A new relative-error inexact ADMM splitting algorithm for convex optimization

M. Marques Alves * Marina Geremia †

April 28, 2022

Abstract

We propose a new relative-error inexact version of the alternating direction method of multipliers (ADMM) for solving convex problems. Our main algorithm is essentially a generalization of [44, Algorithm 1] promoting acceleration through inertial effects on iteration and with a somewhat more flexible relative-error criterion. Contrary to the majority of existing inexact relative-error versions of ADMM, one of the distinctive features of both [44, Algorithm 1] and the main algorithm proposed in this paper relies on the fact that the first subproblem is supposed to be solved exactly whereas the second one allows for inexact computations. We justify the effectiveness of the proposed algorithm through some numerical experiments on regression and classification problems.

2000 Mathematics Subject Classification: 90C25, 90C06, 49J52.

Key words: Convex optimization, ADMM, inexact, relative-error, inertial algorithms.

1 Introduction

Consider the convex problem

$$\underset{x\in\mathcal{H}}{\text{minimize}}\, f(x)+g(Lx), \tag{1}$$

where $f:\mathcal{H}\to(-\infty,\infty]$ and $g:\mathcal{G}\to(-\infty,\infty]$ are lower semicontinuous proper convex functions and $L:\mathcal{H}\to\mathcal{G}$ is a linear operator ($\mathcal{H}$ and $\mathcal{G}$ denote finite-dimensional inner product spaces). Problem (1) appears in different contexts in applied mathematics, including optimization, inverse problems, machine learning, among others.

One of the most popular numerical algorithms for solving (1) is the alternating direction method of multipliers (ADMM) [25, 27, 28], which has now attracted a lot of attention from the numerical optimization community (see, e.g., [9, 11, 12, 13, 14, 15, 21, 22, 29, 31, 32, 33, 35, 40, 44]).

In this paper we propose and study a new inexact version of the (semi-proximal) ADMM allowing relative-error criteria for the solution of the second subproblem (which will appear in the formulation

*Departamento de Matemática, Universidade Federal de Santa Catarina, Florianópolis, Brazil, 88040-900 (maicon.alves@ufsc.br). The work of this author was partially supported by CNPq grant 308036/2021-2.

†Departamento de Matemática, Universidade Federal de Santa Catarina, Florianópolis, Brazil, 88040-900. Departamento de Ensino, Pesquisa e Extensão, Instituto Federal de Santa Catarina (IFSC) (marina.geremia@ifsc.edu.br).

of proposed algorithm) and promoting acceleration through the effects of inertia on the iterations. Our approach is motivated by the recent contribution [44], where a new partially inexact ADMM is formulated for solving two-block separable convex optimization problems.

Motivation. We first note that (1) is clearly equivalent to the separable problem

$$\begin{array}{ll} \text{minimize} & f(x)+g(y), \\ \text{subject to} & Lx-y=0. \end{array} \tag{2}$$

An iteration of the standard (semi-proximal) ADMM [15, 17] for solving (2) can be described as follows: given a starting point $(y_0, z_0) \in \mathcal{G}^2$ and a regularization parameter $\gamma > 0$, iterate for $k \geq 0$:

$$x_{k+1} \in \operatorname*{argmin}_{x\in\mathcal{H}} \left\{ f(x) + \langle z_k \,|\, Lx - y_k\rangle + \frac{\gamma}{2}\|Lx-y_k\|^2 \right\}, \tag{3}$$

$$y_{k+1} = \operatorname*{argmin}_{y\in\mathcal{G}} \left\{ g(y) + \langle z_k \,|\, Lx_{k+1} - y\rangle + \frac{\gamma}{2}\|Lx_{k+1}-y\|^2 + \frac{1}{2\gamma}\|y-y_k\|^2 \right\}, \tag{4}$$

$$z_{k+1} = z_k + \gamma\,(Lx_{k+1} - y_{k+1})\,. \tag{5}$$

We consider here the case in which (3) can be solved exactly and, on the other hand, (4) is supposed to be solved only approximately by some other (inner) algorithm, like, for instance, CG or BFGS, depending on the particular structure of the function $g(\cdot)$ in (2).

With this in mind, we will introduce a notion of relative-error approximate solution for (4) (more details will be given on Section 2). To this end, first note that (4) is an instance of the general family of minimization problems

$$\operatorname*{minimize}_{y\in\mathcal{G}} \left\{ g(y) + \langle z \,|\, Lx - y\rangle + \frac{\gamma}{2}\|Lx-y\|^2 + \frac{1}{2\gamma}\|y-w\|^2 \right\}, \tag{6}$$

where $x \in \mathcal{H}$, $(z, w) \in \mathcal{G}^2$ and $\gamma > 0$ are given (in the case of (4), we have $(z, x, w) = (z_k, x_{k+1}, y_k)$). Moreover, since the function $g(\cdot)$ is convex, we have that (6) is also equivalent to the inclusion/equation system for the pair (y, v):

$$\begin{cases} v \in \partial g(y), \\ \gamma v - \gamma\,[z + \gamma(Lx - y)] + y - w = 0. \end{cases} \tag{7}$$

A formal definition of approximate (inexact) solution of (7) (or, equivalently, (6)) will be given in Definition 2.1 in Section 2; such a notion of approximate solution will allow for errors in both the inclusion and the equation in (7).

The extended-solution set. The Fenchel dual of (1) is

$$\operatorname*{maximize}_{z\in\mathcal{G}} -f^*(-L^*z) - g^*(z), \tag{8}$$

where $f^* : \mathcal{H} \to (-\infty, \infty]$ and $g^* : \mathcal{G} \to (-\infty, \infty]$ denote the Fenchel conjugates of f and g, respectively, and $L^* : \mathcal{G} \to \mathcal{H}$ denotes the adjoint operator of L. Under standard regularity conditions [10] on f, g and L it is well-known that (1) and (8) are, respectively, equivalent to the (monotone) inclusions

$$0 \in \partial f(x) + L^*\partial g(Lx), \tag{9}$$

and

$$0 \in -L\partial f^*(-L^*z) + \partial g^*(z). \tag{10}$$

We make the blanket assumption:

Assumption 1.1. *For the function f and the operator L as in* (1)*, the following holds:*

$$\partial(f^* \circ -L^*) = -L \circ \partial f^* \circ -L^*.$$

Several sufficient conditions for Assumption 1.1 to hold true can be found, e.g., in [10]. Motivated by [44, Eq. (19)], we will consider an extended-solution set $\mathcal{S}$, attached to the pair of inclusions (9)–(10), defined as

$$\mathcal{S} := \{(z, w, s) \in \mathcal{G}^3 \mid s \in \partial(f^* \circ -L^*)(z), \quad w \in \partial g^*(z) \text{ and } s + w = 0\}. \tag{11}$$

Under the Assumption 1.1, it is easy to check that if $(z, w, s) \in \mathcal{S}$, then it follows that there exists $x \in \mathcal{H}$ such that $x \in \partial f^*(-L^*z)$, $s = -Lx$ and x and z are solutions of (9) and (10), respectively.

Throughout this work we will also assume the following.

Assumption 1.2. *We assume the extended solution set $\mathcal{S}$ as in* (11) *is nonempty.*

Inertial algorithms. Iterative algorithms with inertial effects for monotone inclusions (and related topics in optimization, saddle-point, equilibrium problems, etc) were first proposed in the seminal paper [2] and subsequently developed in various directions of research by different authors and research groups (see, e.g., [3, 5, 6, 7, 8, 12, 16] and references therein). Basically, the main idea consists in at a current iterate, say p_k, produce an "inertial effect" by a simple extrapolation:

$$\widehat{p}_k = p_k + \alpha_k(p_k - p_{k-1}),$$

where $\alpha_k \geq 0$, and then generate the next iterate p_{k+1} from $\widehat{p}_k$ instead of p_k (see (17)–(19) below). Our main algorithm, namely Algorithm 1, will benefit from inertial effects on the iteration; see the comments and remarks following Algorithm 1 for more discussions regarding the effects of inertia.

