



A relaxed primal–dual hybrid gradient method with line search

Alex McManus¹ · Stephen Becker¹ · Nicholas Dwork²

Received: 31 March 2025 / Accepted: 8 May 2026
© The Author(s) 2026

Abstract

The primal–dual hybrid gradient method (PDHG) is a versatile optimization method commonly used for large optimization problems, including image reconstruction. With standard PDHG, there are three user-set parameters and performance of the algorithm is highly sensitive to their values. Toward a parameter-free algorithm, we combine two existing line search methods. The first, by Malitsky et al. [1], is over two of the step sizes in the PDHG iterations. We then use the connection between PDHG and the primal–dual form of Douglas–Rachford splitting to construct a line search over PDHG’s relaxation parameter. We demonstrate the efficacy of the combined line search on multiple problems, including a novel inverse problem in magnetic resonance image reconstruction. The method presented in this manuscript is the first parameter-free variant of PDHG. Across all numerical experiments, there were no changes to line search hyperparameters.

Keywords Primal–dual hybrid gradient · Line search · Partial Fourier · Homodyne detection

Stephen Becker and Nicholas Dwork contributed equally to this work.

✉ Alex McManus
alex.mcmanus@colorado.edu
Stephen Becker
stephen.becker@colorado.edu
Nicholas Dwork
nicholas.dwork@cuanschutz.edu

¹ Department of Applied Mathematics, University of Colorado Boulder, 1111 Engineering Drive, Boulder, CO 80309, USA

² Department of Biomedical Informatics, University of Colorado School of Medicine, 1890 N. Revere Court, Aurora, CO 80045, USA

1 Introduction

Large convex optimization problems with non-smooth objective functions are common in image reconstruction. Proximal algorithms are effective methods commonly used to solve these problems [2]. Examples include the Alternating Direction Method of Multipliers [3], the Douglas Rachford method [4], and the Fast Iterative Shrinkage Thresholding Algorithm (FISTA) [5]. This manuscript is focused on the primal–dual hybrid gradient method (PDHG) [6], which is an extremely versatile algorithm capable of solving a wide array of problems. PDHG is a first-order proximal algorithm capable of solving problems of the form

$$x^* \in \operatorname{argmin}_{x \in \mathcal{H}} \{f(x) + g(Ax)\},$$

where $\mathcal{H}$ is a Hilbert space, A is a linear operator, and f and g are closed convex proper functions.

PDHG can efficiently solve problems that are more general than those that can be solved with other proximal splitting methods (e.g., Douglas-Rachford or FISTA) because it only requires the proximal operator of g , rather than the proximal operator of $g \circ A$ which generally only has a closed form solution when AA^* is a scaled identity.

With standard relaxed PDHG, there are three parameters the user needs to set: two step sizes τ and σ , and a relaxation parameter θ . Convergence is guaranteed when $\tau\sigma\|A\|^2 \leq 1$. Therefore, if the spectral norm of A is known or can be estimated, the number of parameters that must be chosen by the user can be reduced to two. The rate of convergence is highly dependent on the parameter values which makes a line search that eliminates user-defined parameters desirable.

Toward that goal, this manuscript presents a combination of two line search techniques over two sets of parameters. Malitsky et al. [1] introduced a line search over the proximal step sizes. By using the connection between PDHG and the primal–dual formulation of the Douglas-Rachford splitting method (PDDR) [7], we incorporate the theory of *averaged operator iterations* (AOI). AOI admits a line search over the relaxation parameter [8]. The remaining user-set parameters are the proximal step sizes in PDHG. In general, this can be left as unity, so there are no user-defined parameters remaining. This manuscript extends work we previously presented in [9].

Other adaptive methods for PDHG include the adaptive method of Goldstein et al. [10] and the preconditioning method of Pock and Chambolle [11]. The adaptive method selects optimization parameters for the next iteration based on heuristics of the current iteration. It does not automatically determine both the step sizes and the relaxation parameter, which is the intent of the method of this manuscript. In this manuscript, we focused on line-search techniques, where a parameter search is conducted in each optimization iteration. Notably, when there is sufficient hardware for parallel processing (e.g., a Central Processing Unit with multiple processing cores) and the inner optimization is parallelizable, then the line-search can occur with negligible time-penalty when compared to an adaptive method. The preconditioning of

[11] is often beneficial and can be applied independently of the parameter selection method, which we hope to do in future work.

This paper is organized as follows. Section 2 contains notation and relevant definitions and facts. Section 3 contains a review of relevant algorithms including the two line search algorithms. Section 4 is about the combined line search; its Sect. 4.1 details convergence thereof. Section 5 shows numerical results on several example problems. Section 6 details an application to magnetic resonance imaging (MRI) reconstruction, where we introduce a new method that unifies compressed sensing with partial Fourier reconstruction via homodyne detection.

2 Preliminaries

2.1 Notation and definitions

We follow standard optimization literature notation conventions; see [12] for further background. Let $\mathcal{H}$ be a real Hilbert space equipped with inner product $\langle x, y \rangle$ and norm $\| \cdot \| = \sqrt{\langle \cdot, \cdot \rangle}$. Let $\Gamma_0(\mathcal{H})$ be the set of proper, lower semicontinuous convex functions from $\mathcal{H} \rightarrow \mathbb{R} \cup \{+\infty\}$. Take $f \in \Gamma_0(\mathcal{H})$. The domain of f is $\text{dom}(f) = \{x \in \mathcal{H} \mid f(x) < +\infty\}$. Define as $2^{\mathcal{H}}$ the *power set* of $\mathcal{H}$, or the set of all subsets of the Hilbert space. Let $\mathcal{I}$ denote the identity operator.

The *Legendre-Fenchel conjugate* of function f is $f^* \in \Gamma_0(\mathcal{H})$, given by

$$f^*(x) = \sup_{y \in \mathcal{H}} \{\langle x, y \rangle - f(y)\}.$$

Note that if $f \in \Gamma_0(\mathcal{H})$, then $f^* \in \Gamma_0(\mathcal{H})$ and $f^{**} = f$ [12].

The *proximal operator* of λf , for any $\lambda > 0$ is defined as [2]:

$$\text{prox}_{\lambda f}(x) = \underset{u \in \mathcal{H}}{\text{argmin}} \left\{ f(u) + \frac{1}{2\lambda} \|u - x\|^2 \right\}.$$

A *set-valued* operator $A : \mathcal{H} \rightarrow 2^{\mathcal{H}}$ assigns to each $x \in \mathcal{H}$ a subset of $\mathcal{H}$ (or an element in $2^{\mathcal{H}}$). The image of x under A is denoted $A(x)$. If $A(x)$ is a singleton, we write $A(x) = y$ instead of $A(x) = \{y\}$. The *graph* of operator A is defined as

$$\text{gr}(A) = \{(x, y) \mid y \in A(x)\}.$$

The inverse of operator A , denoted A^{-1} , is defined with graph

$$\text{gr}(A^{-1}) = \{(y, x) \mid (x, y) \in \text{gr}(A)\}.$$

For operator A , define left- and right-scalar multiplication as

$$(\lambda A)(x) = \{\lambda y : y \in A(x)\}, \quad (A\mu)(x) = \{y : y \in A(\mu x)\}.$$

The following relationships hold between the graph of A and the graphs of its inverse and scalar multiples:

$$\text{gr}(A^{-1}) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \text{gr}(A), \quad \text{gr}(\lambda A\mu) = \begin{bmatrix} \mu^{-1}\mathcal{I} & 0 \\ 0 & \lambda\mathcal{I} \end{bmatrix} \text{gr}(A). \quad (1)$$

Operator A is *monotone* if

$$\langle x - x', y - y' \rangle \geq 0, \quad \forall (x, y), (x', y') \in \text{gr}(A).$$

An operator A is *maximally monotone* if it is both monotone and there are no monotone operators B such that $\text{gr}(A) \subset \text{gr}(B)$.

Let $\mathcal{D}$ be a nonempty subset of $\mathcal{H}$ and let $A : \mathcal{D} \rightarrow \mathcal{H}$. A is *nonexpansive* if it is Lipschitz continuous with constant 1, or

$$\|Ax - Ay\| \leq \|x - y\|, \quad \forall x \in \mathcal{D}, \forall y \in \mathcal{D}.$$

A is *firmly nonexpansive* if

$$\|Ax - Ay\|^2 + \|(\mathcal{I} - A)x - (\mathcal{I} - A)y\|^2 \leq \|x - y\|^2, \quad \forall x \in \mathcal{D}, \forall y \in \mathcal{D}.$$

We say operator A is *averaged* with constant α if there exists a nonexpansive operator $R : \mathcal{D} \rightarrow \mathcal{H}$ such that $A = (1 - \alpha)\mathcal{I} + \alpha R$. A is *firmly nonexpansive* if and only if it is $1/2$ -averaged [12, Remark 4.34].

For a set-valued operator A , define its *resolvent* as:

$$J_{\lambda A} = (\mathcal{I} + \lambda A)^{-1},$$

where $\mathcal{I}$ is the identity. If A is a monotone operator, its resolvent is firmly nonexpansive. Define the *reflected resolvent* of A as:

$$R_{\lambda A} = 2J_{\lambda A} - \mathcal{I}.$$

Importantly, if A is maximally monotone then its reflected resolvent is nonexpansive [12, Corollary 23.11]. The resolvent and reflected resolvents of A satisfy the following identities:

$$J_A(x) + J_{A^{-1}}(x) = x, \quad \text{and} \quad R_A(x) + R_{A^{-1}}(x) = 0. \quad (2)$$

Since $(\gamma A)^{-1} = A^{-1}\gamma^{-1}$ for scalar γ (which can be shown directly from the relationships in (1)),

$$J_{\gamma A}(x) + \gamma J_{\gamma^{-1}A^{-1}}(x/\gamma) = x. \quad (3)$$

Let $A : \mathcal{H} \rightarrow \mathcal{H}$ be such that $AA^* = \alpha \mathcal{I}$, where * denotes the operator's adjoint. Let $f \in \Gamma_0(\mathcal{H})$ and let $g(x) = f(Ax)$. Then

$$\text{prox}_{\gamma g}(x) = x + \frac{1}{\alpha} A^* (\text{prox}_{\alpha \gamma f}(Ax) - Ax).$$

The *subdifferential* of $f \in \Gamma_0(\mathcal{H})$ is a set-valued operator ∂f defined as:

$$\partial f(x) = \{\zeta \in \mathcal{H} \mid f(y) \geq f(x) + \langle \zeta, y - x \rangle \forall y \in \mathcal{H}\}.$$

By the definition of the subdifferential, we have $x^* \in \underset{x}{\operatorname{argmin}} f(x)$ if and only if $0 \in \partial f(x^*)$ —this is known as *Fermat's law* [12, Theorem 16.3]. Therefore, for suitable functions $f \in \Gamma_0(\mathcal{H})$, solving a minimization problem is the same as solving a *monotone inclusion problem*.

