Journal of Scientific Computing (2021) 86:21
https://doi.org/10.1007/s10915-020-01371-1

Acceleration of Primal–Dual Methods by Preconditioning and Simple Subproblem Procedures

Yanli Liu[1] · Yunbei Xu[2] · Wotao Yin[1]

Received: 27 June 2020 / Revised: 30 September 2020 / Accepted: 13 November 2020 /
Published online: 7 January 2021

Abstract

Primal–dual hybrid gradient (PDHG) and alternating direction method of multipliers (ADMM) are popular first-order optimization methods. They are easy to implement and have diverse applications. As first-order methods, however, they are sensitive to problem conditions and can struggle to reach the desired accuracy. To improve their performance, researchers have proposed techniques such as diagonal preconditioning and inexact subproblems. This paper realizes additional speedup about one order of magnitude. Specifically, we choose general (non-diagonal) preconditioners that are much more effective at reducing the total numbers of PDHG/ADMM iterations than diagonal ones. Although the subproblems may lose their closed-form solutions, we show that it suffices to solve each subproblem approximately with a few proximal-gradient iterations or a few epochs of proximal block-coordinate descent, which are simple and have closed-form steps. Global convergence of this approach is proved when the inner iterations are fixed. Our method opens the choices of preconditioners and maintains both low per-iteration cost and global convergence. Consequently, on several typical applications of primal–dual first-order methods, we obtain 4–95× speedup over the existing state-of-the-art.

Keywords Primal–dual hybrid gradient · Alternating direction method of multipliers · Preconditioning · Acceleration

Mathematics Subject Classification 49M29 · 65K10 · 65Y20 · 90C25

✉ Yanli Liu
yanli@math.ucla.edu

Yunbei Xu
yunbei.xu@gsb.columbia.edu

Wotao Yin
wotaoyin@math.ucla.edu

[1] Department of Mathematics, University of California, Los Angeles, CA, USA

[2] Graduate School of Business, Columbia University, New York, NY, USA

1 Introduction

In this paper, we consider the following optimization problem:

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\; f(x)+g(Ax), \tag{1}$$

together with its dual problem:

$$\underset{z\in\mathbb{R}^m}{\text{minimize}}\; f^*(-A^Tz)+g^*(z), \tag{2}$$

where $f:\mathbb{R}^{\mathrm{n}}\to\mathbb{R}\cup\{+\infty\}$ and $g:\mathbb{R}^{\mathrm{m}}\to\mathbb{R}\cup\{+\infty\}$ are closed proper convex, $A\in\mathbb{R}^{m\times n}$, and f^*, g^* are the convex conjugates of f, g, respectively.

Formulations (1) or (2) abstracts many application problems, which include image restoration [59], magnetic resonance imaging [51], network optimization [23], computer vision [45], and earth mover's distance [35]. They can be solved by primal–dual algorithms such as primal–dual hybrid gradient (PDHG) and alternating direction method of multipliers (ADMM).

However, as first-order algorithms, PDHG and ADMM suffer from slow (tail) convergence. They may take thousands of iterations and still struggle reaching just four digits of accuracy. While they have many advantages such as being easy to implement and friendly to parallelization, their sensitivity to problem conditions is their main disadvantage.

To improve the performance of PDHG and ADMM, researchers have tried using preconditioners, which has been widely applied for forward-backward type of methods [10,16,53], as well as other methods [9,17,30,52]. Depending on the application and how one applies splitting, preconditioned PDHG and ADMM may or may not have subproblems with closed-form solutions. When they do not, researchers have studied approximate subproblem solutions to reduce the total running time. In this work, we propose a new way of applying preconditioning that outperforms the existing state-of-the-art.

1.1 Proposed Approach

Simply speaking, we find a way to use non-diagonal preconditioners (thus much fewer iterations) and still have very simple subproblem procedures (thus maintaining the advantages of PDHG and ADMM).

First, we present Preconditioned PDHG (PrePDHG) along with its convergence condition and a performance bound. We propose to choose preconditioners to optimize the bound. In the special case where one preconditioner is trivially fixed as an identity matrix, optimizing the bound gives us the optimal choice of the other preconditioner, which actually reduces PrePDHG to ADMM. This explains why ADMM often takes fewer iterations than PDHG.

Next, we study how to solve PrePDHG subproblems. In all applications we are aware of, only one of the two subproblem is (subject to) ill-conditioned. (After all, we can always apply splitting to gather ill-conditioned components into one subproblem.) Therefore, we choose a non-diagonal preconditioner for the ill-conditioned subproblem and a trivial or diagonal preconditioner for the other subproblem. Again, the pair of preconditioners should be chosen to (nearly) optimize the performance bound. Since the non-diagonal preconditioner introduces dependence between different coordinates, its subproblem generally does not have a closed-form solution. In particular, if the subproblem has an ℓ_1-norm, which is often the reason why PDHG or ADMM is used, it often loses its closed-form solution due to the preconditioner. Therefore, we propose to approximately solve it to satisfy an accuracy condition. Remarkably, there is no need to dynamically stops a subproblem procedure to honor the condition. Instead, the condition is automatically satisfied as long as one applies warm start and a common iterative procedure for some *fixed number* of iterations, which is new in the literature. Common choices of the procedure include proximal gradient descent, FISTA with restart, proximal block coordinate descent, and accelerated block-coordinate-gradient-descent (BCGD) methods (e.g., [1,28,37]). We call this method iPrePDHG (i for "inexact").

Next, we establish the overall convergence of iPrePDHG. To handle the inexact subproblem, we first transform iPrePDHG into an equivalent form and then analyze an Lyapunov function to establish convergence. The technique in our proof appears to be new in the PDHG and ADMM literature.

Finally, we apply our approach to a few applications including image denoising, graph cut, optimal transport, and CT reconstruction. For CT reconstruction, we use a diagonal preconditioner in one subproblem and a non-diagonal preconditioner in the other, which we approximately solve. In each of the other applications, one subproblem uses no preconditioner, and the other uses a non-diagonal preconditioner. Using these preconditioners, we observed iPrePDHG was 4–95 times faster than the existing state-of-the-art.

Since ADMM is a special PrePDHG with one trivial preconditioner, our approach also applies to ADMM. In fact, for three of the above four applications, there are one trivial preconditioner in each, so their iPrePDHG are inexact preconditioned ADMM.

1.2 Related Literature

The main references for PDHG are [11,22,59]. Many problems to which we apply PDHG have separable functions f or g, or both, so the resulting PDHG subproblems often (though not always) have closed-form solutions. When subproblems are simple, we care mainly about the convergence rate of PDHG, which depends on the problem conditioning. To accelerate PDHG, diagonal preconditioning [44] was proposed since its diagonal structure maintains closed-form solutions for the subproblems and, therefore, reduces iteration complexity without making each iteration more difficult. In comparison, non-diagonal preconditioners are much more effective at reducing iteration complexity, but their off-diagonal entries couple different components in the subproblems, causing the lost of closed-form solutions of subproblems.

When a PDHG subproblem has no closed-form solution, one often uses an iterative algorithm to approximately solve it. We call it Inexact PDHG. Under certain conditions, Inexact PDHG still converges to the exact solution. Specifically, Rasch and Chambolle [46] uses three different types of conditions to skillfully control the errors of the subproblems; all those errors need to be summable over all the iterations and thereby requiring the error to diminish asymptotically. In an interesting method from [6,8], one subproblem computes a proximal operator of a convex quadratic function, which can include a preconditioner and still has a closed-form solution involving matrix inversion. This proximal operator is successively applied n times in each iteration, for $n \geq 1$.

ADMM has different subproblems. One of its subproblems minimizes the sum of $f(x)$ and a squared term involving Ax. Only when A has special structures does the subproblem have closed-form solutions. Inexact ADMM refers to the ADMM with at least one of its subproblems inexactly solved. An *absolute error criterion* was introduced in [19], where the subproblem errors are controlled by a summable (thus diminishing) sequence of error tolerances. To simplify the choice of the sequences, a *relative error criterion* was adopted in several later works, where the subproblem errors are controlled by a single parameter multiplying certain quantities that one can compute during the iterations. In [40], the parameters need to be square summable. In [34], the parameters are constants when both objective functions are Lipschitz differentiable. In [20,21], two possible outcomes of the algorithm are described: (i) infinite outer loops and finite inner loops, and (ii) finite outer loops and the last inner loop is infinite, both guaranteeing convergence to a solution. On the other hand, it is unclear how to recognize them. Since there is no bound on the number of inner loops in case (i), one may recognize it as case (ii) and stop the algorithm before it converges.

There are works that apply certain kinds of preconditioning to accelerate ADMM. Paper [24] uses diagonal preconditioning and observes improved performance. After that, non-diagonal preconditioning is analyzed [6,8], which presents effective preconditioners for specific applications. One of their preconditioners needs to be inverted (though not needed in our method). Recently, preconditioning for problems with linear convergence has also been studied with promising numerical performances [25].

Finally, there is another line of work that combines Nesterov-type acceleration technique with primal–dual methods to obtain an accelerated convergence rate of $\mathcal{O}(1/k^2)$. This idea has been successfully applied to PDHG [11,13,31,39], ADMM [7,26,27,32], and linearized ADMM [41,56]. We would like to point out that their contributions are orthogonal to this paper. For example, in order to have simple subproblems in accelerated PDHG, preconditioning is not applied. For accelerated linearized ADMM, certain proximal terms have to be added in the subproblems to guarantee a closed-form solution.

1.3 Organization

The rest of this paper is organized as follows: Sect. 2 establishes notation and reviews basics. In the first part of Sect. 3, we provide a criterion for choosing preconditioners. In its second part, we introduce the condition for inexact subproblems, which can be automatically satisfied by iterating a fixed number of certain inner loops. This method is called iPrePDHG. In the last part of Sect. 3, we establish the convergence of iPrePDHG. Section 4 describes specific preconditioners and reports numerical results. Finally, Sect. 5 concludes the paper.

2 Preliminaries

We use $\|\cdot\|$ for ℓ_2-norm and $\langle\cdot,\cdot\rangle$ for dot product. We use I_n to denote the identity matrix of size $n \times n$. $M \succ 0$ means M is a symmetric, positive definite matrix, and $M \succeq 0$ means M is a symmetric, positive semidefinite matrix.

We write $\lambda_{\min}(M)$ and $\lambda_{\max}(M)$ as the smallest and the largest eigenvalues of M, respectively, and $\kappa(M) = \frac{\lambda_{\max}(M)}{\lambda_{\min}(M)}$ as the condition number of M. For $M \succeq 0$, let $\|\cdot\|_M$ and $\langle\cdot,\cdot\rangle_M$ denote the semi-norm and inner product induced by M, respectively. If $M \succ 0$, $\|\cdot\|_M$ is a norm.

For a proper closed convex function $\phi : \mathbb{R}^{\mathrm{n}} \to \mathbb{R} \cup \{+\infty\}$, its subdifferential at $x \in \mathbf{dom}\phi$ is written as

$$\partial\phi(x) = \{v \in \mathbb{R}^{\mathrm{n}} \mid \phi(z) \geq \phi(x) + \langle v, z - x\rangle, \ \ \forall z \in \mathbb{R}^{\mathrm{n}}\},$$

and its convex conjugate as

$$\phi^*(y) = \sup_{x\in\mathbb{R}^n} \{\langle y, x\rangle - \phi(x)\}.$$

We have $y \in \partial\phi(x)$ if and only if $x \in \partial\phi^*(y)$.

For any $M \succ 0$, we define the extended proximal operator of ϕ as

$$\mathrm{Prox}_\phi^M(x) := \arg\min_{y\in\mathbb{R}^n} \left\{ \phi(y) + \frac{1}{2}\|y-x\|_M^2 \right\}. \tag{3}$$

If $M = \gamma^{-1}I$ for $\gamma > 0$, it reduces to a classic proximal operator.

We also have the following generalization of Moreau's Identity:

Lemma 1 ([15], Theorem 3.1(ii)) *For any proper closed convex function ϕ and $M \succ 0$, we have*

$$x = \mathrm{Prox}_\phi^M(x) + M^{-1}\,\mathrm{Prox}_{\phi^*}^{M^{-1}}(Mx). \tag{4}$$

We say a proper closed function ϕ is a Kurdyka–Lojasiewicz (KL) function if, for each $x_0 \in \mathbf{dom}\phi$, there exist $\eta \in (0,\infty]$, a neighborhood U of x_0, and a continuous concave function $\varphi : [0,\eta) \to \mathbb{R}_+$ such that:

1. $\varphi(0) = 0$,
2. φ is C^1 on $(0,\eta)$,
3. for all $s \in (0,\eta)$, $\varphi'(s) > 0$,
4. for all $x \in U \cap \{x \mid \phi(x_0) < \phi(x) < \phi(x_0)+\eta\}$, the KL inequality holds:

$$\varphi'(\phi(x)-\phi(x_0))\mathrm{dist}(0,\partial\phi(x)) \geq 1.$$

3 Main Results

This section presents the key results of our paper. In Sect. 3.1 we demonstrate how to apply preconditioning to PDHG. Then, we establish rules of preconditioner selection in Sect. 3.2. In Sect. 3.3, we present the proposed method iPrePDHG. Finally, we establish the convergence of iPrePDHG in Sect. 3.4.

Throughout this section, we assume the following regularity assumptions:

Assumption 1

1. $f : \mathbb{R}^n \to \mathbb{R}\cup\{+\infty\}$, $g : \mathbb{R}^m \to \mathbb{R}\cup\{+\infty\}$ are proper closed convex.
2. A primal–dual solution pair $(x^\star, z^\star)$ of (1) and (2) exists, i.e.,

$$\mathbf{0} \in \partial f(x^\star) + A^T z^\star, \quad \mathbf{0} \in \partial g(Ax^\star) - z^\star.$$

Problem (1) also has the following convex-concave saddle-point formulation:

$$\min_{x\in\mathbb{R}^n}\max_{z\in\mathbb{R}^m} \varphi(x,z) := f(x) + \langle Ax, z\rangle - g^*(z). \tag{5}$$

A primal–dual solution pair $(x^\star, z^\star)$ is a solution of (5).

3.1 Preconditioned PDHG

The method of primal–dual hybrid gradient or PDHG [11,59] for solving (1) refers to the iteration

$$\begin{aligned} x^{k+1} &= \mathrm{Prox}_{\tau f}(x^k - \tau A^T z^k), \\ z^{k+1} &= \mathrm{Prox}_{\sigma g^*}(z^k + \sigma A(2x^{k+1} - x^k)). \end{aligned} \tag{6}$$

When $\frac{1}{\tau\sigma} \geq \|A\|^2$, the iterates of (6) converge [11] to a primal–dual solution pair of (1). We can generalize (6) by applying preconditioners $M_1, M_2 \succ 0$ (their choices are discussed below) to obtain Preconditioned PDHG or PrePDHG:

$$\begin{aligned} x^{k+1} &= \operatorname{Prox}_f^{M_1}\left(x^k - M_1^{-1}A^T z^k\right), \\ z^{k+1} &= \operatorname{Prox}_{g^*}^{M_2}\left(z^k + M_2^{-1}A(2x^{k+1} - x^k)\right), \end{aligned} \tag{7}$$

where the extended proximal operators $\operatorname{Prox}_f^{M_1}$ and $\operatorname{Prox}_{g^*}^{M_2}$ are defined in (3). We can obtain the convergence of PrePDHG using the analysis in [12].