Main contributions. We present a theoretical and computational study of a variant of a partially inexact (semi-proximal) ADMM proposed and studied in [44]. Contrary to [44, Algorithm 1], our main algorithm, namely Algorithm 1 below, benefits from the addition of inertial effects; see (17)–(19). The convergence analysis is presented in Theorem 2.5, to which the proof incorporates some elements of [3]. We also mention that the relative-error criterion for Algorithm 1 is somewhat more general than the corresponding one in [44] (see the third remark following Definition 2.1). We justify the effectiveness of our main algorithm through the realization of numerical experiments on some machine learning problems (see Section 3).

Related works. As was already emphasized, this paper is motivated by [44]. Some other related works are as follows. Paper [1] proposes a partially inexact ADMM for which the first subproblem is supposed to be solved inexactly. The analysis of the main algorithm in [1] is performed by viewing it as a special instance of a non-Euclidean version of the hybrid proximal extragradient method [34]. In contrast to this, our main algorithm, namely Algorithm 1 below, assumes the second subproblem is solved inexactly, its convergence analysis is much simpler when compared to [1], and it also incorporates inertial effects in the iteration. Moreover, since (22)–(24) below also allows for errors in ∂g, the

error criterion we propose here is potentially more general than the corresponding one in [1]. Other relative-error inexact versions of ADMM were also previously studied in [46, 47], but we notice that the convergence results were restricted to the analysis of the dual sequences. We also mention that the relative-error inexact variants of the ADMM from [3, 22] only apply to (1) in the particular case of $L = I$, and, additionally, these variants assume the first subproblem to be solved inexactly with the error condition verified only a-posteriori, that is, only after the computation of second subproblem's solution.

Organization of the paper. The material is organized as follows. In Section 2, we present our main algorithm (Algorithm 1), and its asymptotic analysis in Theorem 2.5. Numerical experiments will be presented in Section 3. Appendix A contains some auxiliary results that are useful in the proof of Theorem 2.5.

General notation. We denote by $\mathcal{H}$ and $\mathcal{G}$ finite-dimensional inner product spaces with inner product and induced norm denoted, respectively, by $\langle \cdot \,|\, \cdot \rangle$ and $\|\cdot\| = \sqrt{\langle \cdot \,|\, \cdot \rangle}$. For any set $\mathcal{X}$ we denote by X^n the n-product $\mathcal{X} \times \cdots \times \mathcal{X}$. In $\mathcal{G}^3$, we will consider the inner product and induced norm defined, respectively, by

$$\langle p \,|\, p' \rangle_\gamma := \langle z \,|\, z' \rangle + \langle w \,|\, w' \rangle + \gamma^2 \langle s \,|\, s' \rangle, \qquad \|p\|_\gamma = \sqrt{\langle p \,|\, p \rangle_\gamma}, \tag{12}$$

where $p = (z, w, s), p' = (z', w', s') \in \mathcal{G}^3$ and $\gamma > 0$. The *ε-subdifferential* and the *subdifferential* of a convex function $g : \mathcal{H} \to (-\infty, \infty]$ at $x \in \mathcal{H}$ are defined as $\partial_\varepsilon g(x) := \{u \in \mathcal{H} \mid g(y) \geq g(x) + \langle u \,|\, y - x \rangle - \varepsilon \quad \forall y \in \mathcal{H}\}$ and $\partial g(x) := \partial_0 g(x)$, respectively. For additional details on standard notations and definitions of convex analysis we refer the reader to the references [10, 39].

2 The main algorithm and its convergence analysis

Consider the minimization problem (1), i.e.,

$$\operatorname*{minimize}_{x \in \mathcal{H}} f(x) + g(Lx), \tag{13}$$

where $f : \mathcal{H} \to (-\infty, \infty]$ and $g : \mathcal{G} \to (-\infty, \infty]$ are lower semicontinuous proper convex functions and $L : \mathcal{H} \to \mathcal{G}$ is a linear operator between finite-dimensional inner product spaces $\mathcal{H}$ and $\mathcal{G}$.

In this section we present our main algorithm, namely Algorithm 1 below. This is a partially inexact (the second block is allowed to be solved inexactly) semi-proximal ADMM with relative-error criterion for the second subproblem. Recall the extended solution set $\mathcal{S}$ as in (11) and Assumptions 1.1 and 1.2. The main result of this section is Theorem 2.5 below; the three technical lemmas 2.2, 2.3 and 2.4 will be used in the proof of Theorem 2.5.

Before presenting our main algorithm, as we discussed in the Introduction, we have to formalize the notion of inexact solution that will be used to compute approximate solution for the second subproblem. Recall that the second subproblem of the standard (semi-proximal) ADMM (see (4)) belongs to the general family of minimization problems (6), which is, in particular, equivalent to the inclusion/equation system (7) for the pair (y, v), i.e.,

$$\begin{cases} v \in \partial g(y), \\ \gamma v - \gamma\left[z + \gamma(Lx - y)\right] + y - w = 0. \end{cases} \tag{14}$$

Definition 2.1 (σ-approximate solution of (6))**.** *For $x \in \mathcal{H}$, $(z, w) \in \mathcal{G}^2$ and $\lambda > 0$, a triple $(v, \widetilde{y}, \varepsilon) \in \mathcal{G} \times \mathcal{G} \times \mathbb{R}_+$ is said to be a σ-approximate solution of (6) (or, equivalently, of (7)) if $\sigma \in [0, 1)$ and*

$$\begin{cases} v \in \partial_\varepsilon g(\widetilde{y}), \\ \gamma v - \gamma\left[z + \gamma(Lx - \widetilde{y})\right] + \widetilde{y} - w =: e \\ \|e\|^2 + 2\gamma\varepsilon \leq \sigma^2\left(\gamma^2\|Lx - \widetilde{y}\|^2 + \|\widetilde{y} - w\|^2\right). \end{cases} \tag{15}$$

We will also write

$$\widetilde{y} \approx \operatorname*{argmin}_{y \in \mathcal{G}} \left\{ g(y) + \langle z \,|\, Lx - y\rangle + \frac{\gamma}{2}\|Lx - y\|^2 + \frac{1}{2\gamma}\|y - w\|^2 \right\}$$

meaning that there exists (v, ε) such that $(v, \widetilde{y}, \varepsilon)$ satisfies (15).

We now make some remarks regarding Definition 2.1:

(i) Note that if $\sigma = 0$ in (15), then it follows that $e = 0$ and $\varepsilon = 0$, which is to say that the pair $(\widetilde{y}, v)$ satisfies the inclusion/equation system (7) (recall that $\partial_0 g = \partial g$) and, in particular, $\widetilde{y}$ is the exact solution of (6).

(ii) The error criterion for (7) as in (15) belongs to the class of *relative-error* criteria for proximal-type algorithms. Different variants of such error conditions have been employed for computing approximate solution for (sub) problems for a wide range of algorithms in monotone inclusions, convex optimization, saddle-point problems, etc (see, e.g., [3, 22, 34, 41, 42, 43]).

(iii) The relative-error criterion as in (15) is motivated by a similar one introduced in [44, Definition 2.1]. On the other hand, we mention that (15) is somewhat more general than the criterion of [44, Definition 2.1], since it has the additional term $\|Lx - \widetilde{y}\|^2$ in the right-hand side of the inequality.

