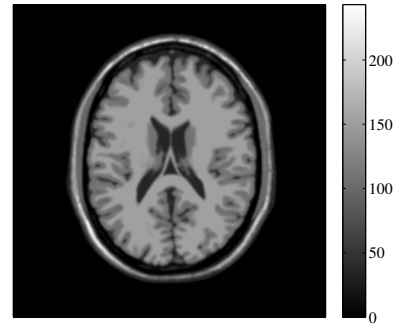
References

- [1] D. L. Collins, A. P. Zijdenbos, V. Kollokian, J. G. Sled, N. J. Kabani, C. J. Holmes, and A. C. Evans, “Design and construction of a realistic digital brain phantom,” *IEEE Trans. Med. Imag.*, vol. 17, pp. 463–8, June 1998.

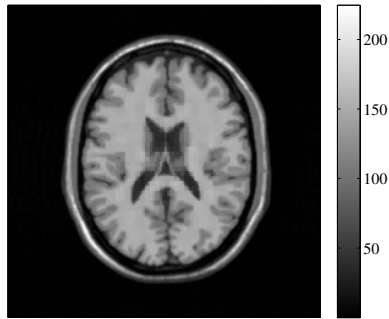
- [2] M. I. Grivich and D. P. Jackson, “The magnetic field of current-carrying polygons: An application of vector field rotations,” *Amer. J. Phys.*, vol. 68, pp. 469–74, May 2000.
- [3] M. J. Allison, S. Ramani, and J. A. Fessler, “Accelerated regularized estimation of MR coil sensitivities using augmented Lagrangian methods,” *IEEE Trans. Med. Imag.*, vol. 32, pp. 556–64, Mar. 2013.
- [4] M. J. Muckley and J. A. Fessler, “Fast MR image reconstruction with orthogonal wavelet regularization via shift-variant shrinkage,” in *Proc. IEEE Intl. Conf. on Image Processing*, 2014. To appear.
- [5] M. J. Muckley, D. C. Noll, and J. A. Fessler, “Fast parallel MR image reconstruction via B1-based, adaptive restart, iterative soft thresholding algorithms (BARISTA),” *IEEE Trans. Med. Imag.*, 2014. Submitted.
- [6] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Found. & Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.