There is no need to compute M_1^{-1} and M_2^{-1} since (7) is equivalent to

$$\begin{aligned} x^{k+1} &= \underset{x\in\mathbb{R}^{\mathrm{n}}}{\arg\min}\left\{f(x) + \langle x - x^k, A^T z^k\rangle + \frac{1}{2}\|x - x^k\|_{M_1}^2\right\}, \\ z^{k+1} &= \underset{z\in\mathbb{R}^{\mathrm{m}}}{\arg\min}\left\{g^*(z) - \langle z - z^k, A(2x^{k+1} - x^k)\rangle + \frac{1}{2}\|z - z^k\|_{M_2}^2\right\}. \end{aligned} \tag{8}$$

3.2 Choice of Preconditioners

In this section, we discuss how to select appropriate preconditioners M_1 and M_2. As a by-product, we show that ADMM corresponds to choosing $M_1 = \frac{1}{\tau}I_n$ and optimally choosing $M_2 = \tau AA^T$, thereby, explaining why ADMM appears to be faster than PDHG.

The following well-known lemma characterizes primal–dual solution pairs of (1) and (2). For completeness, we included its proof in “Appendix A”.

Lemma 2 *Under Assumption 1, (X, Z) is a primal–dual solution pair of (1) if and only if $\varphi(X, z) - \varphi(x, Z) \leq 0$ for any $(x, z) \in \mathbb{R}^{n+m}$, where φ is given in the saddle-point formulation (5).*

We present the following ergodic convergence result, adapted from [12, Theorem 1].

Theorem 1 *Let (x^k, z^k), $k = 0, 1, \ldots, N$ be a sequence generated by PrePDHG (7). Under Assumption 1, if in addition*

$$\tilde{M} := \begin{pmatrix} M_1 & -A^T \\ -A & M_2 \end{pmatrix} \succeq 0, \tag{9}$$

then, for any $x \in \mathbb{R}^{\mathrm{n}}$ and $z \in \mathbb{R}^{\mathrm{m}}$, it holds that

$$\varphi(X^N, z) - \varphi(x, Z^N) \leq \frac{1}{2N}(x - x^0, z - z^0)\begin{pmatrix} M_1 & -A^T \\ -A & M_2 \end{pmatrix}\begin{pmatrix} x - x^0 \\ z - z^0 \end{pmatrix}, \tag{10}$$

where $X^N = \frac{1}{N}\sum_{i=1}^N x^i$ and $Z^N = \frac{1}{N}\sum_{i=1}^N z^i$.

Proof This follows from Theorem 1 of [12] by setting $L_f = 0$, $\frac{1}{\tau}D_x(x, x_0) = \frac{1}{2}\|x - x^0\|_{M_1}^2$, $\frac{1}{\sigma}D_z(z, z_0) = \frac{1}{2}\|z - z^0\|_{M_2}^2$, and $K = A$. Note that in Remark 1 of [12], D_x and D_z need to be 1−strongly convex to ensure their inequality (13) holds, which is exactly our (9). Therefore, we do not need D_x and D_z to be strongly convex. □

Based on the above results, one approach to accelerate convergence is to choose preconditioners M_1 and M_2 to obey (9) and make the right-hand side of (10) as small as possible for all x, z, x_0, z_0. When a pair of preconditioner matrices attains this minimum, we say they are optimal. When one of them is fixed, the other that attains the minimum is also called optimal.

By Schur complement, the condition (9) is equivalent to $M_2 \succeq AM_1^{-1}A^T$. Hence, for any given $M_1 \succ 0$, the optimal M_2 is $AM_1^{-1}A^T$.[1]

Original PDHG (6) corresponds to $M_1 = \frac{1}{\tau}I_n$, $M_2 = \frac{1}{\sigma}I_m$ with τ and σ obeying $\frac{1}{\tau\sigma} \geq \|A\|^2$ for convergence. In "Appendix B", we show that ADMM for problem (1) corresponds to setting $M_1 = \frac{1}{\tau}I_n,\ M_2 = \tau AA^T$, M_2 is optimal since $AM_1^{-1}A^T = \tau AA^T = M_2$ (This is related to, but different from, the result in [11, Sect. 4.3] stating that PDHG is equivalent to a preconditioned ADMM). In the next section, we show that when the $z-$subproblem is solved inexactly, a choice of $M_1 = \frac{1}{\tau}I_n$, $M_2 = \tau AA^T + \theta I_m$ with a small θ guarantees convergence (see Proposition 2).

By using more general pairs of M_1, M_2, we can potentially have even fewer iterations of PrePDHG than ADMM.

3.3 PrePDHG with Fixed Inner Iterations

It wastes time to solve the subproblems in (8) very accurately. It is more efficient to develop a proper condition and stop the subproblem procedure, i.e., *inner iterations*, once the condition is satisfied. It is even better if we can simply fix the number of inner iterations and still guarantee global convergence.

In this subsection, we describe the "bounded relative error" of the z-subproblem in (7) and then show that this can be satisfied by running a fixed number of inner iterations with warm start, uniformly for every outer loop, which is new in the literature.

Definition 1 (*Bounded relative error condition*) Given x^k, x^{k+1} and z^k, we say that the z-subproblem in PrePDHG (7) is solved to a bounded relative error by some iterator S, if there is a constant $c > 0$ such that

$$\mathbf{0} \in \partial g^*(z^{k+1}) + M_2\left(z^{k+1} - z^k - M_2^{-1}A(2x^{k+1} - x^k)\right) + \varepsilon^{k+1}, \tag{11}$$

$$\|\varepsilon^{k+1}\| \leq c\|z^{k+1} - z^k\|. \tag{12}$$

Remarkably, this condition does not need to be checked at run time. For a fixed $c > 0$, the condition can be satisfied by apply warm start and a fixed number of inner iterations using, for example, S being the proximal gradient iteration (Theorem 2). One can also use faster solvers, e.g., FISTA with restart [42], and solvers that suit the subproblem structure, e.g., cyclic proximal BCD (Theorem 3). Although the error in solving z-subproblems appears to be neither summable nor square summable, convergence can still be established. But first, we summarize this method in Algorithm 1.

Theorem 2 *Take Assumption 1. Suppose in iPrePDHG, or Algorithm 1, we choose S as the proximal-gradient step with stepsize $\gamma \in (0, \frac{2\lambda_{\min}(M_2)}{\lambda_{\max}^2(M_2)})$ and repeat it p times, where $p \geq 1$. Then, $z^{k+1} = z_p^{k+1}$ is an approximate solution to the z-subproblem up to a bounded relative error in Definition 1 for*

$$c = c(p) = \frac{\frac{1}{\gamma} + \lambda_{\max}(M_2)}{1 - \rho^p}(\rho^p + \rho^{p-1}), \tag{13}$$

where $\rho = \sqrt{1 - \gamma(2\lambda_{\min}(M_2) - \gamma\lambda_{\max}^2(M_2))} < 1$.

[1] Let $M_2 = AM_1^{-1}A^T + M_0$, where $M_0 \succeq 0$, then the right-hand side of Eq. (10) is minimized when $M_0 = 0$.

Algorithm 1 Inexact Preconditioned PDHG or iPrePDHG

Input: f, g, A in (1), preconditioners M_1 and M_2, initial (x_0, z_0), z-subproblem iterator S, inner iteration number p, max outer iteration number K.
Output: (x^K, z^K)

```
for $k \leftarrow 0, 1, \ldots, K-1$ do
    $x^{k+1} = \text{Prox}_f^{M_1}(x^k - M_1^{-1}A^T z^k)$;
    $z_0^{k+1} = z^k$;
    for $i \leftarrow 0, 1, \ldots, p-1$ do
        $z_{i+1}^{k+1} = S(z_i^{k+1}, x^{k+1}, x^k)$;
    end for
    $z^{k+1} = z_p^{k+1}$;        ▷ which approximates $\text{Prox}_{g^*}^{M_2}(z^k + M_2^{-1}A(2x^{k+1} - x^k))$
end for
```

Proof The z-subproblem in (8) is of the form

$$\underset{z \in \mathbb{R}^m}{\text{minimize}}\, h_1(z) + h_2(z), \tag{14}$$

for $h_1(z) = g^*(z)$ and $h_2(z) = \frac{1}{2}\|z - z^k - M_2^{-1}A(2x^{k+1} - x^k)\|_{M_2}^2$. With our choice of S as the proximal-gradient descent step, the inner iterations are

$$\begin{aligned} z_0^{k+1} &= z^k, \\ z_{i+1}^{k+1} &= \text{Prox}_{\gamma h_1}\left(z_i^{k+1} - \gamma \nabla h_2\left(z_i^{k+1}\right)\right), \quad i = 0, 1, \ldots, p-1, \end{aligned} \tag{15}$$

Concerning the last iterate $z^{k+1} = z_p^{k+1}$, we have from the definition of $\text{Prox}_{\gamma h_1}$ that

$$\mathbf{0} \in \partial h_1\left(z_p^{k+1}\right) + \nabla h_2\left(z_{p-1}^{k+1}\right) + \frac{1}{\gamma}\left(z_p^{k+1} - z_{p-1}^{k+1}\right).$$

Compare this with (11) and use $z^{k+1} = z_p^{k+1}$ to get

$$\varepsilon^{k+1} = \frac{1}{\gamma}\left(z_p^{k+1} - z_{p-1}^{k+1}\right) + \nabla h_2\left(z_{p-1}^{k+1}\right) - \nabla h_2\left(z_p^{k+1}\right).$$

It remains to show that ε^{k+1} satisfies (12).

Let $z_\star^{k+1}$ be the solution of (14), $\alpha = \lambda_{\min}(M_2)$, and $\beta = \lambda_{\max}(M_2)$. Then $h_1(z)$ is convex and $h_2(z)$ is α-strongly convex and β-Lipschitz differentiable. Consequently, [3, Prop. 26.16(ii)] gives

$$\left\|z_i^{k+1} - z_\star^{k+1}\right\| \leq \rho^i \left\|z_0^{k+1} - z_\star^{k+1}\right\|, \quad \forall i = 0, 1, \ldots, p,$$

where $\rho = \sqrt{1 - \gamma(2\alpha - \gamma\beta^2)}$.

Let $a_i = \|z_i^{k+1} - z_\star^{k+1}\|$. Then, $a_i \leq \rho^i a_0$. We can derive

$$\begin{aligned} \|\varepsilon^{k+1}\| &\leq \left(\frac{1}{\gamma} + \beta\right)\left\|z_p^{k+1} - z_{p-1}^{k+1}\right\| \leq \left(\frac{1}{\gamma} + \beta\right)(a_p + a_{p-1}) \\ &\leq \left(\frac{1}{\gamma} + \beta\right)(\rho^p + \rho^{p-1})a_0. \end{aligned} \tag{16}$$

On the other hand, we have

$$\|z^{k+1} - z^k\| \geq a_0 - a_p \geq (1 - \rho^p)a_0. \tag{17}$$

Combining these two equations yields

$$\|\varepsilon^{k+1}\| \leq c\|z^{k+1} - z^k\|,$$

where c is given in (13). □

Theorem 2 uses the iterator S that is the proximal-gradient step. It is straightforward to extend its proof to S being the FISTA step with restart since it is also linearly convergent [42]. We omit the proof.

In our next theorem, we let S be the iterator of one epoch of the cyclic proximal BCD method. A BCD method updates one block of coordinates at a time while fixing the remaining blocks. In one epoch of cyclic BCD, all the blocks of coordinates are sequentially updated, and every block is updated once. In cyclic *proximal* BCD, each block of coordinates is updated by a proximal-gradient step, just like (15) except only the chosen block is updated each time. When h_1 is block separable, each update costs only a fraction of updating all the blocks together. When different blocks are updated one after another, the Gauss–Seidel effect brings more progress. In addition, since the Lipschitz constant of each block gradient of h_2 is typically less than than that of ∇h_2, one can use a larger stepsize γ and get potentially even faster progress. Therefore, the iterator of cyclic proximal BCD is a better choice for S.

In summary, with $h_1(z) = g^*(z)$ and $h_2(z) = \frac{1}{2}\|z - z^k - M_2^{-1}A(2x^{k+1} - x^k)\|_{M_2}^2$, an epoch of cyclic proximal BCD for the $z-$subproblem is written as

$$\begin{aligned} z_0^{k+1} &= z^k, \\ z_{i+1}^{k+1} &= S\left(z_i^{k+1}, x^{k+1}, x^k\right), \quad i = 0, 1, \ldots, p-1, \\ z^{k+1} &= z_p^{k+1}. \end{aligned}$$

where S is the iterator of cyclic proximal BCD. Define

$$T(z) := \mathrm{Prox}_{\gamma h_1(z)}(z - \gamma \nabla h_2(z)), \qquad B(z) := \frac{1}{\gamma}(z - T(z)),$$

and the jth coordinate operator of B:

$$B_j(z) = (0, \ldots, (B(z))_j, \ldots, 0), \quad j = 1, 2, \ldots, l.$$

Then, we have

$$z_{i+1}^{k+1} = S\left(z_i^{k+1}, x^{k+1}, x^k\right) = (I - \gamma B_l)(I - \gamma B_2)\ldots(I - \gamma B_1)z_i^{k+1}.$$

Theorem 3 *Let Assumption 1 hold and g be block separable, i.e., $z = (z_1, z_2, \ldots, z_l)$ and $g(z) = \sum_{j=1}^{l} g_j(z_j)$. Suppose in iPrePDHG, or Algorithm 1, we choose S as the iterator of cyclic proximal BCD with stepsize γ satisfying*

$$0 < \gamma \leq \min\left\{\frac{2\lambda_{\min}(M_2))}{\lambda_{\max}^2(M_2))}, \frac{1 - \sqrt{1 - \gamma\left(2\lambda_{\min}(M_2) - \gamma\lambda_{\max}^2(M_2)\right)}}{4\sqrt{2}\gamma l \lambda_{\max}(M_2)}, \right.$$

$$\left. \frac{1}{4l\lambda_{\max}(M_2)}, \frac{2l\lambda_{\max}(M_2)}{17l\lambda_{\max}(M_2) + 2\left(\frac{1 - \sqrt{1 - \gamma\left(2\lambda_{\min}(M_2) - \gamma\lambda_{\max}^2(M_2)\right)}}{\gamma}\right)^2}\right\},$$

and we set $p \geq 1$. Then, $z^{k+1} = z_p^{k+1}$ is an approximate solution to the z-subproblem up to a bounded relative error in Definition 1 for

$$c = c(p) = \frac{\left(l\lambda_{\max}(M_2) + \frac{1}{\gamma}\right)(\rho^p + \rho^{p-1})}{1 - \rho^p}, \tag{18}$$

where $\rho = 1 - \frac{\left(1-\sqrt{1-\gamma(2\lambda_{\min}(M_2)-\gamma\lambda^2_{\max}(M_2))}\right)^2}{2\gamma} < 1.$

Proof See "Appendix C". □

3.4 Global Convergence of iPrePDHG

In this subsection, we show the convergence of Algorithm 1. Our approach first transforms Algorithm 1 into an equivalent algorithm in Proposition 1 below and then proves its convergence in Theorems 5 and 6 below.