(iv) The error criterion (15) will be used to compute approximate solutions in step 3 of our main algorithm, namely Algorithm 1 below (see (22)–(24)).

(v) As an illustrative example, consider the special case of the LASSO problem [45]

$$\min_{x \in \mathbb{R}^d} \left\{ \frac{1}{2}\|Ax - b\|^2 + \nu\|x\|_1 \right\}, \tag{16}$$

where $A \in \mathbb{R}^{n \times d}$, $b \in \mathbb{R}^n$ and $\nu > 0$. Problem (16) is clearly a special instance of (1) in which $L := I$, $f(x) := \nu\|x\|_1$ and $g(x) := (1/2)\|Ax - b\|^2$ (for more details see Section 3 below). In this case, our inclusion/equation system (7) clearly reduces to

$$v = A^*(Ay - b), \quad \gamma v - \gamma\left[z + \gamma(x - y)\right] + y - w = 0,$$

or, in other words, in this special case, (7) is equivalent to the linear system (operator equation)

$$\Big(\gamma A^*A + (\gamma^2+1)I\Big)y = \gamma\,(A^*b+z) + w.$$

The latter linear system can be solved by the CG algorithm [37], where e as in (15) will simply denote the residual of the system and the inequality in (15) can be used as a stopping criterion for CG.

Next is our main algorithm.

Algorithm 1. Inexact ADMM algorithm for solving (1)

(0) Let $(z_0, w_0, y_0) = (z_{-1}, w_{-1}, y_{-1}) \in \mathcal{G}^3$, $0 \le \alpha, \sigma < 1$ and $\gamma > 0$. Set $k = 0$.

(1) Choose $\alpha_k \in [0, \alpha]$ and let

$$\widehat{z}_k = z_k + \alpha_k(z_k - z_{k-1}), \tag{17}$$
$$\widehat{w}_k = w_k + \alpha_k(w_k - w_{k-1}), \tag{18}$$
$$\widehat{y}_k = y_k + \alpha_k(y_k - y_{k-1}). \tag{19}$$

(2) Compute

$$x_{k+1} \in \operatorname*{argmin}_{x\in\mathcal{H}} \left\{ f(x) + \langle \widehat{z}_k \,|\, Lx - \widehat{y}_k\rangle + \frac{\gamma}{2}\|Lx - \widehat{y}_k\|^2 \right\}. \tag{20}$$

(3) Compute

$$y_{k+1} \approx \operatorname*{argmin}_{y\in\mathcal{G}} \left\{ g(y) + \langle \widehat{z}_k \,|\, Lx_{k+1} - y\rangle + \frac{\gamma}{2}\|Lx_{k+1} - y\|^2 + \frac{1}{2\gamma}\|y - \widehat{w}_k\|^2 \right\} \tag{21}$$

in the sense of Definition 2.1, i.e., compute $(y_{k+1}, v_{k+1}, \varepsilon_{k+1}) \in \mathcal{G}\times\mathcal{G}\times\mathbb{R}_+$ such that

$$v_{k+1} \in \partial_{\varepsilon_{k+1}} g(y_{k+1}), \tag{22}$$
$$\gamma v_{k+1} - \gamma\,[\widehat{z}_k + \gamma(Lx_{k+1} - y_{k+1})] + y_{k+1} - \widehat{w}_k =: e_{k+1}, \tag{23}$$
$$\|e_{k+1}\|^2 + 2\gamma\varepsilon_{k+1} \le \sigma^2\left(\gamma^2\|Lx_{k+1} - \widehat{y}_k\|^2 + \|y_{k+1} - \widehat{w}_k\|^2\right). \tag{24}$$

(4) Set

$$z_{k+1} = \widehat{z}_k + \gamma(Lx_{k+1} - y_{k+1}), \tag{25}$$
$$w_{k+1} = \widehat{w}_k + \gamma(z_{k+1} - v_{k+1}), \tag{26}$$

$k = k+1$ and go to step 1.

We now make some remarks concerning Algorithm 1:

(i) As we mentioned before, Algorithm 1 is closely related to [44, Algorithm 1]. Indeed, by letting $\alpha_k \equiv 0$ in (17)–(19), in which case $(\widehat{z}_k, \widehat{w}_k, \widehat{y}_k) = (z_k, w_k, y_k)$, and by deleting the additional

term $\gamma^2\|Lx_{k+1} - \widehat{y}_k\|^2$ in (24), Algorithm 1 above reduces to Algorithm 1 in the latter reference for solving (1).

(ii) Algorithm 1 is specially designed for instances of (1) in which (20) has a closed-form solution, i.e., for problems in which (20) is easy to solve. In this regard, one example of interest is when $f(\cdot) = \|\cdot\|_1$ and $L = I$, in which case (20) has a unique solution given explicitly by $x_{k+1} = \mathrm{prox}_{\gamma^{-1}\|\cdot\|_1}(\widehat{y}_k - \gamma^{-1}\widehat{z}_k)$. On the other hand, we assume that the computation of y_{k+1} as in (21) demands the use of an (inner) algorithm, which the choice of depends on the particular structure of the function g, and, in this case, one can use (22)–(24) as a stopping criterion for the inner algorithm of choice.

(iii) Recall that we discussed in the Introduction (see "Related works") other ADMM-type algorithms related to Algorithm 1.

(iv) The main result on the convergence of Algorithm 1 is Theorem 2.5 below. Numerical experiments will be presented and discussed in Section 3.

Next we present three technical lemmas – Lemmas 2.2, 2.3 and 2.4 –, which will be useful to prove the main theorem on the convergence of Algorithm 2.5, namely Theorem 2.5 below.

Lemma 2.2. *Consider the sequences evolved by* Algorithm 1, *let* $\mathcal{S}$ *be as in* (11) *and define*

$$p_k = (z_k, w_k, -y_k) \ \text{ and } \ \widehat{p}_k = (\widehat{z}_k, \widehat{w}_k, -\widehat{y}_k), \qquad \forall k \geq 0. \tag{27}$$

Define also, for all $k \geq 0$,

$$\widetilde{p}_{k+1} = (\widetilde{z}_{k+1}, \widetilde{w}_{k+1}, -\widetilde{y}_{k+1}), \tag{28}$$

where

$$\widetilde{z}_{k+1} := \widehat{z}_k + \gamma(Lx_{k+1} - \widehat{y}_k), \quad \widetilde{w}_{k+1} := y_{k+1} \ \text{ and } \ \widetilde{y}_{k+1} := y_{k+1}. \tag{29}$$

Then,

(a) *For all* $k \geq 0$,

$$-Lx_{k+1} \in \partial\left(f^* \circ -L^*\right)(\widetilde{z}_{k+1}). \tag{30}$$

(b) *For all* $k \geq 0$ *and* $p = (z, w, s) \in \mathcal{S}$,

$$\langle p_{k+1} - \widehat{p}_k \,|\, p - \widetilde{p}_{k+1}\rangle_\gamma \geq -\gamma\varepsilon_{k+1}. \tag{31}$$

Proof. (a) We first note that from (20) and the first definition in (29), we obtain $0 \in \partial f(x_{k+1}) + L^*\widetilde{z}_{k+1}$, or, equivalently, $-L^*\widetilde{z}_{k+1} \in \partial f(x_{k+1})$. As $(\partial f)^{-1} = \partial f^*$, the latter inclusion is also equivalent to $x_{k+1} \in \partial f^*(-L^*\widetilde{z}_{k+1})$, which in turn yields $-Lx_{k+1} \in -L\partial f^*(-L^*\widetilde{z}_{k+1})$, which by Assumption 1.1 reduces to (30).