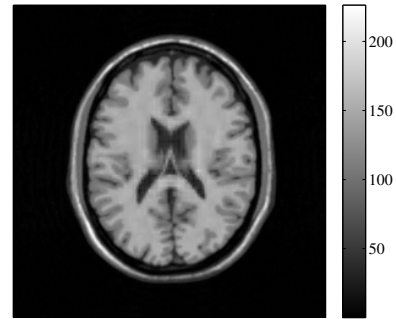
(a) SENSE Map Sum of Squares



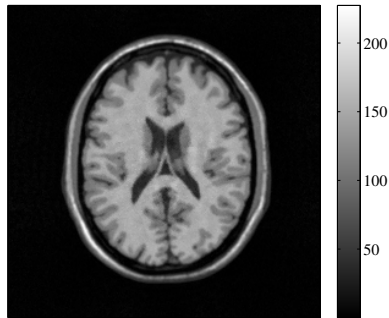
(b) True Image



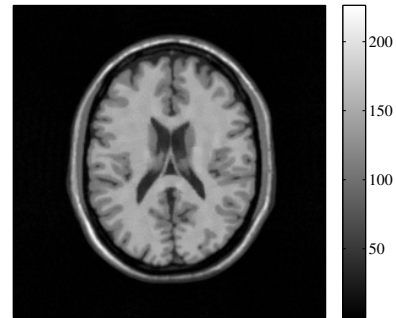
(c) Orthogonal Haar Regularizer



(d) Orthogonal D4 Regularizer

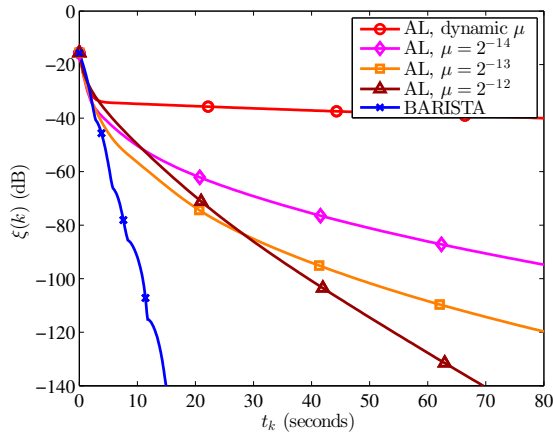


(e) Total Variation Regularizer

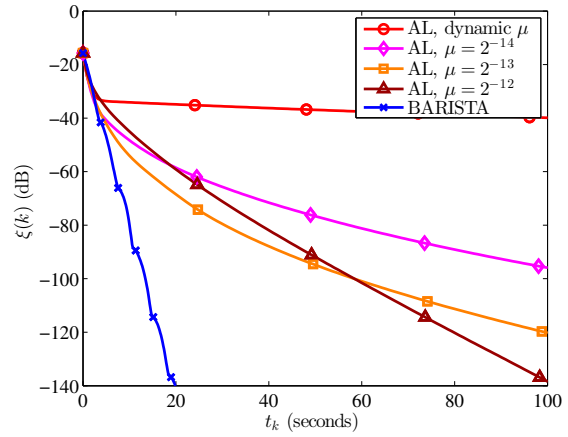


(f) Undecimated Haar Regularizer

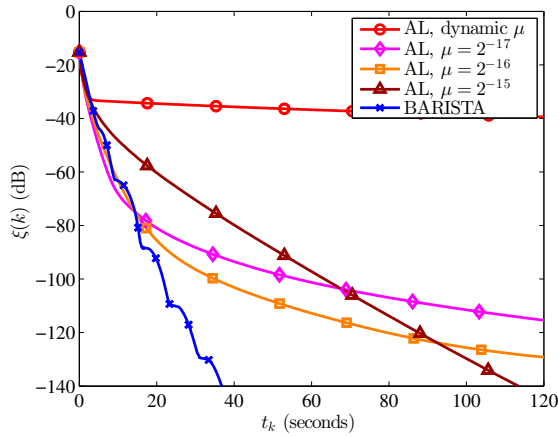
Figure 1: Examples of true and reconstructed images from the simulated data set. (a) Sum of squared absolute values of the estimated sensitivity coil profiles for the numerical phantom experiment with areas outside the brain masked for presentation. (b) The ground truth \mathbf{x} used for the numerical simulations. (c) The reconstructed $\mathbf{x}^{(\infty)}$ using orthogonal Haar wavelet regularization, NRMSE of 2.6%. (d) The reconstructed $\mathbf{x}^{(\infty)}$ using orthogonal Daubechies D4 wavelet regularization, NRMSE of 2.6%. (e) The reconstructed $\mathbf{x}^{(\infty)}$ using total variation regularization, NRMSE of 2.9%. (f) The reconstructed $\mathbf{x}^{(\infty)}$ using 2-level undecimated Haar wavelet regularization, NRMSE of 2.7%.



