INERTIAL PROXIMAL ADMM FOR LINEARLY CONSTRAINED SEPARABLE CONVEX OPTIMIZATION

CAIHUA CHEN*, RAYMOND H. CHAN†, SHIQIAN MA‡, AND JUNFENG YANG§

Abstract. The *alternating direction method of multipliers* (ADMM) is a popular and efficient first-order method that has recently found numerous applications, and the proximal ADMM is an important variant of it. The main contributions of this paper are the proposition and the analysis of a class of inertial proximal ADMMs, which unify the basic ideas of the inertial proximal point method and the proximal ADMM, for linearly constrained separable convex optimization. This class of methods are of inertial nature because at each iteration the proximal ADMM is applied to a point extrapolated at the current iterate in the direction of last movement. The recently proposed inertial primal-dual algorithm [1, Algorithm 3] and the inertial linearized ADMM [2, Eq. (3.23)] are covered as special cases. The proposed algorithmic framework is very general in the sense that the weighting matrices in the proximal terms are allowed to be only positive semidefinite, but not necessarily positive definite as required by existing methods of the same kind. By setting the two proximal terms to zero, we obtain an inertial variant of the classical ADMM, which is new to the best of our knowledge. We carry out a unified analysis for the entire class of methods under very mild assumptions. In particular, convergence, as well as asymptotic $o(1/\sqrt{k})$ and nonasymptotic $O(1/\sqrt{k})$ rates of convergence, are established for the best primal function value and feasibility residues, where k denotes the iteration counter. The global iterate convergence of the generated sequence is established under an additional assumption. We also present extensive experimental results on total variation based image reconstruction problems to illustrate the profits gained by introducing the inertial extrapolation steps.

Key words. *alternating direction method of multipliers* (ADMM), *proximal point method* (PPM), inertial PPM, proximal ADMM, inertial proximal ADMM.

AMS subject classifications. 65K05, 65K10, 65J22, 90C25

1. Introduction.

Let m, n_1 and n_2 be positive integers, $f : \Re^{n_1} \to \Re$ and $g : \Re^{n_2} \to \Re$ be closed convex functions, $\mathcal{X} \subseteq \Re^{n_1}$ and $\mathcal{Y} \subseteq \Re^{n_2}$ be closed convex sets, and $A \in \Re^{m\times n_1}$, $B \in \Re^{m\times n_2}$ and $b \in \Re^m$. In this paper, we consider linearly constrained separable convex optimization problem of the form

$$\min_{x,y}\{f(x) + g(y) : \text{ s.t. } Ax + By = b, x \in \mathcal{X}, y \in \mathcal{Y}\}. \tag{1.1}$$

A very important special case of (1.1) is given by $\min_y\{f(By) + g(y) : y \in \Re^{n_2}\}$, or, equivalently,

$$\min_{x,y}\{f(x) + g(y) : \text{ s.t. } -x + By = 0, x \in \Re^{n_1}, y \in \Re^{n_2}\}. \tag{1.2}$$

The functions f and g in (1.2) are often further assumed to be extended real-valued in order to incorporate additional side constraints. Problems like (1.2) arise from diverse applications such as signal and image reconstruction, compressive sensing and machine learning, etc., see, e.g., [3, 4, 5, 6, 7, 8, 9, 10] and references therein. On the other hand, both A and B can be generic linear operators as well, e.g., in compressive principal component pursuit [11] and matrix decomposition [12], the constraints appear as $\mathcal{A}(x + y) = b$, where $\mathcal{A}$ represents the measurement system. In this paper, we mainly focus on (1.1), though we also diverge to the special case (1.2) when we try to clarify connections between different algorithms.

*International Center of Management Science and Engineering, School of Management and Engineering, Nanjing University, China (Email: chchen@nju.edu.cn).

†Department of Mathematics, The Chinese University of Hong Kong, Hong Kong (Email: rchan@math.cuhk.edu.hk).

‡Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Hong Kong (Email: sqma@se.cuhk.edu.hk).

§Corresponding author. Department of Mathematics, Nanjing University, China (Email: jfyang@nju.edu.cn).

Two classical optimization methods are closely related to this work. They are the proximal *alternating direction method of multipliers* (proximal ADMM, [13, 14, 15]) and the inertial *proximal point method* (inertial PPM, [16, 17, 18]), which we will review very briefly below. We then summarize our main contributions, the notation and the organization of this paper.

1.1. Augmented Lagrangian related methods. Let the augmented Lagrange function associated with (1.1) be defined as

$$\mathcal{L}(x, y, p) := f(x) + g(y) - \langle p, Ax + By - b\rangle + \frac{\beta}{2}\|Ax + By - b\|^2,$$

where $p \in \Re^m$ is the Lagrange multiplier and $\beta > 0$ is a penalty parameter. Given $p^k \in \Re^m$, the classical *augmented Lagrangian method* [19, 20] (abbreviated as ALM) iterates as

$$(x^{k+1}, y^{k+1}) \in \arg\min_{x,y}\{\mathcal{L}(x, y, p^k) : \ x \in \mathcal{X}, y \in \mathcal{Y}\}, \tag{1.3a}$$

$$p^{k+1} = p^k - \beta(Ax^{k+1} + By^{k+1} - b), \tag{1.3b}$$

where "arg min" represents the collection of minimizers. When f and g have structures that one can exploit, it is favorable to utilize the separability of the objective function, rather than applying a joint minimization with (x, y). The ADMM [13, 14] applies alternating minimization with x and y in (1.3a) in a Gauss-Seidel fashion followed by immediate update of the dual variable p in (1.3b). Here we shall present a cyclically equivalent form of the ADMM. Given $(y^k, p^k) \in \mathcal{Y} \times \Re^m$, the ADMM in "$x - p - y$" order updates the variables as follows:

$$x^{k+1} \in \arg\min_{x}\{\mathcal{L}(x, y^k, p^k) : x \in \mathcal{X}\}, \tag{1.4a}$$

$$p^{k+1} = p^k - \beta(Ax^{k+1} + By^k - b), \tag{1.4b}$$

$$y^{k+1} \in \arg\min_{y}\{\mathcal{L}(x^{k+1}, y, p^{k+1}) : y \in \mathcal{Y}\}. \tag{1.4c}$$

Compared to the ALM, an obvious advantage of the ADMM is that it solves simpler subproblems in each round and can utilize the structures of f and g individually. It is well known that the dual sequence $\{p^k\}$ generated by (1.4) converges to an optimal solution of the dual problem if (1.1) possesses a KKT point, but without additional conditions the sequence of primal iterates does not necessarily converge. To improve the primal convergence, Eckstein first proposed in [15] a proximal ADMM by adding some quadratic terms to the subproblems of (1.4). This variant will be discussed in detail in Section 2.

1.2. PPM and its inertial variant. Another closely related method is the PPM [21, 22, 23], which is an approach for finding a zero of a maximal monotone operator T on $\Re^n$. The primary PPM for minimizing a differentiable function $\psi : \ \Re^n \to \Re$ can be interpreted as an implicit one-step discretization method for the *ordinary differential equations* (ODEs) $w' + \nabla\psi(w) = 0$, where $w : \Re \to \Re^n$ is differentiable, w' denotes its derivative, and $\nabla\psi$ is the gradient of ψ.

To accelerate speed of convergence of the PPM, multi-step methods have been proposed in the literature, which can usually be viewed as certain discretizations of the second-order ODEs

$$w'' + \gamma w' + \nabla\psi(w) = 0, \tag{1.5}$$

where $\gamma > 0$ represents a friction parameter. For example, an implicit discretization method was proposed in [17]. Specifically, given w^{k-1} and w^k, the next point w^{k+1} is determined via

$$\frac{w^{k+1} - 2w^k + w^{k-1}}{h^2} + \gamma\frac{w^{k+1} - w^k}{h} + \nabla\psi(w^{k+1}) = 0,$$

which results in an iterative algorithm of the form

$$w^{k+1} = (I + \lambda\nabla\psi)^{-1}(w^k + \alpha(w^k - w^{k-1})), \tag{1.6}$$

where $\lambda = h^2/(1+\gamma h)$, $\alpha = 1/(1+\gamma h)$ and I is the identity operator. Note that (1.6) can be viewed as applying the PPM to the extrapolated point $w^k + \alpha(w^k - w^{k-1})$ and is usually referred as inertial PPM. Subsequently, this inertial technique was extended to solve the maximal monotone operator inclusion problem in [18, 24, 25, 26]. Recently, there are increasing interests in studying inertial type algorithms, e.g., inertial forward-backward splitting methods [27, 28, 29, 30, 31, 32], inertial Douglas-Rachford operator splitting method [33] and inertial ADMM [34]. In particular, when restricted to (1.2), the method in [29] with certain specialized preconditioner reduces to [1, Algorithm 3], which is also a special case of the inertial proximal ADMM proposed in this paper. Global convergence results were obtained there under roughly the same conditions as used in this paper, but convergence rate results were not given. Our inertial algorithms also cover an inertial ADMM as well. A method with the same name was discussed in [34] but we will explain later that the two methods are in fact quite different.

1.3. Contributions. Our main contributions of this paper are twofold. First, we propose a class of inertial variants of the general weighted proximal ADMM (see (2.2) and Algorithm 1), where the weighting matrices are allowed to be positive semidefinite, but not necessarily positive definite. This class of inertial algorithms unify and largely extend the existing inertial primal-dual algorithm [1, Algorithm 3] (i.e., inertial variant of case 3 restricted to the special case (1.2) in Table 2.1) and the inertial linearized ADMM [2, Eq. (3.23)] (i.e., inertial variant of case 4 in Table 2.1). Apart from these two special cases, all the other inertial variants covered in Table 2.1, including case 3 for the generic problem (1.1), are new. In particular, by setting both weighting matrices to zero, we obtain an inertial variant of the original ADMM (see Algorithm 2, which corresponds to inertial variant of case 1 in Table 2.1). Second, under very mild assumptions, we establish global convergence, asymptotic $o(1/\sqrt{k})$ and nonasymptotic $O(1/\sqrt{k})$ rates of convergence for the best primal function value and feasibility residues, where k denotes the iteration counter (see Theorems 4.3, 4.4 and 4.6). The global iterate convergence of the generated sequence is established under an additional assumption (see Theorem 4.7). Furthermore, we evaluate the practical performance of this class of inertial algorithms by comparing them with the corresponding original algorithms on some imaging problems.

1.4. Notation and organization. Our notation is standard, as used above in this section. The standard inner product and ℓ_2-norm are denoted by $\langle\cdot,\cdot\rangle$ and $\|\cdot\|$, respectively. The superscript "^{T}" denotes the matrix/vector transpose operator. The fact that a matrix M is a symmetric and positive semidefinite (resp. positive definite) is denoted by $M \succeq 0$ (resp. $M \succ 0$). For any $M \succeq 0$ of size n-by-n and vectors $u, v \in \Re^n$, we let $\langle u, v\rangle_M := u^T M v$ and $\|u\|_M := \sqrt{\langle u, u\rangle_M}$. The spectral radius of a square matrix M is denoted by $\rho(M)$. The identity matrix of appropriate orders will be denoted by I. Zero matrices and vectors are simply denoted by 0. With a little abuse of notation, the columnwise adhesion of columns vectors x, y and p, i.e., $(x^T, y^T, p^T)^T$, is often denoted by (x, y, p) whenever it will not incur any confusion. Other notation will be introduced as the presentation progresses.

The rest of the paper is organized as follows. In Section 2, we describe a general proximal ADMM, characterize it as a mixed variational inequality, and clarify its connections to some existing methods. In Section 3, we propose a class of inertial proximal ADMMs, whose convergence is analyzed in Section 4. Numerical results and concluding remarks are given, respectively, in Sections 5 and 6.

2. Proximal ADMM. One type of structure that is usually preserved by f and g in many applications is that their proximity operators are easy to evaluate. Let $\gamma > 0$ be a scalar. The proximity operator of a

closed proper convex function $h : \Re^n \to (-\infty, +\infty]$ is defined as

$$\text{prox}_\gamma^h(x) := \arg\min_z \{h(z) + \|z - x\|^2/(2\gamma) : \ z \in \Re^n\}, \ x \in \Re^n. \tag{2.1}$$

In general, (1.4a) and (1.4c) are not easy to solve even when f and g are simple. To avoid inner loop, they are usually modified by linearizing the quadratic term of $\mathcal{L}(x, y, p)$ with respect to x (resp., y) and meanwhile adding a proximal term in ℓ_2-norm. This technique, which we will refer to as proximal-linearization below, has been used extensively in, e.g., [35, 36, 37, 6, 5]. The popular primal-dual algorithm [8, 3] also modifies one subproblem via this proximal-linearization, see [8, 3, 5]. The algorithms resulting from modifying ADMM subproblems via proximal-linearization are special realizations of the following proximal ADMM (generalization of the algorithm in [15]): given $w^k = (x^k, y^k, p^k) \in \mathcal{X} \times \mathcal{Y} \times \Re^m$, iterate as

$$x^{k+1} \in \arg\min_x \{\mathcal{L}(x, y^k, p^k) + \|x - x^k\|_S^2/2 : x \in \mathcal{X}\}, \tag{2.2a}$$

$$p^{k+1} = p^k - \beta(Ax^{k+1} + By^k - b), \tag{2.2b}$$

$$y^{k+1} \in \arg\min_y \{\mathcal{L}(x^{k+1}, y, p^{k+1}) + \|y - y^k\|_T^2/2 : y \in \mathcal{Y}\}, \tag{2.2c}$$

where $S, T \succeq 0$. In particular, by setting $S = (\beta/\tau)I - \beta A^T A$ (resp. $T = (\beta/\eta)I - \beta B^T B$), where $\tau > 0$ (resp. $\eta > 0$), the overlapping of components of x (resp. y) in the quadratic term of $\mathcal{L}(x, y, p)$ can be canceled out. Similar algorithms have different names in the literature, such as split inexact Uzawa, proximal or linearized ADMM, see, e.g., [38, 8, 5, 2]. This general proximal ADMM framework goes back to at least [38], where the focus is *variational inequality* (VI) problem with separable structures. See also [39] for a comprehensive study on the convergence of (2.2).

2.1. Mixed VI characterization. We now present the mixed VI characterization of the primal-dual optimality conditions of (1.1) and the proximal ADMM (2.2). As before, we denote the dual variable by p. Define $\mathcal{W}$, w, θ and F, respectively, by $\mathcal{W} := \mathcal{X} \times \mathcal{Y} \times \Re^m$,

$$w := \begin{pmatrix} x \\ y \\ p \end{pmatrix}, \quad \theta(w) := f(x) + g(y), \quad F(w) := \begin{pmatrix} 0 & 0 & -A^T \\ 0 & 0 & -B^T \\ A & B & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ p \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ b \end{pmatrix}. \tag{2.3}$$

Clearly, F is monotone. Throughout this paper, we assume that the set of KKT points of (1.1), denoted by $\mathcal{W}^*$, is nonempty. Then solving (1.1) amounts to determining a solution of the mixed VI problem: find $w^* \in \mathcal{W}$ such that

$$\theta(w) - \theta(w^*) + \langle w - w^*, F(w^*)\rangle \geq 0, \ \forall w \in \mathcal{W}. \tag{2.4}$$

The following result explains how the proximal ADMM can be interpreted as a proximal-like method applied to (2.4). We omit the proof since it is a simple generalization of the result in [40].