Maximal monotone operators are useful for convex optimization because the subdifferential of any $f \in \Gamma_0(\mathcal{H})$ is maximally monotone. Furthermore, the inverse of the subdifferential is the subdifferential of the conjugate function:

$$(\partial f)^{-1} = \partial f^*.$$

An important fact for the convergence of proximal algorithms is that the proximal operator of a proper function $f \in \Gamma_0(\mathcal{H})$ is the resolvent of the subdifferential:

$$\text{prox}_{\gamma f}(x) = (\mathcal{I} + \gamma \partial f)^{-1} x. \quad (4)$$

Since the subdifferential of $f \in \Gamma_0(\mathcal{H})$ is maximally monotone, (4) is single-valued. Letting $A = \partial f$ for $f \in \Gamma_0(\mathcal{H})$ in (3) yields the *Moreau identity*:

$$\text{prox}_{\gamma f}(x) + \gamma \text{prox}_{\gamma^{-1} f^*}(x/\gamma) = x.$$

2.2 The problem

For the remainder of this manuscript, let $\mathcal{X}$ and $\mathcal{Y}$ be real Hilbert spaces. We are interested in solving the following (*primal*) optimization problem:

$$x^* \in \underset{x \in \mathcal{X}}{\operatorname{argmin}} f(x) + g(Ax), \quad (5)$$

where $f \in \Gamma_0(\mathcal{X})$ and $g \in \Gamma_0(\mathcal{Y})$ have computationally tractable forms of their respective proximal operators and $A : \mathcal{X} \rightarrow \mathcal{Y}$ is a linear operator with known adjoint A^* and finite operator norm:

$$\|A\| = \sup \{\|Ax\| : x \in \mathcal{X} \text{ with } \|x\| \leq 1\} < +\infty.$$

The corresponding dual problem to (5) is:

$$y^* \in \operatorname{argmax}_{y \in \mathcal{Y}} -f^*(-A^*y) - g^*(y). \quad (6)$$

Under certain constraint qualifications such as

$$0 \in \operatorname{sri}(\operatorname{dom}(g) - A(\operatorname{dom}(f))), \quad (7)$$

then the duality gap is zero (i.e. strong duality holds) and the set of solutions to (6) is nonempty [12, Theorem 15.23]. Here $\operatorname{sri}(C)$ is the *strong relative interior* of a set C (see [12, Section 6.2] for more details). In many applications of interest, $\mathcal{X}$ and $\mathcal{Y}$ are finite-dimensional. The condition (7) is then satisfied if

$$(\operatorname{ri} \operatorname{dom}(g)) \cap A(\operatorname{ri} \operatorname{dom}(f)) \neq \emptyset, \quad (8)$$

where ri is the *relative interior* of a set [12, Proposition 15.24]. The condition (8) is easy to show if either f or g is full domain.

The algorithm that is the focus of this manuscript is *primal–dual*, meaning it is applied to the saddle point interpretation of the primal (5) and dual (6) problems:

$$(\hat{x}, \hat{y}) \in \operatorname{argmin}_{x \in \mathcal{X}} \operatorname{argmax}_{y \in \mathcal{Y}} \{\langle Ax, y \rangle + f(x) - g^*(y)\}. \quad (9)$$

More details can be found in, e.g., [6, 12, 13].

The saddle point formulation yields the following optimality conditions, which are the starting point for development of primal–dual algorithms:

$$(\hat{x}, \hat{y}) \in \mathcal{X} \times \mathcal{Y} \text{ such that } \begin{pmatrix} 0 \\ 0 \end{pmatrix} \in \begin{pmatrix} 0 & A^* \\ -A & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \partial f(x) \\ \partial g^*(y) \end{pmatrix}. \quad (10)$$

The Karush-Kuhn-Tucker conditions state that if $(\hat{x}, \hat{y})$ is a solution to (10), then $\hat{x}$ is a solution to (5) and $\hat{y}$ is a solution to (6) [12, Theorem 19.1]. Furthermore, $(\hat{x}, \hat{y})$ is a solution to (9) [12, Proposition 19.20]. Note that the converse is not true in general and the set of solutions to (10) may be empty. Under certain constraint qualifications, such as (8), whenever $\hat{x}$ is a solution to (5) and $\hat{y}$ is a solution to (6), then $(\hat{x}, \hat{y})$ is a solution to (10).

3 Algorithms

3.1 Douglas-Rachford splitting

The standard form of Douglas-Rachford splitting (DR) was originally given by Lions and Mercier [14] to find zeros of a sum of maximally monotone operators F and G ; i.e., to find x such that

$$0 \in F(x) + G(x).$$

The standard version of the algorithm is the fixed point iteration

$$y_{k+1} = \frac{1}{2}y_k + \frac{1}{2}R_{\gamma G}R_{\gamma F}y_k, \quad (11)$$

where γ is a positive constant.

This is commonly presented as the following steps:

$$\begin{aligned} x_{k+1} &= J_{\gamma F}(y_k) \\ w_{k+1} &= J_{\gamma G}(2x_{k+1} - y_k) \\ y_{k+1} &= y_k + w_{k+1} - x_{k+1}. \end{aligned}$$

If we allow for a relaxation parameter $\alpha_k \in (0, 1)$, then (11) is altered as follows:

$$y_{k+1} = (1 - \alpha_k)y_k + \alpha_k R_{\gamma G}R_{\gamma F}y_k. \quad (12)$$

Equivalently,

$$\begin{aligned} x_{k+1} &= J_{\gamma F}(y_k) \\ w_{k+1} &= J_{\gamma G}(2x_{k+1} - y_k) \\ y_{k+1} &= y_k + \rho_k(w_{k+1} - x_{k+1}), \end{aligned} \quad (13)$$

where $\rho_k = 2\alpha_k$.

Results on the convergence of DR are given in, e.g., [4, 12, 14, 15]. The standard assumptions for convergence are the maximal monotonicity of F and G and the existence of a solution. Using relationship (2) on the second step of the iteration (13) yields the iterations for *primal dual* Douglas-Rachford (PDDR):

$$\begin{aligned} x_{k+1} &= J_{\gamma F}(y_k) \\ w_{k+1} &= J_{(\gamma G)^{-1}}(2x_{k+1} - y_k) \\ y_{k+1} &= (1 - \rho_k)y_k + \rho_k(x_{k+1} - w_{k+1}). \end{aligned} \quad (14)$$

By using relationship (2) applied to the fixed point iteration version of DR in (12), we get the fixed point iteration form of PDDR:

$$y_{k+1} = (1 - \alpha_k)y_k - \alpha_k R_{(\gamma G)^{-1}}R_{\gamma F}y_k. \quad (15)$$

3.2 Primal–dual hybrid gradient

The primal–dual hybrid gradient (PDHG) method that solves problems of the form of (10) has the following iterations [6, 16]:

$$\begin{aligned}\bar{x}_{k+1} &= J_{\tau_k F}(x_k - \tau_k A^* z_k) \\ \bar{z}_{k+1} &= J_{\sigma_k G^{-1}}(z_k + \sigma_k A(2\bar{x}_{k+1} - x_k)) \\ x_{k+1} &= x_k + \alpha_k(\bar{x}_{k+1} - x_k) \\ z_{k+1} &= z_k + \alpha_k(\bar{z}_{k+1} - z_k),\end{aligned}\tag{16}$$

where $F = \partial f$, $G^{-1} = \partial g^*$ for proper, lower semicontinuous convex functions $f \in \Gamma_0(\mathcal{X})$ and $g \in \Gamma_0(\mathcal{Y})$, and $A : \mathcal{X} \rightarrow \mathcal{Y}$ is a bounded linear operator with known adjoint. Since $g \in \Gamma_0(\mathcal{Y})$, $g^* \in \Gamma_0(\mathcal{X})$ as well. The resolvents J_F and $J_{G^{-1}}$ are proximal operators of the functions f and g , as in (4).

As written, the algorithm has three parameters for each iteration: τ and σ , which must satisfy $\tau\sigma\|A\|^2 \leq 1$ to guarantee convergence [13], and α , which is a *relaxation parameter* and typically takes values in $(0, 2)$ [12]. Commonly, parameter values are held constant for all iterations. I.e., $\tau_k = \tau$, $\sigma_k = \sigma$, and $\alpha_k = \alpha$ for all k .

Line searches for primal–dual methods are difficult in general. The conditions for a standard line search, e.g., Armijo backtracking line search [17], cannot always be evaluated (e.g., when primal–dual methods are applied to problems where f or g are an indicator function). However, a line search for PDHG was introduced by Malitsky et al. in [1]; it is shown in Alg. 1.

Algorithm 1 PDHG line search [1]

Initialization: Given $x_{k-1}, z_{k-1} \in \mathcal{H}$, $\text{prox}_f, \text{prox}_{g^*}$, $\tau_{k-1} > 0$, $\eta_{k-1} > 0$, $\mu \in (0, 1)$, $\beta > 0$, $\delta \in (0, 1)$.