First, let us show that PrePDHG (7) is equivalent to an algorithm applied on the dual problem (2). This equivalence is analogous to the equivalence between PDHG (6) and Linearized ADMM applied to the dual problem (2), shown in [22]). Specifically, PrePDHG is equivalent to

$$\begin{aligned} z^{k+1} &= \mathrm{Prox}_{g^*}^{M_2}\left(z^k + M_2^{-1}AM_1^{-1}\left(-A^T z^k - y^k + u^k\right)\right), \\ y^{k+1} &= \mathrm{Prox}_{f^*}^{M_1^{-1}}(u^k - A^T z^{k+1}), \\ u^{k+1} &= u^k - A^T z^{k+1} - y^{k+1}. \end{aligned} \tag{19}$$

When $M_1 = \frac{1}{\tau}I$, $M_2 = \lambda I$, (19) reduces to Linearized ADMM, also known as Split Inexact Uzawa [58].

Furthermore, iPrePDHG in Algorithm 1 is equivalent to (19) with inexact subproblems, which we present in Algorithm 2.

Algorithm 2 Inexact Preconditioned ADMM

```
Input: $f, g, A$ in (1), preconditioners $M_1$ and $M_2$,
initial vector $(z_0, y_0, u_0)$, subproblem solver $S$ for the $z$-subproblem in (19), number of inner loops $p$, number of outer iterations $K$.
Output: $(z^K, y^K, u^K)$
for $k \leftarrow 0, 1, \ldots, K-1$ do
    $z_0^{k+1} = z^k$;
    for $i \leftarrow 0, 1, \ldots, p-1$ do
        $z_{i+1}^{k+1} = S(z_i^{k+1}, y^k, u^k)$;
    end for
    $z^{k+1} = z_p^{k+1}$;    ▷ approximate $\mathrm{Prox}_{g^*}^{M_2}(z^k + M_2^{-1}AM_1^{-1}(-A^T z^k - y^k + u^k))$.
    $y^{k+1} = \mathrm{Prox}_{f^*}^{M_1^{-1}}(u^k - A^T z^{k+1})$;
    $u^{k+1} = u^k - A^T z^{k+1} - y^{k+1}$;
end for
```

Proposition 1 *Under Assumption 1 and the transforms $u^k = M_1 x^k$, $y^{k+1} = u^k - A^T z^k - u^{k+1}$, PrePDHG (7) is equivalent to (19), and iPrePDHG in Algorithm 1 is equivalent to Algorithm 2.*

Proof Set $u^k = M_1 x^k$, $y^{k+1} = u^k - A^T z^k - u^{k+1}$. Then (4) and (7) yield

$$y^{k+1} = M_1 x^k - A^T z^k - M_1 x^{k+1} = \mathrm{Prox}_{f^*}^{M_1^{-1}}(u^k - A^T z^k),$$

and

$$\begin{aligned} u^{k+1} &= u^k - A^T z^k - y^{k+1}, \\ z^{k+1} &= \mathrm{Prox}_{g^*}^{M_2}\left(z^k + M_2^{-1} A M_1^{-1}(-A^T z^k - y^{k+1} + u^{k+1})\right). \end{aligned}$$

If the z-update is performed first, then we arrive at (19).

In iPrePDHG or Algorithm 1, we are solving the z-subproblem of PrePDHG (7) approximately to the relative error in Definition 1. This is equivalent to doing the same to the z-subproblem of (19), which yields Algorithm 2. □

Let us define the following generalized augmented Lagrangian:

$$L(z, y, u) = g^*(z) + f^*(y) + \left\langle -A^T z - y, M_1^{-1} u \right\rangle + \frac{1}{2}\|A^T z + y\|^2_{M_1^{-1}}. \tag{20}$$

Inspired by [54], we use (20) as the Lyapunov function to establish convergence of Algorithm 2 and, equivalently, the convergence of Algorithm 1. This appears to be a new proof technique for inexact PDHG and inexact ADMM.

We first establish subsequential convergence of iPrePDHG in Algorithm 1 under the following additional assumptions.

Assumption 2

1. $f(x)$ is μ_f-strongly convex.
2. $g^*(z) + f^*(-A^T z)$ is coercive, i.e., $\lim_{\|z\|\to\infty} g^*(z) + f^*(-A^T z) = \infty$.

To establish convergence of iPrePDHG in Algorithm 1, we also need the following assumption.

Assumption 3 $L(z, y, u)$ is a KL function.

Assumption 3 is true when both $g^*(z)$ and $f^*(y)$ are semi-algebraic, or more generally, definable in an o-minimal structure (more details can be referred to Sect. 2.2 of [2] and Sect. 2.2 of [56] and the references therein).

Note that under Assumptions 2 and 3, it is not necessarily true that PDHG is linearly convergent.[2]

Theorem 4 *Take Assumptions 1 and 2. Choose any preconditioners M_1, M_2 and inner iteration number p such that*

$$C_1 = \frac{1}{2}M_1^{-1} - \frac{\lambda_{max}(M_1)}{\mu_f^2} I_n \succ 0, \tag{21}$$

$$C_2 = M_2 - \frac{1}{2}A M_1^{-1} A^T - c(p) I_m \succ 0, \tag{22}$$

where $c(p)$ depends on the z-subproblem iterator S and M_2 (e.g., (13) and (18)). Define $L^k := L(z^k, y^k, u^k)$. Then, Algorithm 2 satisfies the following sufficient descent and lower boundedness properties, respectively:

$$L^k - L^{k+1} \geq \|y^k - y^{k+1}\|^2_{C_1} + \|z^k - z^{k+1}\|^2_{C_2}, \tag{23}$$

[2] A counterexample can be found at Sect. 6.2.1 of [11]. The ROF functional satisfy these assumptions, and PDHG(ALG1) has $\mathcal{O}(1/N)$ convergence in Table 1. This is also mentioned on page 26 of [11].

$$L^k \geq g^*(z^\star) + f^*(-A^T z^\star) > -\infty. \tag{24}$$

Proof Since the z-subproblem of Algorithm 2 is solved to the bounded relative error in Def. 1, we have

$$\mathbf{0} \in \partial g^*(z^{k+1}) + M_2 \left(z^{k+1} - z^k - M_2^{-1} A M_1^{-1}(-A^T z^k - y^k + u^k) \right) + \varepsilon^{k+1}, \tag{25}$$

where ε^{k+1} satisfies (12):

$$\|\varepsilon^{k+1}\| \leq c(p) \|z^{k+1} - z^k\|. \tag{26}$$

The y and u updates produce

$$\mathbf{0} = \nabla f^*(y^{k+1}) + M_1^{-1}(y^{k+1} - u^k + A^T z^{k+1}) = \nabla f^*(y^{k+1}) - M_1^{-1} u^{k+1}, \tag{27}$$

$$u^{k+1} = u^k - A^T z^{k+1} - y^{k+1}. \tag{28}$$

In order to show (23), let us write

$$\begin{aligned} g^*(z^k) &\geq g^*(z^{k+1}) \\ &\quad + \left\langle M_2(z^k - z^{k+1}) + A M_1^{-1}(-A^T z^k - y^k + u^k) - \varepsilon^{k+1}, z^k - z^{k+1} \right\rangle, \\ f^*(y^k) &\geq f^*(y^{k+1}) + \left\langle M_1^{-1} u^{k+1}, y^k - y^{k+1} \right\rangle. \end{aligned}$$

Assembling these inequalities with (26) gives us

$$\begin{aligned} L^k - L^{k+1} \geq\ & \|z^k - z^{k+1}\|^2_{M_2 - c(p) I_m} \\ & + \left\langle A M_1^{-1}(-A^T z^k - y^k + u^k), z^k - z^{k+1} \right\rangle + \left\langle M_1^{-1} u^{k+1}, y^k - y^{k+1} \right\rangle \\ & + \left\langle -A^T z^k - y^k, M_1^{-1} u^k \right\rangle \\ & - \left\langle A^T z^{k+1} - y^{k+1}, M_1^{-1}(u^k - A^T z^{k+1} - y^{k+1}) \right\rangle \\ & + \frac{1}{2} \|A^T z^k + y^k\|^2_{M_1^{-1}} - \frac{1}{2} \|A^T z^{k+1} + y^{k+1}\|^2_{M_1^{-1}} \end{aligned} \tag{29}$$

$$= \|z^k - z^{k+1}\|^2_{M_2 - c(p) I_m} \tag{A}$$

$$+ \left\langle A M_1^{-1}(-A^T z^k - y^k), z^k - z^{k+1} \right\rangle + \left\langle M_1^{-1} u^{k+1}, y^k - y^{k+1} \right\rangle$$

$$+ \left\langle -y^k, M_1^{-1} u^k \right\rangle - \left\langle -y^{k+1}, M_1^{-1} u^k \right\rangle \tag{B}$$

$$+ \frac{1}{2} \|A^T z^k + y^k\|^2_{M_1^{-1}} - \frac{3}{2} \|A^T z^{k+1} + y^{k+1}\|^2_{M_1^{-1}},$$

where the terms in (A) and (B) simplify to

$$\left\langle A M_1^{-1}(-A^T z^k - y^k), z^k - z^{k+1} \right\rangle + \left\langle M_1^{-1}(-A^T z^{k+1} - y^{k+1}), y^k - y^{k+1} \right\rangle. \tag{30}$$

Apply the following cosine rule on the two inner products above:

$$\langle a - b, a - c \rangle_{M_1^{-1}} = \frac{1}{2} \|a - b\|^2_{M_1^{-1}} + \frac{1}{2} \|a - c\|^2_{M_1^{-1}} - \frac{1}{2} \|b - c\|_{M_1^{-1}}.$$

Set $a = A^T z^k$, $c = A^T z^{k+1}$, and $b = -y^k$ to obtain

$$\begin{aligned} &\left\langle AM_1^{-1}(-A^T z^k - y^k), z^k - z^{k+1} \right\rangle && (31)\\ &= -\frac{1}{2}\|A^T z^k + y^k\|^2_{M_1^{-1}} - \frac{1}{2}\|A^T z^k - A^T z^{k+1}\|^2_{M_1^{-1}} \\ &\quad + \frac{1}{2}\|y^k + A^T z^{k+1}\|^2_{M_1^{-1}}. && (32) \end{aligned}$$

Set $a = y^{k+1}$, $c = y^k$, and $b = -A^T z^{k+1}$ to obtain

$$\begin{aligned} &\left\langle M_1^{-1}(-A^T z^{k+1} - y^{k+1}), y^k - y^{k+1} \right\rangle && (33)\\ &= \frac{1}{2}\|A^T z^{k+1} + y^{k+1}\|^2_{M_1^{-1}} + \frac{1}{2}\|y^k - y^{k+1}\|_{M_1^{-1}} \\ &\quad - \frac{1}{2}\|A^T z^{k+1} + y^k\|^2_{M_1^{-1}}. && (34) \end{aligned}$$

Combining (30), (32), and (34) yields

$$\begin{aligned} L^k - L^{k+1} \geq \|z^k - z^{k+1}\|^2_{M_2 - \frac{1}{2}AM_1^{-1}A^T - c(p)I_m} + \|y^k - y^{k+1}\|^2_{\frac{1}{2}M_1^{-1}} \\ - \|A^T z^{k+1} + y^{k+1}\|^2_{M_1^{-1}}. \qquad (35) \end{aligned}$$

Since f is μ_f-strongly convex, we know that ∇f^* is $\frac{1}{\mu_f}-$Lipschitz continuous. Consequently,

$$\begin{aligned} \|A^T z^{k+1} + y^{k+1}\|^2_{M_1^{-1}} = \|u^k - u^{k+1}\|^2_{M_1^{-1}} &\leq \frac{1}{\lambda_{\min}(M_1^{-1})} \left\| M_1^{-1}(u^k - u^{k+1}) \right\|^2 \\ &\overset{(27)}{\leq} \frac{\lambda_{\max}(M_1)}{\mu_f^2}\|y^k - y^{k+1}\|^2. \qquad (36) \end{aligned}$$

Combining (35) and (36) gives us (23).

Now, to show (24), we use (27) and smoothness of f^* to get

$$f^*(y^k) \geq f^*(-A^T z^k) + \left\langle M_1^{-1}u^k, y^k + A^T z^k \right\rangle - \frac{1}{2\mu_f}\|A^T z^k + y^k\|^2.$$

Hence, we arrive at

$$\begin{aligned} L^k &= g^*(z^k) + f^*(y^k) + \left\langle -A^T z^k - y^k, M_1^{-1}u^k \right\rangle + \frac{1}{2}\|A^T z^k + y^k\|^2_{M_1^{-1}} \\ &\geq g^*(z^k) + f^*(-A^T z^k) + \frac{1}{2}\|A^T z^k + y^k\|^2_{M_1^{-1}} - \frac{1}{2\mu_f}\|A^T z^k + y^k\|^2. \qquad (37) \end{aligned}$$

Since $C_1 \succ 0$ if and only if $\mu_f > \sqrt{2}\lambda_{\max}(M_1)$, (24) follows. □

Remark 1 In Theorem 4, we require $C_2 = M_2 - \frac{1}{2}AM_1^{-1}A^T - c(p)I_m \succ 0$. Recall that p is the number of inner loops applied to solve the $z-$subproblem and $c(p)$ converges linearly to 0. Therefore, if we apply a smaller p, then M_2 needs to be larger. This means that the dual update needs to use a smaller effective stepsize.

Next, we provide a simple choice of M_1, M_2, and p that ensures the positive definiteness of C_1 and C_2 in Theorem 4.

Proposition 2 *In order to ensure* (21) *and* (22)*, it suffices to set* $M_1 = \frac{1}{\tau} I_n$ *where* $\tau < \frac{1}{\sqrt{2}} \mu_f$*,* $M_2 = \tau AA^T + \theta I_m$ *with* $\theta > 0$*, and large* p *such that* $c(p) < \theta$*.*

Proof Since $M_1 = \frac{1}{\tau} I_n$, it is evident that $C_1 \succ 0$ if and only if $\tau < \frac{1}{\sqrt{2}} \mu_f$. With $M_1 = \frac{1}{\tau} I_n$ and $M_2 = \tau AA^T + \theta I_m$, we have

$$C_2 = \frac{1}{2} \tau AA^T + (\theta - c(p)) I_m.$$

As we have seen in Theorem 2, and 3, $c(p) = \mathcal{O}(\tau^p)$ with some $\tau \in [0, 1)$ for S being proximal gradient or cyclic proximal BCD. Therefore, there exists p_0 such that $C_2 \succ 0$ for any $p \geq p_0$.

Remark 2 With a small $\theta > 0$, the choices of M_1 and M_2 given in Proposition 2 is close to the "ADMM choice" $M_1 = \frac{1}{\tau} I_n$ and $M_2 = \tau AA^T$, where M_2 is optimal (see Sect. 3.2).

We are now ready to show convergence of Algorithm 1.

Theorem 5 *Take Assumptions* 1 *and* 2*. Then,* (x^k, z^k) *in Algorithm 1 are bounded, and any cluster point is a primal–dual solution pair of* (1) *and* (2)*.*

Proof According to Theorem 1, it is sufficient to show that $\{M_1^{-1} u^k, z^k\}$ is bounded, and its cluster points are primal–dual solution pairs of (1).