THEOREM 2.1. *The new point $w^{k+1} = (x^{k+1}, y^{k+1}, p^{k+1})$ generated by the proximal ADMM (2.2) from a given $w^k = (x^k, y^k, p^k) \in \mathcal{W}$ satisfies*

$$w^{k+1} \in \mathcal{W}, \ \theta(w) - \theta(w^{k+1}) + \left\langle w - w^{k+1}, F(w^{k+1}) + G(w^{k+1} - w^k)\right\rangle \geq 0, \ \forall w \in \mathcal{W}, \tag{2.5}$$

where G is defined by

$$G := \begin{pmatrix} S & 0 & 0 \\ 0 & \beta B^T B + T & -B^T \\ 0 & -B & \frac{1}{\beta} I \end{pmatrix}. \tag{2.6}$$

TABLE 2.1
Some special cases of the proximal ADMM (2.2) *for* (1.1). *G is given in* (2.6).

Case	(1.4a)	(1.4c)	S	T	$G \succ 0$
1	intact	intact	0	0	never
2	prox-linearize	intact	$\frac{\beta}{\tau}I - \beta A^TA$	0	never
3	intact	prox-linearize	0	$\frac{\beta}{\eta}I - \beta B^TB$	never
4	prox-linearize	prox-linearize	$\frac{\beta}{\tau}I - \beta A^TA$	$\frac{\beta}{\eta}I - \beta B^TB$	$0 < \tau < 1/\rho(A^TA)$ $0 < \eta < 1/\rho(B^TB)$
5	$+\frac{1}{2}\|x - x^k\|_S^2$	$+\frac{1}{2}\|y - y^k\|_T^2$	$S \succeq 0$	$T \succeq 0$	depends

2.2. Related methods. Clearly, we recover the original ADMM by letting $S = 0$ and $T = 0$ in (2.2). If we merely proximal-linearize (1.4a) at x^k, the resulting algorithm corresponds to (2.2) with $S = \frac{\beta}{\tau}I - \beta A^TA$ and $T = 0$. Alternatively, we can proximal-linearize (1.4c) and keep (1.4a) intact, in which case the resulting algorithm corresponds to (2.2) with $S = 0$ and $T = \frac{\beta}{\eta}I - \beta B^TB$. Clearly, $G \succ 0$ never holds for these three cases. On the other hand, if we proximal-linearize both subproblems simultaneously, the resulting algorithm corresponds to (2.2) with $S = \frac{\beta}{\tau}I - \beta A^TA$ and $T = \frac{\beta}{\eta}I - \beta B^TB$. In this case, G is indeed positive definite if $\tau, \eta > 0$ are sufficiently small. These special cases of (2.2) are summarized in Table 2.1.

We now focus on (1.2) temporarily. Due to $A = -I$ and resort to the relation $(\partial f)^{-1} = \partial f^*$, see, e.g., [41], where ∂f and f^* denote, respectively, the subdifferential operator and the convex conjugate of f, we can eliminate the x variable in w in (2.4). When applying ADMM (1.4) to (1.2), there is no need to modify (1.4a) since it already amounts to evaluating the proximity operator of f. It is only necessary to proximal-linearize (1.4c). The resulting algorithm corresponds to case 3 in Table 2.1 and appears as

$$x^{k+1} = \mathrm{prox}^f_{1/\beta}(By^k - p^k/\beta), \tag{2.7a}$$

$$p^{k+1} = p^k - \beta(-x^{k+1} + By^k), \tag{2.7b}$$

$$y^{k+1} = \mathrm{prox}^g_{\eta/\beta}\left(y^k - \eta B^T(By^k - x^{k+1} - p^{k+1}/\beta)\right). \tag{2.7c}$$

From [3, 5], (2.7) is equivalent to the primal-dual algorithm [42, 8, 3]. Subsequently, it was shown in [43] that (2.7) can be explained as a weighted PPM. Specifically, $w^{k+1} = (y^{k+1}, p^{k+1})$ generated by (2.7) from a given $w^k = (y^k, p^k)$ satisfies (2.5) with $\mathcal{W}$, w, θ, F and G defined, respectively, as $\mathcal{W} = \Re^{n_2} \times \Re^m$, $w = (y, p)$, $\theta(w) = g(y) + f^*(-p)$,

$$F(w) = \begin{pmatrix} 0 & -B^T \\ B & 0 \end{pmatrix}\begin{pmatrix} y \\ p \end{pmatrix} \quad \text{and} \quad G = \begin{pmatrix} \frac{\beta}{\eta}I & -B^T \\ -B & \frac{1}{\beta}I \end{pmatrix}. \tag{2.8}$$

Clearly, G defined in (2.8) is symmetric and positive definite if $0 < \eta < 1/\rho(B^TB)$. In this sense, the primal-dual algorithm [42, 8, 3] for (1.2) also falls into the framework of PPM as specified in (2.5).

2.3. Motivation of this paper. Recently, we proposed in [2, Eq. (3.23)] an inertial variant of case 4 in Table 2.1. Global iterate convergence and certain convergence rate results were established under the condition that the weighting matrix G defined in (2.6) is positive definite. Our numerical results have shown that the inertial extrapolation steps can accelerate convergence in practice. When restricted to (1.2), we can directly extend the results in [2] to the inertial primal-dual algorithm [1, Algorithm 3], because the primal-dual algorithm [42, 8, 3] is an application of a weighted PPM [43].

We emphasize that the convergence guarantee of [2, Eq. (3.23)] and [1, Algorithm 3] depends heavily on the positive definiteness of G, which, however, when restricted to (1.1) never holds for the first three cases

in Table 2.1. On the other hand, in many applications the matrices A and B have special structures and, as a result, both ADMM subproblems (1.4a) and (1.4c) can be solved exactly and conveniently without any approximation or modification. For such cases, it is not desired to modify either ADMM subproblem. It is thus desirable to consider inertial variant of the original ADMM, the convergence of which, however, cannot be covered by existing results. This motivates the current work.

3. Inertial proximal ADMM. We now present our inertial proximal ADMM. At each iteration, the inertial proximal ADMM first extrapolates at the current point in the direction of last movement and then applies the proximal ADMM to the extrapolated point. The overall algorithm is summarized below.

ALGORITHM 1 (Inertial proximal ADMM). *Given $S \succeq 0$, $T \succeq 0$, $(x^0, y^0, p^0) \in \mathcal{W}$, $\beta > 0$ and a sequence of nonnegative parameters $\{\alpha_k\}_{k=0}^{\infty}$. Let $(x^{-1}, y^{-1}, p^{-1}) = (x^0, y^0, p^0)$. For $k \geq 0$, iterate as*

$$(\bar{x}^k, \bar{y}^k, \bar{p}^k) = (x^k, y^k, p^k) + \alpha_k(x^k - x^{k-1}, y^k - y^{k-1}, p^k - p^{k-1}), \tag{3.1a}$$

$$x^{k+1} \in \arg\min_{x \in \mathcal{X}} \mathcal{L}(x, \bar{y}^k, \bar{p}^k) + \|x - \bar{x}^k\|_S^2/2, \tag{3.1b}$$

$$p^{k+1} = \bar{p}^k - \beta(Ax^{k+1} + B\bar{y}^k - b), \tag{3.1c}$$

$$y^{k+1} \in \arg\min_{y \in \mathcal{Y}} \mathcal{L}(x^{k+1}, y, p^{k+1}) + \|y^k - \bar{y}^k\|_T^2/2. \tag{3.1d}$$

Let $\bar{w}^k := w^k + \alpha_k(w^k - w^{k-1})$. According to Theorem 2.1, $w^{k+1} := (x^{k+1}, y^{k+1}, p^{k+1})$ generated by (3.1) satisfies

$$w^{k+1} \in \mathcal{W},\ \theta(w) - \theta(w^{k+1}) + \left\langle w - w^{k+1}, F(w^{k+1}) + G(w^{k+1} - \bar{w}^k) \right\rangle \geq 0,\ \forall w \in \mathcal{W}, \tag{3.2}$$

where $\mathcal{W}$, w, θ and F are defined in (2.3), and G is given by (2.6).

Clearly, the proposed algorithmic framework (3.1) unifies [1, Algorithm 3] and [2, Eq. (3.23)]. It applies to the general problem (1.1), rather than (1.2) only, and contains inertial variants of linearized ADMM with either one or both of the subproblems being proximal-linearized. Moreover, S and T are not restricted to the special form $c_1 I - c_2 A^T A$ or $c_1 I - c_2 B^T B$ for some constants $c_1, c_2 > 0$ associated with the linearized ADMM, but can be generic positive semidefinite matrices. These are all new features of the proposed algorithm framework. Our unified analysis for the entire class of algorithms and the convergence results established under the relaxed condition $S \succeq 0$ and $T \succeq 0$ are new to the literature.

By setting $S = 0$ and $T = 0$ in (3.1), we obtain an inertial ADMM, which is summarized below.

ALGORITHM 2 (Inertial ADMM). *Given $(y^0, p^0) \in \mathcal{Y} \times \Re^m$, $\beta > 0$ and a sequence of nonnegative parameters $\{\alpha_k\}_{k=0}^{\infty}$. Let $(y^{-1}, p^{-1}) = (y^0, p^0)$. For $k \geq 0$, iterate as*

$$(\bar{y}^k, \bar{p}^k) = (y^k, p^k) + \alpha_k(y^k - y^{k-1}, p^k - p^{k-1}), \tag{3.3a}$$

$$x^{k+1} \in \arg\min_{x \in \mathcal{X}} \mathcal{L}(x, \bar{y}^k, \bar{p}^k), \tag{3.3b}$$

$$p^{k+1} = \bar{p}^k - \beta(Ax^{k+1} + B\bar{y}^k - b), \tag{3.3c}$$

$$y^{k+1} \in \arg\min_{y \in \mathcal{Y}} \mathcal{L}(x^{k+1}, y, p^{k+1}). \tag{3.3d}$$

Though Algorithm 2 has exactly the same name as [34, Algorithm 5], we need to point out that the two algorithms are in fact quite different. Here we will not write out the algorithm of [34, Algorithm 5] as it will be tedious. We only point out the main differences between the two algorithms. First, [34, Algorithm 5] is designed for solving (1.2) only, while our Algorithm 2 solves the general problem (1.1). Second, [34, Algorithm 5] is an application of the inertial Douglas-Rachford splitting method [33] applied to the dual

problem of (1.2), while our Algorithm 2 is not. Third, the parametric conditions assumed in [34, Algorithm 5] to ensure global convergence are very different from ours (see Section 4).

4. Convergence analysis. In this section, we establish global convergence and convergence rate results in the best function value and feasibility residues of the proposed inertial proximal ADMM. In particular, the obtained results apply to the inertial ADMM described in Algorithm 2 and extend those in [2]. We make the following assumption on the sequence of parameters $\{\alpha_k\}_{k=0}^{\infty}$.

ASSUMPTION 1. *Assume that $\{\alpha_k\}_{k=0}^{\infty}$ is chosen such that (i) for all $k \geq 0$, $0 \leq \alpha_k \leq \alpha$ for some $\alpha \in [0, 1)$, and (ii) the sequence of points $\{w^k\}_{k=0}^{\infty}$ generated by (3.1), or equivalently, (3.2), satisfies*

$$\sum_{k=0}^{\infty} \alpha_k \|w^k - w^{k-1}\|_G^2 < \infty. \tag{4.1}$$

We note that one way to ensure Assumption 1 in practice is to determine $\{\alpha_k\}_{k=0}^{\infty}$ adaptively. Alternatively, it is simultaneously satisfied if $\{\alpha_k\}_{k=0}^{\infty}$ satisfies some further conditions, see, e.g., [18, Prop. 2.1], [44, Sec. 2], [25, Prop. 2.5] and Proposition 4.5 in Section 4. We first give some lemmas which are useful in our analysis.

LEMMA 4.1. *Let $\{w^k\}_{k=0}^{\infty}$ be generated by the inertial proximal ADMM given in Algorithm 1. Then, for any $w^* \in \mathcal{W}^*$, it holds that*

$$\langle A(x^{k+1} - x^*),\, p^{k+1} - p^* \rangle \geq \langle x^{k+1} - x^*,\, S(x^{k+1} - \bar{x}^k) \rangle. \tag{4.2}$$

Proof. Let $k \geq 0$ and $x \in \mathcal{X}$. It follows from the optimality condition of (3.1b) that

$$f(x) - f(x^{k+1}) + \left\langle x - x^{k+1},\, -A^T p^{k+1} + S(x^{k+1} - \bar{x}^k) \right\rangle \geq 0. \tag{4.3}$$

Since $w^* \in \mathcal{W}^*$, by setting $w = (x, y^*, p^*)$ in (2.4) we obtain

$$f(x) - f(x^*) + \left\langle x - x^*,\, -A^T p^* \right\rangle \geq 0. \tag{4.4}$$

Setting $x = x^*$ in (4.3) and $x = x^{k+1}$ in (4.4), and adding them together, we get (4.2) immediately. □

The following lemma gathers several useful facts which facilitate the convergence analysis of the proposed inertial proximal ADMM. Since its proof follows essentially from [18, 44, 24], we omit the details.

LEMMA 4.2. *Suppose that $\{\alpha_k\}_{k=0}^{\infty}$ satisfies Assumption 1. Let $\{w^k\}_{k=0}^{\infty}$ be generated by the inertial proximal ADMM given in Algorithm 1. The following two statements hold.*

(i) Let $\bar{w}^k := w^k + \alpha_k(w^k - w^{k-1})$, then

$$\sum_{k=0}^{\infty} \|w^{k+1} - \bar{w}^k\|_G^2 < \infty, \tag{4.5}$$

and hence $\lim_{k\to\infty} \|w^{k+1} - \bar{w}^k\|_G = 0$.

(ii) For any $w^* \in \mathcal{W}^*$, $\lim_{k\to\infty} \|w^k - w^*\|_G$ exists, and furthermore, it holds that

$$\|w^k - w^*\|_G^2 \leq \|w^0 - w^*\|_G^2 + \frac{2}{1-\alpha} \sum_{j=1}^{\infty} \alpha_j \|w^j - w^{j-1}\|_G^2,\ \forall\ k \geq 1. \tag{4.6}$$

With Lemma 4.2 at hand and by using the special structures of (1.1) and (3.1), we are able to establish the following theorem on the feasibility and objective convergence of the inertial proximal ADMM.