Main iteration:

1. Compute
$$x_k = \text{prox}_{\tau_{k-1} f}(x_{k-1} - \tau_{k-1} A^* z_{k-1})$$
2. Choose $\tau_k \in [\tau_{k-1}, \tau_{k-1}\sqrt{1 + \eta_{k-1}}]$ and run $\triangleright$ By default we choose $\tau_k = \tau_{k-1}\sqrt{1 + \eta_{k-1}}$

Linesearch:

- 2.a. Compute
$$\eta_k = \frac{\tau_k}{\tau_{k-1}}$$
$$\bar{x}_k = x_k + \eta_k(x_k - x_{k-1})$$
$$z_k = \text{prox}_{\beta\tau_k g^*}(z_{k-1} + \beta\tau_k A\bar{x}_k)$$
- 2.b. Break linesearch if
$$\sqrt{\beta}\tau_{k+1}\|A^* z_k - A^* z_{k-1}\| \leq \delta\|z_k - z_{k-1}\|$$

and return $z_k, \bar{x}_k$

Otherwise, set $\tau_k \leftarrow \mu\tau_k$ and go to 2.a.

End of linesearch

3.3 The relationship between PDDR and PDHG

The PDDR iterations (14) applied to the following *modified problem* of (17) yield the same iterations as PDHG applied to the original problem (5) [7]:

$$(\hat{x}, \hat{y}) \in \operatorname{argmin}_{x \in \mathcal{X}, y \in \mathcal{Y}} \tilde{f}(x, y) + \tilde{g}(x, y) \quad (17)$$

with modified functions

$$\tilde{f}(x, y) = f(x) + \delta_{\{0\}}(y), \quad \tilde{g}(x, y) = g(Ax + By).$$

Here, $\delta_{\{0\}}$ is an indicator function specifying that the argument is 0 and B is chosen to satisfy $AA^* + BB^* = \frac{1}{\theta} \mathcal{I}$ for some $\theta > 0$. (Note that an indicator takes on a value of 0 when its argument satisfies its condition; otherwise, it takes on a value of ∞ .) As an example, if A is an $n \times n$ matrix with $\frac{1}{\theta} \geq \|A\|_2^2$, let $B = (\frac{1}{\theta} \mathcal{I} - AA^*)^{1/2}$ where $(\cdot)^{1/2}$ is the Cholesky factorization. Defining these modified functions and applying the iterations of PDDR (14) to problem (17) yields the PDHG iterations of (16) [7]. In [7], the choice of θ is left to the user. The relationship between θ and the τ, σ parameters from PDHG is $\tau\sigma = \theta$ [7].

As can be seen in [7], the relationship between the variables of PDDR in (14) when applied to (17) and PDHG in (16) when applied to (5) is

$$y_k = (x_k - \tau_k A^* z_k, -\tau_k B^* z_k). \quad (18)$$

This relationship is what allows us to tie the two existing line searches together. An iteration of PDHG is the same as one iteration of PDDR with the relationship above.

3.4 The averaged operator iteration line search

An *averaged operator iteration* (AOI) for a nonexpansive operator S is an iteration of the form

$$y_{k+1} = (1 - \alpha)y_k + \alpha Sy_k = y_k + \alpha(Sy_k - y_k). \quad (19)$$

All of the algorithms discussed in this section so far can be written as an AOI. The AOI form of Douglas-Rachford is given in (11) where $S = R_{\gamma G} R_{\gamma F}$. The AOI form

of PDDR is given in (15) with $S = -R_{(\gamma G)^{-1}} R_{\gamma F}$. In both cases, S is nonexpansive as it is a composition of reflected resolvents, which are themselves nonexpansive [12, Proposition 4.4].

A line search for averaged operator iterations was introduced by Gisselson et al. in [8]. The pseudocode for the algorithm is presented in Alg. 2.

Algorithm 2 AOI line search [8]

Initialization: Choose $y_0 \in \mathcal{H}$, $\alpha_{\max} > 0$, $\bar{\alpha} \in (0, \alpha_{\max})$, $\varepsilon > 0$, $\mu \in (0, 1)$.

Main iteration:

1. Compute

$$r_k = Sy_k - y_k$$

$$\tilde{y}_k = y_k + \bar{\alpha} r_k$$

$$\bar{r}_k = S\tilde{y}_k - \tilde{y}_k$$

2. Set $\alpha_k = \alpha_{\max}$ and run

Linesearch:

2.a. Compute

$$y_{k+1} = y_k + \alpha_k r_k$$

$$r_{k+1} = Sy_{k+1} - y_{k+1}$$

2.b. Break linesearch if

$$\|r_{k+1}\| = \|Sy_{k+1} - y_{k+1}\| \leq (1 - \varepsilon)\|\bar{r}_k\|$$

Otherwise, set $\alpha_k \leftarrow \mu\alpha_k$ and go to 2.a. If $\alpha_k = \bar{\alpha}$, terminate the line search.

End of linesearch

AOI with line search requires the user to set a nominal step size $\bar{\alpha}$. This can be set to $\frac{1}{2}$ to ensure the *nominal* step corresponds to a fixed-point iteration of a firmly nonexpansive operator. In order, the steps of Alg. 2 compute the current residual, compute the *nominal* next step with the nominal step size, and then the residual with the nominal step size. The line search then iterates over the possible values of α_k in the final step. Either the step size selected is $\alpha_k = \bar{\alpha}$, the nominal step, or $\alpha_k \in (\bar{\alpha}, \alpha_{\max}]$ is chosen such that

$$\|r_{k+1}\|_2 = \|Sy_{k+1} - y_{k+1}\|_2 \leq (1 - \varepsilon)\|\bar{r}_k\|_2, \quad (20)$$

where $\varepsilon > 0$ and $\alpha_{\max} > \bar{\alpha}$ are fixed algorithm parameters.

The AOI iterations of (19) can be interpreted as steps of length α in the direction of the residual $Sx_k - x_k$. Taking a large step in this direction may be beneficial and that is what this line search aims to accomplish. In practice, $\alpha_{\max}$ is set much larger than $\bar{\alpha}$ and progressively shrunk to test if a large step in the residual direction reduces the norm of the residual more than the nominal step.

4 The line search

The connection between PDDR as an AOI has been shown and the relationship between the variables of PDDR and PDHG are presented in (18). The strategy of the proposed line search is as follows: use the AOI line search to find an advantageous relaxation parameter, where for each proposed relaxation parameter we convert the variables to those of PDHG and use the line search of Malitsky et al. to find proximal step sizes τ and σ . We refer to these steps, respectively, as the “outer” and “inner” line searches, respectively.

We propose a *relaxed primal–dual hybrid gradient method with line search*, or rPDHG. With our proposed method, the outer AOI line search calls the inner PDHG line search at each step. Pseudocode for the Outer Line Search is shown in Alg. 3, and pseudocode for the Inner Line Search (which is equivalent to the Maltisky PDHG line search) is shown in Alg. 4.

The proposed line search inherits some parameters from the AOI and PDHG line searches. The β parameter is the ratio between the two proximal step sizes τ_k and σ_k , which we leave as unity in experiments but include in the algorithm for completeness. Numerical experiments have shown that a choice of $\beta = 1$ reliably leads to good convergence.

Algorithm 3 Relaxed PDHG with line search-outer line search

Initialization: Choose $x_0, z_0 \in \mathcal{H}$, $\alpha_{\max} > 0$, $\bar{\alpha} \in (0, \alpha_{\max})$, $\varepsilon > 0$, $\mu \in (0, 1)$, $\tau_0 > 0$, A, B such that $AA^* + BB^* = \frac{1}{\mu} \mathcal{L}$.

Main iteration:

1. Compute

$$[x_{k+1/2}, z_{k+1/2}, \tau_k] = \text{Inner Line Search}(x_{k-1}, z_{k-1}, \tau_{k-1}, \bar{\alpha}).$$

2. Compute

$$\begin{aligned} \bar{x}_{k+1} &= \text{prox}_{\tau_k f}(x_{k+1/2} - \tau_k A^* z_{k+1/2}), \\ \bar{z}_{k+1} &= \text{prox}_{\beta \tau_k g^*}(z_{k+1/2} + \beta \tau_k A(2 * \bar{x}_{k+1} - x_{k+1/2})), \end{aligned}$$

3. Compute

$$\bar{r}_k = \begin{bmatrix} (\bar{x}_k - \tau_k A^* \bar{z}_k) - (x_{k+1/2} - \tau_k A^* z_{k+1/2}) \\ -\tau_k B^* \bar{z}_k - (-\tau_k A(2 * \bar{x}_{k+1} - x_{k+1/2})) \end{bmatrix}.$$

4. Set $\alpha_k = \alpha_{\max}$ and run

Linesearch:

4.a. Compute

$$[x_{k+1/2}, z_{k+1/2}, \tau_k] = \text{Inner Line Search}(x_{k-1}, z_{k-1}, \tau_{k-1}, \alpha_k), \quad (21)$$

$$\bar{x}_{k+1} = \text{prox}_{\tau_k f}(x_{k+1/2} - \tau_k A^* z_{k+1/2}), \quad (22)$$

$$\bar{z}_{k+1} = \text{prox}_{\beta \tau_k g^*}(z_{k+1/2} + \beta \tau_k A(2 * \bar{x}_{k+1} - x_{k+1/2})), \quad (23)$$

$$r_{k+1} = \begin{bmatrix} (\bar{x}_{k+1} - \tau_k A^* \bar{z}_{k+1}) - (x_{k+1/2} - \tau_k A^* z_{k+1/2}) \\ -\tau_k B^* \bar{z}_{k+1} - (-\tau_k A(2 * \bar{x}_{k+1} - x_{k+1/2})) \end{bmatrix}. \quad (24)$$

4.b. Break line search if

$$\|r_{k+1}\| \leq (1 - \varepsilon)\|\bar{r}_k\|$$

Otherwise, set $\alpha_k \leftarrow \mu \alpha_k$ and go to 4.a. If $\alpha_k = \bar{\alpha}$, terminate the line search.

End of line search

Algorithm 4 Relaxed PDHG with line search-inner line search

Initialization: Given $x_{k-1}, z_{k-1} \in \mathcal{H}$, $\text{prox}_f, \text{prox}_{g^*}$, $\alpha_k \in [\bar{\alpha}, \alpha_{max}]$, $\tau_{k-1} > 0$, $\theta_{k-1} > 0$, A, B such that $AA^* + BB^* = \frac{1}{\theta} \mathcal{I}$, $\mu \in (0, 1)$, $\beta > 0$, $\delta \in (0, 1)$.