(b) As $\big(\partial_{\varepsilon_{k+1}} g\big)^{-1} = \partial_{\varepsilon_{k+1}} g^*$, we have that (22) is equivalent to the inclusion

$$y_{k+1} \in \partial_{\varepsilon_{k+1}} g^*(v_{k+1}). \tag{32}$$

As $p = (z, w, s) \in \mathcal{S}$, according to the definition of $\mathcal{S}$ in (11), we have $s \in \partial\,(f^* \circ -L^*)\,(z)$ and $w \in \partial g^*(z)$. The latter inclusions combined with (30) and (32) and the monotonicity of $\partial(f^* \circ -L^*)$ and ∂g^* yield

$$\langle s + Lx_{k+1} \,|\, z - \widetilde{z}_{k+1}\rangle \geq 0 \;\text{ and }\; \langle w - y_{k+1} \,|\, z - v_{k+1}\rangle \geq -\varepsilon_{k+1}. \tag{33}$$

To prove (31), note that from (27),(28), (25), (26), the assumption $p = (z, w, s) \in \mathcal{S}$ (combined with the definition of $\mathcal{S}$ as in (11)) and (12), we find

$$\begin{aligned}
\langle p_{k+1} - \widehat{p}_k \,|\, p - \widetilde{p}_{k+1}\rangle_\gamma &= \langle z_{k+1} - \widehat{z}_k \,|\, z - \widetilde{z}_{k+1}\rangle + \langle w_{k+1} - \widehat{w}_k \,|\, w - \widetilde{w}_{k+1}\rangle + \gamma^2\langle -y_{k+1} + \widehat{y}_k \,|\, s + \widetilde{y}_{k+1}\rangle \\
&= \gamma\langle Lx_{k+1} - y_{k+1} \,|\, z - \widetilde{z}_{k+1}\rangle + \gamma\langle z_{k+1} - v_{k+1} \,|\, w - y_{k+1}\rangle \\
&\quad + \gamma\langle \gamma y_{k+1} - \gamma\widehat{y}_k \,|\, w - y_{k+1}\rangle \\
&= \gamma\langle s + Lx_{k+1} \,|\, z - \widetilde{z}_{k+1}\rangle + \gamma\langle w - y_{k+1} \,|\, z - \widetilde{z}_{k+1}\rangle + \gamma\langle z_{k+1} - v_{k+1} \,|\, w - y_{k+1}\rangle \\
&\quad + \gamma\langle \gamma y_{k+1} - \gamma\widehat{y}_k \,|\, w - y_{k+1}\rangle \\
&= \gamma\langle s + Lx_{k+1} \,|\, z - \widetilde{z}_{k+1}\rangle + \gamma\langle z - \widetilde{z}_{k+1} + z_{k+1} - v_{k+1} + \gamma y_{k+1} - \gamma\widehat{y}_k \,|\, w - y_{k+1}\rangle \\
&= \gamma\left[\langle s + Lx_{k+1} \,|\, z - \widetilde{z}_{k+1}\rangle + \langle z - v_{k+1} - (\widetilde{z}_{k+1} - z_{k+1}) + \gamma\,(y_{k+1} - \widehat{y}_k) \;|\, w - y_{k+1}\rangle\right] \\
&= \gamma\left[\langle s + Lx_{k+1} \,|\, z - \widetilde{z}_{k+1}\rangle + \langle z - v_{k+1} - \gamma\,(y_{k+1} - \widehat{y}_k) + \gamma\,(y_{k+1} - \widehat{y}_k) \;|\, w - y_{k+1}\rangle\right] \\
&= \gamma\left[\langle s + Lx_{k+1} \,|\, z - \widetilde{z}_{k+1}\rangle + \langle z - v_{k+1} \,|\, w - y_{k+1}\rangle\right] \\
&\geq -\gamma\varepsilon_{k+1},
\end{aligned}$$

where the latter inequality follows from (33). □

Lemma 2.3. *Consider the sequences evolved by* Algorithm 1 *and let* $\{\widehat{p}_k\}$ *be as in* (27)*. Then, for all* $k \geq 0$ *and* $p = (z, w, s) \in \mathcal{S}$*,*

$$\|p - \widehat{p}_k\|_\gamma^2 - \|p - p_{k+1}\|_\gamma^2 \geq (1 - \sigma^2)\Big(\gamma^2\|Lx_{k+1} - \widehat{y}_k\|^2 + \|y_{k+1} - \widehat{w}_k\|^2\Big).$$

Proof. From the well-known identity $\|a - b\|_\gamma^2 - \|a - c\|_\gamma^2 = \|d - b\|_\gamma^2 - \|d - c\|_\gamma^2 + 2\langle c - b \,|\, a - d\rangle_\gamma$, with $a = p$, $b = \widehat{p}_k$, $c = p_{k+1}$ and $d = \widetilde{p}_{k+1}$, and Lemma 2.2, we find

$$\begin{aligned}
\|p - \widehat{p}_k\|_\gamma^2 - \|p - p_{k+1}\|_\gamma^2 &= \|\widetilde{p}_{k+1} - \widehat{p}_k\|_\gamma^2 - \|\widetilde{p}_{k+1} - p_{k+1}\|_\gamma^2 + 2\langle p_{k+1} - \widehat{p}_k \,|\, p - \widetilde{p}_{k+1}\rangle_\gamma \\
&\geq \|\widetilde{p}_{k+1} - \widehat{p}_k\|_\gamma^2 - \|\widetilde{p}_{k+1} - p_{k+1}\|_\gamma^2 - 2\gamma\varepsilon_{k+1}.
\end{aligned} \tag{34}$$

In view of the second definition in (27), (28) and (29), we also find

$$\begin{aligned}
\|\widetilde{p}_{k+1} - \widehat{p}_k\|_\gamma^2 &= \|\widetilde{z}_{k+1} - \widehat{z}_k\|^2 + \|\widetilde{w}_{k+1} - \widehat{w}_k\|^2 + \gamma^2\|\widetilde{y}_{k+1} - \widehat{y}_k\|^2 \\
&= \gamma^2\|Lx_{k+1} - \widehat{y}_k\|^2 + \|y_{k+1} - \widehat{w}_k\|^2 + \gamma^2\|y_{k+1} - \widehat{y}_k\|^2.
\end{aligned} \tag{35}$$

Now using (28), (29), (25) and (26), we obtain

$$\begin{aligned}\|\widetilde{p}_{k+1}-p_{k+1}\|_\gamma^2 &= \|\widetilde{z}_{k+1}-z_{k+1}\|^2+\|\widetilde{w}_{k+1}-w_{k+1}\|^2+\gamma^2\|\widetilde{y}_{k+1}-y_{k+1}\|^2\\ &= \gamma^2\|y_{k+1}-\widehat{y}_k\|^2+\|y_{k+1}-\widehat{w}_k-\gamma\left(z_{k+1}-v_{k+1}\right)\|^2+\gamma^2\|y_{k+1}-y_{k+1}\|^2\\ &= \gamma^2\|y_{k+1}-\widehat{y}_k\|^2+\|\gamma v_{k+1}-\gamma\underbrace{\left[\widehat{z}_k+\gamma\left(Ax_{k+1}-y_{k+1}\right)\right]}_{z_{k+1}}+y_{k+1}-\widehat{w}_k\|^2\\ &= \gamma^2\|y_{k+1}-\widehat{y}_k\|^2+\|e_{k+1}\|^2. \end{aligned} \tag{36}$$