THEOREM 4.3 (Convergence). *Suppose that $\{\alpha_k\}_{k=0}^{\infty}$ satisfies Assumption 1. Let $\{w^k\}_{k=0}^{\infty}$ be generated by the inertial proximal ADMM given in Algorithm 1. Then, we have the following results.*

(i) $\sum_{k=1}^{\infty}\|Ax^k+By^k-b\|^2<\infty$, and hence $\lim_{k\to\infty}\|Ax^k+By^k-b\|=0$;

(ii) The objective function value $f(x^k)+g(y^k)$ converges to the optimal value of (1.1) as $k\to\infty$.

Proof. (i) From (3.1c), $S,T\succeq 0$ and the definition of G, we obtain that

$$\|Ax^{k+1}+By^{k+1}-b\|^2=\|(By^{k+1}-p^{k+1}/\beta)-(B\bar{y}^k-\bar{p}^k/\beta)\|^2\le\frac{1}{\beta}\|w^{k+1}-\bar{w}^k\|_G^2. \tag{4.7}$$

The conclusion $\sum_{k=1}^{\infty}\|Ax^k+By^k-b\|^2<\infty$ follows from (4.7) and (4.5).

(ii) Let $w^*=(x^*,y^*,p^*)\in\mathcal{W}^*$. It follows from setting $w=(x^k,y^k,p^*)$ in (2.4) and the definition of F in (2.3) that

$$f(x^k)+g(y^k)-f(x^*)-g(y^*)\ge\langle p^*,\,Ax^k+By^k-b\rangle. \tag{4.8}$$

Therefore, it follows from $\lim_{k\to\infty}(Ax^k+By^k-b)=0$ that

$$\liminf_{k\to\infty}\big(f(x^k)+g(y^k)\big)\ge f(x^*)+g(y^*). \tag{4.9}$$

On the other hand, by setting $w=w^*$ in (3.2) we obtain that

$$f(x^*)+g(y^*)-f(x^{k+1})-g(y^{k+1})\ge-\langle p^*,\,Ax^{k+1}+By^{k+1}-b\rangle+\langle w^{k+1}-w^*,\,G(w^{k+1}-\bar{w}^k)\rangle. \tag{4.10}$$

It follows from $Ax^k+By^k\to b$, $\|w^{k+1}-\bar{w}^k\|_G\to 0$ and the boundness of $\{\|w^k-w^*\|_G\}_{k=0}^{\infty}$ that

$$\limsup_{k\to\infty}\big(f(x^k)+g(y^k)\big)\le f(x^*)+g(y^*), \tag{4.11}$$

which, together with (4.9), completes the proof of (ii). □

The following theorem establishes certain asymptotic convergence results of the proposed inertial proximal ADMM. Specifically, parts (ii) and (iii) of the theorem present asymptotic $o(1/\sqrt{k})$ convergence rate results measured by the best residues in primal feasibility and function values, respectively. These results are consequences of the structures of (1.1) and the iterative scheme (3.1), as well as part (i) of the theorem, the validity of which had been implied by the analysis in [18] for the inertial PPM. We note that little-o convergence results already appeared in, e.g., [45] for parallel multi-block ADMM.

THEOREM 4.4 (Asymptotic convergence rate). *Suppose that $\{\alpha_k\}_{k=0}^{\infty}$ satisfies Assumption 1. Let $\{w^k\}_{k=0}^{\infty}$ be generated by the inertial proximal ADMM given in Algorithm 1. Then, there hold as $k\to\infty$,*

(i) $\min_{1\le i\le k}\|w^i-\bar{w}^{i-1}\|_G=o(1/\sqrt{k})$,

(ii) $\min_{1\le i\le k}\|Ax^i+By^i-b\|=o(1/\sqrt{k})$,

(iii) $\min_{1\le i\le k}|f(x^i)+g(y^i)-f(x^*)-g(y^*)|=o(1/\sqrt{k})$.

Proof. Let $1\le i\le k$ be arbitrarily fixed. Part (i) follows directly from (4.5) and the Cauchy principle. Part (ii) follows immediately by noting (4.7). From (4.8) and (4.10), we know that

$$|f(x^i)+g(y^i)-f(x^*)-g(y^*)|\le|\langle p^*,\,Ax^i+By^i-b\rangle|+|\langle w^i-w^*,\,G(w^i-\bar{w}^{i-1})\rangle|, \tag{4.12}$$

which, together with (i), (ii) and Lemma 4.2, completes the proof of (iii). □

REMARK 1. *We note that the asymptotic $o(1/\sqrt{k})$ convergence result of the best function value residue given in Part (iii) of Theorem 4.4 alone does not indicate a convergence speed of the algorithm because the proposed inertial proximal ADMM is an infeasible one in general. However, since we also establish the same result for the best feasibility residual, a combination of the two results given in Parts (ii) and (iii) of Theorem 4.4 implies an asymptotic $o(1/\sqrt{k})$ convergence speed of the algorithm. Similar remarks also apply to the nonasymptotic convergence results given below in Theorem 4.6.*

The results given in Theorem 4.4 are asymptotic in the sense that they hold only when $k \to \infty$. To derive some nonasymptotic convergence results, we further assume that $\{\alpha_k\}_{k=0}^{\infty}$ is monotonically nondecreasing and bounded above by some $0 \le \alpha < 1/3$. In fact, these conditions on $\{\alpha_k\}_{k=0}^{\infty}$ also ensure the validity of Assumption 1. The following proposition summarizes this fact and an additional bound result, which are very useful in our nonasymptotic convergence analysis. Since its proof stems from previous results in [18, 44, 25], we omit the details.

PROPOSITION 4.5. *Suppose that, for all $k \ge 0$, $0 \le \alpha_k \le \alpha_{k+1} \le \alpha$ for some $0 \le \alpha < 1/3$. Let $\{w^k\}_{k=0}^{\infty}$ be generated by the inertial proximal ADMM Algorithm 1. Then Assumption 1 is valid. Furthermore, it holds for any $w^* \in \mathcal{W}^*$ that*

$$\sum_{k=1}^{\infty} \|w^k - w^{k-1}\|_G^2 \le \frac{2\|w^0 - w^*\|_G^2}{1-3\alpha}. \tag{4.13}$$

Now, we are ready to establish our nonasymptotic convergence results.

THEOREM 4.6 (Nonasymptotic convergence rate). *Suppose that $0 \le \alpha_k \le \alpha_{k+1} \le \alpha < \frac{1}{3}$ for all k. Let $\{w^k\}_{k=0}^{\infty}$ be generated by the inertial proximal ADMM given in Algorithm 1. Then, it holds for any $k \ge 1$ and $w^* = (x^*, y^*, p^*) \in \mathcal{W}^*$ that*

(i) $\min\limits_{1\le i\le k} \|w^i - \bar{w}^{i-1}\|_G \le C_1/\sqrt{k}$,

(ii) $\min\limits_{1\le i\le k} \|Ax^i + By^i - b\| \le C_2/\sqrt{k}$,

(iii) $\min\limits_{1\le i\le k} |f(x^i) + g(y^i) - f(x^*) - g(y^*)| \le C_1 \left(\|p^*\|/\sqrt{\beta} + C_3\right)/\sqrt{k}$,

where $C_1 := 2\sqrt{\frac{1+\alpha^2}{1-3\alpha}}\|w^0 - w^*\|_G$, $C_2 := C_1/\sqrt{\beta}$ and $C_3 := \sqrt{1 + \frac{4\alpha}{(1-\alpha)(1-3\alpha)}}\|w^0 - w^*\|_G$.

Proof. It follows from the definition of $\bar{w}^k$ and (4.13) that

$$\begin{aligned}
\sum_{i=1}^{k} \|w^i - \bar{w}^{i-1}\|_G^2 &\le 2\left(\sum_{i=1}^{k} \|w^i - w^{i-1}\|_G^2 + \alpha^2 \sum_{i=1}^{k} \|w^{i-1} - w^{i-2}\|_G^2\right) \\
&\le 2\left(\sum_{i=1}^{\infty} \|w^i - w^{i-1}\|_G^2 + \alpha^2 \sum_{i=1}^{\infty} \|w^{i-1} - w^{i-2}\|_G^2\right) \\
&= 2(1+\alpha^2) \sum_{i=1}^{\infty} \|w^i - w^{i-1}\|_G^2 \le \frac{4(1+\alpha^2)\|w^0 - w^*\|_G^2}{1-3\alpha},
\end{aligned} \tag{4.14}$$

where the "=" follows from $w^0 = w^{-1}$. Part (i) follows immediately from (4.14), and part (ii) follows from (4.7). To prove part (iii), we first note from (4.6) and (4.13) that, for any $i \ge 1$,

$$\|w^i - w^*\|_G^2 \le \|w^0 - w^*\|_G^2 + \frac{2}{1-\alpha} \sum_{j=1}^{\infty} \alpha_j \|w^j - w^{j-1}\|_G^2 \le \left(1 + \frac{4\alpha}{(1-\alpha)(1-3\alpha)}\right) \|w^0 - w^*\|_G^2 = C_3^2.$$

This, together with (4.12), completes the proof of part (iii). □

REMARK 2. *For the proximal ADMM* (2.2)*, the quantity* $\|w^k - w^{k-1}\|_G$ *is monotonically nonincreasing with* k*. However, this property does not hold for its inertial variant* (3.1)*. As a result, we are not able to remove the "*$\min_{1\le i\le k}$*" in our results. We note that either with or without the "*$\min_{1\le i\le k}$*", a nonasymptotic* $O(1/\sqrt{k})$ *convergence rate would imply that an* ε*-accuracy solution in the sense that* $\|w^k - \bar{w}^{k-1}\|_G \le \varepsilon$ *is obtainable within no more than* $O(1/\varepsilon^2)$ *iterations.*

REMARK 3. *Assume that* $\alpha_k = 0$ *for all* k*. Then the "*$\min_{1\le i\le k}$*" can be removed by setting* $i = k$ *in Theorems 4.4 and 4.6. If further restricted to the ADMM* (1.4)*, i.e.,* $S = 0$ *and* $T = 0$*, then the asymptotic* $o(1/\sqrt{k})$ *and the nonasymptotic* $O(1/\sqrt{k})$ *convergence rate results given in Theorems 4.4 and 4.6 coincide with those given in [46, Theorem 13, Theorem 15]. It is well-known that the ADMM is a dual application of the Douglas-Rachford splitting method, and it has been shown in [46, Theorem 8, Sec. 6.1.1] that the* $o(1/\sqrt{k})$ *convergence rate of Douglas-Rachford splitting method measured in fixed point residue is tight. It is also pointed out in [46] that the Douglas-Rachford splitting method (including the ADMM as a special case) can be nearly as slow as the subgradient method in the nonergodic sense.*

Note that Theorem 4.3 does not ensure the iterate convergence of $\{w^k\}_{k=0}^{\infty}$. In fact, the iterate convergence of $\{w^k\}_{k=0}^{\infty}$ can be guaranteed under some further conditions. The convergence result given in the next theorem cannot be derived from analytic techniques analogous to those existing in the literature for inertial type methods. Its validity relies on the structure of the problem and the iterative scheme considered in this paper.

THEOREM 4.7 (Convergence). *Suppose that* $\{\alpha_k\}_{k=0}^{\infty}$ *satisfies Assumption 1. Let* $\{w^k\}_{k=0}^{\infty}$ *be generated by the inertial proximal ADMM given in Algorithm 1. Then,* $\{(S+\beta A^TA)x^k\}_{k=1}^{\infty}$, $\{(T+\beta B^TB)y^k\}_{k=1}^{\infty}$ *and* $\{p^k\}_{k=1}^{\infty}$ *are all bounded. Furthermore, if* $S + \beta A^TA \succ 0$ *and* $T + \beta B^TB \succ 0$*, then* $\{w^k\}_{k=0}^{\infty}$ *converges to a member of* $\mathcal{W}^*$ *as* $k \to \infty$*.*

Proof. For any $w^* \in \mathcal{W}^*$, it follows from Lemma 4.2 that $\lim_{k\to\infty}\|w^k - w^*\|_G$ exists. Thus, the sequence $\{Gw^k\}_{k=0}^{\infty}$ must be bounded. As a result, it follows from the definition of G in (2.6) that the sequences $\{Sx^k\}_{k=0}^{\infty}$, $\{Ty^k\}_{k=0}^{\infty}$ and $\{By^k - p^k/\beta\}_{k=0}^{\infty}$ must be all bounded. This, together with part (ii) of Theorem 4.3, implies the boundedness of $\{Ax^k + p^k/\beta\}_{k=0}^{\infty}$. Moreover, we know from (4.2) that

$$\langle A(x^k - x^*),\, p^k - p^*\rangle \ge \langle x^k - x^*,\, S(x^k - \bar{x}^{k-1})\rangle \ge -(\|x^k - x^*\|_S^2 + \|x^k - \bar{x}^{k-1}\|_S^2)/2,$$

for $k \ge 1$. By further considering the boundedness of $\{Sx^k\}_{k=0}^{\infty}$, we deduce that $\langle A(x^k - x^*),\, p^k - p^*\rangle$ is bounded from below for $k \ge 1$. Then, by the elementary equality

$$\|A(x^k - x^*)\|^2 + \|(p^k - p^*)\beta\|^2 = \|(Ax^k + p^k/\beta) - (Ax^* + p^*/\beta)\|^2 - (2/\beta)\langle A(x^k - x^*),\, p^k - p^*\rangle,$$

it follows that $\{Ax^k\}_{k=0}^{\infty}$ and $\{p^k\}_{k=0}^{\infty}$ are bounded, and so is the sequence $\{By^k\}_{k=0}^{\infty}$ due to the fact that $\lim_{k\to\infty}(Ax^k + By^k - b) = 0$. Therefore, $\{(S + \beta A^TA)x^k\}_{k=0}^{\infty}$, $\{(T + \beta B^TB)y^k\}_{k=0}^{\infty}$ and $\{p^k\}_{k=0}^{\infty}$ are all bounded.