Main iteration:

1. Compute

$$\hat{x}_k = \text{prox}_{\tau_{k-1}f}(x - \tau_{k-1}A^*z_{k-1})$$
2. Choose $\tau_k \in [\tau_{k-1}, \tau_{k-1}\sqrt{1 + \theta_{k-1}}]$ and run ▷ By default we choose
 $\tau_k = \tau_{k-1}\sqrt{1 + \theta_{k-1}}$
- Line search:**
 - 2.a. Compute

$$\theta_k = \frac{\tau_k}{\tau_{k-1}}$$

$$\bar{x}_k = \hat{x}_k + \theta_k(\hat{x}_k - x_{k-1})$$

$$\bar{z}_k = \text{prox}_{\beta\tau_k g^*}(z_{k-1} + \beta\tau_k K\bar{x}_k)$$
 - 2.b. Break line search if

$$\sqrt{\beta\tau_k}\|A^*\bar{z}_k - A^*z_{k-1}\| \leq \delta\|\bar{z}_k - z_{k-1}\|.$$

Otherwise, set $\tau_k \leftarrow \mu\tau_k$ and go to 2.a.

End of line search

3. Compute

$$x_k = (1 - 2\alpha_k)x_{k-1} + 2\alpha_k\bar{x}_k$$

$$z_k = (1 - 2\alpha_k)z_{k-1} + 2\alpha_k\bar{z}_k.$$

Return $x_k, z_k, \tau_k, \theta_k$.

4.1 Convergence

Convergence of the outer (AOI) line search is detailed in the original paper [8]. To briefly summarize, let k_0 denote the number of times the outer line search activates (that is, chooses $\alpha_k > \bar{\alpha}$). If k_0 is finite, let $k_{\max}$ denote the final iteration on which $\alpha_k > \bar{\alpha}$. For all $k > k_{\max}$, the iteration is

$$y_{k+1} = y_k + \bar{\alpha}(S_k y_k - y_k).$$

For $\bar{\alpha} \in (0, 1)$ this corresponds exactly to an iteration of PDHG with Malitsky's line search. As shown in [1, Lemma 1], this line search always terminates. Convergence is equivalent to convergence of PDHG with line search.

For each iteration k that $\alpha_k > \bar{\alpha}$ the line search condition requires that (for $\varepsilon \in (0, 1)$)

$$\|r_{k+1}\| = \|S_k y_k - y_k\| \leq (1 - \varepsilon)\|\bar{r}_k\| \leq (1 - \varepsilon)\|r_k\|,$$

which yields

$$\|r_{k+1}\| \leq (1 - \varepsilon)^{k_0}\|r_0\|.$$

If $k_0 \rightarrow \infty$ as $k \rightarrow \infty$, then $\|r_{k+1}\| \rightarrow 0$.

4.2 Line search activation

Conducting the outer line search is computationally costly. If there were an inexpensive test that could be run to determine whether or not line search would be beneficial, then one could activate line search only in these instances to save time. In [8], Gisselson et al. suggest a test based on the displacement vectors of the trajectory: $v_k = y_k - y_{k-1}$. They suggest activating the line search when v_k is approximately aligned with v_{k+1} ; i.e., when the following is true

$$\frac{v_k^T v_{k-1}}{\|v_k\|_2 \|v_{k-1}\|_2} > 1 - \hat{\varepsilon}$$

for some small $\hat{\varepsilon} > 0$.

We hypothesize that this heuristic was useful because the objective function of the problem in their experiments was a strongly-convex quadratic where the gradient becomes smaller quickly as one approaches the optimal point. The ℓ_1 norm used as the objective function in our application does not have this property, and the line search was almost always activated.

Instead, we propose the following: the line search will be activated if either 1) it is the first iteration, 2) line search was activated in the previous iteration and α_k did not equal $\bar{\alpha}$, or 3) $\|r_k\|_2 / \|r_{k-1}\|_2 < 1 - \hat{\varepsilon}$ for some small $\hat{\varepsilon} > 0$. For the results presented in this manuscript, $\hat{\varepsilon} = 0.05$. In words, the third condition means that line search should be activated when the size of the residual for the current iteration is smaller than that of the previous iteration. We based this on (20), where we are using r_{k-1} as a surrogate for $\bar{r}_k$.

4.3 Limitations with matrix free implementations

The line search presented may not be suitable for every application. For example, in image processing applications, A may operate on $n \times n$ images, so $A : \mathbb{R}^{n^2} \rightarrow \mathbb{R}^{n^2}$. For modestly sized n , this can quickly exceed memory limitations for standard computers if A is stored in dense matrix form. Many of these operators work matrix-free, but finding a suitable B for the proposed line search may require storing B as a dense matrix where B is at least as large as A . An example of this phenomenon is shown in Sect. 5.3, where the image had to be scaled down so B could fit in memory.

For certain operators A , such as the discrete gradient operator (used in Sects. 5.2 and 5.3), the operator itself is sparse. In such cases, large memory and performance gains are possible using numerical implementations of sparse matrices.

When A is matrix free, it may be possible to find a matrix-free implementation of B ; an example is shown in Sect. 6. Alternatively, one may be able to evaluate B matrix-free for arbitrary A by using the Lanczos method [18], but it is unclear whether this would work in practice due to the tradeoff between accuracy and cost.

5 Numerical experiments

We present numerical results for the following problems:

- A generalized LASSO-style regularized least squares problem,
- a one-dimensional (signal) total variation denoising problem,
- a two-dimensional (image) total variation denoising problem, and
- a novel magnetic resonance image (MRI) reconstruction problem using compressed sensing.

For the numerical experiments, we compare the proposed line search against:

- Standard PDHG iterations (16),
- PDHG with the line search of Malitsky et al. as in Alg. 1,
- PDDR applied to the modified problem using the AOI formulation and line search as in Alg. 2.

Extensive grid searches were conducted to choose the optimal parameters for these algorithms, used in the comparison to the proposed method. For the γ parameter of the AOI version of PDDR and the τ parameter of PDHG, we used an evenly-spaced grid in log-space. For each parameter choice, the algorithm was run to completion and the parameter with the smallest final objective value was chosen. All numerical experiments were conducted on a desktop PC with an Intel Core i7-7700k CPU (4.20 GHz) and 64 GB DDR4 RAM. The code was written and executed in MATLAB.

For all experiments, we set $\theta = (0.9)/(\|A\|_2^2)$ and $\beta = 1$. Experiments have shown that the value of θ does not affect convergence so long as $\theta \leq 1/\|A\|_2^2$ is satisfied. More accurately, the effect of θ is subsumed by other parameters in the line search so the specific value of θ does not have an observable effect. Experiments have also shown that so long as the value of β is not extreme – e.g., $10^{-3} < \beta < 10^3$ – the specific value of β only modestly impacts convergence. A value of $\beta = 1$ has reliably given good convergence. An overview of the experiments concerning β and θ is in Sec. 5.4.

Grid search allowed us to find a set of parameters that yielded a best performing optimization for each problem. Effectively, the grid search provided an oracle that allowed us to compare how well the adaptive method converges to standard PDHG if an oracle had chosen the best parameters from the grid. Other methods for selecting hyperparameters, such as optimization step sizes or relaxation variables, include Nelder-Mead optimization, simulated annealing, Bayesian optimization, and successive halving [19]. The purpose of each of these methods is to reduce the computational cost of parameter selection while attaining a high-quality outcome; they are likely viable for many practical applications. Unfortunately, they are generally not guaranteed to achieve an optimal solution. Grid search, while computationally costly, chooses the best parameters on the selected grid. In order to compare the method presented in this manuscript to the best possible performance of PDHG, we compared to parameters chosen with grid search.

5.1 LASSO

Consider the unconstrained generalized LASSO problem:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \|x - b\|_2^2 + \lambda \|Ax\|_1. \quad (25)$$

Here $A \in \mathbb{R}^{n \times n}$, $x \in \mathbb{R}^n$, and $b \in \mathbb{R}^n$. We write $f(x) = \frac{1}{2} \|x - b\|_2^2$ and $g(x) = \lambda \|x\|_1$. The elements of A and b are randomly generated from a standard normal distribution. The matrix B is chosen by setting $\theta = 0.9/\|A\|_2^2$ and solving for B from $BB^* = \frac{1}{\theta} \mathcal{I} - AA^*$ via Cholesky factorization. For the results presented, we set $n = 1000$. After searching over parameters, we compare the best convergence rate for each method. This is shown in Fig. 1. For rPDHG, however, we set $\beta = 1$ and chose a $\tau_0 > 0$, avoiding a costly parameter search.

5.2 1D-TV denoising

Consider the one-dimensional total variation denoising problem (cf. [20]). We add white noise to a piecewise-constant vector b of length n . We then solve the problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \|x - b\|_2^2 + \lambda \|\nabla x\|_1,$$

where $\lambda > 0$ is a chosen parameter and $\nabla : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is the one-dimensional finite difference operator:

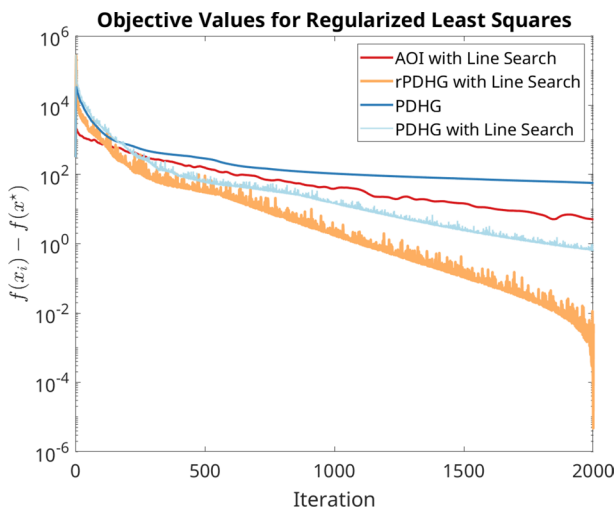


Fig. 1 A comparison of the objective values from the different optimization algorithms as they solve the regularized least squares problem in (25)

$$\nabla x_i = \begin{cases} |x_1 - x_n|, & i = 1 \\ |x_i - x_{i-1}|, & i = 2, \dots, n. \end{cases}$$

The signal is created by specifying a number of segments num_{segs} and length of segments len_{segs} and letting $n = num_{\text{segs}} * len_{\text{segs}}$. Then b is created by taking num_{segs} random integers $z \in [-5, 5]$ and creating a step function from len_{segs} . White noise is then added to this to create the measurements b . For this experiment, we chose $\lambda = 1$.