Direct use of (35), (36) and (24) yields

$$\begin{aligned}\|\widetilde{p}_{k+1}-\widehat{p}_k\|_\gamma^2-\|\widetilde{p}_{k+1}-p_{k+1}\|_\gamma^2-2\gamma\varepsilon_{k+1} &= \gamma^2\|Lx_{k+1}-\widehat{y}_k\|^2+\|y_{k+1}-\widehat{w}_k\|^2-\left[\|e_{k+1}\|^2+2\gamma\varepsilon_{k+1}\right]\\ &\geq (1-\sigma^2)\Big(\gamma^2\|Lx_{k+1}-\widehat{y}_k\|^2+\|y_{k+1}-\widehat{w}_k\|^2\Big),\end{aligned}$$

which, in turn, when combined with (34) finishes the proof of the lemma. □

Lemma 2.4. *Consider the sequences evolved by* Algorithm 1 *and, for an arbitrary* $p=(z,w,s)\in\mathcal{S}$, *define*

$$h_k=\|p_k-p\|_\gamma^2 \qquad \forall k\geq -1. \tag{37}$$

Then $h_0=h_{-1}$ *and, for all* $k\geq 0$,

$$h_{k+1}-h_k-\alpha_k(h_k-h_{k-1})+(1-\sigma^2)\Big(\gamma^2\|Lx_{k+1}-\widehat{y}_k\|^2+\|y_{k+1}-\widehat{w}_k\|^2\Big)\leq\alpha_k(1+\alpha_k)\|p_k-p_{k-1}\|_\gamma^2,$$

i.e., $\{h_k\}$ *satisfies the assumptions of* Lemma A.1 *where, for all* $k\geq 0$,

$$s_{k+1}:=(1-\sigma^2)\Big(\gamma^2\|Lx_{k+1}-\widehat{y}_k\|^2+\|y_{k+1}-\widehat{w}_k\|^2\Big), \tag{38}$$

$$\delta_k:=\alpha_k(1+\alpha_k)\|p_k-p_{k-1}\|_\gamma^2. \tag{39}$$

Proof. From (17)–(19) and the definition of $\widehat{p}_k$ as in (27), we obtain

$$\widehat{p}_k=p_k+\alpha_k(p_k-p_{k-1}), \tag{40}$$

which is clearly equivalent to

$$p_k-p=\frac{1}{1+\alpha_k}\left(\widehat{p}_k-p\right)+\frac{\alpha_k}{1+\alpha_k}\left(p_{k-1}-p\right).$$

Now using the well-known identity $\|tx+(1-t)y\|_\gamma^2=t\|x\|_\gamma^2+(1-t)\|y\|_\gamma^2-t(1-t)\|x-y\|_\gamma^2$ with $t=1/(1+\alpha_k)$, $x=\widehat{p}_k-p$ and $y=p_{k-1}-p$, we find

$$\|p_k-p\|_\gamma^2=\frac{1}{1+\alpha_k}\|\widehat{p}_k-p\|_\gamma^2+\frac{\alpha_k}{1+\alpha_k}\|p_{k-1}-p\|_\gamma^2-\frac{\alpha_k}{(1+\alpha_k)^2}\|\widehat{p}_k-p_{k-1}\|_\gamma^2,$$

which, in turn, when combined with the fact that $\widehat{p}_k - p_{k-1} = (1+\alpha_k)(p_k - p_{k-1})$ (see (40)) and after some simple algebraic manipulations yields

$$\|\widehat{p}_k - p\|_\gamma^2 = (1+\alpha_k)\underbrace{\|p_k - p\|_\gamma^2}_{h_k} - \alpha_k \underbrace{\|p_{k-1} - p\|_\gamma^2}_{h_{k-1}} + \alpha_k(1+\alpha_k)\|p_k - p_{k-1}\|_\gamma^2.$$

The desired result now follows from the above displayed equation, Lemma 2.3 and the definition of h_k in (37). □

Next we present our main result on the asymptotic behavior of Algorithm 1.

Theorem 2.5 (Convergence of Algorithm 1). *Consider the sequences evolved by* Algorithm 1 *and let* $\emptyset \neq \mathcal{S}$ *be as in* (11). *Assume that*

$$\sum_{k=0}^{\infty} \alpha_k \Big(\|z_k - z_{k-1}\|^2 + \|w_k - w_{k-1}\|^2 + \gamma^2 \|y_k - y_{k-1}\|^2 \Big) < \infty. \tag{41}$$

Then there exists $(z_\infty, w_\infty, s_\infty) \in \mathcal{S}$ *such that*

$$z_k \to z_\infty, \quad w_k \to w_\infty \ \text{ and } \ y_k \to -s_\infty. \tag{42}$$

Additionally, we also have

$$v_k \to z_\infty, \quad \widetilde{z}_k \to z_\infty, \quad y_k \to w_\infty \ \text{ and } \ Lx_k \to -s_\infty, \tag{43}$$

where $\widetilde{z}_k$ *is as in* (29).

Proof. We start by making a few remarks. First, from (41), (12), the definition of p_k as in (27) and the fact that $\alpha_k(1+\alpha_k) \leq 2\alpha_k$ (because $0 \leq \alpha_k < 1$), we conclude that $\sum_{k=0}^{\infty} \delta_k < \infty$, where δ_k is as in (39), which, in turn, combined with Lemmas 2.4 and A.1 (below) gives

$$\lim_{k\to\infty} h_k \ \text{ exists and } \ \sum_{k=1}^{\infty} s_k < \infty, \tag{44}$$

where h_k and s_{k+1} are as in (37) and (38), respectively. Using the second statement in (44), (38), (24) and (23), we find

$$Lx_{k+1} - \widehat{y}_k \to 0, \quad y_{k+1} - \widehat{w}_k \to 0, \quad v_{k+1} - \underbrace{[\widehat{z}_k + \gamma(Lx_{k+1} - y_{k+1})]}_{z_{k+1}} \to 0 \ \text{ and } \ \varepsilon_{k+1} \to 0. \tag{45}$$

From the second and third statements in (45), (26) and a standard argument based on the triangle inequality we also find

$$y_{k+1} - w_{k+1} \to 0. \tag{46}$$

Second, from (41) and the fact that $\alpha_k^2 \leq \alpha_k$, we obtain

$$\lim_{k\to\infty} \alpha_k \|z_k - z_{k-1}\| = \lim_{k\to\infty} \alpha_k \|w_k - w_{k-1}\| = \lim_{k\to\infty} \alpha_k \|y_k - y_{k-1}\| = 0,$$

which in turn combined with the definitions of $\widehat{z}_k$, $\widehat{w}_k$ and $\widehat{y}_k$ as in (17)–(19) yields

$$\widehat{z}_k - z_k \to 0, \quad \widehat{w}_k - w_k \to 0 \ \text{ and } \ \widehat{y}_k - y_k \to 0. \tag{47}$$

Third, note that the first definition in (29) and the first statement in (45) yield

$$\widetilde{z}_{k+1} - \widehat{z}_k \to 0. \tag{48}$$

Now, let $p_k = (z_k, w_k, -y_k)$ be as in (27). Note that using the first statement in (44), the definition of h_k as in (37) and Lemma A.2 below, it follows that to prove the convergence of $\{p_k\}$ to some element in $\mathcal{S}$ (and hence the statement in (42)) it suffices to show that every cluster point of $\{p_k\}$ belongs to $\mathcal{S}$. To this end, let $p_\infty = (z_\infty, w_\infty, s_\infty) \in \mathcal{G}^3$ be a cluster point of $\{p_k\}$ (we know from (44) and (37) that $\{p_k\}$ is bounded), i.e., let z_∞, w_∞ and $-s_\infty$ be cluster points of $\{z_k\}$, $\{w_k\}$ and $\{y_k\}$, respectively. That said, let $\{k_j\}$ be an increasing sequence of indexes such that

$$z_{k_j} \to z_\infty, \quad w_{k_j} \to w_\infty \ \text{ and } \ y_{k_j} \to -s_\infty. \tag{49}$$