Now, we assume that both $S + \beta A^TA$ and $T + \beta B^TB$ are positive definite. In this case, it is clear that $\{w^k\}_{k=0}^{\infty}$ is bounded and must have a limit point. Suppose that w^* is any limit point of $\{w^k\}_{k=0}^{\infty}$ and $w^{k_j} \to w^*$ as $j \to \infty$. Since $\mathcal{W}$ is closed, $w^* \in \mathcal{W}$. Furthermore, by taking the limit over $k = k_j \to \infty$ in (3.2) and noting that $G(w^{k_j} - \bar{w}^{k_j-1}) \to 0$, we obtain

$$\theta(w) - \theta(w^*) + \langle w - w^*, F(w^*)\rangle \ge 0.$$

Since w can vary arbitrarily in $\mathcal{W}$, we conclude that $w^* \in \mathcal{W}^*$. That is, any limit point of $\{w^k\}_{k=0}^{\infty}$ must also lie in $\mathcal{W}^*$. It remains to show the uniqueness of the limit points of $\{w^k\}_{k=0}^{\infty}$. Suppose that $w_\ell^* = (x_\ell^*, y_\ell^*, p_\ell^*)$,

$\ell = 1, 2$, are two limit points of $\{w^k\}_{k=0}^{\infty}$ and $\lim_{j\to\infty} w^{i_j} = w_1^*$, $\lim_{j\to\infty} w^{k_j} = w_2^*$. By Lemma 4.2, $\lim_{k\to\infty} \|w^k - w_\ell^*\|_G$ exists for $\ell = 1, 2$. Assume that $\lim_{k\to\infty} \|w^k - w_\ell^*\|_G = v_\ell$ for $\ell = 1, 2$. By taking the limit over $k = i_j \to \infty$ and $k = k_j \to \infty$ in the equality

$$\|w^k - w_1^*\|_G^2 - \|w^k - w_2^*\|_G^2 = \|w_1^* - w_2^*\|_G^2 + 2\langle w_1^* - w_2^*, w_2^* - w^k\rangle_G,$$

we obtain $v_1 - v_2 = -\|w_1^* - w_2^*\|_G^2 = \|w_1^* - w_2^*\|_G^2$. Thus, $\|w_1^* - w_2^*\|_G = 0$. Since G is positive semidefinite, this implies that $Gw_1^* = Gw_2^*$, or equivalently, $Sx_1^* = Sx_2^*$, $Ty_1^* = Ty_2^*$ and $By_1^* - p_1^*/\beta = By_2^* - p_2^*/\beta$. Since $Ax_1^* + By_1^* = Ax_2^* + By_2^* = b$, it follows that $Ax_1^* + p_1^*/\beta = Ax_2^* + p_2^*/\beta$. On the other hand, by the definition of w_1^* and w_2^*, it follows that

$$f(x_1^*) - f(x_2^*) + \left\langle x_1^* - x_2^*, -A^T p_2^* \right\rangle \geq 0 \quad \text{and} \quad f(x_2^*) - f(x_1^*) + \left\langle x_2^* - x_1^*, -A^T p_1^* \right\rangle \geq 0.$$

By adding the two inequalities above together, we get $\langle A(x_1^* - x_2^*), p_1^* - p_2^*\rangle \geq 0$, and thus

$$\|A(x_1^* - x_2^*)\|^2 + \|(p_1^* - p_2^*)/\beta\|^2 \leq \|(Ax_1^* + p_1^*/\beta) - (Ax_2^* + p_2^*/\beta)\|^2 = 0.$$

It therefore holds that $Ax_1^* = Ax_2^*$, $p_1^* = p_2^*$, and hence $By_1^* = By_2^*$. This together with $Sx_1^* = Sx_2^*$ and $Tx_1^* = Tx_2^*$ implies that $(S + \beta A^T A)(x_1^* - x_2^*) = 0$ and $(T + \beta B^T B)(y_1^* - y_2^*) = 0$. Since $S + \beta A^T A$ and $T + \beta B^T B$ are positive definite, we deduce that $x_1^* = x_2^*$ and $y_1^* = y_2^*$. Therefore, $\{w^k\}_{k=0}^{\infty}$ converges to some point in $\mathcal{W}^*$ as $k \to \infty$. □

We give the following remarks on the convergence results presented in Theorem 4.7.

REMARK 4. *The conditions $S + \beta A^T A \succ 0$ and $T + \beta B^T B \succ 0$ to ensure the iterate convergence of $\{w^k\}_{k=0}^{\infty}$ are in fact not sufficient to ensure the positive definiteness of G in (2.6). For monotone operator inclusion problem, iterate convergence cannot be guaranteed in general under the relaxed condition that G is only positive semidefinite, although part of the results existing in the literature can indeed be remained. The reason that we are able to establish iterate convergence under the relaxed condition that G is only positive semidefinite is because we are restricted to the convex optimization problem (1.1) which has useful structures.*

REMARK 5. *The conditions $S + \beta A^T A \succ 0$ and $T + \beta B^T B \succ 0$ to ensure the iterate convergence of $\{w^k\}_{k=0}^{\infty}$ are in fact very mild. This can be seen by assuming $S = 0$ and $T = 0$. In this case, these conditions are essentially requiring that both A and B have full column rank, which are commonly assumed to ensure solution uniqueness of (1.4a) and (1.4c) and to guarantee the iterate convergence, see, e.g., [9], where the focus is the special case (1.2). See also remarks in [4].*

REMARK 6. *As mentioned in Section 2, the inertial proximal ADMM reduces to the inertial linearized ADMM [2, Eq. (3.23)] if $S = \frac{\beta}{\tau} I - \beta A^T A$ and $T = \frac{\beta}{\eta} I - \beta B^T B$. By Theorem 4.7, we know that the conditions $0 < \tau \leq 1/\rho(A^T A)$ and $0 < \eta \leq 1/\rho(B^T B)$ suffice for the iterate convergence. This, somewhat, closes the gap between the convergence requirements $0 < \tau < 1/\rho(A^T A)$ and $0 < \eta < 1/\rho(B^T B)$ in [2] for inertial linearized ADMM and $0 < \tau \leq 1/\rho(A^T A)$ and $0 < \eta \leq /\rho(B^T B)$ in [47, 39] for linearized ADMM.*

REMARK 7. *The iterate convergence given in Theorem 4.7 is stronger than convergence in function value for the accelerated methods in [48, 37], which in fact can also be viewed as inertial type methods. Our stronger result is obtained at the cost of more restrictive conditions on $\{\alpha_k\}_{k=0}^{\infty}$.*

5. Numerical results. In this section, we present numerical results to compare the performance of the proximal ADMM (2.2) and its inertial variant (3.1). We carried out two sets of experiments. In the first set of experiments, we concentrate on a constrained total variation (TV) minimization problem for image

reconstruction from incomplete Walsh-Hadamard coefficients. The problem is in the form of (1.2) and the resulting ADMM subproblem (1.4c) cannot be easily solved. We thus compare the linearized ADMM (2.7) with its inertial variant, i.e., $S = 0$ and $T = \frac{\beta}{\eta} I - \beta B^T B$ in (2.2) and (3.1), respectively. Since this linearized ADMM is equivalent to the well-known primal-dual algorithm by Chambolle and Pock [3], we will refer to (2.7) and its inertial variant as CP and iCP, respectively. Note that iCP is exactly the inertial primal-dual algorithm [1, Algorithm 3]. The performance of CP relative to other state-of-the-art algorithms is well illustrated in the literature, see, e.g., [3, 6, 7, 49]. In the second set of experiments, we compare the original ADMM (1.4), i.e., $S = 0$ and $T = 0$ in (2.2), with its inertial variant (3.3) (abbreviated as iADMM) on an unconstrained TV regularization problem for image reconstruction from incomplete wavelet coefficients, for which both ADMM subproblems are easily solvable. Since the problem is unconstrained, for this set of experiments we also present results on the evolution of objective function values as the iteraton/CPU time proceeds and compare with CP and iCP. All algorithms were implemented in MATLAB, and the experiments were performed with Microsoft Windows 8 and MATLAB v7.13 (R2011b), running on a 64-bit Lenovo laptop with an Intel Core i7-3667U CPU at 2.00 GHz and 8 GB of memory.

5.1. Compressive image reconstruction based on TV minimization. In compressive image reconstruction, one tries to recover an image from a number of its linear measurements, similar as in compressive sensing. The reconstruction is realized via TV minimizations, which have been widely used since the pioneering work [50] and have shown to give favorable results with well-preserved edges. Another very important reason of the popularity of TV minimizations for image restoration is the availability of very fast numerical algorithms, see, e.g., [51, 52, 53]. Exact reconstruction of piecewise constant images from their incomplete frequencies via TV minimization was first obtained in [54]. Lately, it was shown in [55] that an image can be accurately recovered to within its best s-term approximation of its gradient from approximately $O(s \log(n^2))$ nonadaptive linear measurements, where the underlying image is of size n-by-n.

In the following, we let $B^{(1)}, B^{(2)} \in \Re^{n^2 \times n^2}$ be the first-order global forward finite difference matrices (with certain boundary conditions assumed) in the horizontal and the vertical directions, respectively. Let $B_i \in \Re^{2 \times n^2}$, $i = 1, 2, \ldots, n^2$, be the corresponding first-order local forward finite difference operator at the ith pixel, i.e., each B_i is a two-row matrix formed by stacking the ith rows of $B^{(1)}$ and $B^{(2)}$. Let $y^* \in \Re^{n^2}$ be an original n-by-n image, whose columns are stacked in an upper-left to lower-right order to form a vector of length n^2. Given a set of linear measurements $b = \mathcal{A} y^* \in \Re^q$, where $\mathcal{A} : \Re^{n^2} \to \Re^q$ is a linear operator, the theory developed in [55] guarantees that one can reconstruct y^* from $\mathcal{A}$ and b to within a certain high accuracy, as long as $\mathcal{A}$ satisfies certain technical conditions. Specifically, to reconstruct y^* from $\mathcal{A}$ and b, one seeks an image that fits the observation data and meanwhile has the minimum TV norm, i.e., a solution of the following TV minimization problem

$$\min_{y \in \Re^{n^2}} \sum_{i=1}^{n^2} \|B_i y\| + \iota_{\{y:\ \mathcal{A}y = b\}}(y). \tag{5.1}$$

Here $\iota_{\mathcal{Y}}(y)$ denotes the indicator function of a set $\mathcal{Y}$, i.e., $\iota_{\mathcal{Y}}(y)$ is equal to 0 if $y \in \mathcal{Y}$ and ∞ otherwise. For $x_j \in \Re^{n^2}$, $j = 1, 2$, we define

$$x := \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \Re^{2n^2}, \quad \mathbf{x}_i := \begin{pmatrix} (x_1)_i \\ (x_2)_i \end{pmatrix} \in \Re^2,\ i = 1, 2, \ldots, n^2, \quad B := \begin{pmatrix} B^{(1)} \\ B^{(2)} \end{pmatrix} \in \Re^{2n^2 \times n^2}. \tag{5.2}$$

Note that $x = (x_1, x_2)$ and $\{\mathbf{x}_i : i = 1, 2, \ldots, n^2\}$ denote the same set of variables.

Let $f : \Re^{n^2} \to (-\infty, \infty]$ and $g : \Re^{2n^2} \to (-\infty, \infty)$ be, respectively, defined as

$$f(x) := f(x_1, x_2) = \sum\nolimits_{i=1}^{n^2} \|\mathbf{x}_i\|,\ x = (x_1, x_2) \in \Re^{2n^2}, \tag{5.3a}$$

$$g(y) := \iota_{\{y:\ \mathcal{A}y=b\}}(y),\ y \in \Re^{n^2}. \tag{5.3b}$$

Then, (5.1) can be rewritten as $\min_{y\in\Re^{n^2}} f(By)+g(y)$, which is clearly in the form of (1.2) after introducing the constraints $-x + By = 0$. Let $\mathcal{A}^*$ be the adjoint operator of $\mathcal{A}$ and $\mathcal{I}$ be the identity operator. In our experiments, the linear operator $\mathcal{A}$ satisfies $\mathcal{A}\mathcal{A}^* = \mathcal{I}$. Therefore, the proximity operator of g is given by

$$\text{prox}^g(y) = y + \mathcal{A}^*(b - \mathcal{A}y),\ y \in \Re^{n^2}. \tag{5.4}$$

Note that the proximity operator of an indicator function reduces to the orthogonal projection onto the underlying set. The proximity parameter is omitted because it is irrelevant in this case. On the other hand, with the convention $0/0 = 0$, the proximity operator of "$\|\cdot\|$" is given by

$$\text{prox}_\eta^{\|\cdot\|}(\mathbf{x}_i) = \max\{\|\mathbf{x}_i\| - \eta, 0\} \times \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|},\ \mathbf{x}_i \in \Re^2,\ \eta > 0. \tag{5.5}$$

Furthermore, it is easy to observe from (5.3a) that f is separable with respect to $\mathbf{x}_i$ and thus the proximity operator of f can also be expressed explicitly. Therefore, the proximity operators of f and g defined in (5.3) are both easy to evaluate. As a result, CP and iCP are easy to implement.

5.2. Experimental data. In the first set of experiments, the linear operator $\mathcal{A}$ is set to be randomized partial Walsh-Hadamard transform matrix, whose rows are randomly chosen and columns are randomly permuted. Therefore, it holds that $\mathcal{A}\mathcal{A}^* = \mathcal{I}$. Specifically, the Walsh-Hadamard transform matrix of order 2^j is defined recursively as

$$H_{2^0} = [1], H_{2^1} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \ldots, H_{2^j} = \begin{bmatrix} H_{2^{j-1}} & H_{2^{j-1}} \\ H_{2^{j-1}} & -H_{2^{j-1}} \end{bmatrix}.$$

It can be shown that $H_{2^j} H_{2^j}^T = 2^j I$. In our experiments, the linear operator $\mathcal{A}$ contains randomly selected rows from $2^{j/2} H_{2^j}$, where $2^{j/2}$ is a normalization factor. It is worth pointing out that for some special linear operators (5.1) (and its denoising variants when the observation data contains noise) can be solved by the classical ADMM (1.4) without proximal-linearizing any of the subproblems, as long as the constraints are wisely treated and the finite difference operations are assumed to satisfy appropriate boundary conditions. In these cases, the y-subproblem can usually be solved by fast transforms, see, e.g., [56, 52, 57]. However, in our setting, the matrices $B^T B$ and $\mathcal{A}^*\mathcal{A}$ cannot be diagonalized simultaneously, no matter what boundary conditions are assumed for B. Therefore, when solving (5.1) by the classical ADMM, the y-subproblem is not easily solvable. In contrast, CP and iCP can be easily implemented to solve (5.1).

We tested 12 images, most of which are obtained from the USC-SIPI image database[1]. The image sizes are 256-by-256, 512-by-512 and 1024-by-1024, each of which contains 4 images. The tested images, together with their names in the database, are given in Figure 5.1.