Here $f(x) = \frac{1}{2}\|x - b\|_2^2$, $g(x) = \lambda\|x\|_1$, and A is the finite difference operator ∇ . Creating B for the line search requires first creating a matrix representation of A and then creating B by choosing $\theta < 1/\|A\|^2$ and solving for B from $BB^* = (1/\theta)(I - AA^*)$, via Cholesky factorization. Figure 2 shows the ground truth and the noisy function. Figure 3 shows the objective values for each iteration of the four different algorithms along with the residuals of the final answer produced by the algorithms. For the three algorithms other than rPDHG, we performed a grid search for all parameters and identified the set of parameters that yielded the lowest objective value. For rPDHG, however, we simply set $\beta_0 = 1$ and choose a $\tau_0 > 0$, avoiding a costly parameter search. In Fig. 3, the residual values are highest where there are discontinuities in the underlying function.

5.3 2D-TV denoising

The two-dimensional total variation denoising problem follows the standard ROF model [20, 21]:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2}\|x - b\|_2^2 + \|\nabla x\|_{2,1}. \quad (26)$$

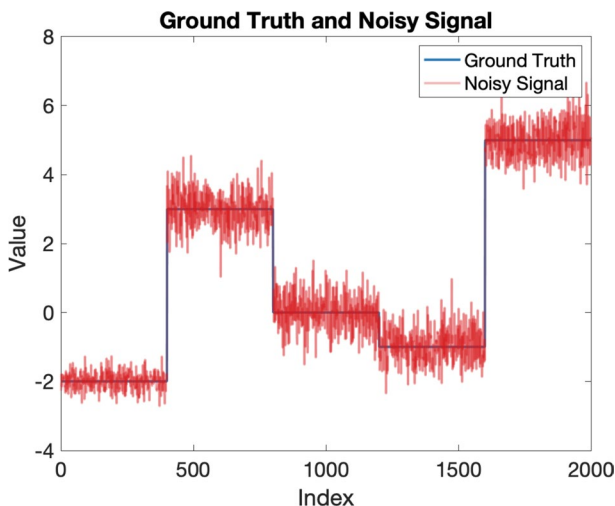


Fig. 2 The ground truth and noisy signals used for the one-dimensional total variation denoising problem

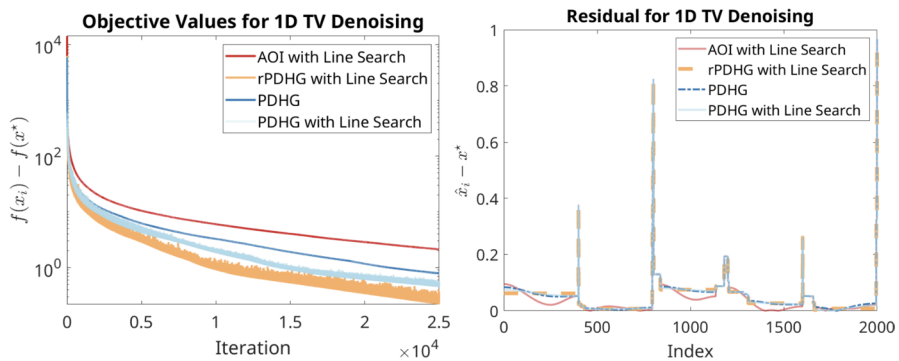


Fig. 3 (Left) A comparison of the objective values of different algorithms solving the one-dimensional total variation denoising problem. The line search proposed in this manuscript converges with fewer iterations than either constituent line search alone. The objective value jumps often as different step sizes are chosen. The raw objective value for each step is shown for academic purposes; in practice, one would choose the best objective value so far. The PDHG line performs better than standard PDHG. The AOI line search alone converges to a poor solution. (Right) A comparison of the best answers generated by the different optimization algorithms after 1000 iterations, plotted as the residual



Fig. 4 The cameraman image used for the two-dimensional total variation denoising problem. On the right is the image with white noise added

The experiments conducted for this manuscript used the `cameraman.tif` image in MATLAB. The image was read in as a 256×256 double and scaled so that its maximum value is 1. The image was then downsized to 77×77 pixels for memory considerations. (As discussed previously in Sect. 4.3, for image processing tasks the B matrix can have a large memory footprint. Therefore, to run this example on a desktop computer the image was scaled down.) Noise was generated as a 77×77 matrix where the value of each element was drawn from a standard normal distribution and multiplied by 0.08. The original image and the noised image are shown in Fig. 4.

The discrete gradient operator ∇ is defined $\nabla : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}^{n \times m \times 2}$:

$$\nabla x_{i,j}^1 = \begin{cases} x_{i+1,j} - x_{i,j}, & i = 1, \dots, m-1 \\ x_{1,j} - x_{n,j}, & i = m \end{cases} \quad (27)$$

$$\nabla x_{i,j}^2 = \begin{cases} x_{i,j+1} - x_{i,j}, & i = 1, \dots, n-1 \\ x_{i,n} - x_{i,1}, & i = n. \end{cases} \quad (28)$$

The $\ell_{2,1}$ norm is defined as

$$\|\nabla x\|_{2,1} = \sum_{i=1}^m \sum_{j=1}^n |(\nabla x_{i,j}^1)^2 + (\nabla x_{i,j}^2)^2|^{1/2}.$$

Similar to the one-dimensional problem, the discrete gradient operator is our A operator for the minimization problem. To calculate B , we first turn A into a matrix from its matrix-free form and then choose $\theta < 1/\|A\|^2$ and solve for B from $BB^* = 1/\theta I - AA^*$ via Cholesky factorization. A comparison of the objective values for the different algorithms used is in Fig. 5 (left). PDHG and the line search for the AOI formulation of PDDR converge slowly. The line search proposed in this manuscript and the Malitsky line search decrease both tend to fluctuate towards a solution, rather than decreasing monotonically. For a fixed number of iterations, rPDHG with line search has the lowest error. The final output of the proposed algorithm is Fig. 5 (right).

The vibrations in the loss of rPDHG with line search as a function of iteration are a feature of the inner line search. The choice of τ (step 2 in Alg. 4) influences how aggressive the vibrations are (data not shown). Heuristically, this allows the line search to explore the solution space around the previous iterate.

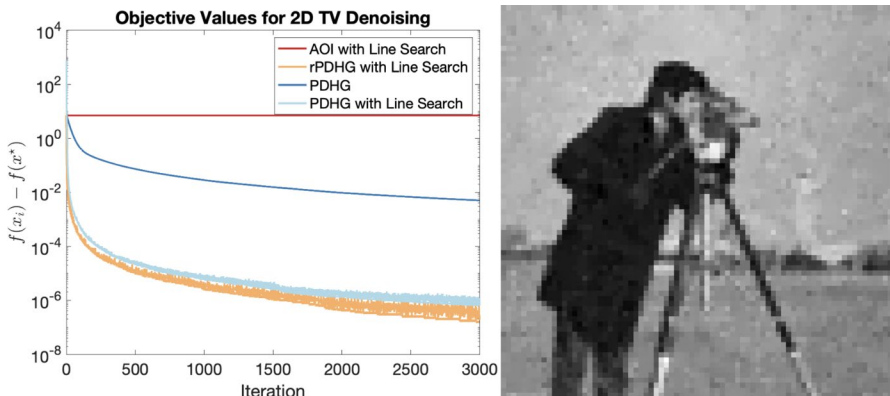


Fig. 5 (Left) A comparison of the objective values of the different optimization algorithms used for the two-dimensional total variation denoising problem. The AOI line search converged to a poor solution. The line search proposed in this manuscript took fewer iterations to converge to any given level of error and did not require searching over any parameters. The comparison against PDHG and PDHG with line search is done against their best sets of parameters. (Right) The final output of rPDHG, the line search proposed in this manuscript, solving the two dimensional TV denoising problem as in (26)

5.4 Sensitivity to β and θ

To assess the robustness of the proposed line-search, we performed a sensitivity analysis with respect to its two user-facing scalar parameters, β and θ .

Recall that $\beta > 0$ is the ratio of the two proximal step sizes from PDHG, while $\theta > 0$ is the constant in the relation

$$AA^* + BB^* = \frac{1}{\theta}I.$$

In all previous experiments we used the default values $\beta = 1$ and $\theta = 0.9/\|A\|_2^2$; this subsection verifies that these choices are not critical.

We evaluated the *ID-TV* denoising problem across a logarithmic range of β and θ values and display three representative cases in Fig. 6. For β we plot the low and high extremes $\{10^{-4}, 10^4\}$ along with the mid-range 10^0 , and for θ we show the extremes $\{10^{-8}/\|A\|_2^2, 0.9/\|A\|_2^2\}$. All intermediate values behaved similarly and are omitted for clarity.

As seen in Fig. 6, convergence is insensitive to orders-of-magnitude changes in either parameter. Thus the default pair $(\beta, \theta) = (1, 0.9/\|A\|_2^2)$ is a robust and reliable choice, and we use it throughout our experiments.

6 Application to magnetic resonance imaging reconstruction

The motivating idea behind rPDHG with line search is an application to accelerated magnetic resonance imaging (MRI) reconstruction. We present a novel combination of acceleration methods, compressed sensing [23] and partial Fourier with homodyne detection [24]. This problem can be written in the form of (5).