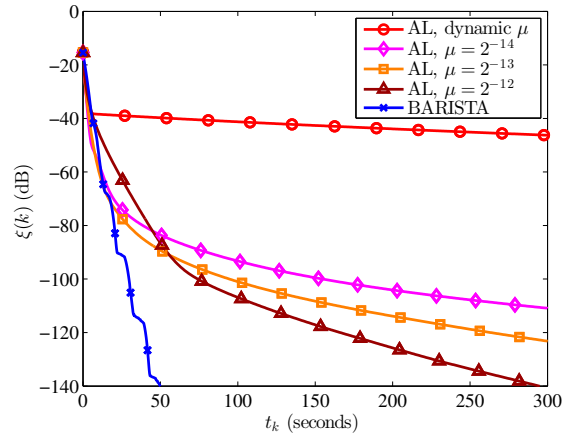
(a) Orthogonal Haar Regularizer



(b) Orthogonal D4 Regularizer

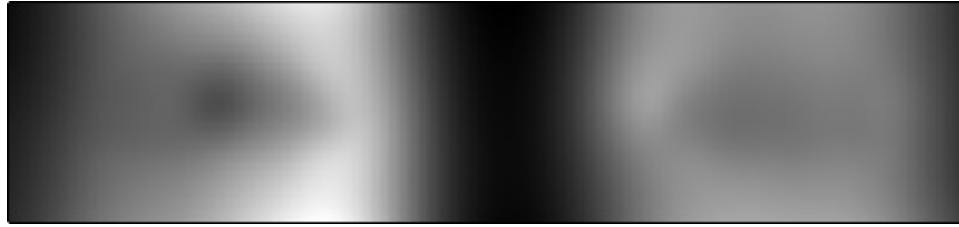


(c) Total Variation Regularizer

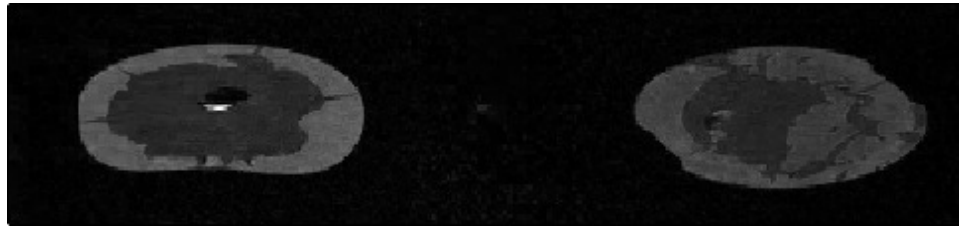


(d) Undecimated Haar Regularizer

Figure 2: Convergence speed results from fully synthetic data. Markers are placed at 30-iteration intervals for all algorithms. (a) Orthogonal Haar wavelet regularizer. (b) Orthogonal Daubechies D4 wavelet regularizer. (c) Anisotropic total variation regularizer. (d) Undecimated Haar wavelet regularizer.



(a) SENSE Map Sum of Squares



(b) Orthogonal Haar Regularizer



(c) Orthogonal D4 Regularizer

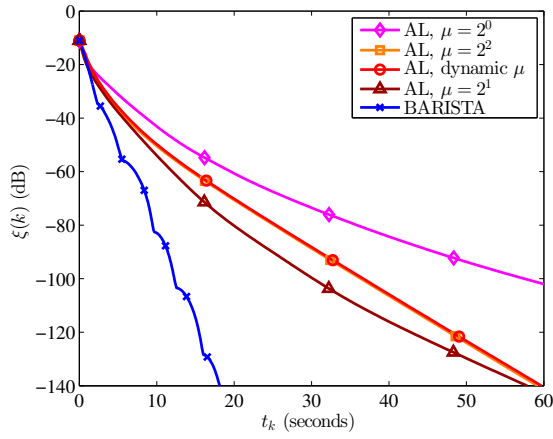


(d) Total Variation Regularizer

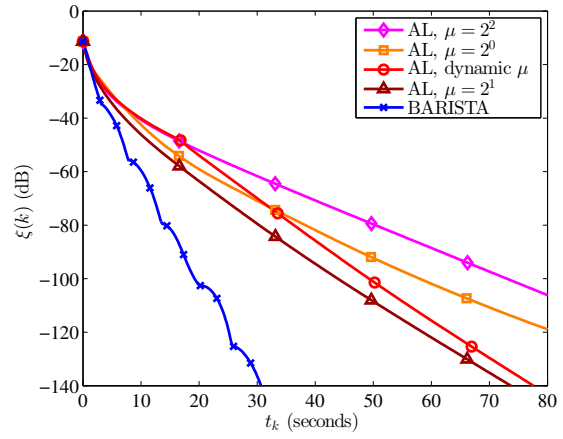


(e) Undecimated Haar Regularizer

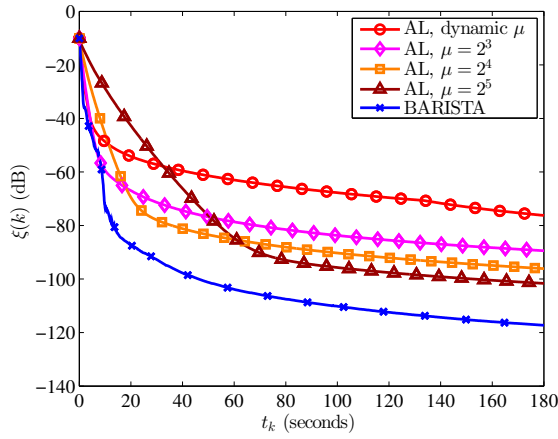
Figure 3: Examples of reconstructed images from the breast phantom imaging data set. (a) Sum of squared absolute values of the estimated sensitivity coil profiles for the real breast phantom. (b) The reconstructed $\mathbf{x}^{(\infty)}$ using orthogonal Haar wavelet regularization. (c) The reconstructed $\mathbf{x}^{(\infty)}$ using orthogonal Daubechies D4 wavelet regularization. (d) The reconstructed $\mathbf{x}^{(\infty)}$ using total variation regularization. (e) The reconstructed $\mathbf{x}^{(\infty)}$ using 2-level undecimated Haar wavelet regularization.