5.3. Parameters, initialization, stopping rules, etc. The parameters common to CP and iCP are β and η, for which we used the same set of values. In our experiments, periodic boundary conditions are assumed for the finite difference operations. It is easy to show that $\rho(B^T B) = 8$. We set $\beta = 5$ and $\eta = 0.125 = 1/\rho(B^T B)$ uniformly for all tests, which may be suboptimal but perform favorably for imaging

[1] `http://sipi.usc.edu/database/`

FIG. 5.1. *Tested images from the USC-SIPI image database. The image sizes from the first to the third row are* 256×256, 512×512 *and* 1024×1024, *respectively.*

problems with appropriately scaled data. In particular, this setting satisfies the convergence requirement of both algorithms. The extrapolation parameter α_k for iCP was set to be 0.28 and held constant. This value of α_k is determined based on experiences. Note that constant strategy for α_k was also used in the recent work [1], where $\alpha_k \equiv \alpha \in \{0, 1/12, 1/6, 1/4, 1/3\}$ were tested for a matrix game problem. We will present some experimental results to compare the performance of iCP with different constant values of α_k. We also note that some experimental observations are given in [29], indicating that the feasible range of α_k may depend on the relative magnitudes of η and β. Nevertheless, how to select α_k adaptively to achieve faster convergence remains a research issue. Here our main goal is to illustrate the effect of the extrapolation steps. In our experiments, we initialized $y^0 = \mathcal{A}^* b$ and $p^0 = 0$ for both algorithms. From [43] and discussions in Section 2.2, the CP algorithm is an application of PPM to the mixed VI (2.4) with $w = (y, p)$. It is clear from (2.5) that a solution is already obtained if $w^{k+1} = (y^{k+1}, p^{k+1}) = (y^k, p^k) = w^k$. Moreover, it follows from (4.7) that the feasibility residue is always dominated by $\|w^{k+1} - \bar{w}^k\|_G/\beta$. As a result, it is justifiable to terminate CP by

$$\frac{\|(y^{k+1}, p^{k+1}) - (y^k, p^k)\|}{1 + \|(y^k, p^k)\|} < \varepsilon, \tag{5.6}$$

where $\varepsilon > 0$ is a tolerance parameter, and $\|(y,p)\| := \sqrt{\|y\|^2 + \|p\|^2}$. For iCP, the same can be said, except that (y^k, p^k) needs to be replaced by $(\bar{y}^k, \bar{p}^k)$. Thus, we terminated iCP by

$$\frac{\|(y^{k+1}, p^{k+1}) - (\bar{y}^k, \bar{p}^k)\|}{1 + \|(\bar{y}^k, \bar{p}^k)\|} < \varepsilon. \tag{5.7}$$

The quantities in (5.6) and (5.7) can be viewed as optimality residues in a relative sense. The tolerance parameter ε will be specified later. To evaluate the quality of recovered images, we used the signal-to-noise ratio (SNR), which is defined as

$$\text{SNR} := 20 \times \log_{10} \frac{\|\tilde{y} - y^*\|}{\|y - y^*\|}. \tag{5.8}$$

Here y^* and y represent the original and the recovered images, and $\tilde{y}$ denotes the mean intensity of y^*. Note that, for this set of experiment, the constraint $\mathcal{A}y = b$ is always satisfied at each iteration and for all algorithms. Therefore, we only report the objective function value $\sum_i \|B_i y\|$, denoted by TV(y), but not the data fidelity $\|\mathcal{A}y - b\|$.

5.4. Reconstruction results from incomplete Walsh-Hadamard coefficients. Recall that the image is of size n-by-n, and the number of measurements is denoted by q. For each image, we tested four levels of measurements, that is $q/n^2 \in \{20\%, 40\%, 60\%, 80\%\}$. In the experimental results, besides SNR and objective function value, we also present this feasibility residue, measured by the infinity norm $\|x - By\|_\infty$, and the number of iterations required by the algorithms (denoted, respectively, by It1 and It2 for CP and iCP) to meet the condition (5.6) or (5.7). We do not present the CPU time results for comparison because the per-iteration cost of the algorithms is roughly identical and the consumed CPU time is basically proportional to the respective number of iterations. Detailed experimental results for $\varepsilon = 10^{-2}, 10^{-3}$ and 10^{-4} are given in Tables 5.1–5.3, respectively. Note that in Tables 5.1–5.3 the results for TV(x) and $\|x - By\|_\infty$ are given in scientific notation, where the first number denotes the significant digit and the second denotes the power.

It can be seen from Tables 5.1–5.3 that, to obtain solutions satisfying the aforementioned conditions, iCP is generally faster than CP. Specifically, within our setting the numbers of iterations consumed by iCP range from 70%–80% of those consumed by CP (see the last columns in the tables). In most cases, iCP obtained results with slightly better final objective function values and feasibility residues. The quality of recovered images is also slightly better in terms of SNR. By comparing results between different tables, we see that solutions with high accuracy in optimization point of view generally imply better image quality measured by SNR. This could imply that solving the problem to a certain high accuracy is in some sense necessary for better recovery, though the improvement of image quality could be small when the solution is already very accurate. It can also be observed from the results that both algorithms converge very fast at the beginning stage and slow down afterwards. In particular, to improve the solution quality by one more digit of accuracy (measured by optimality residue defined in (5.6)-(5.7)), the number of iterations could be multiplied by a few times, which is probably a common feature of first-order optimization algorithms. The fact is that in most cases one does not need to solve imaging problems to extremely high accuracy, because the recovered results hardly have any difference detectable by human eyes when they are already accurate enough. For example, for 8-bit images, we only need accuracy up to 3 decimal points. In words, the inertial technique accelerates the original algorithm to some extent without increasing the total computational cost.

To better visualize the improvement of iCP over CP, we reorganized the results given in Tables 5.1-5.3 and presented them in Figure 5.2. For each measurement level q/n^2 and image size n, we accumulated the number of iterations for different images and took an average. The results for $\varepsilon = 10^{-2}, 10^{-3}$ and 10^{-4}

TABLE 5.1

Reconstruction results from incomplete Walsh-Hadamard coefficients with $\varepsilon = 10^{-2}$ ($\beta = 5, \eta = 0.125, \alpha_k \equiv \alpha = 0.28$).

			CP				iCP				
q/n^2	n	image	TV(y)	$\|x - By\|_\infty$	SNR	It1	TV(y)	$\|x - By\|_\infty$	SNR	It2	$\frac{\text{It2}}{\text{It1}}$
20%	256	4.1.03	1.1319 3	7.2942 -3	4.87	65	1.1241 3	4.4170 -3	4.98	50	0.77
		4.1.05	1.3151 3	5.8069 -3	3.87	49	1.3069 3	5.8904 -3	3.96	38	0.78
		lena	2.1665 3	6.7751 -3	2.60	49	2.1605 3	7.5283 -3	2.63	37	0.76
		camera	2.5009 3	8.2455 -3	3.76	55	2.4940 3	4.2202 -3	3.79	41	0.75
	512	4.2.06	6.8642 3	5.7185 -3	1.49	41	6.8356 3	5.4164 -3	1.52	32	0.78
		4.2.07	6.1689 3	6.1500 -3	1.84	46	6.1445 3	6.1123 -3	1.89	36	0.78
		elaine	5.7648 3	7.0215 -3	2.27	47	5.7475 3	7.8325 -3	2.33	37	0.79
		brain	3.7925 3	3.3133 -3	5.36	66	3.7599 3	3.3699 -3	5.43	51	0.77
	1024	5.3.01	2.6227 4	7.1226 -3	2.29	51	2.6126 4	5.3319 -3	2.31	39	0.76
		5.3.02	3.2031 4	6.9707 -3	2.07	44	3.1954 4	5.7347 -3	2.12	35	0.80
		3.2.25	2.9271 4	5.7448 -3	2.60	40	2.9168 4	6.6754 -3	2.68	32	0.80
		7.2.01	1.5247 4	5.1647 -3	1.79	43	1.5138 4	5.3732 -3	1.84	35	0.81
40%	256	4.1.03	1.3488 3	4.2212 -3	7.86	69	1.3412 3	3.9813 -3	7.99	52	0.75
		4.1.05	1.7393 3	6.6140 -3	6.40	47	1.7350 3	9.0815 -3	6.52	37	0.79
		lena	2.8201 3	9.1814 -3	4.18	52	2.8181 3	4.7483 -3	4.22	39	0.75
		camera	3.1159 3	1.0701 -2	5.66	59	3.1119 3	3.6160 -3	5.69	44	0.75
	512	4.2.06	9.6638 3	6.2966 -3	2.82	37	9.6525 3	5.6120 -3	2.87	29	0.78
		4.2.07	8.6189 3	8.5462 -3	3.54	42	8.6143 3	7.2810 -3	3.61	33	0.79
		elaine	8.0715 3	9.2392 -3	3.15	43	8.0612 3	8.1932 -3	3.19	33	0.77
		brain	4.6370 3	9.4767 -3	4.39	65	4.6091 3	4.3974 -3	4.43	50	0.77
	1024	5.3.01	3.5028 4	8.1347 -3	3.75	48	3.5008 4	1.0318 -2	3.80	37	0.77
		5.3.02	4.4279 4	6.2587 -3	4.22	40	4.4227 4	7.6498 -3	4.27	31	0.78
		3.2.25	4.0071 4	5.7304 -3	5.39	38	3.9977 4	6.8326 -3	5.46	29	0.76
		7.2.01	2.1930 4	5.9025 -3	3.18	37	2.1870 4	5.7002 -3	3.25	30	0.81
60%	256	4.1.03	1.4182 3	5.5252 -3	13.44	77	1.4100 3	7.2768 -3	13.49	58	0.75
		4.1.05	2.0491 3	8.1584 -3	8.57	41	2.0462 3	9.5036 -3	8.72	32	0.78
		lena	3.2443 3	9.8217 -3	8.62	50	3.2451 3	1.0342 -2	8.75	38	0.76
		camera	3.4820 3	1.2103 -2	7.12	55	3.4808 3	4.9759 -3	7.20	42	0.76
	512	4.2.06	1.1612 4	8.1289 -3	4.21	34	1.1603 4	8.5716 -3	4.25	26	0.76
		4.2.07	1.0185 4	9.6775 -3	5.66	37	1.0188 4	1.1501 -2	5.75	29	0.78
		elaine	9.8019 3	1.0034 -2	7.56	38	9.7987 3	1.0745 -2	7.64	29	0.76
		brain	5.0578 3	1.0896 -2	4.15	59	5.0421 3	6.4967 -3	4.22	47	0.80
	1024	5.3.01	4.0490 4	1.1646 -2	6.00	43	4.0489 4	1.0927 -2	6.06	33	0.77
		5.3.02	5.3012 4	7.7387 -3	8.60	35	5.2984 4	7.6171 -3	8.71	27	0.77
		3.2.25	4.7548 4	6.9208 -3	8.00	33	4.7513 4	8.2037 -3	8.15	26	0.79
		7.2.01	2.7099 4	7.3859 -3	5.29	32	2.7047 4	7.9069 -3	5.33	25	0.78
80%	256	4.1.03	1.5117 3	1.9780 -2	15.84	47	1.5104 3	1.0129 -2	16.61	39	0.83
		4.1.05	2.2963 3	9.2757 -3	11.84	34	2.2947 3	9.0494 -3	12.06	27	0.79
		lena	3.4791 3	1.4328 -2	12.92	45	3.4796 3	1.4387 -2	13.11	34	0.76
		camera	3.6207 3	1.4613 -2	9.57	48	3.6203 3	1.5002 -2	9.67	38	0.79
	512	4.2.06	1.3283 4	9.8529 -3	9.40	27	1.3282 4	1.0326 -2	9.49	21	0.78
		4.2.07	1.1962 4	9.9285 -3	6.04	29	1.1967 4	1.0784 -2	6.11	23	0.79
		elaine	1.1196 4	1.3529 -2	9.11	31	1.1198 4	1.2973 -2	9.20	24	0.77
		brain	5.2027 3	1.3958 -2	14.92	54	5.1933 3	1.2994 -2	15.45	44	0.81
	1024	5.3.01	4.4290 4	1.7137 -2	9.40	36	4.4303 4	1.4849 -2	9.48	28	0.78
		5.3.02	5.9732 4	9.4555 -3	10.75	28	5.9721 4	9.5126 -3	10.86	22	0.79
		3.2.25	5.3345 4	7.3499 -3	12.81	27	5.3338 4	9.4321 -3	13.07	22	0.81
		7.2.01	3.1427 4	1.1171 -2	7.20	26	3.1409 4	8.6650 -3	7.27	21	0.81

TABLE 5.2

Reconstruction results from incomplete Walsh-Hadamard coefficients with $\varepsilon = 10^{-3}$ ($\beta = 5, \eta = 0.125, \alpha_k \equiv \alpha = 0.28$).