Since L^k is nonincreasing, (37) tells us that

$$g^*(z^k) + f^*(-A^T z^k) + \frac{1}{2} \|A^T z^k + y^k\|^2_{M_1^{-1}} \leq L^0 < +\infty.$$

Since $g^*(z) + f^*(-A^T z)$ is coercive, $\{z^k\}$ is bounded, and, by the boundedness of $\{A^T z^k + y^k\}$, $\{y^k\}$ is also bounded. Furthermore, (27) gives us

$$\left\| M_1^{-1}(u^k - u^0) \right\| \leq \frac{1}{\mu_f} \|y^k - y^0\|.$$

Therefore, $\{M_1^{-1} u^k\}$ is bounded, too.

Let (z^c, y^c, u^c) be a cluster point of $\{z^k, y^k, u^k\}$. We shall show (z^c, y^c, u^c) is a saddle point of $L(z, y, u)$, i.e.,

$$\mathbf{0} \in \partial L(z^c, y^c, u^c), \tag{38}$$

or equivalently,

$$\mathbf{0} \in \partial g^*(z^c) - AM_1^{-1} u^c, \quad \mathbf{0} = \nabla f^*(y^c) - M_1^{-1} u^c, \quad \mathbf{0} = A^T z^c + y^c,$$

which ensures $(M_1^{-1} u^c, z^c)$ to be a primal–dual solution pair of (1).

In order to show (38), we first notice that (20) gives

$$\begin{aligned}
\partial_x L(z^{k+1}, y^{k+1}, u^{k+1}) &= \partial g^*(z^{k+1}) - AM_1^{-1} u^{k+1} + AM_1^{-1}(A^T z^{k+1} + y^{k+1}), \\
\nabla_y L(z^{k+1}, y^{k+1}, u^{k+1}) &= \nabla f^*(y^{k+1}) - M_1^{-1} u^{k+1} + M_1^{-1}(A^T z^{k+1} + y^{k+1}), \\
\nabla_u L(z^{k+1}, y^{k+1}, u^{k+1}) &= M_1^{-1}(-A^T z^{k+1} - y^{k+1}).
\end{aligned}$$

Comparing these with the optimality conditions (25), (27), and (28), we have

$$d^{k+1} = \left(d_z^{k+1}, d_y^{k+1}, d_u^{k+1}\right) \in \partial L\left(z^{k+1}, y^{k+1}, u^{k+1}\right), \tag{39}$$

where

$$\begin{aligned} d_z^{k+1} &= M_2(z^k - z^{k+1}) + 2AM_1^{-1}(u^k - u^{k+1}) - AM_1^{-1}(u^{k-1} - u^k) - \varepsilon^{k+1}, \\ d_y^{k+1} &= M_1^{-1}(u^k - u^{k+1}), \\ d_u^{k+1} &= M_1^{-1}(u^{k+1} - u^k). \end{aligned} \tag{40}$$

Since (23) and (24) imply $z^k - z^{k+1}$, $y^k - y^{k+1} \to \mathbf{0}$, (27) gives $u^k - u^{k+1} \to \mathbf{0}$. Combine these with (12), we have $d^k \to \mathbf{0}$.

Finally, let us take a subsequence $\{z^{k_s}, y^{k_s}, u^{k_s}\} \to (z^c, y^c, u^c)$. Since $d^{k_s} \to \mathbf{0}$ as $s \to +\infty$, [48, Def. 8.3] and [48, Prop. 8.12] yield (38), which tells us that $(M_1^{-1}u^c, z^c)$ is a primal–dual solution pair of (1).

Following the axiomatic approach developed in [2] for decent algorithms on KL functions, we can show that the whole sequence (x^k, z^k) in Algorithm 1 converges to a primal–dual solution pair. This approach has also been applied in [5] for KL-based Lagrangian optimization.

Theorem 6 *Take Assumptions* 1, 2, *and* 3. *Then,* $\{x^k, z^k\}$ *in Algorithm 1 converges to a primal–dual solution pair of* (1).

Proof By Theorem 5, we can take $\{z^{k_s}, y^{k_s}, u^{k_s}\} \to (z^c, y^c, u^c)$ as $s \to \infty$. Since L is a KL function, we can prove the convergence of $\{z^k, y^k, u^k\}$ to $\{z^c, y^c, u^c\}$ following [2]. Specifically, let us first verify that conditions H1, H2, and H3 of [2] are satisfied for $v^k := (z^k, y^k, u^k)$ and $L(v^k)$.

First, (23) gives

$$L(v^{k+1}) + \lambda_{\min}(C_1)\|y^k - y^{k+1}\|^2 + \lambda_{\min}(C_2)\|z^k - z^{k+1}\|^2 \le L(v^k). \tag{41}$$

By (27) and the $\frac{1}{\mu_f}$−Lipschitz differentiability of f^*, we know that

$$\frac{1}{2}\|y^k - y^{k+1}\|^2 \ge \frac{\mu_f^2}{2}\left\|M_1^{-1}u^k - M_1^{-1}u^{k+1}\right\|^2. \tag{42}$$

Combine (41) with (42), we know that there exists $a > 0$ such that

$$L(v^{k+1}) + a\|v^{k+1} - v^k\|^2 \le L(v^k).$$

which satisfies condition H1 of [2].

From (39) and (40), we know that $d^{k+1} \in \partial L(v^{k+1})$ satisfies

$$\|d^{k+1}\| \le b\|v^{k+1} - v^k\|$$

for some $b > 0$, which satisfies condition H2 of [2].

Next, let us verify that condition H3 of [2] also holds true.

Recall that we have taken $\{z^{k_s}, y^{k_s}, u^{k_s}\} \to (z^c, y^c, u^c)$ as $s \to \infty$. Note that $L(z^{k_s}, y^{k_s}, u^{k_s})$ is monotonic nonincreasing and lower bounded due to Theorem 4, which implies the convergence of $L(z^{k_s}, y^{k_s}, u^{k_s})$. Since L is lower semicontinuous, we have

$$L(z^c, y^c, u^c) \le \lim_{s\to\infty} L(z^{k_s}, y^{k_s}, u^{k_s}). \tag{43}$$

Since the only potentially discontinuous term in L is g^*, we have

$$\lim_{s\to\infty} L(z^{k_s}, y^{k_s}, u^{k_s}) - L(z^c, y^c, u^c) \le \limsup_{s\to\infty} g^*(z^{k_s}) - g^*(z^c). \tag{44}$$

By (25), we know that

$$g^*(z^c) \geq g^*(z^{k_s}) + \Big\langle M_2(z^{k_s-1} - z^{k_s}) \\ + AM_1^{-1}(-A^T z^{k_s-1} - y^{k_s-1} + u^{k_s-1}) - \varepsilon^{k_s}, z^c - z^{k_s} \Big\rangle,$$

Then, by Theorem 4, we further get $z^{k_s-1} - z^{k_s} \to \mathbf{0}$. Since $z^{k_s} \to z^c$ and $\{z^k, y^k, u^k\}$ is bounded, we obtain

$$\limsup_{s\to\infty} g^*(z^{k_s}) - g^*(z^c) \leq 0.$$

Combining this with (43) and (44), we conclude that

$$\lim_{s\to\infty} L(z^{k_s}, y^{k_s}, u^{k_s}) = L(z^c, y^c, u^c),$$

which satisfies condition H3 of [2].

Finally, since the conditions H1, H2, and H3 are satisfied, we can follow the proof of Theorem 2.9 of [2] to establish the convergence of $v^k = (z^k, y^k, u^k)$ to (z^c, y^c, u^c), which is a critical point of $L(z, y, u)$. By (40), we further now that $\{M_1^{-1}u^k, z^k\}$ converges to a primal–dual solution pair of (1), which is exactly $\{x^k, z^k\}$ in Algorithm 1 according to Theorem 1.

Remark 3 In order to remove the strong convexity assumption in Assumption 2, we also establish the ergodic convergence of iPrePDHG in the case where $g^* = 0$, and gradient descent is applied to solve the z-subproblem inexactly. See "Appendix D" for details.

4 Numerical Experiments

In this section, we compare our iPrePDHG (Algorithm 1) with (original) PDHG (6), diagonally-preconditioned PDHG (DP-PDHG) [44], accelerated PDHG (APDHG) [11], and accelerated linearized ADMM (ALADMM) [56]. We consider four popular applications of PDHG: TV-L^1 denoising, graph cuts, estimation of earth mover's distance, and CT reconstruction.

For the preconditioners M_1 and M_2 in iPrePDHG, we choose $M_1 = \frac{1}{\tau} I_n$ and $M_2 = \tau AA^T + \theta I$ as suggested in Proposition 2, which corresponds to ADMM and M_2 is nearly optimal for small θ (see Sect. 3.2). Although f may not be strongly convex in our experiments, we still observe significant speedups compared to other algorithms.

When we write these examples in the form of (1), the matrix A (or a part of A) is one of the following operators:

Case 1: 2D discrete gradient operator $D : \mathbb{R}^{M\times N} \to \mathbb{R}^{2M\times N}$:
For images of size $M \times N$ and grid stepsize h, we have

$$(Du)_{i,j} = \begin{pmatrix} (Du)^1_{i,j} \\ (Du)^2_{i,j} \end{pmatrix},$$

where

$$(Du)^1_{i,j} = \begin{cases} \frac{1}{h}(u_{i+1,j} - u_{i,j}) & \text{if } i < M, \\ 0 & \text{if } i = M, \end{cases}$$

$$(Du)^2_{i,j} = \begin{cases} \frac{1}{h}(u_{i,j+1} - u_{i,j}) & \text{if } j < N, \\ 0 & \text{if } j = N. \end{cases}$$

Case 2: 2D discrete divergence operator: div: $\mathbb{R}^{2M\times N} \to \mathbb{R}^{M\times N}$ given by

$$\mathrm{div}(p)_{i,j} = h\left(p^1_{i,j} - p^1_{i-1,j} + p^2_{i,j} - p^2_{i,j-1}\right),$$

where $p = (p^1, p^2)^T \in \mathbb{R}^{2M\times N}$, $p^1_{0,j} = p^1_{M,j} = 0$ and $p^2_{i,0} = p^2_{i,N} = 0$ for $i = 1, \dots, M$, $j = 1, \dots, N$.

in Algorithm 1, we can take S as the iterator of FISTA. To take the advantage of the finite-difference structure of these operators, we also let S be the iterator of cyclic proximal BCD. We split $\{1, 2, \dots m\}$ into 2 blocks (for case 2) or 4 blocks (for case 1), which are inspired by the popular red-black ordering [49] for solving sparse linear system.

According to Theorem 3, running finitely many epochs of cyclic proximal BCD gives us a bounded relative error in Definition 1. We expect that this solver brings fast overall convergence. Specifically, when $g^* = 0$, the z-subproblem in PrePDHG reduces to a linear system with a structured sparse matrix AA^T. Therefore, proximal gradient descent amounts to the Richardson method [47,49], and cyclic proximal BCD amounts to the Gauss–Seidel method and the Successive Overrelaxation (SOR) method [49,55], which are typically faster.

The following two claims tell us that with specific block partitions, the cyclic proximal BCD steps have a closed-form, so Algorithm 1 is easy to implement. Furthermore, each execution of BCD step can use parallel computing.

Claim When $A = \mathrm{div}$ (i.e. $A^T = -D$) and $M_2 = \tau AA^T$, for $z \in \mathbb{R}^{M\times N}$, we separate z into two block z_b, z_r where

$$z_b := \{z_{i,j} \mid i + j \text{ is even}\},\ z_r := \{z_{i,j} \mid i + j \text{ is odd}\},$$

for $1 \le i \le M$, $1 \le j \le N$. If $g(z) = \Sigma_{i,j} g_{i,j}(z_{i,j})$ and $prox_{\gamma g^*_{i,j}}$ have closed-form solutions for all $1 \le i \le M$, $1 \le j \le N$ and $\gamma > 0$, then S as the iterator of cyclic proximal BCD in Algorithm 1 has a closed-form and computing S is parallelizable.

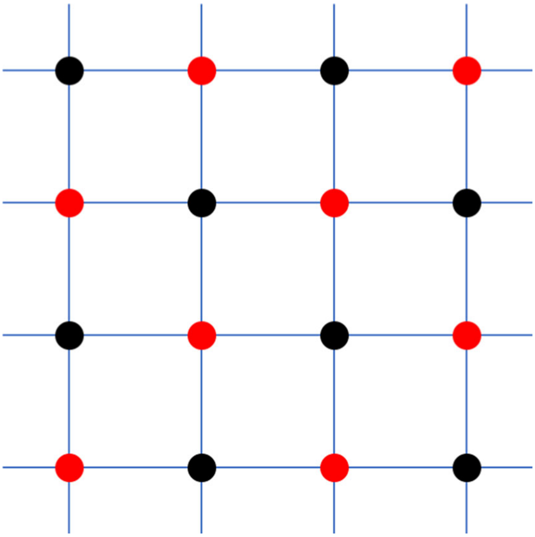


Fig. 1 Two-block ordering in Claim 4

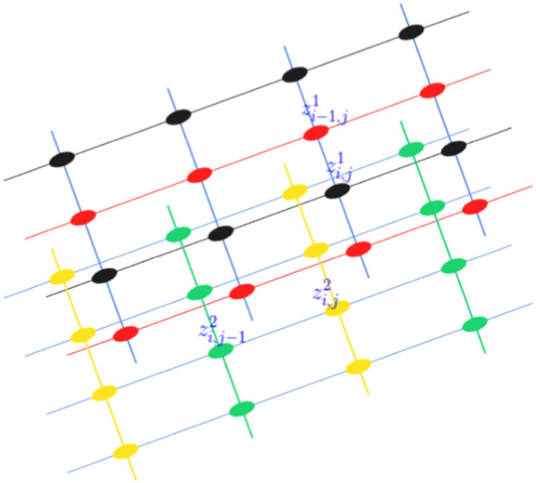


Fig. 2 Four-block ordering in Claim 4

Proof As illustrated in Fig. 1, every black node is connected to its neighbor red nodes, so we can update all the coordinates corresponding to the black nodes in parallel, while those corresponding to the red nodes are fixed, and vice versa. See "Appendix E" for a complete explanation. □

Claim When $A = D$ (i.e. $A^T = -\mathrm{div}$) and $M_2 = \tau AA^T$, for $z = (z^1, z^2)^T \in \mathbb{R}^{2M\times N}$, we separate z into four blocks z_b, z_r, z_y and z_g, where

$$z_b = \left\{z^1_{i,j} \mid i \text{ is odd}\right\}, \quad z_r = \left\{z^1_{i,j} \mid i \text{ is even}\right\},$$
$$z_y = \left\{z^2_{i,j} \mid j \text{ is odd}\right\}, \quad z_g = \left\{z^2_{i,j} \mid j \text{ is even}\right\},$$

for $1 \le i \le M$, $1 \le j \le N$. If $g(z) = \Sigma_{i,j} g_{i,j}(z_{i,j})$ and all $prox_{\gamma g^*_{i,j}}$ have closed-form solutions for all $1 \le i \le M$, $1 \le j \le N$ and $\gamma > 0$, then S as the iterator of cyclic proximal BCD in Algorithm 1 has a closed-form and computing S is parallelizable.