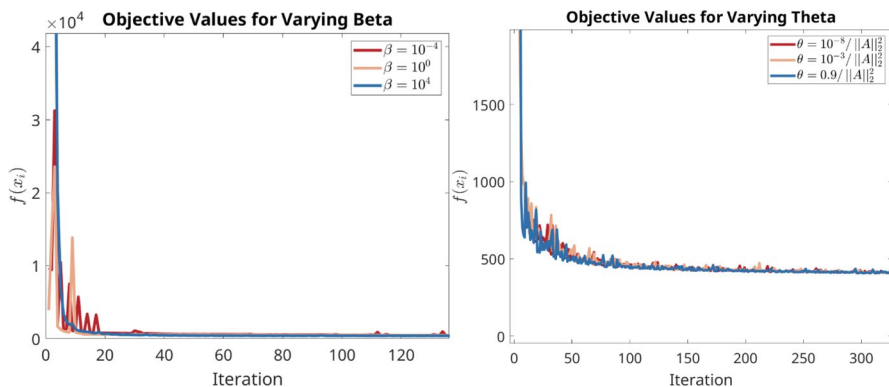


Fig. 6 (Left) Sensitivity of β on the 1D-TV denoising problem. Varying β changes convergence only marginally. (Right) Sensitivity of θ on the same problem. The method is insensitive to θ as long as the required condition $\theta \leq 1/\|A\|_2^2$ is satisfied

6.1 MRI preliminaries

An MRI machine is a Fourier sensing device. An acquisition model for the data from the MRI machine b is

$$b = \mathcal{F}\{x\} + \text{noise} \quad (29)$$

where $\mathcal{F}$ represents the 2 dimensional Fourier transform, x is the subject being imaged, and there is additive noise in the acquisition process [25]. If a sufficiently large amount of data is collected to satisfy the Nyquist-Shannon sampling theorem independently in each direction, then the image can be reconstructed with an inverse Discrete Fourier Transform: $\hat{x} = F^{-1} b$.

The scan time of MRI is proportional to the number of samples acquired. Acquiring a fully sampled grid takes a long time; to accelerate the acquisition, one can reduce the number of samples. That is, there is some data mask D (a matrix comprised of a subset of rows of the identity matrix) such that

$$b = D F x + \text{noise} \in \mathbb{C}^{nmp},$$

where the data is acquired on an $n \times m$ Cartesian grid and $p \in (0, 1]$ represents the undersampling factor corresponding to the number of rows of D .

Reconstructing the image given the acquired samples is then a noisy, underdetermined inverse problem. Common approaches to solving this problem include compressed sensing [26–28] and machine learning techniques [29, 30].

Matrix-free implementations of the algorithm and problem statement are desirable due to memory limitations and to take advantage of the Fast Fourier Transform. A parameter-free version of PDHG is also desirable so the MRI technologist does not have to set a parameter by hand which can prevent the reconstruction of a high-quality image within a reasonable number of iterations.

6.2 Compressed sensing with homodyne detection

Compressed sensing incorporates the *a priori* knowledge that natural images are approximately sparse after an appropriate linear transformation (e.g., a discrete Wavelet transform) to reconstruct high-quality images from fewer samples than those that would be required by the Nyquist-Shannon sampling theorem [23, 27]. Partial Fourier reconstruction incorporates the fact that the Fourier transform of a real image is Hermitian symmetric into the reconstruction process [31, 32]. Though MR images are not generally real due to variances in the B_0 magnetic field and non-linearities in the B_1 magnetic field, the phase is almost all low-bandwidth. Partial Fourier with homodyne detection takes advantage of this fact [24] by estimating the phase from a low-pass filtered version of the image. Once estimated, phase correction makes the data real and the assumptions of partial Fourier reconstruction are satisfied.

Partial Fourier sampling collects data with a sampling pattern consisting of the top ν portion, where $\nu > 0.5$. Common values of ν or $\nu = 5/8$ and $\nu = 9/16$. Homodyne

detection uses a low-pass filtered version of the image to estimate the phase, Φ . Once the phase is estimated, the image is reconstructed with homodyne detection using

$$x = P_{\Phi} b = \text{Re} [\Phi F^{-1} \mathcal{R} b],$$

where $\mathcal{R}$ is a weighting ramp that smoothly transitions from 2 to 0, $\Phi = \exp(-i \angle(M_s))$, $M_s = F^{-1} L b$, and L is a low-pass rectangular filter [33].

The crucial realization is that once the phase Φ is estimated, the P_{Φ} operator—which implements reconstruction with homodyne detection—is a linear transformation with a computationally efficient adjoint. This allows us to combine the phase estimate Φ into a compressed sensing image reconstruction as the following optimization problem:

$$\underset{\xi}{\text{minimize}} \quad \|\Psi P_{\Phi} \xi\|_1 \quad \text{subject to} \quad \|D F \xi - b\|_2 \leq \epsilon,$$

where ξ is a vector of the top ν Fourier values for the reconstructed image, Ψ is a sparsifying transformation, and $\epsilon > 0$ is a noise bound. We write this as an unconstrained problem by using an indicator function to enforce data consistency:

$$\xi^* \in \underset{\xi}{\text{argmin}} \quad \mathbb{I}(M \xi \in B_{\epsilon}[b]) + \|\Psi P_{\Phi} \xi\|_1, \quad (30)$$

where $\mathbb{I}$ is an indicator function that requires the Fourier values ξ at the measured spatial frequencies, specified by sampling mask D , to be within an ϵ ball of the collected data b . Setting $\epsilon = 0$ enforces data consistency completely. The second term contains the sparsifying transform Ψ and the homodyne operator P_{Φ} . By letting $A = \Psi P_{\Phi}$, $f(\xi) = \mathbb{I}(M \xi \in B_{\epsilon}[b])$ and $g(\xi) = \|\Psi P_{\Phi} \xi\|_1$, we see that this problem is of the form (5) so we solve it via rPDHG with line search. Once f^* is determined by solving (30), the image is reconstructed according to $x^* = P_{\Phi} \xi^*$. For our experiments, Ψ was the Daubechies-4 discrete wavelet transform [34].

6.2.1 Matrix-free implementation

For memory efficiency, we write $A = \Psi P_{\Phi}$ in its full, matrix-form implementation:

$$A = \Psi \circ \text{Re} \circ \Phi \circ F^{-1} \circ R \circ D^*. \quad (31)$$

Here $D^* : \mathbb{C}^{n \times m} \rightarrow \mathbb{C}^{n \times m}$ is a zero-filling operator. The linear transformation D selects only the indices at which data was collected (non-zero in the sampling mask) so D^* returns a zero-filled full-sized image. The linear transformation $R : \mathbb{C}^{n \times m} \rightarrow \mathbb{C}^{n \times m}$ applies a weighting matrix, $\mathcal{F} : \mathbb{C}^{n \times m} \rightarrow \mathbb{C}^{n \times m}$ is the two dimensional discrete Fourier transform, $\Phi : \mathbb{C}^{n \times m} \rightarrow \mathbb{C}^{n \times m}$ applies the phase estimate. $\text{Re} : \mathbb{C}^{n \times m} \rightarrow \mathbb{R}^{n \times m}$ takes the real part, and $\Psi : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}^{n \times m}$ is the orthogonal discrete wavelet transform.

If A were implemented as a matrix, we could choose $\theta < 1/\|A\|^2$ and let $B = (\frac{1}{\theta}\mathcal{I} - AA^*)^{1/2}$, where $(\cdot)^{1/2}$ indicates the Cholesky factorization of a matrix. In this case, $\frac{1}{\theta}\mathcal{I} - AA^*$ is symmetric and positive definite and the Cholesky factorization is well-defined. However, for this application, the implementation of A is matrix-free—we write it out in the form of (31) for analysis. Note that $AA^* = \Psi P_\Phi P_\Phi^* \Psi^*$ is *not* equal to a scaled identity. Therefore, to use the line search presented above for an image of reasonable size we need a matrix-free implementation of B such that $AA^* + BB^* = \frac{1}{\theta}\mathcal{I}$.

Let B be defined as $B = \Psi \circ \text{Re} \circ \Phi \circ F^{-1} \circ R \circ Q$, where $Q : \mathbb{C}^{n \times m} \rightarrow \mathbb{C}^{n \times m}$ is defined as $(\frac{1}{\theta}\mathcal{I} - RD^*DR)^{1/2}$. The operator $RD^*DR : \mathbb{C}^{n \times m} \rightarrow \mathbb{C}^{n \times m}$ is symmetric so Q is the symmetric square root. In practice, this square root is cheap to compute as the matrix is diagonal. This form of B was found by writing A in matrix form as in (31), forming AA^* and setting $BB^* = (1/\theta)(\mathcal{I} - AA^*)$. Note that R is symmetric. Similar to above, we write B out in its matrix form to aid in analysis, but B has a natural matrix-free implementation. With this definition of B , and understanding that we are using unitary Fourier and wavelet transformations, we can apply the primal–dual form of Douglas-Rachford to the problem (17) and recover the PDHG iterations.

6.3 Results

We first compare our proposed line search method against standard PDHG and PDHG with line search [1]. A variable density sampling mask with a sampling burden of 0.3 is chosen. This corresponds to Fig. 7B. Intersecting with the partial Fourier mask of Fig. 7A to get Fig. 7C yields an *effective* sampling burden of approximately 0.17. (I.e., the number of samples of the reduced sampling pattern divided by those required to satisfy the Nyquist-Shannon sampling theorem was approximately 0.17.) We solve the problem (30) using each method with the same sampling mask. Each method is run for 4000 iterations and the images from multiple coils are combined

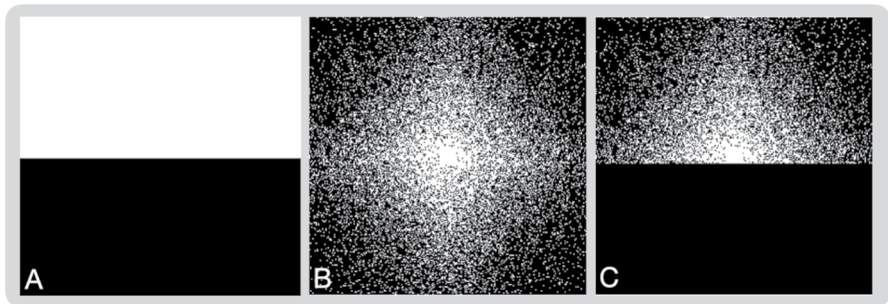


Fig. 7 Sampling masks for three different methods of acceleration, where white indicates collected samples. A) is the sampling mask associated with partial Fourier sampling, where the entire top half plus some number of lines around the origin are collected. B) is the sampling mask associated with compressed sensing using a variable density Poisson disc sampling mask [22]. C) is the sampling mask for the novel combination of partial Fourier and compressed sensing reconstruction presented in this manuscript. It is the intersection of masks A) and B)

into a single image using the method of Roemer [35]. Comparing the reconstructions on ankle data in Fig. 8 center and right shows that the proposed line search yields the same quality reconstruction without a lengthy parameter search. The difference images show that the primary differences are down in the fine structure near the toes and along other areas where the phase is changing rapidly.