			CP				iCP				
q/n^2	n	image	TV(y)	$\|x - By\|_\infty$	SNR	It1	TV(y)	$\|x - By\|_\infty$	SNR	It2	$\frac{\text{It2}}{\text{It1}}$
20%	256	4.1.03	1.0592 3	9.0554 -4	9.33	329	1.0586 3	7.0139 -4	9.44	243	0.74
		4.1.05	1.2514 3	9.8182 -4	6.05	242	1.2508 3	7.4895 -4	6.10	179	0.74
		lena	2.0240 3	1.4969 -3	4.43	318	2.0236 3	1.0133 -3	4.44	231	0.73
		camera	2.3622 3	1.2124 -3	5.92	330	2.3618 3	8.5057 -4	5.94	241	0.73
	512	4.2.06	6.6077 3	1.0215 -3	2.40	217	6.6059 3	8.5628 -4	2.42	160	0.74
		4.2.07	5.8261 3	1.7749 -3	3.70	271	5.8246 3	1.7558 -3	3.74	199	0.73
		elaine	5.4348 3	1.2689 -3	3.97	267	5.4331 3	1.0330 -3	4.00	196	0.73
		brain	3.5531 3	1.4475 -3	8.42	383	3.5509 3	1.0458 -3	8.45	278	0.73
	1024	5.3.01	2.4855 4	1.5801 -3	3.69	290	2.4848 4	1.5348 -3	3.71	211	0.73
		5.3.02	3.1250 4	1.7648 -3	2.86	195	3.1244 4	1.4521 -3	2.88	145	0.74
		3.2.25	2.8511 4	1.5011 -3	3.72	179	2.8501 4	1.2207 -3	3.75	133	0.74
		7.2.01	1.4674 4	1.1333 -3	2.83	212	1.4667 4	9.6846 -4	2.87	159	0.75
40%	256	4.1.03	1.2637 3	8.7982 -4	17.13	358	1.2631 3	6.4152 -4	17.18	265	0.74
		4.1.05	1.6953 3	1.8062 -3	8.88	197	1.6950 3	1.2055 -3	8.97	147	0.75
		lena	2.6837 3	1.3561 -3	6.19	295	2.6835 3	1.0678 -3	6.21	215	0.73
		camera	3.0109 3	1.3347 -3	9.22	289	3.0109 3	1.0672 -3	9.28	211	0.73
	512	4.2.06	9.4002 3	1.6715 -3	4.16	192	9.3995 3	1.3027 -3	4.18	141	0.73
		4.2.07	8.2590 3	1.7877 -3	6.11	242	8.2584 3	1.4797 -3	6.14	177	0.73
		elaine	7.7316 3	2.0480 -3	4.97	232	7.7309 3	1.6872 -3	5.00	170	0.73
		brain	4.4112 3	2.0597 -3	7.85	332	4.4100 3	1.5813 -3	7.89	242	0.73
	1024	5.3.01	3.3361 4	1.9296 -3	6.11	283	3.3359 4	1.8139 -3	6.13	206	0.73
		5.3.02	4.3547 4	1.6664 -3	5.44	171	4.3544 4	1.2812 -3	5.47	127	0.74
		3.2.25	3.9377 4	1.4296 -3	7.25	159	3.9372 4	1.2021 -3	7.29	118	0.74
		7.2.01	2.1404 4	2.1191 -3	4.81	186	2.1401 4	1.6644 -3	4.86	138	0.74
60%	256	4.1.03	1.3827 3	1.5129 -3	21.78	260	1.3826 3	9.4321 -4	22.12	193	0.74
		4.1.05	2.0087 3	1.9442 -3	12.48	182	2.0086 3	1.4960 -3	12.58	135	0.74
		lena	3.1255 3	1.6699 -3	16.48	283	3.1256 3	1.7352 -3	16.62	207	0.73
		camera	3.3504 3	2.4634 -3	14.46	326	3.3498 3	1.7600 -3	14.44	234	0.72
	512	4.2.06	1.1364 4	2.4834 -3	5.48	180	1.1364 4	1.9204 -3	5.49	132	0.73
		4.2.07	9.8460 3	2.8274 -3	9.07	200	9.8463 3	2.1139 -3	9.13	147	0.73
		elaine	9.4950 3	3.1823 -3	10.79	199	9.4948 3	2.4538 -3	10.83	146	0.73
		brain	4.8640 3	3.0694 -3	7.05	290	4.8640 3	3.1651 -3	7.11	213	0.73
	1024	5.3.01	3.8842 4	4.1921 -3	9.13	273	3.8843 4	2.3870 -3	9.16	199	0.73
		5.3.02	5.2337 4	2.2509 -3	11.43	158	5.2337 4	1.7885 -3	11.50	117	0.74
		3.2.25	4.7037 4	1.9918 -3	10.37	137	4.7035 4	2.0416 -3	10.43	102	0.74
		7.2.01	2.6668 4	2.7801 -3	7.06	159	2.6668 4	2.1974 -3	7.10	118	0.74
80%	256	4.1.03	1.4787 3	1.2684 -3	29.97	217	1.4785 3	1.6997 -3	29.34	162	0.75
		4.1.05	2.2664 3	2.9759 -3	19.08	171	2.2664 3	2.3118 -3	19.17	123	0.72
		lena	3.3928 3	2.5510 -3	22.10	229	3.3928 3	2.0750 -3	22.12	167	0.73
		camera	3.5637 3	6.0327 -3	13.63	211	3.5642 3	3.4959 -3	13.77	156	0.74
	512	4.2.06	1.3092 4	3.4162 -3	12.37	143	1.3093 4	3.3944 -3	12.42	105	0.73
		4.2.07	1.1693 4	5.4809 -3	9.34	163	1.1693 4	4.2707 -3	9.38	119	0.73
		elaine	1.0980 4	4.2817 -3	12.57	159	1.0980 4	3.3699 -3	12.61	116	0.73
		brain	5.0233 3	1.5505 -3	34.64	275	5.0219 3	1.5675 -3	34.09	206	0.75
	1024	5.3.01	4.3206 4	5.4126 -3	13.58	205	4.3207 4	4.3222 -3	13.61	149	0.73
		5.3.02	5.9350 4	4.1668 -3	13.04	120	5.9350 4	3.1152 -3	13.07	88	0.73
		3.2.25	5.3086 4	2.9624 -3	15.55	103	5.3086 4	2.0553 -3	15.64	77	0.75
		7.2.01	3.1127 4	5.8998 -3	8.74	128	3.1129 4	4.5796 -3	8.77	95	0.74

TABLE 5.3

Reconstruction results from incomplete Walsh-Hadamard coefficients with $\varepsilon = 10^{-4}$ ($\beta = 5, \eta = 0.125, \alpha_k \equiv \alpha = 0.28$).

			CP				iCP				
q/n^2	n	image	TV(y)	$\|x - By\|_\infty$	SNR	It1	TV(y)	$\|x - By\|_\infty$	SNR	It2	$\frac{\text{It2}}{\text{It1}}$
20%	256	4.1.03	1.0529 3	1.4547 -4	11.76	1241	1.0528 3	1.0827 -4	11.79	908	0.73
		4.1.05	1.2450 3	1.7284 -4	6.75	953	1.2449 3	1.1230 -4	6.75	694	0.73
		lena	2.0075 3	1.8010 -4	5.30	1272	2.0074 3	1.3062 -4	5.31	920	0.72
		camera	2.3487 3	2.2856 -4	7.17	1277	2.3486 3	1.6444 -4	7.18	924	0.72
	512	4.2.06	6.5656 3	2.1007 -4	3.57	1016	6.5654 3	1.5352 -4	3.58	737	0.73
		4.2.07	5.7503 3	2.2159 -4	7.05	1595	5.7500 3	1.9751 -4	7.06	1146	0.72
		elaine	5.3811 3	1.8871 -4	5.78	1250	5.3809 3	1.5154 -4	5.79	905	0.72
		brain	3.5307 3	1.8038 -4	10.45	1379	3.5305 3	1.4380 -4	10.48	1007	0.73
	1024	5.3.01	2.4471 4	2.6258 -4	6.80	1873	2.4470 4	1.9095 -4	6.81	1350	0.72
		5.3.02	3.1125 4	1.9095 -4	3.74	926	3.1124 4	1.4926 -4	3.75	672	0.73
		3.2.25	2.8419 4	2.1814 -4	4.33	726	2.8418 4	1.6618 -4	4.34	531	0.73
		7.2.01	1.4567 4	2.6271 -4	5.21	1209	1.4567 4	1.8932 -4	5.24	881	0.73
40%	256	4.1.03	1.2613 3	1.4225 -4	18.57	866	1.2613 3	9.7930 -5	18.63	659	0.76
		4.1.05	1.6906 3	2.1066 -4	10.92	689	1.6905 3	1.7779 -4	10.96	505	0.73
		lena	2.6725 3	1.9869 -4	7.18	956	2.6725 3	1.6135 -4	7.19	692	0.72
		camera	2.9844 3	1.8506 -4	15.42	1382	2.9843 3	1.3780 -4	15.44	1001	0.72
	512	4.2.06	9.3340 3	2.5729 -4	6.54	1022	9.3339 3	1.9041 -4	6.56	741	0.73
		4.2.07	8.1870 3	2.0627 -4	10.52	1179	8.1870 3	1.5139 -4	10.56	855	0.73
		elaine	7.6796 3	2.9240 -4	7.09	1015	7.6795 3	2.2082 -4	7.10	736	0.73
		brain	4.3411 3	1.8608 -4	18.22	1774	4.3409 3	1.3406 -4	18.31	1293	0.73
	1024	5.3.01	3.2940 4	3.0227 -4	10.09	1463	3.2940 4	2.2911 -4	10.11	1059	0.72
		5.3.02	4.3426 4	3.0691 -4	6.76	735	4.3425 4	2.3810 -4	6.78	535	0.73
		3.2.25	3.9299 4	2.8261 -4	8.27	587	3.9298 4	2.2508 -4	8.29	431	0.73
		7.2.01	2.1271 4	2.8676 -4	7.44	1043	2.1271 4	2.3184 -4	7.46	761	0.73
60%	256	4.1.03	1.3813 3	2.1023 -4	23.71	651	1.3812 3	1.4074 -4	23.75	494	0.76
		4.1.05	2.0061 3	2.7249 -4	14.10	508	2.0061 3	1.9179 -4	14.15	377	0.74
		lena	3.1180 3	3.0073 -4	17.80	828	3.1180 3	2.3795 -4	17.78	601	0.73
		camera	3.3238 3	2.9028 -4	20.34	1126	3.3237 3	2.3618 -4	20.29	820	0.73
	512	4.2.06	1.1338 4	3.0901 -4	6.75	582	1.1338 4	2.4203 -4	6.77	424	0.73
		4.2.07	9.7655 3	3.8127 -4	16.08	998	9.7655 3	3.0599 -4	16.12	725	0.73
		elaine	9.4659 3	4.3193 -4	13.04	679	9.4659 3	3.0643 -4	13.06	496	0.73
		brain	4.7679 3	2.9519 -4	23.57	1747	4.7678 3	2.4006 -4	23.81	1278	0.73
	1024	5.3.01	3.8615 4	8.4962 -4	12.40	1018	3.8615 4	6.0623 -4	12.43	739	0.73
		5.3.02	5.2243 4	6.5600 -4	12.61	608	5.2243 4	4.5762 -4	12.60	444	0.73
		3.2.25	4.6980 4	4.5021 -4	11.83	484	4.6980 4	3.5987 -4	11.85	356	0.74
		7.2.01	2.6581 4	4.1830 -4	8.96	731	2.6581 4	3.0788 -4	8.98	536	0.73
80%	256	4.1.03	1.4781 3	2.4197 -4	30.76	419	1.4781 3	1.7573 -4	30.80	327	0.78
		4.1.05	2.2649 3	3.8634 -4	20.65	425	2.2649 3	3.6174 -4	20.66	314	0.74
		lena	3.3916 3	3.4517 -4	22.44	447	3.3916 3	4.2157 -4	22.44	331	0.74
		camera	3.5313 3	3.3864 -4	27.10	970	3.5312 3	2.7673 -4	26.92	707	0.73
	512	4.2.06	1.3069 4	5.8567 -4	14.79	519	1.3069 4	4.1928 -4	14.82	380	0.73
		4.2.07	1.1585 4	1.1237 -3	20.78	918	1.1586 4	8.7215 -4	20.86	666	0.73
		elaine	1.0928 4	1.0021 -3	19.55	826	1.0928 4	7.3748 -4	19.57	600	0.73
		brain	5.0193 3	1.6601 -4	35.21	689	5.0192 3	1.4387 -4	35.21	549	0.80
	1024	5.3.01	4.2953 4	9.6281 -4	22.98	986	4.2953 4	7.7170 -4	23.03	717	0.73
		5.3.02	5.9293 4	7.7142 -4	15.46	452	5.9293 4	5.5496 -4	15.50	331	0.73
		3.2.25	5.3062 4	8.1420 -4	16.79	336	5.3062 4	5.9169 -4	16.81	249	0.74
		7.2.01	3.1059 4	6.7967 -4	9.43	558	3.1059 4	5.4514 -4	9.43	409	0.73

are given in Figure 5.2. By comparing the three plots in Figure 5.2, we see that the number of iterations increased from a few dozens to around one thousand when the accuracy tolerance ε was decreased from 10^{-2} to 10^{-4}. From the results we can also observe that, on average, both algorithms are stable in the sense that the consumed number of iterations do not vary much for different image sizes.

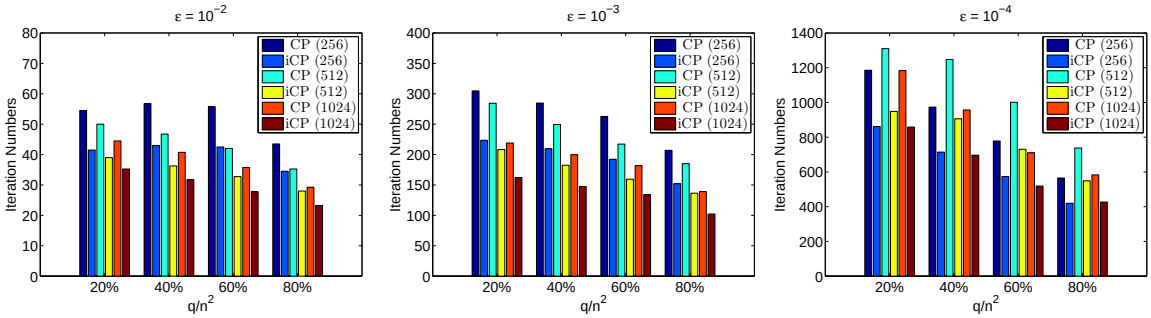

FIG. 5.2. *Comparison results of CP and iCP on different image sizes and stopping tolerance ($n = 256, 512, 1024$, and from left to right $\varepsilon = 10^{-2}, 10^{-3}, 10^{-4}$, respectively).*

We also examined the performance of iCP with different constant strategies of the inertial extrapolation stepsize α_k. In particular, for $n = 1024$ we tested $\alpha_k \equiv \alpha \in \{0.05, 0.15, 0.25, 0.35\}$. The results are given in Figure 5.3. It can be seen from the results that, for the four tested α values, larger ones generally give better performance. Recall that, according to our analysis, iCP is guaranteed to converge under the condition $0 \leq \alpha_k \leq \alpha_{k+1} \leq \alpha < 1/3$ for all k. Indeed, we have observed that iCP either slows down or becomes unstable for large values of α, say, larger than 0.3, especially when the number of measurements is relatively small. This is the main reason that we set α_k a constant value that is near 0.3 but not larger. Similar discussions for compressive principal component pursuit problems can be found in [2].

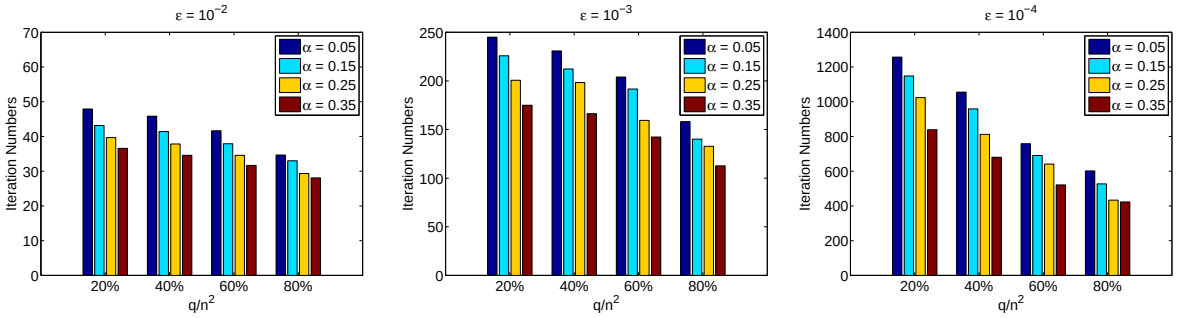

FIG. 5.3. *Comparison results of iCP on different $\alpha_k \equiv \alpha$ and stopping tolerance ($\alpha \in \{0.05, 0.15, 0.25, 0.35\}$, and from left to right $\varepsilon = 10^{-2}, 10^{-3}, 10^{-4}$, respectively).*

5.5. Image reconstruction from incomplete wavelet coefficients. Image reconstruction from incomplete wavelet coefficients is also known as wavelet domain inpainting. Given a set of randomly selected wavelet coefficients $f = PWy^* + \omega \in \Re^q$, where $y^* \in \Re^{n^2}$ denotes the original image, W is an orthonormal wavelet transform, P represents a selection operator which contains q randomly selected rows of the identity matrix of size n^2, and ω contains additive Gaussian noise. In this section, we keep all the notation defined in Section 5.1. In particular, the finite difference operators and the notation defined in (5.2) remain effective.