Direct use of (46) and the second and third statements in (49) give $w_\infty = -s_\infty$, i.e., $s_\infty + w_\infty = 0$. In view of (49) and (47), we also have

$$\widehat{z}_{k_j} \to z_\infty, \quad \widehat{w}_{k_j} \to w_\infty \ \text{ and } \ \widehat{y}_{k_j} \to -s_\infty, \tag{50}$$

which, in particular, when combined with the first and second statements in (45) yields

$$Lx_{k_j+1} \to -s_\infty \quad \text{and} \quad y_{k_j+1} \to w_\infty. \tag{51}$$

From (48) and the first statement in (50) we also have $\widetilde{z}_{k_j+1} \to z_\infty$, which combined with Lemma 2.2(a) (with $k = k_j$), the fact that the graph of $\partial(f^* \circ -L^*)$ is closed and the first statement in (51) yields $s_\infty \in \partial(f^* \circ -L^*)(z_\infty)$. As a consequence, according to the definition of $\mathcal{S}$ as in (11), to prove that $(z_\infty, w_\infty, s_\infty) \in \mathcal{S}$, it remains to verify that $w_\infty \in \partial g^*(z_\infty)$. To this end, recall first that from (22) we know that $y_{k_j+1} \in \partial_{\varepsilon_{k_j+1}} g^*(v_{k_j+1})$, which combined with the second statement in (51), the first statement in (49), the third and fourth statements in (45) as well as with the closedness of the graph of ε-subdifferential of g^* gives the desired result, namely $w_\infty \in \partial g^*(z_\infty)$.

Altogether, we have proved that every cluster point of $\{p_k\}$ belongs to $\mathcal{S}$ and so, as we explained above, it guarantees that $\{p_k\}$ converges to some element in $\mathcal{S}$, i.e., here we finish the proof of (42).

Finally, the proof of (43) follows trivially from (42), (45), (47) and (48). □

We now make a few remarks concerning Theorem 2.5:

(i) A sufficient condition for (41) is as follows: for some $0 < \theta < 1$ and $k_0 \geq 1$,

$$\alpha_k \leq \min\left\{\alpha, \frac{\theta^k}{\|z_k - z_{k-1}\|^2 + \|w_k - w_{k-1}\|^2 + \gamma^2\|y_k - y_{k-1}\|^2}\right\}, \qquad \forall k \geq k_0; \tag{52}$$

here we adopt the convention $1/0 = \infty$.

(ii) As we discussed in the Introduction (following Assumption 1.1), under standard regularity conditions on (1), the result $(z_\infty, w_\infty, s_\infty)$ as in Theorem 2.5, gives that there exists $x_\infty \in \mathcal{H}$ such that $x_\infty \in \partial f^*(-L^* z_\infty)$, $s_\infty = -Lx_\infty$ and x_∞ and z_∞ are solutions of (9) and (10), respectively. Moreover, the third statements in (42) and (43) give, in particular, that $Lx_k - y_k \to 0$.

3 Numerical Experiments

This section presents some numerical experiments on the LASSO and logistic regression problems, which are both instances of the minimization problem (1). We compared the following algorithms: the inexact relative-error ADMM from [44, Algorithm 1] and Algorithm 1 from this paper, which we denote as *admm_inexact* and *admm_inexact_inertial*, respectively. We implemented both algorithms in Python 3.7 and, for both algorithms and all problem classes, used the same stopping criterion, namely

$$\operatorname{dist}_\infty\big(0, \partial f(x_k) + \partial g(x_k)\big) \leq \varepsilon, \tag{53}$$

where $\operatorname{dist}_\infty(0, S) := \inf\{\|s\|_\infty \,|\, s \in S\}$ and ε is a tolerance parameter set to 10^{-6}.

The inertial parameter α_k (as in step 1 of Algorithm 1) is updated according to the rule (52) with $\theta = 0.99$ and $k_0 = 1$. More precisely, we choose α_k as

$$\alpha_k = \min\left\{\alpha, \frac{(0.99)^k}{\|z_k - z_{k-1}\|^2 + \|w_k - w_{k-1}\|^2 + \gamma^2\|y_k - y_{k-1}\|^2}\right\}, \quad \forall k \geq 1,$$

where $0 \leq \alpha < 1$.

3.1 The LASSO problem

In this subsection, we perform numerical experiments on the LASSO problem (as already discussed in (16)), namely

$$\min_{x \in \mathbb{R}^d} \left\{\frac{1}{2}\|Ax - b\|^2 + \nu\|x\|_1\right\}, \tag{54}$$

where $A \in \mathbb{R}^{n \times d}$, $b \in \mathbb{R}^n$ and $\nu > 0$. For the data matrix A and the vector b, we used two categories of non-artificial datasets:

> *Gene expression*: This category consists of six standard cancer DNA microarray datasets from [18], which have dense and wide matrices A, with the number of rows $n \in [42, 102]$ and the number of columns $d \in [2000, 6033]$. These problems are called *brain* (with $n = 42$ and $d = 5597$), *colon* (with $n = 62$ and $d = 2000$), *leukemia* (with $n = 72$ and $d = 3571$), *lymphoma* (with $n = 62$ and $d = 4026$), *prostate* (with $n = 102$ and $d = 6033$) and *srbct* (with $n = 63$ and $d = 2308$).
>
> *Single-Pixel camera*: This category consists of four compressed image sensing datasets from [20], which have dense and wide matrices A, with $n \in \{410, 1638\}$ and $d \in \{1024, 4096\}]$. These problems are called *Ball64_singlepixcam* (with $n = 1638$ and $d = 4096$), *Logo64_singlepixcam* (with $n = 1638$ and $d = 4096$), *Mug32_singlepixcam* (with $n = 410$ and $d = 1024$) and *Mug128_singlepixcam* (with $n = 410$ and $d = 1024$).

We implemented both algorithms *admm_inexact* and *admm_inexact_inertial* in Python 3.7, combined with a CG procedure to approximately solve the subproblems (21); see also the fifth remark following Definition 2.1. As usual (see, e.g., [3]), we solved the (easy) subproblem (20) by using the standard-soft thresholding operator. We also set $\alpha = 0.2$, $\sigma = 0.99$ and $\gamma = 1$. Moreover, as in [13],

Table 1: Outer iterations for the LASSO.

Problem	admm_inexact $_{(iteration1)}$	admm_inexact_inertial $_{(iteration2)}$	$\frac{iteration2}{iteration1}$
Ball64_singlepixcam	374	**296**	0.7914
Logo64_singlepixcam	382	**311**	0.8141
Mug32_singlepixcam	311	**246**	0.7909
Mug128_singlepixcam	994	**939**	0.9447
Brain	4227	**3391**	0.8022
Colon	767	**489**	0.6375
Leukemia	833	**671**	0.8055
Lymphoma	1232	**996**	0.8084
Prostate	2806	**2263**	0.8064
srbct	677	**543**	0.8021
Geometric mean	880.21	**701.74**	0.7972

we set the regularization parameter ν as $0.1\|A^T b\|_\infty$, and scaled the vector b and the columns of matrix A to have unit l_2-norm.

Table 1 shows the number of outer iterations required by each algorithm on each problem instance. Table 2 shows the cumulative total number of inner iterations required by the CG algorithm for solving (21). Table 3 shows runtimes in seconds demanded by each algorithm to achieve the prescribed tolerance as in (53). Figure 1 shows the same results graphically.