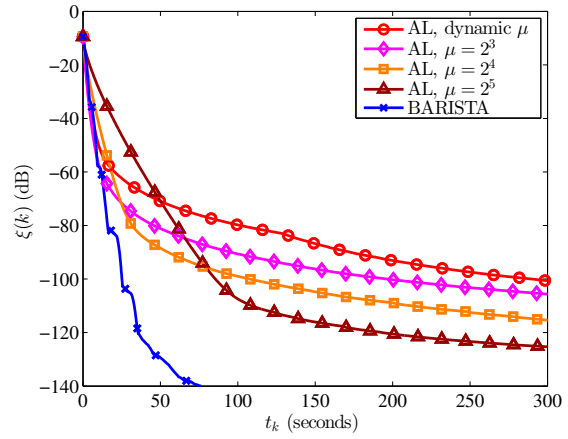
(a) Orthogonal Haar Regularizer



(b) Orthogonal D4 Regularizer

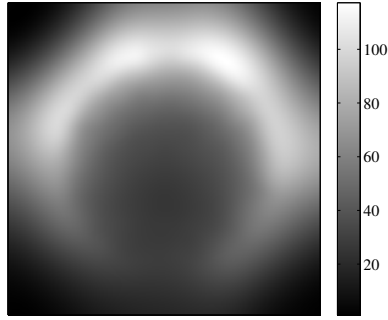


(c) Total Variation Regularizer

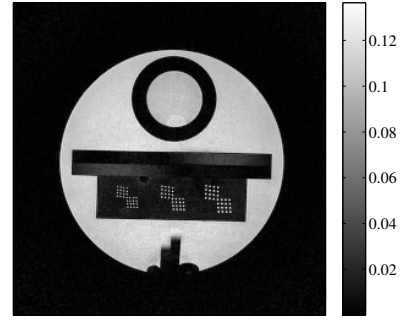


(d) Undecimated Haar Regularizer

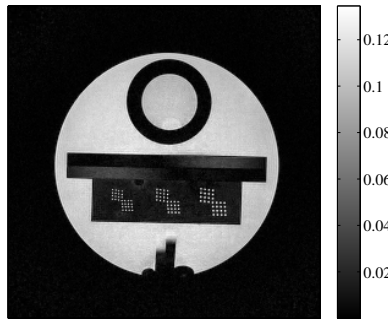
Figure 4: Convergence speed results from breast phantom imaging data. Markers are placed at 50-iteration intervals. (a) Orthogonal Haar wavelet regularizer. (b) Orthogonal Daubechies D4 wavelet regularizer. (c) Anisotropic total variation regularizer. (d) Undecimated Haar wavelet regularizer.