The TV model to reconstruct y^* from f is

$$\min_y \sum\nolimits_{i=1}^{n^2} \|B_i y\| + \frac{\mu}{2}\|PWy - f\|^2, \tag{5.9}$$

where $\mu > 0$ is a weighting parameter dependent on the noise level. By introducing auxiliary variables, (5.9) can be equivalently transformed to

$$\min_{x,y,z} \left\{ \sum\nolimits_{i=1}^{n^2} \|\mathbf{x}_i\| + \frac{\mu}{2}\|Pz - f\|^2 : \text{s.t. } -\begin{pmatrix} x \\ z \end{pmatrix} + \begin{pmatrix} B \\ W \end{pmatrix} y = 0 \right\}, \tag{5.10}$$

which is clearly in the form of (1.2). When solving (5.10) by the classical ADMM (1.4), both subproblems can be solved exactly via either shrinkage operators or fast Fourier and/or wavelet transforms due to the special structures of the underlying functions and linear operators, as long as periodic boundary conditions are assumed for B, see, e.g., [58]. Denote the dual variable of (5.10) by p. According to (2.5) and (2.6) (note that both S and T are zero matrices in this case), an optimal solution is already obtained if $(y^{k+1}, p^{k+1}) = (y^k, p^k)$ (resp. $(y^{k+1}, p^{k+1}) = (\bar{y}^k, \bar{p}^k)$) for ADMM (resp. iADMM). Therefore, similar as in the first set of experiments, we terminated ADMM and iADMM by (5.6) and (5.7), respectively, where the tolerance parameter ε was set to be 10^{-3}. The parameters β and α_k remain the same values as used in Section 5.3, i.e., $\beta = 5$ and $\alpha_k \equiv \alpha = 0.28$. The variables y and p are initialized at $\mathcal{A}^* b$ and 0, respectively, for both algorithms. In our experiments, we used the Haar wavelet transform provided by the Rice Wavelet Toolbox [59] with its default settings. The noise ω was random Gaussian with mean zero and standard deviation 10^{-3}. The weighting parameter μ was set to be 10^3. Recall that the number of measurements is q and the sample ratio is q/n^2. Detailed experimental results for $q/n^2 \in \{20\%, 40\%, 60\%, 80\%\}$ are reported in Table 5.4 for the set of images in Figure 5.1, where the final values of objective function (Obj), SNR and the number of iterations (denoted by It1 and It2 for ADMM and iADMM, respectively) are given.

Roughly speaking, similar conclusion as given in Section 5.4 can be drawn from the results in Table 5.4, i.e., to obtain solutions of approximately the same accuracy, iADMM is generally faster than ADMM. Specifically, to obtain solutions with approximately the same SNR values, the number of iterations consumed by iADMM is on average about 80% of that consumed by ADMM. The final objective function values obtained by iADMM are also slightly better in most cases. For different constant values of α, the results are similar to those presented in Figure 5.3. Thus, the detailed results are omitted here.

Note that (5.9) is an unconstrained optimization, and it is thus appropriate to compare different optimization algorithms by examining the evolution behavior of objective function values and SNR values as the iteration/CPU time proceeds. In this experiment, besides the results of ADMM and iADMM, we also present those of CP and iCP. First, by introducing auxiliary variable x only, we can transform (5.9) to

$$\min_{x,y} \left\{ \sum\nolimits_{i=1}^{n^2} \|\mathbf{x}_i\| + \frac{\mu}{2}\|PWy - f\|^2 : \text{s.t. } -x + By = 0 \right\}. \tag{5.11}$$

Let $f(x) = \sum_{i=1}^{n^2} \|\mathbf{x}_i\|$ and $g(y) = \frac{\mu}{2}\|PWy - f\|^2$. Then, the CP algorithm (2.7) can be applied. By the orthonormality of W, it is easy to show that, for any $\gamma > 0$, the proximity operator of g is given by

$$\text{prox}_\gamma^g(y) = W^T(\gamma\mu P^T P + I)^{-1}(\gamma\mu P^T f + Wy), \quad \forall y \in \Re^{n^2}.$$

Since P contains certain rows of the identity matrix, $P^T P$ is a diagonal matrix and the cost to evaluate $(\gamma\mu P^T P + I)^{-1}$ is negligible. As a result, the main cost for computing $\text{prox}_\gamma^g(y)$ is two fast Wavelet transforms. In this experiment, the parameter η in (2.7) was set to be 0.124, which guarantees convergence since

TABLE 5.4

Reconstruction results from incomplete wavelet coefficients ($\varepsilon = 10^{-3}$, $\beta = 5$, $\alpha_k \equiv \alpha = 0.28$).

			ADMM			iADMM			
q/n^2	n	image	Obj	SNR	It1	Obj	SNR	It2	$\frac{\text{It2}}{\text{It1}}$
20%	256	4.1.03	6.7480 2	4.22	214	6.7033 2	4.23	176	0.82
		4.1.05	1.0252 3	5.05	160	1.0245 3	5.05	118	0.74
		lena	1.6258 3	5.56	148	1.6254 3	5.56	107	0.72
		camera	1.6459 3	7.17	176	1.6448 3	7.17	130	0.74
	512	4.2.06	5.5076 3	6.65	151	5.5061 3	6.64	110	0.73
		4.2.07	4.8557 3	7.11	147	4.8515 3	7.09	107	0.73
		elaine	4.7290 3	6.89	149	4.7263 3	6.89	108	0.72
		brain	2.1597 3	7.64	148	2.1618 3	7.64	112	0.76
	1024	5.3.01	1.9263 4	8.46	139	1.9270 4	8.46	101	0.73
		5.3.02	2.5287 4	4.05	129	2.5281 4	4.05	93	0.72
		3.2.25	2.2791 4	2.85	140	2.2781 4	2.85	101	0.72
		7.2.01	1.1828 4	8.17	146	1.1829 4	8.17	110	0.75
40%	256	4.1.03	9.8197 2	7.37	146	9.7438 2	7.35	113	0.77
		4.1.05	1.5226 3	7.98	103	1.5210 3	8.03	79	0.77
		lena	2.3489 3	9.58	89	2.3481 3	9.58	66	0.74
		camera	2.4159 3	10.57	115	2.4156 3	10.58	91	0.79
	512	4.2.06	8.5346 3	9.37	92	8.5299 3	9.37	69	0.75
		4.2.07	7.3840 3	10.66	91	7.3807 3	10.66	69	0.76
		elaine	7.1243 3	10.43	99	7.1194 3	10.43	73	0.74
		brain	3.1941 3	10.17	108	3.1916 3	10.16	83	0.77
	1024	5.3.01	2.8631 4	11.75	91	2.8627 4	11.76	68	0.75
		5.3.02	3.8727 4	6.46	82	3.8725 4	6.47	61	0.74
		3.2.25	3.4924 4	5.73	87	3.4913 4	5.74	64	0.74
		7.2.01	1.8953 4	10.95	100	1.8953 4	10.94	77	0.77
60%	256	4.1.03	1.2102 3	12.46	110	1.1945 3	12.32	86	0.78
		4.1.05	1.8575 3	11.07	79	1.8575 3	11.07	63	0.80
		lena	2.8393 3	12.42	71	2.8381 3	12.45	55	0.77
		camera	2.9276 3	13.71	82	2.9273 3	13.70	69	0.84
	512	4.2.06	1.0813 4	11.69	69	1.0807 4	11.71	55	0.80
		4.2.07	9.3606 3	13.53	71	9.3508 3	13.56	54	0.76
		elaine	8.9962 3	13.91	72	8.9959 3	13.94	56	0.78
		brain	3.9928 3	13.88	81	3.9899 3	13.87	71	0.88
	1024	5.3.01	3.5350 4	14.75	67	3.5342 4	14.76	52	0.78
		5.3.02	4.8778 4	9.36	63	4.8769 4	9.36	48	0.76
		3.2.25	4.3767 4	8.72	65	4.3744 4	8.72	49	0.75
		7.2.01	2.4526 4	13.40	79	2.4516 4	13.39	64	0.81
80%	256	4.1.03	1.3674 3	18.20	81	1.3589 3	18.03	68	0.84
		4.1.05	2.1786 3	15.29	54	2.1752 3	15.25	43	0.80
		lena	3.2458 3	18.06	52	3.2414 3	18.14	44	0.85
		camera	3.2936 3	18.57	68	3.2876 3	18.52	54	0.79
	512	4.2.06	1.2629 4	15.69	56	1.2615 4	15.71	47	0.84
		4.2.07	1.1066 4	17.93	53	1.1052 4	18.03	44	0.83
		elaine	1.0568 4	18.02	52	1.0567 4	18.07	43	0.83
		brain	4.5818 3	18.57	64	4.5763 3	18.55	50	0.78
	1024	5.3.01	4.0889 4	18.75	52	4.0867 4	18.75	42	0.81
		5.3.02	5.6924 4	12.91	51	5.6913 4	12.91	41	0.80
		3.2.25	5.0931 4	12.43	51	5.0929 4	12.49	41	0.80
		7.2.01	2.9298 4	16.82	62	2.9260 4	16.82	53	0.85

$\rho(B^TB) = 8$. All other parameters remain the same as prescribed in Section 5.3. To better understanding the average performance of different algorithms, we tested all the 12 images given in Figure 5.1, ran each algorithm for 500 iterations, and took an average on the results. The performance of ADMM, iADMM, CP and iCP are given in Figure 5.4, where the evolution results of objective function values with respect to iteration and CPU time are presented. Besides, the evolution results of SNR values and the relative difference $\|w^{k+1} - \bar{w}^k\|/(1 + \|\bar{w}^k\|)$ (ADMM and CP correspond to $\alpha_k \equiv 0$ and thus $\bar{w}^k = w^k$) with respect to iteration are also given.

It is easy to observe from these results that CP and iCP are faster than ADMM and iADMM only at the beginning by very few (roughly, less than 20) iterations and fall behind very quickly. After about 100 iterations, CP and iCP catch up with ADMM and iADMM gradually. A plausible explanation for the faster speed to ADMM and iADMM compared to CP and iCP is that, due to the data structure, ADMM/iADMM can solve each subproblem exactly, while CP/iCP approximates one subproblem via proximal-linearization. Another observation is that inertial algorithms are generally faster than their corresponding original algorithms in both decreasing the objective function values and increasing the SNR values, which can be seen from the first and the fourth plots, respectively. The faster speed of inertial algorithms in decreasing function values is presented in an alternative way in the second plot, where the ratios of function values attained by inertial algorithms divided by those attained by their corresponding original algorithms are plotted as the iteration proceeds. It can be seen that the ratios are mostly less than 1, especially in the first few dozens of iterations. This observation may suggest that inertial algorithms are more advantageous to attain low to medium accuracy solutions. The results of function values versus CPU time in the third plot appear roughly the same with those for function values versus iteration. This is predictable since the extra computations in inertial algorithms are not significant. It is also apparent from the first four plots that all the compared algorithms obtained solutions of approximately the same accuracy measured by objective function values and SNR. The last plot in Figure 5.4 demonstrates how the relative difference $\|w^{k+1} - \bar{w}^k\|/(1 + \|\bar{w}^k\|)$ decreases with respect to iteration. It can be seen that $\|w^{k+1} - \bar{w}^k\|/(1 + \|\bar{w}^k\|)$ decreases smoothly for all the tested algorithms and decreases faster for inertial algorithms than the corresponding original ones. This also justifies the suitability of the stopping criteria (5.6) and (5.7).

6. Concluding remarks. In this paper, by combining the inertial techniques and the proximal ADMM, we proposed and analyzed a class of inertial proximal ADMMs, which unify and extend two existing algorithms [1, Algorithm 3] and [2, Eq. (3.23)]. This class of methods are of inertial nature because at each iteration the proximal ADMM is applied to a point extrapolated at the current iterate in the direction of last movement. Under very mild assumptions, we established the global iterate convergence for the entire class of algorithms. Compared to existing methods of the same kind, we only require the weighting matrices to be positive semidefinite, but not positive definite. In particular, by setting both weighting matrices to be zero, we obtained an inertial ADMM. In comparison to the recently proposed inertial ADMM in [34], our proposed algorithm framework is not only more simple and intuitive but also more general. Moreover, the conditions imposed by us to guarantee global convergence are simpler than those assumed in [34]. Based on the pioneering analysis in [18] and by using the structures of (1.1) and the iterative scheme (3.1), we established certain asymptotic $o(1/\sqrt{k})$ and nonasymptotic $O(1/\sqrt{k})$ convergence rate results on the best primal objective and feasibility residues. Our preliminary implementation of the algorithms and extensive experimental results on TV based image reconstruction problems have shown that inertial algorithms are generally faster than the corresponding original ones. Note that, compared to the original algorithms, the corresponding inertial ones do not require much extra computational cost except the linear cost to obtain

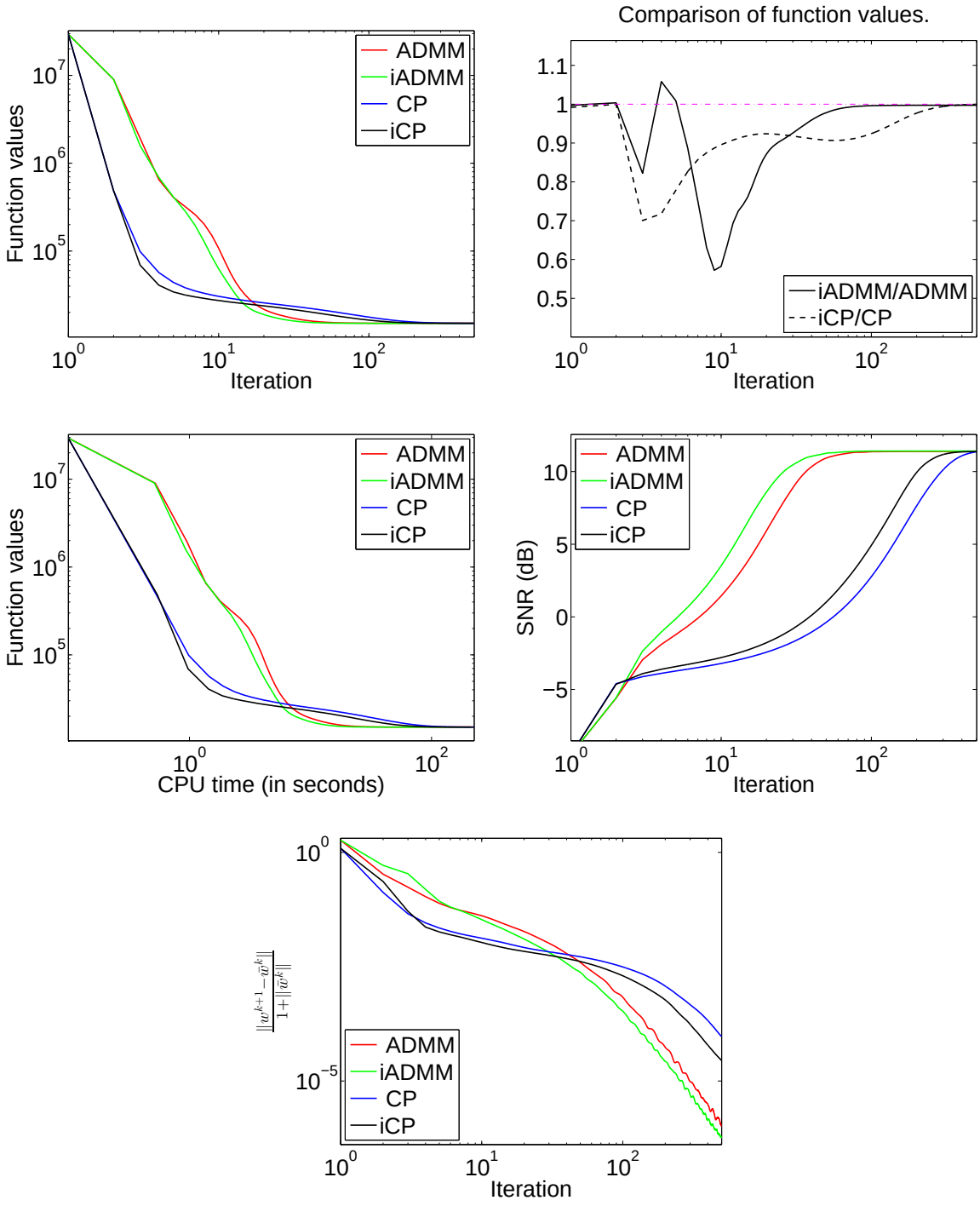

FIG. 5.4. *From top-left to bottom-right in Figure 5.4, the plots are, respectively, objective function values versus iteration, the ratios of function values attained by inertial algorithms divided by those attained by their corresponding original algorithms versus iteration, objective function values versus CPU time (in seconds), SNR values (in dB) versus iteration, and relative difference* $\|w^{k+1} - \bar{w}^k\|/(1 + \|\bar{w}^k\|)$ *versus iteration.*

the inertial variable $\bar{w}^k$. Admittedly, inertial algorithms need to store one more variable, i.e., $\bar{w}^k$, at each step, which could be costly for large scale problems.