3.2 The logistic regression problem

This subsection presents numerical experiments on the ℓ_1–regularized logistic regression problem [26, 36]

$$\min_{(w,v)\in\mathbb{R}^{n-1}\times\mathbb{R}} \left\{ \sum_{i=1}^{q} \log\left(1+\exp(-b_i(a_i^T w+v))\right) + \nu\|w\|_1 \right\}, \tag{55}$$

using training datasets consisting of q pairs (a_i, b_i), where $a_i \in \mathbb{R}^{n-1}$ is a feature vector, $b_i \in \{-1, +1\}$ is the corresponding label, $w \in \mathbb{R}^{n-1}$ represents a weighting of the feature and $v \in \mathbb{R}$ represents a kind of bias.

Problem (55) is clearly a special instance of (1) with $x = (v, w)$, L the identity operator and

$$f(v,w) := \nu\|w\|_1 \quad \text{and} \quad g(v,w) := \sum_{i=1}^{q} \log\left(1+\exp(-b_i(a_i^T w+v))\right). \tag{56}$$

We solved the (easy) subproblem (20) by using the standard soft-thresholding operator. Analogously to [3, 23], to approximately solve the second subproblem corresponding to $g(\cdot)$, namely (21), we used the limited-memory BFGS (L-BFGS) method.

We selected three cancer DNA microarray non-artificial datasets from the *Gene expression* collection, as described in Subsection 3.1, for which $b_i \in \{-1, 1\}$ for $i = 1, \ldots, n$. In addition, we also

Table 2: Total inner iterations for the LASSO.

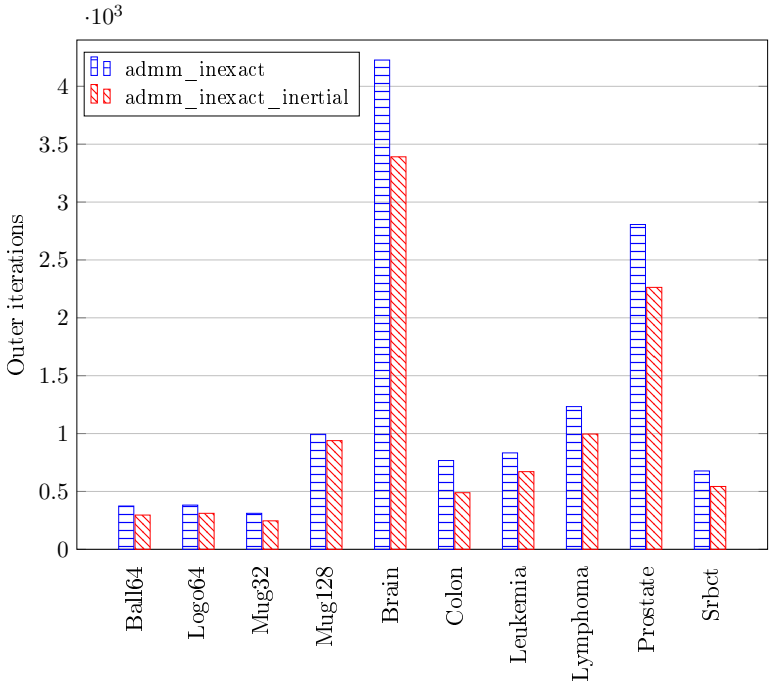
Problem	admm_inexact ($iteration1$)	admm_inexact_inertial ($iteration2$)	$\frac{iteration2}{iteration1}$
Ball64_singlepixcam	801	**620**	0.7741
Logo64_singlepixcam	817	**652**	0.7981
Mug32_singlepixcam	622	**432**	0.6945
Mug128_singlepixcam	2054	**1911**	0.9304
Brain	35848	**19720**	0.5501
Colon	6257	**3353**	0.5359
Leukemia	7421	**4631**	0.6241
Lymphoma	13706	**8194**	0.5978
Prostate	23552	**14562**	0.6183
srbct	5715	**3748**	0.6558
Geometric mean	4374.71	**2924.09**	0.6684

Table 3: Runtimes in seconds for the LASSO.

Problem	admm_inexact ($iteration1$)	admm_inexact_inertial ($iteration2$)	$\frac{iteration2}{iteration1}$
Ball64_singlepixcam	0.0648	**0.0598**	0.9228
Logo64_singlepixcam	0.0578	**0.0448**	0.7751
Mug32_singlepixcam	0.0019	**0.0009**	0.4737
Mug128_singlepixcam	0.5505	**0.4489**	0.8154
Brain	0.0163	**0.0126**	0.7731
Colon	0.0069	**0.0049**	0.7101
Leukemia	0.0119	**0.0098**	0.8235
Lymphoma	0.0139	**0.0109**	0.7842
Prostate	0.0195	**0.0121**	0.6205
srbct	0.0089	**0.0079**	0.8876
Geometric mean	0.0204	**0.0153**	0.7478

Figure 1: Comparison of performance in LASSO problems

(a) Outer iterations



(b) Total inner iterations

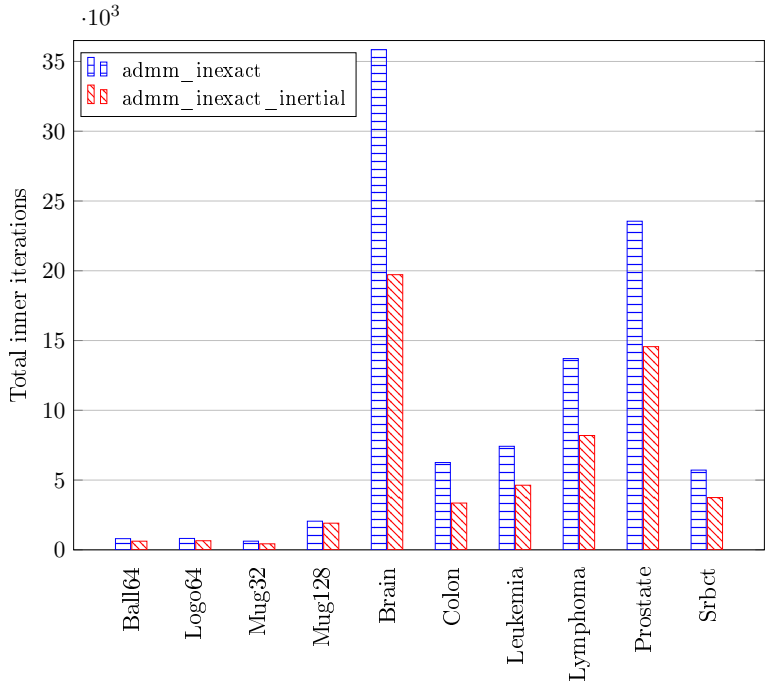


Table 4: Outer iterations for logistic regression problems.

Problem	admm_inexact ($iteration1$)	admm_inexact_inertial ($iteration2$)	$\frac{iteration2}{iteration1}$
Colon	4343	**2936**	0.6761
Leukemia	2231	**1562**	0.7001
Prostate	3100	**2124**	0.6852
Arcene	419	**301**	0.7184
a9a	2310	**1524**	0.6597
Geometric mean	1961.98	**1349.05**	0.6874

Table 5: Total inner iterations for logistic regression problems.