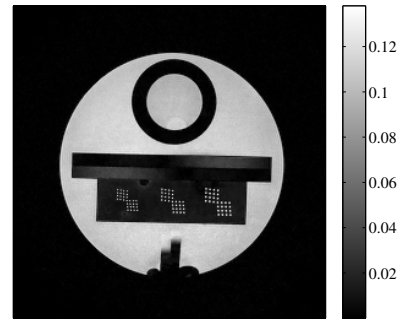
(a) SENSE Map Sum of Squares



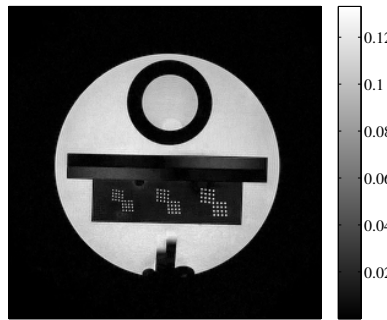
(b) Orthogonal Haar Regularizer



(c) Orthogonal D4 Regularizer

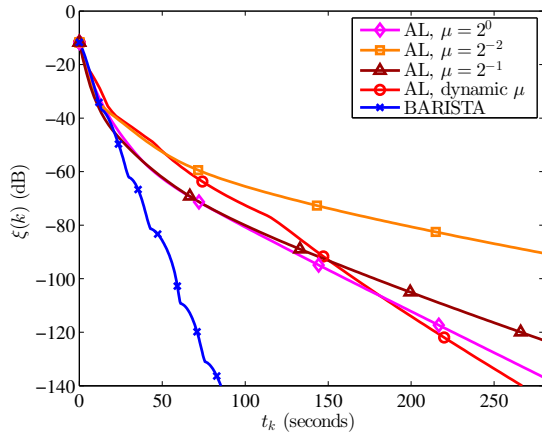


(d) Total Variation Regularizer

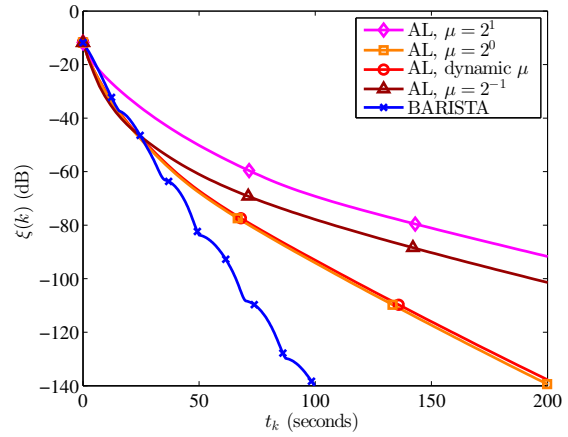


(e) Undecimated Haar Regularizer

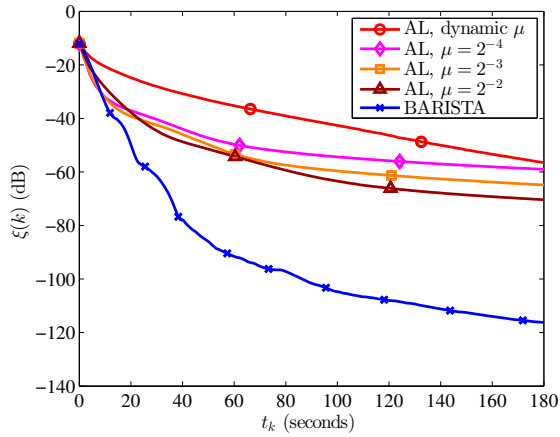
Figure 5: Examples of sum of SENSE maps and reconstructed images from the ACR phantom data set. (a) Sum of squared absolute values of the estimated sensitivity coil profiles for the ACR phantom data set. (c) The reconstructed $\mathbf{x}^{(\infty)}$ using orthogonal Haar wavelet regularization. (d) The reconstructed $\mathbf{x}^{(\infty)}$ using orthogonal Daubechies D4 wavelet regularization. (e) The reconstructed $\mathbf{x}^{(\infty)}$ using total variation regularization. (f) The reconstructed $\mathbf{x}^{(\infty)}$ using 2-level undecimated Haar wavelet regularization.



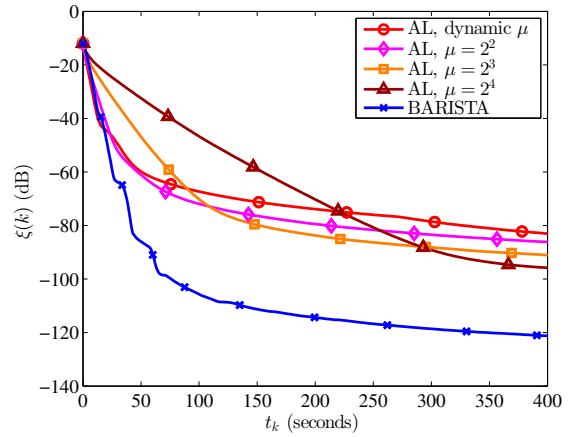
(a) Orthogonal Haar Regularizer



(b) Orthogonal D4 Regularizer

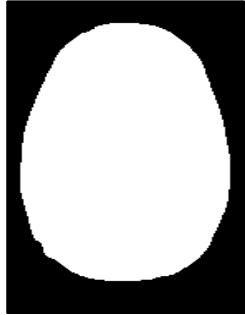


(c) Total Variation Regularizer

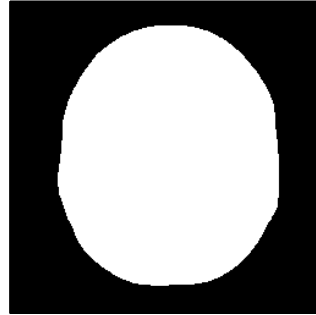


(d) Undecimated Haar Regularizer

Figure 6: Convergence speed results from ACR phantom imaging data. Markers are placed at 30-iteration intervals. (a) Orthogonal Haar wavelet regularizer. (b) Orthogonal Daubechies D4 wavelet regularizer. (c) Anisotropic total variation regularizer. (d) Undecimated Haar wavelet regularizer.