We emphasize that our main contributions are the proposition of a class of inertial proximal ADMMs and their convergence analysis. Compared to the accelerated methods [48, 60, 37] which guarantee convergence in function values, our iterate convergence results are stronger at the cost of more restrictive inertial step sizes. In our experiments, the extrapolation steplength α_k was set to be constant. How to select α_k adaptively such that the overall performance is stable and more efficient deserves further investigation. Though some experimental observations on the dependence of the feasible range of α_k and the relative magnitudes of η and β have been observed in [29], theoretical explanations and deep insights are undoubtedly desired. Moreover, the requirement that $\{\alpha_k\}_{k=0}^{\infty}$ is nondecreasing seems not reasonable either. Interesting topics for future research may include relaxing the conditions on $\{\alpha_k\}_{k=0}^{\infty}$, improving the convergence results and proposing modified inertial type algorithms so that the extrapolation stepsize can be significantly enlarged.

Acknowledgement. We thank three anonymous referees for their thoughtful and insightful comments, which improved the paper greatly. C. H. Chen was supported in part by Natural Science Foundation of Jiangsu Province under project grant No. BK20130550. R. H. Chan was supported by CUHK DAG Grant 4053007 and HKRGC Grant CUHK40041, CUHK2/CRF/11G and AoE/M-05/12. S. Q. Ma was supported by Hong Kong Research Grants Council General Research Fund Early Career Scheme (Project ID: CUHK 439513). J. F. Yang was supported by the Natural Science Foundation of China (NSFC-11371192), the Fundamental Research Funds for the Central Universities (20620140574), the Program for New Century Excellent Talents in University (NCET-12-0252), and the Key Laboratory for Numerical Simulation of Large Scale Complex Systems of Jiangsu Province. This work was done while J. F. Yang was visiting the Chinese University of Hong Kong.

REFERENCES

[1] A. Chambolle and T. Pock, "On the ergodic convergence rates of a first-order primal-dual algorithm," *preprint*, September, 2014.

[2] C. Chen, S. Ma, and J. Yang, "A general inertial proximal point method for mixed variational inequality problem," *arXiv preprint arXiv:1407.8238, under 2nd reound review*, 2014.

[3] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, 2011.

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[5] R. Shefi and M. Teboulle, "Rate of convergence analysis of decomposition methods based on the proximal method of multipliers for convex minimization," *SIAM J. Optim.*, vol. 24, no. 1, pp. 269–297, 2014.

[6] J. Yang and Y. Zhang, "Alternating direction algorithms for ℓ_1-problems in compressive sensing," *SIAM J. Sci. Comput.*, vol. 33, no. 1, pp. 250–278, 2011.

[7] J. Yang and X. Yuan, "Linearized augmented lagrangian and alternating direction methods for nuclear norm minimization," *Mathematics of Computation*, vol. 82, no. 2, pp. 301–329, 2013.

[8] E. Esser, X. Zhang, and T. F. Chan, "A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science," *SIAM J. Imaging Sci.*, vol. 3, no. 4, pp. 1015–1046, 2010.

[9] J. Eckstein, "Splitting methods for monotone operators with applications to parallel optimization," Ph.D. dissertation, Massachusetts Institute of Technology, 1989.

[10] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation.* Prentice Hall Inc., 1989.

[11] J. Wright, A. Ganesh, K. Min, and Y. Ma, "Compressive principal component pursuit," *Information and Inference*, vol. 2, no. 1, pp. 32–68, 2013.

[12] A. Agarwal, S. Negahban, and M. J. Wainwright, "Noisy matrix decomposition via convex relaxation: Optimal rates in high dimensions," *The Annals of Statistics*, vol. 40, no. 2, pp. 1171–1197, 2012.

[13] R. Glowinski and A. Marrocco, "Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité, d'une classe de problèmes de Dirichlet non linéaires," *R.A.I.R.O., R2*, vol. 9, no. R-2, pp. 41–76, 1975.

[14] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers and Mathematics with Applications*, vol. 2, no. 1, pp. 17–40, 1976.

[15] J. Eckstein, "Some saddle-function splitting methods for convex programming," *Optimization Methods and Software*, vol. 4, no. 1, pp. 75–83, 1994.

[16] H. Attouch and F. Alvarez, "The heavy ball with friction dynamical system for convex constrained minimization problems," in *Optimization (Namur, 1998)*, ser. Lecture Notes in Econom. and Math. Systems. Springer, Berlin, 2000, vol. 481, pp. 25–35.

[17] F. Alvarez, "On the minimizing property of a second order dissipative system in Hilbert spaces," *SIAM J. Control Optim.*, vol. 38, no. 4, pp. 1102–1119 (electronic), 2000.

[18] F. Alvarez and H. Attouch, "An inertial proximal method for maximal monotone operators via discretization of a nonlinear oscillator with damping," *Set-Valued Anal.*, vol. 9, no. 1-2, pp. 3–11, 2001, wellposedness in optimization and related topics (Gargnano, 1999).

[19] M. R. Hestenes, "Multiplier and gradient methods," *J. Optimization Theory Appl.*, vol. 4, pp. 303–320, 1969.

[20] M. J. D. Powell, "A method for nonlinear constraints in minimization problems," in *Optimization (Sympos., Univ. Keele, Keele, 1968)*. London: Academic Press, 1969, pp. 283–298.

[21] J.-J. Moreau, "Proximité et dualité dans un espace hilbertien," *Bull. Soc. Math. France*, vol. 93, pp. 273–299, 1965.

[22] B. Martinet, "Régularisation d'inéquations variationnelles par approximations successives," *Rev. Française Informat. Recherche Opérationnelle*, vol. 4, no. Ser. R-3, pp. 154–158, 1970.

[23] R. T. Rockafellar, "Monotone operators and the proximal point algorithm," *SIAM J. Control Optimization*, vol. 14, no. 5, pp. 877–898, 1976.

[24] A. Moudafi and E. Elissabeth, "Approximate inertial proximal methods using the enlargement of maximal monotone operators," *International Journal of Pure and Applied Mathemtics*, vol. 5, no. 3, pp. 283–299, 2003.

[25] F. Alvarez, "Weak convergence of a relaxed and inertial hybrid projection-proximal point algorithm for maximal monotone operators in Hilbert space," *SIAM J. Optim.*, vol. 14, no. 3, pp. 773–782 (electronic), 2004.

[26] P.-E. Maingé and A. Moudafi, "A proximal method for maximal monotone operators via discretization of a first order dissipative dynamical system," *J. Convex Anal.*, vol. 14, no. 4, pp. 869–878, 2007.

[27] P. Ochs, Y. Chen, T. Brox, and T. Pock, "ipiano: Inertial proximal algorithm for non-convex optimization," *manuscript*, 2014.

[28] P. Ochs, T. Brox, and T. Pock, "ipiasco: Inertial proximal algorithm for strongly convex optimization," *manuscript*, 2014.

[29] D. A. Lorenz and T. Pock, "An inertial forward-backward algorithm for monotone inclusions," *J. Math. Imaging Vis.*, 2014.

[30] H. Attouch, J. Peypouquet, and P. Redont, "A dynamical approach to an inertial forward-backward algorithm for convex minimization," *SIAM J. Optim.*, vol. 24, no. 1, pp. 232–256, 2014.

[31] R. I. Bot and E. R. Csetnek, "An inertial tseng's type proximal algorithm for nonsmooth and nonconvex optimization problems," *arXiv preprint arXiv:1406.0724*, 2014.

[32] R. I. Bot, E. R. Csetnek, and S. László, "An inertial forward-backward algorithm for the minimization of the sum of two nonconvex functions," *arXiv preprint arXiv:1410.0641*, 2014.

[33] R. I. Bot, E. R. Csetnek, and C. Hendrich, "Inertial douglas-rachford splitting for monotone inclusion problems," *arXiv preprint arXiv:1403.3330*, 2014.

[34] R. I. Bot and E. R. Csetnek, "An inertial alternating direction method of multipliers," *arXiv preprint arXiv:1404.4582*, 2014.

[35] Y. Nesterov, "Gradient methods for minimizing composite objective function," 2007.

[36] E. T. Hale, W. Yin, and Y. Zhang, "Fixed-point continuation for ℓ_1-minimization: Methodology and convergence," *SIAM Journal on Optimization*, vol. 19, no. 3, pp. 1107–1130, 2008.

[37] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[38] B. He, L.-Z. Liao, D. Han, and H. Yang, "A new inexact alternating directions method for monotone variational inequalities," *Math. Program.*, vol. 92, no. 1, Ser. A, pp. 103–118, 2002.

[39] M. Fazel, T. K. Pong, D. F. Sun, and P. Tseng, "Hankel matrix rank minimization with applications to system identification and realization," *SIAM J. Matrix Analysis and Applications, to appear*, 2013.

[40] X. Cai, G. Gu, B. He, and X. Yuan, "A proximal point algorithm revisit on the alternating direction method of multipliers," *Sci. China Math.*, vol. 56, no. 10, pp. 2179–2186, 2013.

[41] R. T. Rockafellar, *Convex analysis*, ser. Princeton Mathematical Series, No. 28. Princeton University Press.

[42] A. Chambolle, V. Caselles, D. Cremers, M. Novaga, and T. Pock, "An introduction to total variation for image analysis," *Theoretical foundations and numerical methods for sparse recovery*, vol. 9, pp. 263–340, 2010.

[43] B. He and X. Yuan, "Convergence analysis of primal-dual algorithms for a saddle-point problem: From contraction perspective," *SIAM Journal on Imaging Sciences*, vol. 5, no. 1, pp. 119–149, 2012.

[44] A. Moudafi and M. Oliny, "Convergence of a splitting inertial proximal method for monotone operators," *J. Comput. Appl. Math.*, vol. 155, no. 2, pp. 447–454, 2003.

[45] W. Deng, M.-J. Lai, and W. Yin, "On the o(1/k) convergence and parallelization of the alternating direction method of multipliers," *arXiv preprint arXiv:1312.3040*, 2013.

[46] D. Davis and W. Yin, "Convergence rate analysis of several splitting schemes," *arXiv preprint arXiv:1406.4834*, 2014.

[47] C. Chen, "Numerical algorithms for a class of matrix norm approximation problems," Ph.D. dissertation, Nanjing University, 2012.

[48] Y. E. Nesterov, "A method for solving the convex programming problem with convergence rate $O(1/k^2)$," *Dokl. Akad. Nauk SSSR*, vol. 269, no. 3, pp. 543–547, 1983.

[49] X. Wang and X. Yuan, "The linearized alternating direction method of multipliers for dantzig selector," *SIAM Journal on Scientific Computing*, vol. 34, no. 5, pp. 2792–2811, 2012.

[50] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1, pp. 259–268, 1992.

[51] A. Chambolle, "An algorithm for total variation minimization and applications," *Journal of Mathematical imaging and vision*, vol. 20, no. 1-2, pp. 89–97, 2004.

[52] Y. Wang, J. Yang, W. Yin, and Y. Zhang, "A new alternating minimization algorithm for total variation image reconstruction," *SIAM Journal on Imaging Sciences*, vol. 1, no. 3, pp. 248–272, 2008.

[53] T. Goldstein and S. Osher, "The split bregman method for l1-regularized problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 323–343, 2009.

[54] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *Information Theory, IEEE Transactions on*, vol. 52, no. 2, pp. 489–509, 2006.

[55] D. Needell and R. Ward, "Stable image reconstruction using total variation minimization," *SIAM Journal on Imaging Sciences*, vol. 6, no. 2, pp. 1035–1058, 2013.

[56] M. K. Ng, R. H. Chan, and W.-C. Tang, "A fast algorithm for deblurring models with neumann boundary conditions," *SIAM Journal on Scientific Computing*, vol. 21, no. 3, pp. 851–866, 1999.

[57] J. Yang, Y. Zhang, and W. Yin, "A fast alternating direction method for tvl1-l2 signal reconstruction from partial fourier data," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 4, no. 2, pp. 288–297, 2010.

[58] R. H. Chan, J. Yang, and X. Yuan, "Alternating direction method for image inpainting in wavelet domains," *SIAM J. Imaging Sci.*, vol. 4, no. 3, pp. 807–826, 2011.

[59] R. Baraniuk, H. Choi, R. Neelamani, V. Ribeiro, J. Romberg, H. Guo, F. Fernandes, B. Hendricks, R. Gopinath, M. Long, J. E. Odegard, and D. Wei, "Rice wavelet toolbox, version 2.4," 2002. [Online]. Available: http://www.dsp.rice.edu/software/rice-wavelet-toolbox

[60] O. Güler, "New proximal point algorithms for convex minimization," *SIAM J. Optim.*, vol. 2, no. 4, pp. 649–664, 1992.