Proof In Fig. 2, the 4 blocks are in 4 different colors. The coordinates corresponding to nodes of the same color can be updated in parallel, while the rest are fixed. See "Appendix E" for details. □

In the following sections, PDHG denotes original PDHG in (6) without any preconditioning; DP-PDHG denotes the diagonally-preconditioned PDHG in [44], APDHG is the accelerated PDHG in [11], ALADMM is the accelerated linearized ADMM proposed in [56]. PrePDHG denotes Preconditioned PDHG in (7) where the $(k+1)$th z-subproblem is solved until $\frac{\|z^k - z^{k+1}\|_2}{\max\{1, \|z^{k+1}\|_2\}} < 10^{-5}$ using the TFOCS [4] implementation of FISTA; iPrePDHG (Inner: BCD) and iPrePDHG (Inner: FISTA) denote our iPrePDHG in Algorithm 1 with the iterator S being cyclic proximal BCD or FISTA, respectively. All the experiments were performed on MATLAB R2018a on a MacBook Pro with a 2.5 GHz Intel i7 processor and 16 GB of 2133 MHz LPDDR3 memory.

A comparison between PDHG and DP-PDHG is presented in [44] on TV-L^1 denoising and graph cuts, and in [50] on CT reconstruction. A PDHG algorithm is proposed to estimate earth mover's distance (or optimal transport) in [35]. In order to provide a direct comparison, we use their problem formulations.

4.1 Graph Cuts

The total-variation-based graph cut model involves minimizing a weighted TV energy:

$$\begin{array}{ll} \text{minimize} & \|D_w u\|_1 + \langle u, \omega^u \rangle \\ \text{subject to} & 0 \le u \le 1, \end{array}$$

where $w^u \in \mathbb{R}^{M\times N}$ is a vector of unary weights, $w^b \in \mathbb{R}^{2MN}$ is a vector of binary weights, and $D_w = \operatorname{diag}(w^b)D$ for D being the 2D discrete gradient operator with $h = 1$. Specifically, we have $w^u_{i,j} = \alpha(\|I_{i,j} - \mu_f\|^2 - \|I_{i,j} - \mu_b\|^2)$, $w^{b,1}_{i,j} = \exp(-\beta|I_{i+1,j} - I_{i,j}|)$, and $w^{b,2}_{i,j} = \exp(-\beta|I_{i,j+1} - I_{i,j}|)$.

To formulate this problem as (1), we take $f(u) = \langle u, w^u\rangle + \delta_{[0,1]}(u)$, $A = D_w$, and g as the ℓ_1−norm $g(z) = \sum_{i=1}^{2MN} |z_i|$.

In our experiment, the input image[3] has a size 660 × 720. We set $\alpha = 1/2$, $\beta = 10$, $\mu_f = [0; 0; 1]$ (for the blue foreground) and $\mu_b = [0; 1; 0]$ (for the green background). We run all algorithms until $\delta^k := \frac{|\Phi^k - \Phi^\star|}{|\Phi^\star|} < 10^{-8}$, where Φ^k is the objective value at the kth iteration and Φ^* is the optimal objective value obtained by running CVX[4] [18].

We summarize the test results in Table 1. For APDHG and ALADMM, the best results of $\mu \in \{10, 1, 0.1, 0.01, 0.001\}$ are presented, and the rest of their parameters are set as suggested in [11,56], respectively. For iPrePDHG, the best results of $\tau \in \{10, 1, 0.1, 0.01, 0.001\}$ and $p \in \{1, 2, 3, 10, 20, 30\}$ are presented, where the step size of cyclic proximal BCD was chosen as $\gamma = \frac{1}{\|M_2\|}$. Thanks to the efficiency of cyclic proximal BCD on the subproblems, we can simply apply 2 inner loops to achieve a superior performance. It is also worth mentioning that its number of outer iterations is close to that of PrePDHG, which solves z-subproblem much more accurately. In the last row of Table 1, we take $M_2 = \tau D_w D_w^T + \theta I_m$ with $\theta > 0$ as suggest in Proposition 2, the performance is similar to that of $\theta = 0$. In practice, we recommend simply taking $\theta = 0$. Finally, we would like to mention that the number of inner iterations are not exactly proportional to the runtime, this is because Matlab handles operations of sparse matrices in a pretty efficient way, and the runtime of the other parts of the tested algorithms is not negligible.

The input image can be found in Fig. 3. For all the tested algorithms, the output images look similar, therefore, we only present the output image of iPrePDHG in Fig. 4. This is also the case for the other tests in this paper.

4.2 Total Variation Based Image Denoising

The following problem is known as the (discrete) TV-L^1 model for image denoising:

$$\text{minimize}_u \quad \Phi(u) = \|Du\|_1 + \lambda\|u - b\|_1,$$

where D is the 2D discrete gradient operator with $h = 1$, $b \in \mathbb{R}^{1024\times 1024}$ is a noisy input image with noise level 0.15 (see Fig. 5), and $\lambda = 1$ is a regularization parameter. We run the

[3] https://www.shutterstock.com/image-photo/many-blue-hydrangea-flowers-growing-garden-174945887.

[4] Note that by the default setting of CVX, Φ^* given by CVX has an absolute error of $\mathcal{O}(10^{-8})$, while $\Phi^* = \mathcal{O}(10^5)$. Therefore, Φ^* is accurate enough for our tests. This is also the case for the other experiments in this paper.

Table 1 Performance of PDHG, DP-PDHG, ADMM, and iPrePDHG on the graph cut model

Method	Outer Iter	Inner Iter	Runtime (s)	Parameters
PDHG	5529	5529	140.5777	$\tau = 1, M_1 = 1/\tau I_n$, $M_2 = \tau\|D_w\|^2 I_m$
DP-PDHG	3571	3571	104.5392	$M_1 = \text{diag}(\Sigma_i \lvert D_{w\,i,j}\rvert)$, $M_2 = \text{diag}(\Sigma_j \lvert D_{w\,i,j}\rvert)$
PrePDHG (ADMM)	282	19,961	938.3787	$\tau = 10, M_1 = 1/\tau I_n$ $M_2 = \tau D_w D_w^T$
APDHG	–	–	$> 10^4$	$\tau_0 = 1, M_1^k = \frac{1}{\tau_k} I_n$ $M_2^k = \tau_k \|D_w\|^2 I_m, \mu = 1$
ALADMM	17,583	17,583	643.7214	$\mu = 0.001$
iPrePDHG (Inner: FISTA)	734	14,680	216.1936	$\tau = 10, M_1 = \frac{1}{\tau} I_n$, $M_2 = \tau D_w D_w^T, p = 20$
iPrePDHG (Inner: BCD)	**411**	**822**	**14.9663**	$\tau = 10, M_1 = \frac{1}{\tau} I_n$, $M_2 = \tau D_w D_w^T, p = 2$
iPrePDHG (Inner: BCD)	**402**	**804**	**14.7687**	$\tau = 10, M_1 = \frac{1}{\tau} I_n$, $M_2 = \tau D_w D_w^T + \theta I_m$, $\theta = 0.1, p = 2$

The results of iPrePHDG (inner: BCD) are in bold as it perfroms the best

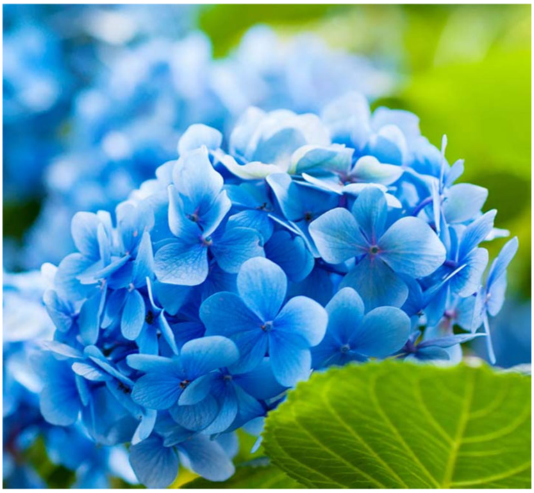


Fig. 3 Input image of graph cut with a size 660 × 720

algorithms until $\frac{|\Phi^k - \Phi^\star|}{|\Phi^\star|} < 10^{-6}$, where Φ^k is the kth objective value and Φ^* is the optimal objective value obtained by CVX [18].[5]

To formulate as (1), we take $f(u) = \lambda\|u - b\|_1$, $g(z) = \|z\|_1$, and $A = D$.

Observed performance is summarized in Table 2, where the best results for $\mu, \tau \in \{10, 1, 0.1, 0.01, 0.001\}$ and $p \in \{1, 2, 3, 5, 10, 20\}$ are presented (Again, the step size of

[5] http://www.hlevkin.com/TestImages/man.bmp.

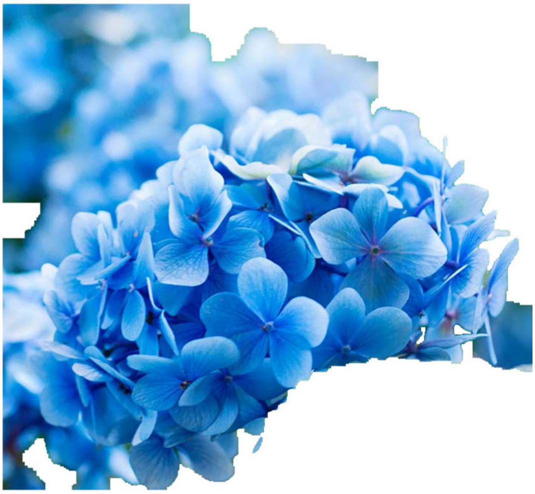


Fig. 4 Output of graph cut by our iPrePDHG (Inner: BCD), where the flower part has been extracted

cyclic proximal BCD has been chosen as $\gamma = \frac{1}{\|M_2\|}$). Our iPrePDHG (Inner: BCD) is significantly faster than the other algorithms. Finally, the denoised image can be found in Fig. 6.

When taking $\theta = 0.1$, we get nearly identical results. This is because $\theta > 0$ adds a proximal term $\frac{\theta}{2}\|z - z^k\|^2$ in the z-subproblem (see Eq. (8)), whose gradient at z^k is 0. Since $p = 1$ and cyclic proximal BCD is initialized exactly at z^k, we get the same iterates as that of $\theta = 0$. In practice, we recommend simply taking $\theta = 0$.

Remarkably, our algorithm uses fewer outer iterations than PrePDHG under the stopping criterion $\frac{\|z^k - z^{k+1}\|_2}{\max\{1, \|z^{k+1}\|_2\}} < 10^{-5}$, as this kind of stopping criteria may become looser as z^k is closer to $z^\star$. In this example, $\frac{\|z^k - z^{k+1}\|_2}{\max\{1, \|z^{k+1}\|_2\}} < 10^{-5}$ only requires 1 inner iteration of FISTA when Outer Iter ≥ 368, while as high as 228 inner iterations on average during the first 100 outer iterations. In comparison, our algorithm uses fewer outer iterations while each of them also costs less.

In addition, the diagonal preconditioner given in [44] appears to help very little when $A = D$. In fact, $M_1 = \mathrm{diag}(\Sigma_i |A_{i,j}|)$ will be $4I_n$ and $M_2 = \mathrm{diag}(\Sigma_j |A_{i,j}|)$ will be $2I_m$ if we ignore the Neumann boundary condition. Therefore, DP-PDHG performs even worse than PDHG.

4.3 Earth Mover's Distance

Earth mover's distance is useful in image processing, computer vision, and statistics [33,38, 43]. A recent method [35] to compute earth mover's distance is based on

$$\begin{array}{ll} \text{minimize} & \|m\|_{1,2} \\ \text{subject to} & \mathrm{div}(m) + \rho^1 - \rho^0 = 0, \end{array}$$

where $m \in \mathbb{R}^{2M \times N}$ is the sought flux vector on the $M \times N$ grid, and ρ^0, ρ^1 represents two mass distributions on the $M \times N$ grid. The setting in our experiment here is the same with that in [35], i.e. $M = N = 256$, $h = \frac{N-1}{4}$, and for ρ^0 and ρ^1 see Fig. 8.

To formulate as (1), we take $f(m) = \|m\|_{1,2}$, $g(z) = \delta_{\{\rho^0 - \rho^1\}}(z)$, and $A = \mathrm{div}$.

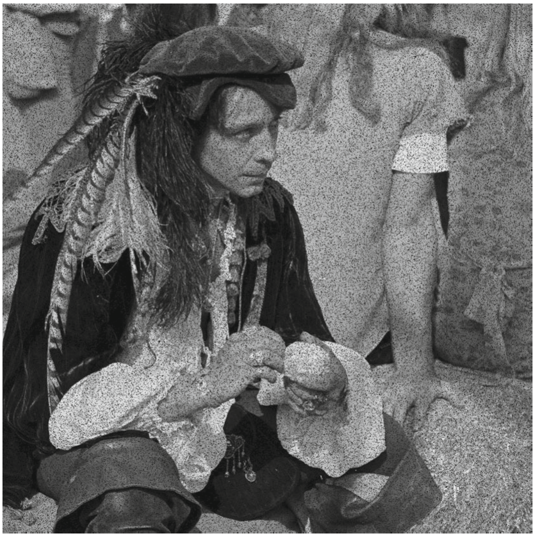


Fig. 5 Noisy input image for the TV-L^1 denoising model with 1024×1024 and noise level 0.15

Table 2 Test of PDHG, DP-PDHG, ADMM, and iPrePDHG on the TV-L^1 denoising model.

Method	Outer Iter	Inner Iter	Runtime (s)	Parameters
PDHG	2990	2990	114.2576	$\tau = 0.01, M_1 = \frac{1}{\tau} I_n,$ $M_2 = \tau \|D\|^2 I_m$
DP-PDHG	8856	8856	329.7890	$M_1 = \text{diag}(\Sigma_i \|D_{i,j}\|),$ $M_2 = \text{diag}(\Sigma_j \|D_{i,j}\|)$
PrePDHG (ADMM)	962	30,242	5641.0435	$\tau = 0.1, M_1 = 1/\tau I_n$ $M_2 = \tau D D^T$
APDHG	1696	1696	76.4154	$\tau_0 = 1, M_1^k = \frac{1}{\tau_k} I_n,$ $M_2^k = \tau_k \|D\|^2 I_m, \mu = 1$
ALADMM	1921	1921	127.1235	$\mu = 1$
iPrePDHG (Inner: FISTA)	564	2820	79.3684	$\tau = 0.01, M_1 = \frac{1}{\tau} I_n,$ $M_2 = \tau D D^T, p = 5$
iPrePDHG (Inner: BCD)	**541**	**541**	**26.2704**	$\tau = 0.01, M_1 = \frac{1}{\tau} I_n,$ $M_2 = \tau D D^T, p = 1$
iPrePDHG (Inner: BCD)	**541**	**541**	**26.2951**	$\tau = 0.01, M_1 = \frac{1}{\tau} I_n,$ $M_2 = \tau D D^T + \theta I_m$ $p = 1, \theta = 0.1$

The results of iPrePHDG (inner: BCD) are in bold as it perfroms the best

Since the iterates m^k may not satisfy the linear constraint, the objective $\Phi(m) = I_{\{m|\text{div}(m)=\rho^0-\rho^1\}} + \|m\|_{1,2}$ is not comparable. Instead, we compare $\|m^k\|_{1,2}$ and the constraint violation until $k = 100{,}000$ outer iterations in Fig. 7, where we set $\tau = 3 \times 10^{-6}$ as in [35], and $\sigma = \frac{1}{\tau\|\text{div}\|^2}$. For iPrePDHG (Inner: BCD), we set $M_1 = \tau^{-1} I_n$, $M_2 = \tau \text{div}\text{div}^{\text{T}}$, BCD stepsize $\gamma = \frac{1}{\|M_2\|}$, and number of BCD epochs $p = 2$.