Figure 9 is an example using brain data with the same undersampling pattern applied. On the right is the reconstruction using standard PDHG. In the center is the novel combination of compressed sensing with homodyne detection using the line search presented in this paper. Similar to the ankle data, the same quality reconstruction is achieved without a parameter search.

A comparison of the objective values for reconstruction of the ankle data with and without the line search is presented in Fig. 10. Let f_o be the objective function the algorithm is minimizing (e.g. (30)). For these plots, let x^* be the optimal point obtained after letting the algorithm run for ten thousand iterations. The objective value plotted in Fig. 10 is $f_o(x) - f_o(x^*)$, or the difference between the current iteration and optimal. As mentioned previously, there is no guarantee that each iterate is feasible. To calculate the objective value we first use the proximal operator of the indicator function for the constraint to make sure the iterate is feasible and then calculate the objective value. The data was run with three different methods: standard

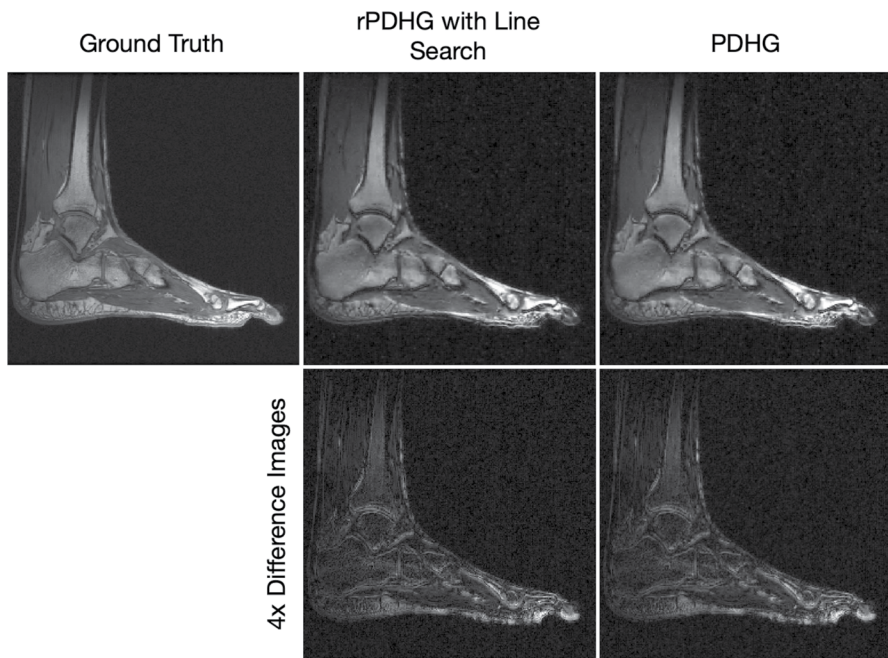


Fig. 8 Comparison of a fully-sampled reconstruction of a sagittal slice of an ankle to two methods of reconstruction from data undersampled according to sampling mask Fig. 7C. (Center) Reconstruction of the undersampled ankle data using the formulation of the problem in (30), solved with the line search presented in this manuscript. (Right) This reconstruction was made using the formulation of the problem in (30), solved with the PDHG method. Bottom row shows 4 times the magnitude difference between the accelerated reconstructions and the fully-sampled reconstruction

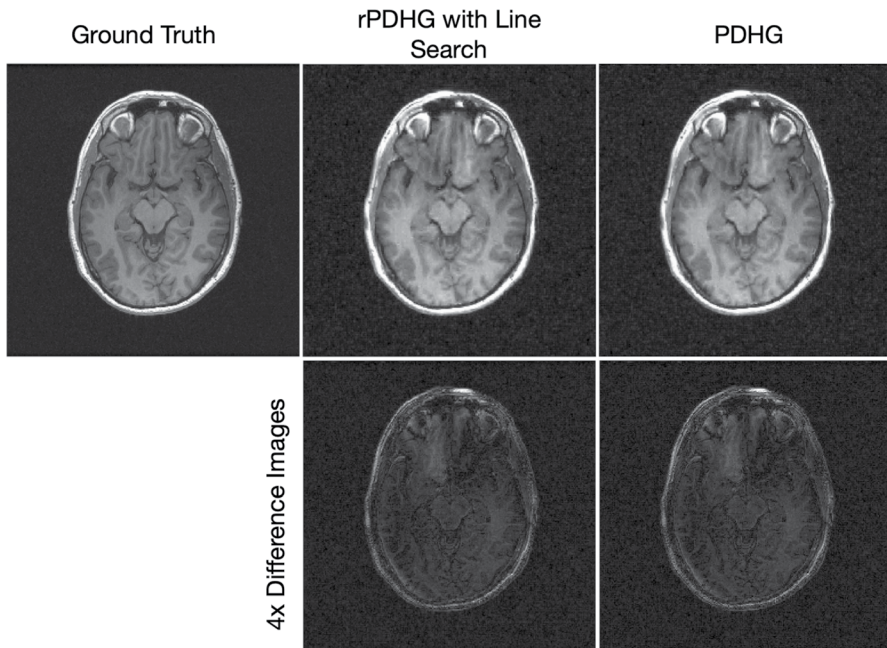


Fig. 9 Comparison of a fully-sampled reconstruction of an axial slice of a brain to reconstruction from data undersampled according to sampling mask Fig. 7C. (Center) Reconstruction of the undersampled ankle data using the formulation of the problem in (30), solved with the line search presented in this manuscript. (Right) Reconstruction of the undersampled ankle data using the formulation of the problem in (30), solved with the standard PDHG method given in (16). Bottom row shows 4 times the magnitude difference between the accelerated reconstructions and the fully-sampled reconstruction. The reconstruction using the new line search shows the same quality reconstruction without requiring a lengthy parameter search

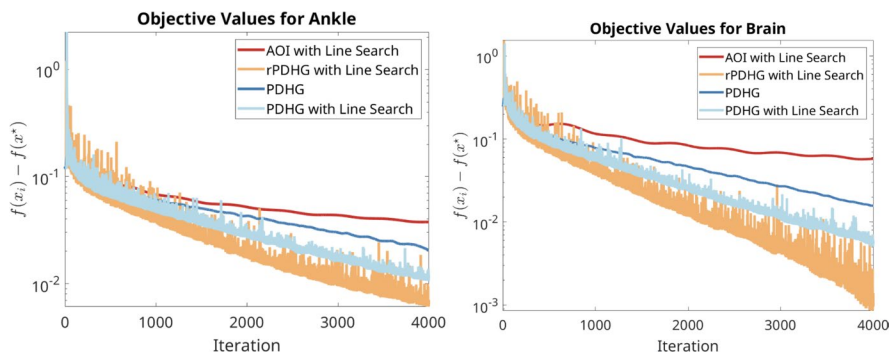


Fig. 10 Objective value plots for the reconstruction of the same ankle data as Fig. 8 (left) and the brain data as in Fig. 9 (right). This compares the performance of rPDHG algorithm with standard PDHG, PDHG with Malitsky's line search, and the averaged operator iteration line search applied to primal-dual Douglas Rachford. Over time, the line search finds a better solution. The line search jumps in objective value a lot as it chooses different step sizes. The objective value at each iteration is shown for academic purposes, though in practice at each iteration the best so far would be chosen

Table 1 An example of the total timing difference between the other algorithms tested and rPDHG with line search (wLS) given the need for a parameter search for other algorithms

Algorithm	Time of 1000 Iterations (s)	Time of Parameter Search (s)	Total time (s)
PDHG	38.44	653.53	691.97
PDHG wLS	98.37	1672.29	1770.66
AOI wLS	86.67	1473.39	1560.06
rPDHG wLS	417.8	0	417.8

The time to 1000 iterations given a set of parameters is longer for rPDHG with line search, but the total time is less

PDHG without the line search, PDHG with the Malitsky line search, and relaxed PDHG with the line search and the activation criterion discussed in Sec. 4.2. The objective of the relaxed PDHG algorithm jumps as different step sizes are accepted and converges to a solution better than the PDHG algorithm. The objective values for all iterations are shown for academic purposes; in practice, at any iteration, the result with the best objective value so far would be chosen. The best PDHG run is shown after a lengthy parameter search. The rPDHG run requires no such parameter search. Table 1 is an example of the timing difference given the parameter search. The time to 1000 iterations is longer for rPDHG but the time to find the correct set of parameters is zero.

On average, the run time per iteration of rPDHG with line search is approximately $10\times$ that of PDHG. However, as shown in Table 1, the benefits of rPDHG with line search are that it avoids a search of parameter values. Note that the inner line search over the step sizes is pleasingly parallelizable; additional processing cores permit a search over a larger number of line search parameter values without incurring additional processing time.