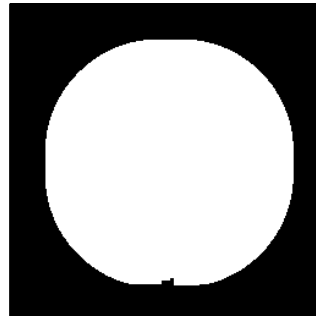
(a) *In vivo* brain mask



(b) BrainWeb numerical phantom mask

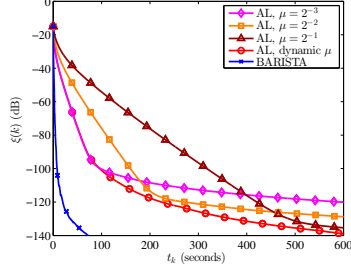


(c) Breast phantom mask

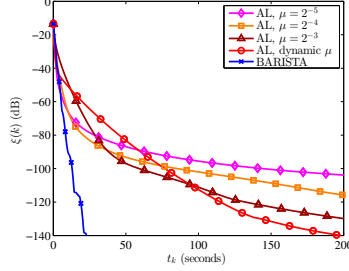


(d) ACR phantom mask

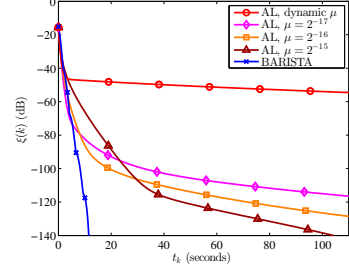
Figure 7: Examples of image-domain masks for each of the four data sets. (a) Mask for the *in vivo* brain data set. (b) Mask for the BrainWeb numerical phantom. (c) Mask for the breast phantom. (d) Mask for the ACR phantom.



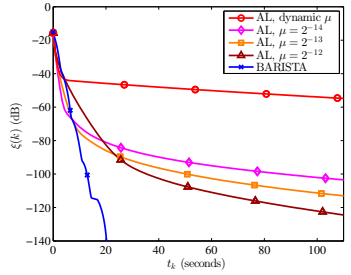
(a) *In vivo* brain, total variation



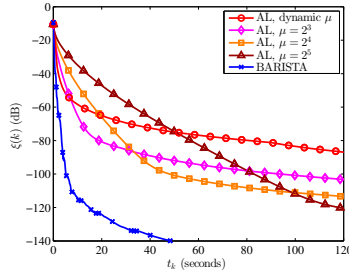
(b) *In vivo* brain, undecimated wavelets



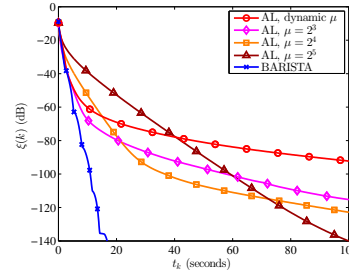
(c) BrainWeb phantom, total variation



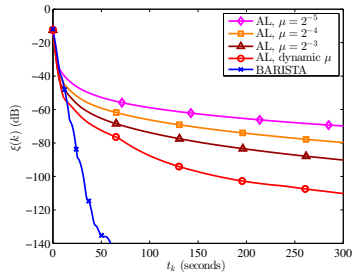
(d) BrainWeb phantom, undecimated wavelets



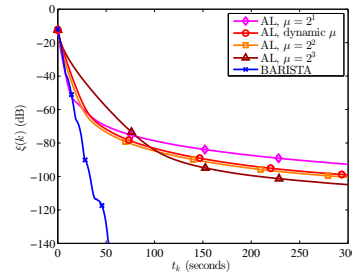
(e) Breast phantom, total variation



(f) Breast phantom, undecimated wavelets



(g) ACR phantom, total variation



(h) ACR phantom, undecimated wavelets

Figure 8: Convergence speed results from ACR phantom imaging data. Markers are placed at 30-iteration intervals, except for the *in vivo* brain data set where they are placed at 100-iteration intervals. (a) *In vivo* brain data, total variation regularizer. (b) *In vivo* brain data, undecimated Haar wavelet regularizer. (c) BrainWeb numerical phantom, total variation regularizer. (d) BrainWeb numerical phantom, undecimated Haar wavelet regularizer. (e) Breast phantom, total variation regularizer. (f) Breast phantom, undecimated Haar wavelet regularizer. (g) ACR phantom, total variation regularizer. (h) ACR phantom, undecimated Haar wavelet regularizer.