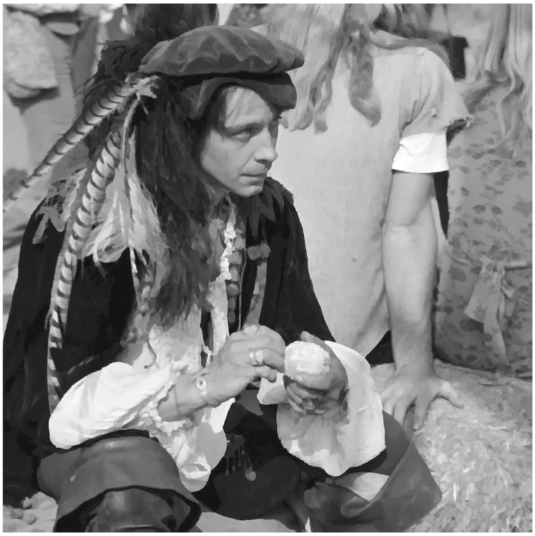


Fig. 6 Denoised image by iPrePDHG (Inner: BCD)

In Fig. 7, we can see that our iPrePDHG provides much lower constraint violation and much more faithful earth mover's distance $\|m\|_{1,2}$ at any given runtime. Figure 8 shows the solution obtained by our iPrePDHG (Inner: BCD), where m is the flux that moves the standing cat ρ^1 into the crouching cat ρ^0. For our iPrePDHG with $M_2 = \tau \text{div}\ \text{div}^T + \theta I_m$, the performance is very similar when a small θ is applied. In practice, we recommend simply taking $\theta = 0$.

DP-PDHG, ALADMM, and PrePDHG are extremely slow in this example and are not reported in Fig. 7. Similar to Sect. 4.2, when $A = \text{div}$, the diagonal preconditioners proposed in [44] are approximately equivalent to fixed constant parameters $\tau = \frac{1}{2h}, \sigma = \frac{1}{4h}$ and they lead to extremely slow convergence. As for PrePDHG, it suffers from the high cost per outer iteration.

It is worth mentioning that unlike [35], the algorithms in our experiments are not parallelized. On the other hand, in our iPrePDHG (Inner: BCD), iterator S can be parallelized (which we did not implement). Therefore, one can expect a further speedup by a parallel implementation.

4.4 CT Reconstruction

We test solving the following optimization problem for CT image reconstruction:

$$\text{minimize} \quad \Phi(u) = \tfrac{1}{2}\|Ru - b\|_2^2 + \lambda\|Du\|_1, \tag{45}$$

where $R \in \mathbb{R}^{13032\times 65536}$ is a system matrix for 2D fan-beam CT with a curved detector, $b = Ru_{\text{true}} \in \mathbb{R}^{13032}$ is a vector of line-integration values, and we want to reconstruct $u_{\text{true}} \in \mathbb{R}^{MN}$, where $M = N = 256$. D is the 2D discrete gradient operator with $h = 1$, and $\lambda = 1$ is a regularization parameter. By using the *fancurvedtomo* function from the AIR Tools II package [29], we generate a test problem where the projection angles are $0°, 10°, \dots, 350°$, and for all the other input parameters we use the default values.

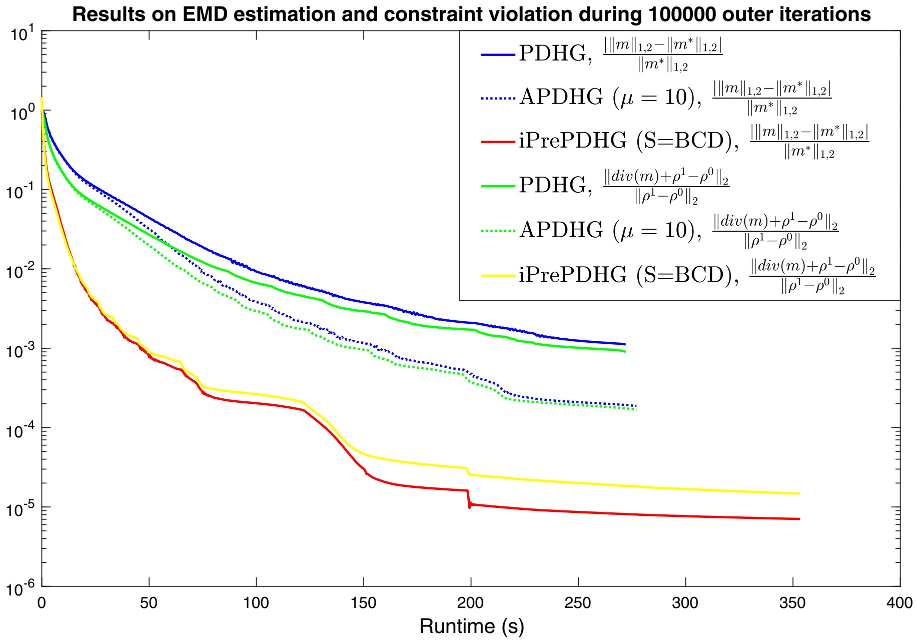
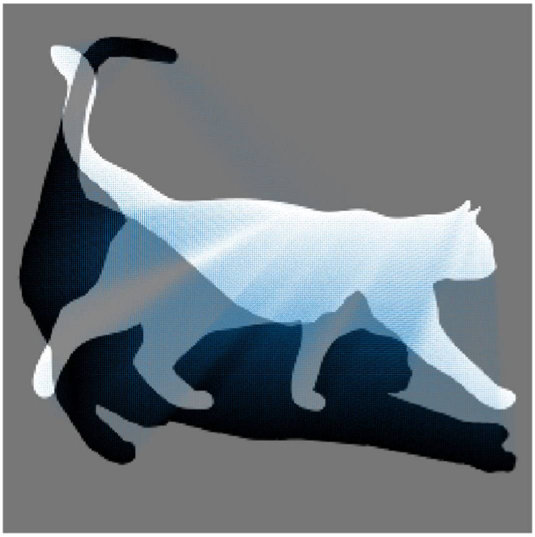


Fig. 7 Comparison of PDHG and iPrePDHG on the EMD estimation problem over 100,000 outer iterations

Fig. 8 Mass distributions ρ^0, ρ^1 for EMD estimation. ρ^0 is the white standing cat, and ρ^1 is the black crouching cat. Both images are 256×256, and the earth mover's distance between ρ^0 and ρ^1 is 0.6718



Following [50], we formulate the problem (45) in the form of (1) by taking

$$g\begin{pmatrix} p \\ q \end{pmatrix} = \frac{1}{2}\|p-b\|_2^2 + \lambda\|q\|_1, \quad f(u) = 0, \quad A = \begin{pmatrix} R \\ D \end{pmatrix}, \tag{46}$$

By using this formulation, we avoids inverting the matrices R and D.

Table 3 Performance of PDHG, DP-PDHG, ADMM, and iPrePDHG on CT reconstruction

Method	Outer	Inner	Runtime (s)	Parameters
PDHG	364,366	364,366	3663.0348	$\tau = 0.001$, $M_1 = \frac{1}{\tau} I_n$, $M_2 = \tau \|A\|^2 I_m$
DP-PDHG	70,783	70,783	713.9865	$M_1 = \text{diag}(\Sigma_i \|A_{i,j}\|)$, $M_2 = \text{diag}(\Sigma_j \|A_{i,j}\|)$
PrePDHG (ADMM)	–	–	$> 10^4$	$\tau = 0.01$, $M_1 = \frac{1}{\tau} I_n$, $M_2 = \tau A A^T$
APDHG	31289	31289	333.1747	$\tau_0 = 0.001$, $M_1^k = \frac{1}{\tau_k} I_n$, $M_2^k = \tau_k \|A\|^2 I_m$, $\mu = 1$
ALADMM	22,286	22,286	342.3022	$\mu = 10$
iPrePDHG (Inner: FISTA)	–	–	$> 10^4$	$\tau = 0.001$, $M_1 = \frac{1}{\tau} I_n$, $M_2 = \tau A A^T$, $p = 1, 2$, or 3
iPrePDHG (Inner: FISTA)	–	–	$> 10^4$	$\tau = 0.01$, $M_1 = \frac{1}{\tau} I_n$, $M_2 = \tau A A^T$, $p = 100$
iPrePDHG (Inner: BCD)	**587**	**1174**	**7.5365**	$\tau = 0.01$, $M_1 = \frac{2}{\tau} I_n$, $p = 2$, $M_2 = \begin{pmatrix} \tau \|R\|^2 I_{m-2n} & 0 \\ 0 & \tau D D^T \end{pmatrix}$
iPrePDHG (Inner: BCD)	**586**	**1172**	**7.2112**	$\tau = 0.01$, $M_1 = \frac{2}{\tau} I_n$, $p = 2$, $M_2 = \begin{pmatrix} \tau \|R\|^2 I_{m-2n} & 0 \\ 0 & \tau D D^T + \theta I_{2n} \end{pmatrix}$
iPrePDHG (Inner: BCD)	**858**	**1716**	**10.3517**	$\tau = 0.01$, $p = 2$ $M_1 = \text{diag}(\Sigma_i \|R_{i,j}\|) + \frac{1}{\tau} I_n$, $M_2 = \begin{pmatrix} \text{diag}(\Sigma_j \|R_{i,j}\|) & 0 \\ 0 & \tau D D^T \end{pmatrix}$
iPrePDHG (Inner: BCD)	**857**	**1714**	**10.3123**	$\tau = 0.01$, $p = 2$ $M_1 = \text{diag}(\Sigma_i \|R_{i,j}\|) + \frac{1}{\tau} I_n$, $M_2 = \begin{pmatrix} \text{diag}(\Sigma_j \|R_{i,j}\|) & 0 \\ 0 & \tau D D^T + \theta I_{2n} \end{pmatrix}$

The results of iPrePHDG (inner: BCD) are in bold as it perfroms the best

Since the block structure of AA^T is rather complicated, if we naively choose $M_1 = \frac{1}{\tau} I_n$ and $M_2 = \tau A A^T$ like in the previous three experiments, it becomes hard to find a fast subproblem solver for the z-subproblem. In Table 3, we report a TFOCS implementation of FISTA for solving the z-subproblem and the overall convergence is very slow.

Instead, we propose to choose

$$M_1 = \frac{2}{\tau} I_n, \quad M_2 = \begin{pmatrix} \tau \|R\|^2 I_{m-2n} & 0 \\ 0 & \tau D D^T + \theta I_{2n} \end{pmatrix} \tag{47}$$

or

$$M_1 = \text{diag}(\Sigma_i |R_{i,j}|) + \frac{1}{\tau} I_n, \quad M_2 = \begin{pmatrix} \text{diag}(\Sigma_j |R_{i,j}|) & 0 \\ 0 & \tau D D^T + \theta I_{2n} \end{pmatrix} \tag{48}$$

for some small $\theta \geq 0$. These choices satisfy (9), and have simple block structures, a fixed epoch of S as cyclic proximal BCD iterator gives fast overall convergence. Note that (48) is a little slower but avoids the need of estimating $\|R\|$.

We summarize the numerical results in Table 3. All the algorithms are executed until $\delta^k := \frac{|\Phi^k - \Phi^\star|}{|\Phi^\star|} < 10^{-4}$, where Φ^k is the objective value at the kth iteration and Φ^* is the optimal objective value obtained by calling CVX. The best results of $\mu, \tau \in \{10, 1, 0.1, 0.01, 0.001\}$ and $p \in \{1, 2, 3\}$ are summarized in Table 3. As in the previous experiments, $\theta = 0.1$ gives similar performances for iPrePDHG (Inner: BCD). In practice, we recommend simply taking $\theta = 0$. For iPrePDHG (Inner: FISTA) with $M_2 = \tau AA^T$, the result for $p = 100$ is also reported (here we use the TFOCS implementation of FISTA).

5 Conclusions

We have developed an approach to improve the performance of PDHG and ADMM in this paper. Our approach uses effective preconditioners to significantly reduce the number of iterations. In general, most effective preconditioners are non-diagonal and cause very difficult subproblems in PDHG and ADMM, so previous arts are restrictive with less effective diagonal preconditioners. However, we deal with those difficult subproblems by “solving” them highly inexactly, running just very few epochs of proximal BCD iterations with warm start. In all of our numerical tests, our algorithm needs relatively few outer iterations (due to effective preconditioners) and has the shortest total running time, achieving 4–95 times speedup over the state-of-the-art.

Theoretically, we show a fixed number of inner iterations suffice for global convergence though a new relative error condition. The number depends on various factors but is easy to choose in all of our numerical results.

There are still open questions left for us to address in the future: (a) Depending on problem structures, there are choices of preconditioners that are better than $M_1 = \frac{1}{\tau} I_n$, $M_2 = \tau AA^T$ (the ones that lead to ADMM if the subproblems are solved exactly). For example, in CT reconstruction, our choices of M_1 and M_2 have much faster overall convergence. (b) Is it possible to show Algorithm 1 converges even with S chosen as the iterator of faster solvers like APCG [36], NU_ACDM [1], and A2BCD [28]? (c) In general, how to accelerate a broader class of algorithms by integrating effective preconditioning and cheap inner loops while still ensuring global convergence?

Funding The work of Yanli Liu and Wotao Yin was supported in part by NSF DMS-1720237 and ONR N000141712162.

Availability of Data and Materials The images used in the numerical experiments are available in our MATLAB code. All data used in the numerical experiments are available. 1. The input image in the graph cut experiment (Sect. 4.1) is available at https://www.shutterstock.com/image-photo/many-blue-hydrangea-flowers-growing-garden-174945887 2. The input image in the TV-L^1 experiment (Sect. 4.2) is available at http://www.hlevkin.com/TestImages/man.bmp 3. The input image in the Earth mover’s distance experiment (Sect. 4.3) is available in the published article [35] and its supplementary code http://www.math.snu.ac.kr/\protect\unhbox\voidb@x\penalty\@M\ernestryu/code/OMT/OMT_code.zip 4. The input image in the CT reconstruction experiment (Sect. 4.4) is generated by the AIR Tools II package [29].

Compliance with Ethical Standards

Conflict of interest The authors declare that there is no conflict of interests.

Code Availability The MATLAB code of the experiments is available on GitHub (https://github.com/xuyunbei/Inexact-preconditioning).

A Proof of Lemma 2

Proof If (X, Z) is a primal–dual solution pair of (1), then

$$-A^T Z \in \partial f(X), \quad AX \in \partial g^*(Z).$$

Hence, for any $(x, z) \in \mathbb{R}^{n+m}$ we have

$$f(x) \geq f(X) + \langle -A^T Z, x - X\rangle, \quad g^*(z) \geq g^*(Z) + \langle AX, z - Z\rangle.$$

Adding them together yields $\varphi(X, z) - \varphi(x, Z) \leq 0$.