7 Conclusion

A line search for the primal–dual hybrid gradient method was proposed. Motivated by the connection between the Douglas-Rachford algorithm and the primal–dual hybrid gradient method, this line search expresses PDHG as an averaged operator iteration. An existing line search for averaged operator iterations is then applied to the relaxation parameter. This is combined with an existing line search over the proximal step sizes in PDHG. This new combination of line searches is essentially parameter free, allowing for quality solutions without a lengthy parameter search. Several numerical examples are demonstrated showing the convergence performance. The efficacy of the line search is presented on a novel combination of magnetic resonance imaging reconstruction techniques.

Appendix A: Computation of B for the MRI problem

For the compressed sensing with homodyne reconstruction problem, A is given as

$$A = \Psi \circ \text{Re} \circ \Phi \circ F^{-1} \circ R \circ D^*. \quad (28)$$

To use relaxed PDHG with line search, we need a B such that $AA^* + BB^* = \frac{1}{\theta}\mathcal{I}$ for some $\theta > 0$. From (28), we need

$$\begin{aligned} (\Psi \circ \text{Re} \circ \Phi \circ F^{-1} \circ R \circ D^*) \circ (\Psi \circ \text{Re} \circ \Phi \circ F^{-1} \circ R \circ D^*)^* + BB^* &= \frac{1}{\theta}\mathcal{I} \\ (\Psi \circ \text{Re} \circ \Phi \circ F^{-1} \circ R \circ D^*) \circ (D \circ R^* \circ (F^{-1})^* \circ \Phi^* \circ \text{Re}^* \circ \Psi^*) + BB^* &= \frac{1}{\theta}\mathcal{I}. \end{aligned} \quad (29)$$

If we write $B = \Psi \circ \text{Re} \circ \Phi \circ F^{-1} \circ Q$ for unknown Q , we can write (29) as

$$\begin{aligned} &\underbrace{(\Psi \circ \text{Re} \circ \Phi \circ F^{-1} \circ R \circ D^*)}_A \circ \underbrace{(D \circ R^* \circ (F^{-1})^* \circ \Phi^* \circ \text{Re}^* \circ \Psi^*)}_{A^*} \\ &+ \underbrace{(\Psi \circ \text{Re} \circ \Phi \circ F^{-1} \circ Q)}_B \circ \underbrace{(Q^* \circ (F^{-1})^* \circ \Phi^* \circ \text{Re}^* \circ \Psi^*)}_{B^*} = \frac{1}{\theta}\mathcal{I}. \end{aligned}$$

Factoring out the similar terms yields

$$(\Psi \circ \text{Re} \circ \Phi \circ F^{-1}) \circ (RD^*DR^* + QQ^*) \circ ((F^{-1})^* \circ \Phi^* \circ \text{Re}^* \circ \Psi^*) = \frac{1}{\theta}\mathcal{I}. \quad (30)$$

We are using the unitary Fourier transform F and orthogonal wavelet transform Ψ . The phase estimate Φ is unitary—it is a diagonal matrix of unit complex exponentials. The transpose of the Re operator is $\text{Re}^* : \mathbb{R}^{n \times m} \rightarrow \mathbb{C}^{n \times m}$ and maps $a \rightarrow a + 0i$, so $\text{Re} \circ \text{Re}^* : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}^{n \times m}$ is the identity operator. Therefore if $RD^*DR^* + QQ^* = \frac{1}{\theta}\mathcal{I}$ then (30) is satisfied. Noting that R is Hermitian, this condition becomes

$$QQ^* = \frac{1}{\theta}\mathcal{I} - RD^*DR. \quad (31)$$

The matrix $\frac{1}{\theta}\mathcal{I} - RD^*DR$ is diagonal so taking a symmetric square root is computationally efficient.

Acknowledgements The authors thank Daniel O'Connor for many useful conversations related to the work of this manuscript.

Author contributions Not applicable

Funding This study was funded by the University of Colorado School of Medicine.

Data availability Not applicable

Code availability Code is available at the following Github repository: <https://github.com/axmanmuscle/rPDHGwls>.

Materials availability Not applicable.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

Ethics approval and consent to participate Not applicable.

Consent for publication Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Malitsky, Y., Pock, T.: A first-order primal–dual algorithm with linesearch. *SIAM J. Optim.* **28**(1), 411–432 (2018)
2. Parikh, N., Boyd, S.: Proximal algorithms. *Found. Trends Optim.* **1**(3), 127–239 (2014)
3. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al.: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* **3**(1), 1–122 (2011)
4. Eckstein, J., Bertsekas, D.P.: On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Program.* **55**, 293–318 (1992)
5. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imag. Sci.* **2**(1), 183–202 (2009)
6. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imag. Vis.* **40**, 120–145 (2011)
7. O'Connor, D., Vandenberghe, L.: On the equivalence of the primal–dual hybrid gradient method and Douglas–Rachford splitting. *Math. Program.* **179**(1), 85–108 (2020)
8. Giselsson, P., Fält, M., Boyd, S.: Line search for averaged operator iteration. In: *IEEE 55th Conference on Decision and Control (CDC)*, pp. 1015–1022. IEEE (2016)
9. McManus, A., Becker, S.R., O'Connor, D., Dwork, N.: Compressed sensing with homodyne detection. In: *2024 IEEE Conference on Computational Imaging Using Synthetic Apertures (CISA)*, pp. 01–05. IEEE (2024)
10. Goldstein, T., Li, M., Yuan, X., Esser, E., Baraniuk, R.: Adaptive primal–dual hybrid gradient methods for saddle-point problems (2013). *arXiv preprint*
11. Pock, T., Chambolle, A.: Diagonal preconditioning for first order primal–dual algorithms in convex optimization. In: *2011 International Conference on Computer Vision*, pp. 1762–1769. IEEE (2011)
12. Bauschke, H.H., Combettes, P.L.: *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, Berlin (2011)
13. Condat, L.: A primal–dual splitting method for convex optimization involving Lipschitzian, Proximinal and linear composite terms. *J. Optim. Theory Appl.* **158**(2), 460–479 (2013)
14. Lions, P.-L., Mercier, B.: Splitting algorithms for the sum of two nonlinear operators. *SIAM J. Numer. Anal.* **16**(6), 964–979 (1979)
15. Bauschke, H.H., Moursi, W.M.: The Douglas–Rachford algorithm for two (not necessarily intersecting) affine subspaces. *SIAM J. Optim.* **26**(2), 968–985 (2016)
16. Pock, T., Cremers, D., Bischof, H., Chambolle, A.: An algorithm for minimizing the Mumford–Shah functional. In: *IEEE 12th International Conference on Computer Vision*, pp. 1133–1140. IEEE (2009)

17. Armijo, L.: Minimization of functions having Lipschitz continuous first partial derivatives. *Pac. J. Math.* **16**(1), 1–3 (1966)
18. Amsel, N., Chen, T., Greenbaum, A., Musco, C., Musco, C.: Nearly optimal approximation of matrix functions by the Lanczos method (2024). [arXiv:org/abs/2303.03358](https://arxiv.org/abs/2303.03358)
19. Li, L., Jamieson, K., Rostamizadeh, A., Gonina, K., Hardt, M., Recht, B., Talwalkar, A.: Massively parallel hyperparameter tuning (2018)
20. Chambolle, A.: An algorithm for total variation minimization and applications. *J. Math. Imag. Vis.* **20**, 89–97 (2004)
21. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physica D* **60**(1–4), 259–268 (1992)
22. Dwork, N., Baron, C.A., Johnson, E.M., O'Connor, D., Pauly, J.M., Larson, P.E.: Fast variable density Poisson-disc sample generation with directional variation for compressed sensing in MRI. *Magn. Reson. Imag.* **77**, 186–193 (2021)
23. Lustig, M., Donoho, D.L., Santos, J.M., Pauly, J.M.: Compressed sensing MRI. *IEEE Signal Process. Mag.* **25**(2), 72–82 (2008)
24. Noll, D.C., Nishimura, D.G., Macovski, A.: Homodyne detection in magnetic resonance imaging. *IEEE Trans. Med. Imaging* **10**(2), 154–163 (1991)
25. Nishimura, D.G.: *Principles of Magnetic Resonance Imaging*. Lulu.com (2010)
26. Lustig, M., Donoho, D., Pauly, J.M.: Sparse MRI: the application of compressed sensing for rapid MR imaging. *Mag. Resonan. Med.* **58**(6), 1182–1195 (2007)
27. Candès, E.J., Wakin, M.B.: An introduction to compressive sampling. *IEEE Signal Process. Mag.* **25**(2), 21–30 (2008)
28. Dwork, N., O'Connor, D., Baron, C.A., Johnson, E.M., Kerr, A.B., Pauly, J.M., Larson, P.E.: Utilizing the wavelet transform's structure in compressed sensing. *SIViP* **15**, 1407–1414 (2021)
29. Sandino, C.M., Cheng, J.Y., Chen, F., Mardani, M., Pauly, J.M., Vasanawala, S.S.: Compressed sensing: from research to clinical practice with deep neural networks: shortening scan times for magnetic resonance imaging. *IEEE Signal Process. Mag.* **37**(1), 117–127 (2020)
30. Knoll, F., Hammernik, K., Zhang, C., Moeller, S., Pock, T., Sodickson, D.K., Akcakaya, M.: Deep-learning methods for parallel magnetic resonance imaging reconstruction: a survey of the current approaches, trends, and issues. *IEEE Signal Process. Mag.* **37**(1), 128–140 (2020)
31. McGibney, G., Smith, M., Nichols, S., Crawley, A.: Quantitative evaluation of several partial Fourier reconstruction algorithms used in MRI. *Magn. Reson. Med.* **30**(1), 51–59 (1993)
32. Bydder, M., Robson, M.D.: Partial Fourier partially parallel imaging. *Mag. Resonan. Med.* **53**(6), 1393–1401 (2005)
33. Pauly, J.: Partial k-space reconstruction (2005). Preprint
34. Dwork, N., Larson, P.E.: Utilizing the structure of a redundant dictionary comprised of wavelets and curvelets with compressed sensing. *J. Electron. Imag.* **31**(6), 063043–063043 (2022)
35. Roemer, P.B., Edelstein, W.A., Hayes, C.E., Souza, S.P., Mueller, O.M.: The NMR phased array. *Magn. Reson. Med.* **16**(2), 192–225 (1990)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.