On the other hand, if $\varphi(X, z) - \varphi(x, Z) \leq 0$ for any $(x, z) \in \mathbb{R}^{n+m}$, then

$$\langle AX, z\rangle + f(X) - g^*(z) - \langle Ax, Z\rangle - f(x) + g^*(Z) \leq 0 \quad \text{for any } (x, z) \in \mathbb{R}^{n+m}.$$

Taking $x = X$ yields $\langle AX, z - Z\rangle - g^*(z) + g^*(Z) \leq 0$, so $AX \in \partial g^*(Z)$; Similarly, taking $z = Z$ gives $\langle AX - Ax, Z\rangle + f(X) - f(x) \leq 0$, so $-A^T Z \in \partial f(X)$. As a result, (X, Z) is a primal–dual solution pair of (1). □

B ADMM as a Special Case of PrePDHG

In this section we show that if we choose $M_1 = \frac{1}{\tau}$ and $M_2 = \tau AA^T$ in PrePDHG (7), then it is equivalent to ADMM on the primal problem (1).

By Theorem 1 of [57], we know that ADMM is primal–dual equivalent, in the sense that one can recover primal iterates from dual iterates and vice versa. Therefore, it suffices to show that $M_1 = \frac{1}{\tau}$ and $M_2 = \tau AA^T$ in PrePDHG (7) on the primal problem is equivalent to ADMM on the dual problem (2).

In Theorem 1 we have shown that, under an appropriate change of variables, PrePDHG on the primal is equivalent to applying (19) to the dual. As a result, we just need to demonstrate that the latter is exactly ADMM on the dual when $M_1 = \frac{1}{\tau} I_n$ and $M_2 = \tau AA^T$.

For the z-update in (19), we have

$$\begin{aligned}
z^{k+1} &= \underset{z\in\mathbb{R}^m}{\arg\min} \left\{ g^*(z) - \tau\langle z - z^k, A(-A^T z^k - y^k + u^k)\rangle + \frac{\tau}{2}\|z - z^k\|^2_{AA^T} \right\} \\
&= \underset{z\in\mathbb{R}^m}{\arg\min} \left\{ g^*(z) - \tau\langle z - z^k, A(-y^k + u^k)\rangle + \frac{\tau}{2}\|z\|^2_{AA^T} \right\} \\
&= \underset{z\in\mathbb{R}^m}{\arg\min} \left\{ g^*(z) + \tau\langle z, A(y^k - u^k)\rangle + \frac{\tau}{2}\|A^T z\|^2 \right\} \\
&= \underset{z\in\mathbb{R}^m}{\arg\min} \left\{ g^*(z) + \tau\langle A^T z, -u^k\rangle + \frac{\tau}{2}\|A^T z + y^k\|^2 \right\} \\
&= \underset{z\in\mathbb{R}^m}{\arg\min} \left\{ g^*(z) + \tau\langle -A^T z - y^k, u^k\rangle + \frac{\tau}{2}\|A^T z + y^k\|^2 \right\}.
\end{aligned} \tag{49}$$

and for the y-update we have

$$y^{k+1} = \mathrm{Prox}_{f^*}^{M_1^{-1}}(u^k - A^T z^{k+1})$$

$$
\begin{aligned}
&= \underset{y\in\mathbb{R}^{\mathrm{n}}}{\arg\min}\left\{f^*(y)+\frac{\tau}{2}\|y-u^k+A^Tz^{k+1}\|^2\right\}\\
&= \underset{y\in\mathbb{R}^{\mathrm{n}}}{\arg\min}\left\{f^*(y)+\tau\langle -A^Tz^{k+1}-y, u^k\rangle+\frac{\tau}{2}\|A^Tz^{k+1}+y\|^2\right\}. \qquad (50)
\end{aligned}
$$

Define $v^k=\tau u^k$, (49), (50), and the $u-$update in (19) become

$$
\begin{aligned}
z^{k+1} &= \underset{z\in\mathbb{R}^{\mathrm{m}}}{\arg\min}\left\{g^*(z)+\langle -A^Tz-y^k, v^k\rangle+\frac{\tau}{2}\|A^Tz+y^k\|^2\right\},\\
y^{k+1} &= \underset{y\in\mathbb{R}^{\mathrm{n}}}{\arg\min}\left\{f^*(y)+\langle -A^Tz^{k+1}-y, v^k\rangle+\frac{\tau}{2}\|A^Tz^{k+1}+y\|^2\right\},\\
v^{k+1} &= v^k-\tau(A^Tz^{k+1}+y^{k+1}),
\end{aligned}
$$

which are ADMM iterations on the dual problem (2).

C Proof of Theorem 3: Bounded Relative Error when *S* is the Iterator of Cyclic Proximal BCD

The z-subproblem in (7) has the form

$$\min_{z\in\mathbb{R}^{\mathrm{m}}} h_1(z)+h_2(z),$$

where $h_1(z)=g^*(z)=\sum_{j=1}^{l}g_j^*(z_j)$, and $h_2(z)=\frac{1}{2}\|z-z^k-M_2^{-1}A(2x^{k+1}-x^k)\|_{M_2}^2$. And $z^{k+1}=z_p^{k+1}$ is given by

$$
\begin{aligned}
z_0^{k+1} &= z^k,\\
z_{i+1}^{k+1} &= S\left(z_i^{k+1},x^{k+1},x^k\right), \quad i=0,1,\dots,p-1,
\end{aligned}
$$

Here, S is the iterator of cyclic proximal BCD. Define

$$
\begin{aligned}
T(z) &= \mathrm{Prox}_{\gamma h_1(z)}(z-\gamma\nabla h_2(z)),\\
B(z) &= \frac{1}{\gamma}(z-T(z)),
\end{aligned}
$$

and the jth coordinate operator of B:

$$B_j(z)=(0,\dots,(B(z))_j,\dots,0), \quad j=1,2,\dots,l.$$

Then, we have

$$z_{i+1}^{k+1}=S\left(z_i^{k+1},x^{k+1},x^k\right)=(I-\gamma B_l)(I-\gamma B_2)\dots(I-\gamma B_1)z_i^{k+1}.$$

By [3, Prop. 26.16(ii)], we know that $T(z)$ is a contraction with coefficient $\rho_0=\sqrt{1-\gamma(2\lambda_{\min}(M_2)-\gamma\lambda_{\max}^2(M_2))}$. We know that for $\forall z_1,z_2\in\mathbb{R}^{\mathrm{m}}$ and $\mu_0=\frac{1-\rho_0}{\gamma}$,

$$
\begin{aligned}
\langle B(z_1)-B(z_2),z_1-z_2\rangle &= \frac{1}{\gamma}\|z_1-z_2\|^2-\frac{1}{\gamma}\langle T(z_1)-T(z_2),z_1-z_2\rangle\\
&\geq \mu_0\|z_1-z_2\|^2,
\end{aligned}
$$

Let $z_\star^{k+1} = \arg\min_{z\in\mathbb{R}^m}\{h_1(z)+h_2(z)\}$. For [14, Thm 3.5], we have

$$\left\| z_i^{k+1} - z_\star^{k+1} \right\| \le \rho^i \left\| z_0^{k+1} - z_\star^{k+1} \right\|, \quad \forall i = 1, 2, \ldots, p. \tag{51}$$

where $\rho = 1 - \frac{\gamma\mu_0^2}{2}$.

Let $y_j = (I - \gamma B_j)\ldots(I-\gamma B_1) z_{p-1}^{k+1}$ for $j = 1, \ldots, l$ and $y_0 = z_{p-1}^{k+1}$. Note that $(z_p^{k+1})_j = (y_j)_j$ for $j = 1, 2, \ldots, l$, and the blocks of y_j satisfies

$$(y_j)_t = \begin{cases} \left(\operatorname{Prox}_{\gamma g^*}\left(y_{j-1} - \gamma\nabla h_2(y_{j-1})\right)\right)_t, & \text{if } t = j \\ (y_{j-1})_t, & \text{otherwise.} \end{cases}$$

On the other hand, we have

$$\operatorname{Prox}_{\gamma g^*}\left(y_{j-1} - \gamma\nabla h_2(y_{j-1})\right) = \underset{y\in\mathbb{R}^m}{\arg\min}\left\{ g^*(y) + \frac{1}{2\gamma}\|y - y_{j-1} + \gamma\nabla h_2(y_{j-1})\|^2 \right\}.$$

Since g^* and $\|\cdot\|^2$ are separable, we obtain

$$\mathbf{0} \in \partial g_j^*((y_j)_j) + \frac{1}{\gamma}\Big((y_j)_j - (y_{j-1})_j + \gamma\big(\nabla h_2(y_{j-1})\big)_j\Big), \quad \forall j = 1, 2, \ldots, l,$$

or equivalently,

$$\mathbf{0} \in \partial g_j^*\left(\left(z_p^{k+1}\right)_j\right) + \frac{1}{\gamma}\Big(\left(z_p^{k+1}\right)_j - \left(z_{p-1}^{k+1}\right)_j + \gamma\big(\nabla h_2(y_{j-1})\big)_j\Big), \quad \forall j = 1, 2, \ldots, l.$$

Therefore,

$$\mathbf{0} \in \partial g^*\left(z_p^{k+1}\right) + \frac{1}{\gamma}\Big(z_p^{k+1} - z_{p-1}^{k+1} + \gamma\xi_p\Big), \quad \forall j = 1, 2, \ldots, l,$$

where $(\xi_p)_j = \big(\nabla h_2(y_{j-1})\big)_j$ for $j = 1, 2, \ldots, l$. Comparing this with (11), we obtain

$$\varepsilon^{k+1} = \xi_p - \nabla h_2\left(z_p^{k+1}\right) + \frac{1}{\gamma}\left(z_p^{k+1} - z_{p-1}^{k+1}\right).$$

Notice that the first $j-1$ blocks of y_{j-1} are the same with those of $y_l = z_p^{k+1}$, and the rest of the blocks are the same with those of $y_0 = z_{p-1}^{k+1}$, so we have

$$\begin{aligned} \|\varepsilon^{k+1}\| &\le \sum_{j=1}^{l} \lambda_{\max}(M_2)\left\| y_{j-1} - z_p^{k+1} \right\| + \frac{1}{\gamma}\left\| z_p^{k+1} - z_{p-1}^{k+1} \right\| \\ &\le l\lambda_{\max}(M_2)\left\| z_{p-1}^{k+1} - z_p^{k+1} \right\| + \frac{1}{\gamma}\left\| z_p^{k+1} - z_{p-1}^{k+1} \right\| \\ &\le \left(l\lambda_{\max}(M_2) + \frac{1}{\gamma}\right)\left(\left\| z_p^{k+1} - z_\star^{k+1} \right\| + \left\| z_{p-1}^{k+1} - z_\star^{k+1} \right\|\right) \end{aligned}$$

Combine this with (51)

$$\|\varepsilon^{k+1}\| \le \left(l\lambda_{\max}(M_2) + \frac{1}{\gamma}\right)(\rho^p + \rho^{p-1})\left\| z_0^{k+1} - z_\star^{k+1} \right\|. \tag{52}$$

Combining

$$\|z^{k+1} - z^k\| = \left\| z_p^{k+1} - z_0^{k+1} \right\|$$

$$\geq \left\| z_0^{k+1} - z_\star^{k+1} \right\| - \left\| z_p^{k+1} - z_\star^{k+1} \right\|$$
$$\geq (1 - \rho^p) \left\| z_0^{k+1} - z_\star^{k+1} \right\|$$

with (52), we obtain

$$\|\varepsilon^{k+1}\| \leq \frac{\left(l\lambda_{\max}(M_2) + \frac{1}{\gamma}\right)(\rho^p + \rho^{p-1})}{1 - \rho^p} \|z^{k+1} - z^k\|.$$

D Ergodic Convergence of iPrePDHG when $g^* = 0$ and S Being Gradient Descent

In this section, we present an ergodic convergence result for iPrePDHG, which does not require f to be strongly convex.

Theorem 7 *Assume that $g^* = 0$ and the $z-$subproblem is solved by applying p iterations of gradient descent with warm-start and stepsize $\gamma \in (0, \frac{1}{\lambda_{max}(M_2)})$. Let $(x^k, z^k), k = 0, 1, \ldots, N$ be a sequence generated by iPrePDHG in Algorithm 1. Under Assumption 1, if in addition*

$$\tilde{M} := \begin{pmatrix} M_1 & -A^T \\ -A & M_{2,p} \end{pmatrix} \succeq 0,$$

where $M_{2,p} = M_2^{-1}(I - (I - \gamma M_2)^p)$. Then, for any $x \in \mathbb{R}^{\mathrm{n}}$ and $z \in \mathbb{R}^{\mathrm{m}}$, it holds that

$$\varphi(X^N, z) - \varphi(x, Z^N) \leq \frac{1}{2N}(x - x^0, z - z^0) \begin{pmatrix} M_1 & -A^T \\ -A & M_{2,p} \end{pmatrix} \begin{pmatrix} x - x^0 \\ z - z^0 \end{pmatrix},$$

where $X^N = \frac{1}{N}\sum_{i=1}^N x^i$ and $Z^N = \frac{1}{N}\sum_{i=1}^N z^i$.

Proof Essentially, we would like to show that when $g^* = 0$ and p iterations of gradient descent with warm-start are applied to the $z-$subproblem, we are effectively applying $M_{2,p} = M_2^{-1}(I - (I - \gamma M_2)^p)$ as a preconditioner. Therefore, the desired result follows immediately by applying Theorem 1.

For that purpose, we first recall from Algorithm 1 that the $z-$subproblem of iPrePDHG is

$$z^{k+1} \approx \underset{z \in \mathbb{R}^{\mathrm{m}}}{\arg\min} \left\{ g^*(z) - \langle z - z^k, A(2x^{k+1} - x^k) \rangle + \frac{1}{2}\|z - z^k\|_{M_2}^2 \right\}$$
$$= \underset{z \in \mathbb{R}^{\mathrm{m}}}{\arg\min} \left\{ -\langle z - z^k, A(2x^{k+1} - x^k) \rangle + \frac{1}{2}\|z - z^k\|_{M_2}^2 \right\}$$

When applying warm-started gradient descent with stepsize γ to solve it, we have

$$z_0^{k+1} = z^k,$$
$$z_{i+1}^{k+1} = z_i^{k+1} - \gamma \left(M_2(z_i^{k+1} - z^k) - A(2x^{k+1} - x^k) \right), \quad i = 0, 1, \ldots, p-1.$$

As a result we have the following recursion

$$z_{i+1}^{k+1} - z^k = (I - \gamma M_2)\left(z_i^{k+1} - z^k\right) + \gamma A(2x^{k+1} - x^k),$$

which leads to

$$\begin{aligned} z_p^{k+1} &= z^k + \left(\sum_{i=1}^{p} (I - \gamma M_2)^{p-i} \right) \gamma A(2x^{k+1} - x^k) \\ &= z^k + M_2^{-1} \left(I - (I - \gamma M_2)^p \right) A(2x^{k+1} - x^k) \\ &= z^k - M_{2,p}^{-1} A(2x^{k+1} - x^k), \end{aligned}$$

where we have applied $M_{2,p} = M_2^{-1} \left(I - (I - \gamma M_2)^p \right)$. Therefore, z_p^{k+1} is the exact solution of the following problem:

$$z^{k+1} = \operatorname*{arg\,min}_{z \in \mathbb{R}^{\mathrm{m}}} \left\{ -\langle z - z^k, A(2x^{k+1} - x^k) \rangle + \frac{1}{2} \| z - z^k \|_{M_{2,p}}^2 \right\}.$$

□

E Two-Block Ordering and Four-Block Ordering

According to (8), when $M_2 = \tau A A^T$, the z-subproblem of Algorithm 1 is

$$z^{k+1} = \operatorname*{arg\,min}_{z \in \mathbb{R}^{\mathrm{m}}} \left\{ g^*(z) - \langle z - z^k, A(2x^{k+1} - x^k) \rangle + \frac{\tau}{2} \| A^T (z - z^k) \|_2^2 \right\}. \tag{53}$$

Let us prove the claim for two-block ordering first. In that claim, $A = \operatorname{div} \in \mathbb{R}^{MN \times 2MN}$ and $z \in \mathbb{R}^{MN}$. Following the definition of the sets z_b and z_r, we separate the MN columns of $A^T = -D$ into two blocks L_b, L_r by associating them with z_b and z_r, respectively. Therefore, we have $A^T z = L_b z_b + L_r z_r$ for any $z \in \mathbb{R}^{MN}$.

By the red-black ordering in Fig. 1, different columns of L_b are orthogonal one another, so ${L_b}^T L_b$ is diagonal. Similarly, ${L_r}^T L_r$ is also diagonal.

Define $c^k = -A(2x^{k+1} - x^k)$, and let b be the set of black nodes and r the set of red nodes. We can rewrite (53) as

$$\begin{aligned} z^{k+1} = \operatorname*{arg\,min}_{z_b, z_r \in \mathbb{R}^{MN/2}} & \left\{ g_b^*(z_b) + g_r^*(z_r) + \left\langle z_b, c_b^k \right\rangle + \left\langle z_r, c_r^k \right\rangle \right. \\ & \left. + \frac{\tau}{2} \left\| L_b \left(z_b - z_b^k \right) + L_r \left(z_r - z_r^k \right) \right\|_2^2 \right\}, \end{aligned} \tag{54}$$

where $g_b^*(z_b) = \sum_{(i,j) \in b} g_{i,j}^*(z_{i,j})$, $g_r^*(z_r) = \sum_{(i,j) \in r} g_{i,j}^*(z_{i,j})$, and c_b^k, c_r^k are the coordinates of c^k associated with z_b and z_r, respectively.

Applying cyclic proximal BCD to black and red blocks with stepsize γ yields

$$z_b^{k+\frac{t+1}{p}} = \operatorname{Prox}_{\gamma g_b^*} \left(z_b^{k+\frac{t}{p}} - \gamma \left(c_b^k + \tau L_b^T L_b \left(z_b^{k+\frac{t}{p}} - z_b^k \right) + \tau L_b^T L_r \left(z_r^{k+\frac{t}{p}} - z_r^k \right) \right) \right), \tag{55}$$

$$z_r^{k+\frac{t+1}{p}} = \operatorname{Prox}_{\gamma g_r^*} \left(z_r^{k+\frac{t}{p}} - \gamma \left(c_r^k + \tau L_r^T L_b \left(z_b^{k+\frac{t+1}{p}} - z_b^k \right) + \tau L_r^T L_r \left(z_r^{k+\frac{t}{p}} - z_r^k \right) \right) \right), \tag{56}$$

for $t = 0, 1, \ldots, p - 1$, where p is the number of inner iterations in Algorithm 1.

Since $\text{Prox}_{\gamma g_b^*} = \sum_{(i,j)\in b} \text{Prox}_{\gamma g_{i,j}^*}$, $\text{Prox}_{\gamma g_r^*} = \sum_{(i,j)\in r} \text{Prox}_{\gamma g_{i,j}^*}$ and $\text{Prox}_{\gamma g_{(i,j)}^*}$ are closed-form, (55) and (56) have closed-form solutions. Furthermore, the updates within each block can be done in parallel.

The proof of the second claim is similar. When $A = D$, we separate the columns of A^T into four blocks L_b, L_r, L_y, L_g by associating them with z_b, z_r, z_y ,z_g, respectively. Therefore, we have $A^T z = L_b z_b + L_r z_r + L_y z_y + L_g z_g$ for all $z \in \mathbb{R}^{2MN}$. Similarly, by the block design in Fig. 2, cyclic proximal BCD iterations have closed-form solutions, and updates within each block can be executed in parallel.

References

1. Allen-Zhu, Z., Qu, Z., Richtárik, P., Yuan, Y.: Even faster accelerated coordinate descent using non-uniform sampling. In: International Conference on Machine Learning, pp. 1110–1119 (2016)
2. Attouch, H., Bolte, J., Svaiter, B.F.: Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods. Math. Program. **137**(1–2), 91–129 (2013)
3. Bauschke, H.H., Combettes, P.L., et al.: Convex Analysis and Monotone Operator Theory in Hilbert Spaces, vol. 2011. Springer, Berlin (2017)
4. Becker, S.R., Candès, E.J., Grant, M.C.: Templates for convex cone problems with applications to sparse signal recovery. Math. Program. Comput. **3**(3), 165 (2011)
5. Bolte, J., Sabach, S., Teboulle, M.: Nonconvex Lagrangian-based optimization: monitoring schemes and global convergence. Math. Oper. Res. **43**(4), 1210–1232 (2018)
6. Bredies, K., Sun, H.: Preconditioned Douglas–Rachford splitting methods for convex–concave saddle-point problems. SIAM J. Numer. Anal. **53**(1), 421–444 (2015)
7. Bredies, K., Sun, H.: Accelerated Douglas–Rachford methods for the solution of convex–concave saddle-point problems. arXiv preprint arXiv:1604.06282 (2016)
8. Bredies, K., Sun, H.: A proximal point analysis of the preconditioned alternating direction method of multipliers. J. Optim. Theory Appl. **173**(3), 878–907 (2017)
9. Briceño-Arias, L.M., Combettes, P.L.: A monotone+ skew splitting model for composite monotone inclusions in duality. SIAM J. Optim. **21**(4), 1230–1250 (2011)
10. Briceño-Arias, L.M., Davis, D.: Forward–backward–half forward algorithm for solving monotone inclusions. SIAM J. Optim. **28**(4), 2839–2871 (2018)
11. Chambolle, A., Pock, T.: A first-order primal–dual algorithm for convex problems with applications to imaging. J. Math. Imaging Vis. **40**(1), 120–145 (2011)
12. Chambolle, A., Pock, T.: On the ergodic convergence rates of a first-order primal–dual algorithm. Math. Program. **159**(1–2), 253–287 (2016)
13. Chen, Y., Lan, G., Ouyang, Y.: Optimal primal–dual methods for a class of saddle point problems. SIAM J. Optim. **24**(4), 1779–1814 (2014)
14. Chow, Y.T., Wu, T., Yin, W.: Cyclic coordinate-update algorithms for fixed-point problems: analysis and applications. SIAM J. Sci. Comput. **39**(4), A1280–A1300 (2017)
15. Combettes, P.L., Reyes, N.N.: Moreau's decomposition in Banach spaces. Math. Program. **139**(1–2), 103–114 (2013)
16. Combettes, P.L., Vũ, B.C.: Variable metric forward–backward splitting with applications to monotone inclusions in duality. Optimization **63**(9), 1289–1318 (2014)
17. Condat, L.: A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms. J. Optim. Theory Appl. **158**(2), 460–479 (2013)
18. CVX Research, I.: CVX: Matlab software for disciplined convex programming, version 2.0. http://cvxr.com/cvx (2012) Accessed 15 Dec 2018
19. Eckstein, J., Bertsekas, D.P.: On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators. Math. Program. **55**(1–3), 293–318 (1992)
20. Eckstein, J., Yao, W.: Approximate ADMM algorithms derived from Lagrangian splitting. Comput. Optim. Appl. **68**(2), 363–405 (2017)
21. Eckstein, J., Yao, W.: Relative-error approximate versions of Douglas–Rachford splitting and special cases of the ADMM. Math. Program. **170**, 1–28 (2017)
22. Esser, E., Zhang, X., Chan, T.F.: A general framework for a class of first order primal–dual algorithms for convex optimization in imaging science. SIAM J. Imaging Sci. **3**(4), 1015–1046 (2010)

23. Feijer, D., Paganini, F.: Stability of primal–dual gradient dynamics and applications to network optimization. Automatica **46**(12), 1974–1981 (2010)
24. Giselsson, P., Boyd, S.: Diagonal scaling in Douglas–Rachford splitting and ADMM. In: 2014 IEEE 53rd Annual Conference on Decision and Control (CDC), pp. 5033–5039. IEEE (2014)
25. Giselsson, P., Boyd, S.: Linear convergence and metric selection for Douglas–Rachford splitting and ADMM. IEEE Trans. Autom. Control **62**(2), 532–544 (2017)
26. Goldfarb, D., Ma, S., Scheinberg, K.: Fast alternating linearization methods for minimizing the sum of two convex functions. Math. Program. **141**(1–2), 349–382 (2013)
27. Goldstein, T., O'Donoghue, B., Setzer, S., Baraniuk, R.: Fast alternating direction optimization methods. SIAM J. Imaging Sci. **7**(3), 1588–1623 (2014)
28. Hannah, R., Feng, F., Yin, W.: A2BCD: an asynchronous accelerated block coordinate descent algorithm with optimal complexity. arXiv preprint arXiv:1803.05578 (2018)
29. Hansen, P.C., Jørgensen, J.S.: Air tools II: algebraic iterative reconstruction methods, improved implementation. Numer. Algorithms **79**(1), 107–137 (2018)
30. He, B., Yuan, X.: Convergence analysis of primal–dual algorithms for a saddle-point problem: from contraction perspective. SIAM J. Imaging Sci. **5**(1), 119–149 (2012)
31. He, Y., Monteiro, R.D.: An accelerated HPE-type algorithm for a class of composite convex–concave saddle-point problems. SIAM J. Optim. **26**(1), 29–56 (2016)
32. Kadkhodaie, M., Christakopoulou, K., Sanjabi, M., Banerjee, A.: Accelerated alternating direction method of multipliers. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 497–506 (2015)
33. Levina, E., Bickel, P.: The earth mover's distance is the mallows distance: some insights from statistics. In: null, p. 251. IEEE (2001)
34. Li, M., Liao, L.Z., Yuan, X.: Inexact alternating direction methods of multipliers with logarithmic–quadratic proximal regularization. J. Optim. Theory Appl. **159**(2), 412–436 (2013)
35. Li, W., Ryu, E.K., Osher, S., Yin, W., Gangbo, W.: A parallel method for earth mover's distance. J. Sci. Comput. **75**(1), 182–197 (2018)
36. Lin, Q., Lu, Z., Xiao, L.: An accelerated proximal coordinate gradient method. Z. Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K.Q. (Eds.) In: Advances in Neural Information Processing Systems, vol. 27, pp. 3059–3067 Curran Associates, Inc. (2014)
37. Lin, Q., Lu, Z., Xiao, L.: An accelerated randomized proximal coordinate gradient method and its application to regularized empirical risk minimization. SIAM J. Optim. **25**(4), 2244–2273 (2015)
38. Métivier, L., Brossier, R., Mérigot, Q., Oudet, E., Virieux, J.: Measuring the misfit between seismograms using an optimal transport distance: application to full waveform inversion. Geophys. Suppl. Mon. Not. R. Astron. Soc. **205**(1), 345–377 (2016)
39. Nesterov, Y.: Smooth minimization of non-smooth functions. Math. Program. **103**(1), 127–152 (2005)
40. Ng, M.K., Wang, F., Yuan, X.: Inexact alternating direction methods for image recovery. SIAM J. Sci. Comput. **33**(4), 1643–1668 (2011)
41. Ouyang, Y., Chen, Y., Lan, G., Pasiliao Jr., E.: An accelerated linearized alternating direction method of multipliers. SIAM J. Imaging Sci. **8**(1), 644–681 (2015)
42. O'donoghue, B., Candes, E.: Adaptive restart for accelerated gradient schemes. Found. Comput. Math. **15**(3), 715–732 (2015)
43. Pele, O., Werman, M.: Fast and robust earth mover's distances. In: ICCV, vol. 9, pp. 460–467 (2009)
44. Pock, T., Chambolle, A.: Diagonal preconditioning for first order primal–dual algorithms in convex optimization. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 1762–1769. IEEE (2011)
45. Pock, T., Cremers, D., Bischof, H., Chambolle, A.: An algorithm for minimizing the Mumford–Shah functional. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 1133–1140. IEEE (2009)
46. Rasch, J., Chambolle, A.: Inexact first-order primal–dual algorithms. arXiv preprint arXiv:1803.10576 (2018)
47. Richardson, L.F.: Ix. the approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam. Phil. Trans. R. Soc. Lond. A **210**(459–470), 307–357 (1911)
48. Rockafellar, R.T., Wets, R.J.B.: Variational Analysis, vol. 317. Springer, Berlin (2009)
49. Saad, Y.: Iterative Methods for Sparse Linear Systems, vol. 82. SIAM, Philadelphia (2003)
50. Sidky, E.Y., Jørgensen, J.H., Pan, X.: Convex optimization problem prototyping for image reconstruction in computed tomography with the Chambolle–Pock algorithm. Phys. Med. Biol. **57**(10), 3065 (2012)
51. Valkonen, T.: A primal–dual hybrid gradient method for nonlinear operators with applications to MRI. Inverse Probl. **30**(5), 055012 (2014)

52. Vũ, B.C.: A splitting algorithm for dual monotone inclusions involving cocoercive operators. Adv. Comput. Math. **38**(3), 667–681 (2013)
53. Vũ, B.C.: A variable metric extension of the forward–backward–forward algorithm for monotone operators. Numer. Funct. Anal. Optim. **34**(9), 1050–1065 (2013)
54. Wang, Y., Yin, W., Zeng, J.: Global convergence of ADMM in nonconvex nonsmooth optimization. J. Sci. Comput. **78**(1), 29–63 (2019)
55. Wright, S.J.: Coordinate descent algorithms. Math. Program. **151**(1), 3–34 (2015)
56. Xu, Y.: Accelerated first-order primal–dual proximal methods for linearly constrained composite convex programming. SIAM J. Optim. **27**(3), 1459–1484 (2017)
57. Yan, M., Yin, W.: Self equivalence of the alternating direction method of multipliers. In: Glowinski, R., Osher, S., Yin, W. (Eds.). Splitting Methods in Communication, Imaging, Science, and Engineering, pp. 165–194. Springer, Berlin (2016)
58. Zhang, X., Burger, M., Osher, S.: A unified primal–dual algorithm framework based on Bregman iteration. J. Sci. Comput. **46**(1), 20–46 (2011)
59. Zhu, M., Chan, T.: An efficient primal–dual hybrid gradient algorithm for total variation image restoration. UCLA CAM Report 08–34 (2008)