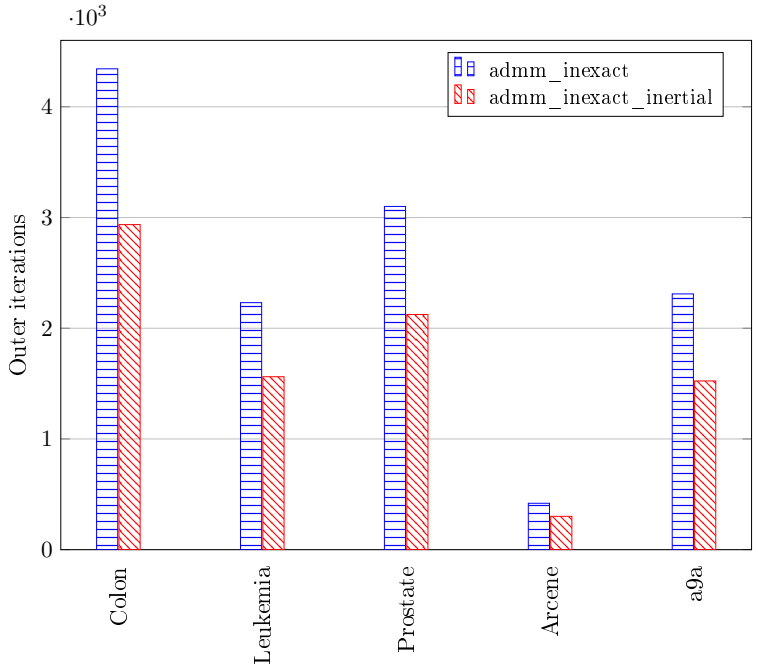
Problem	admm_inexact ($iteration1$)	admm_inexact_inertial ($iteration2$)	$\frac{iteration2}{iteration1}$
Colon	19344	**14218**	0.7351
Leukemia	11493	**7235**	0.6295
Prostate	26637	**14845**	0.5603
Arcene	2683	**2297**	0.8561
a9a	17201	**10987**	0.6387
Geometric mean	12227.22	**8263.79**	0.6758

used the a9a ($n = 32561$ and $d = 123$) and Arcene ($n = 900$ and $d = 10000$) datasets from [24] and [30], respectively. Both of these datasets are sparse and are available from the UCI Machine Learning Repository [19] (where a9a is called adult). We set the regularization parameter ν as $0.1\|A^T b\|_\infty$ and scaled the columns of matrix A to have unit l_2-norm. We also set $\alpha = 0.36$, $\sigma = 0.99$ and $\gamma = 1$.

Tables 4, 5 and 6 show the outer iterations, the total inner iterations and the runtimes, respectively. These results are also graphically summarized in Figure 2.

Figure 2: Comparison of performance in logistic regression problems

(a) Outer iterations



(b) Total inner iterations

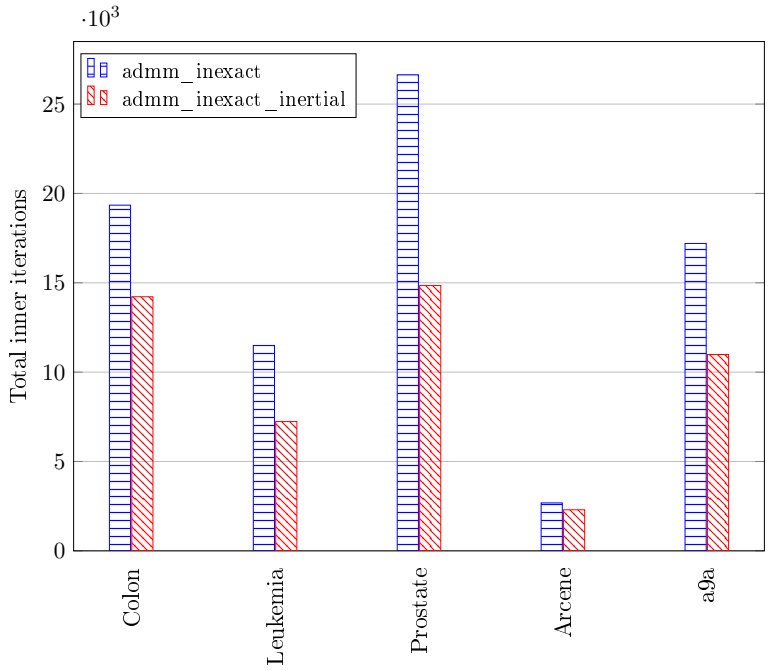


Table 6: Runtimes in seconds for logistic regression problems.

Problem	admm_inexact $_{(iteration1)}$	admm_inexact_inertial $_{(iteration2)}$	$\frac{iteration2}{iteration1}$
Colon	0.0149	**0.0105**	0.7047
Leukemia	0.0369	**0.0178**	0.4824
Prostate	0.0596	**0.0467**	0.7836
Arcene	0.2139	**0.2007**	0.9383
a9a	2.8205	**1.0811**	0.3833
Geometric mean	0.1146	**0.0717**	0.6256

A Auxiliary results

The following lemma was essentially proved by Alvarez and Attouch in [2, Theorem 2.1] (see also [4, Lemma A.4]).

Lemma A.1. *Let the sequences $\{h_k\}$, $\{s_k\}$, $\{\alpha_k\}$ and $\{\delta_k\}$ in $[0,\infty)$ and $\alpha \in \mathbb{R}$ be such that $h_0 = h_{-1}$, $0 \leq \alpha_k \leq \alpha < 1$ and*

$$h_{k+1} - h_k + s_{k+1} \leq \alpha_k(h_k - h_{k-1}) + \delta_k \qquad \forall k \geq 0. \tag{57}$$

The following hold:

(a) *For all $k \geq 1$,*

$$h_k + \sum_{j=1}^{k} s_j \leq h_0 + \frac{1}{1-\alpha}\sum_{j=0}^{k-1} \delta_j. \tag{58}$$

(b) *If $\sum_{k=0}^{\infty} \delta_k < \infty$, then $\lim_{k\to\infty} h_k$ exists, i.e., the sequence $\{h_k\}$ converges to some element in $[0,\infty)$.*

Lemma A.2 (Opial [38])**.** *Let $\mathcal{H}$ be a finite dimensional inner product space, let $\emptyset \neq \mathcal{S} \subset \mathcal{H}$ and let $\{p_k\}$ be a sequence in $\mathcal{H}$ such that every cluster point of $\{p_k\}$ belongs to $\mathcal{S}$ and $\lim_{k\to\infty} \|p_k - p\|$ exists for every $p \in \mathcal{S}$. Then $\{p_k\}$ converges to a point in $\mathcal{S}$.*

References

[1] V. A. Adona, M. L. N. Gonçalves, and J. G. Melo. A partially inexact proximal alternating direction method of multipliers and its iteration-complexity analysis. *J. Optim. Theory Appl.*, 182(2):640–666, 2019.

[2] F. Alvarez and H. Attouch. An inertial proximal method for maximal monotone operators via discretization of a nonlinear oscillator with damping. *Set-Valued Anal.*, 9(1-2):3–11, 2001.

[3] M. M. Alves, J. Eckstein, M. Geremia, and J.G. Melo. Relative-error inertial-relaxed inexact versions of Douglas-Rachford and ADMM splitting algorithms. *Comput. Optim. Appl.*, 75(2):389–422, 2020.

[4] M. M. Alves and R. T. Marcavillaca. On inexact relative-error hybrid proximal extragradient, forward-backward and Tseng's modified forward-backward methods with inertial effects. *Set-Valued Var. Anal.*, 28(2):301–325, 2020.

[5] H. Attouch. Fast inertial proximal ADMM algorithms for convex structured optimization with linear constraint. *Minimax Theory Appl.*, 6(1):1–24, 2021.

[6] H. Attouch and A. Cabot. Convergence of a relaxed inertial proximal algorithm for maximally monotone operators. *Math. Program.*, 184(1-2, Ser. A):243–287, 2020.

[7] H. Attouch, A. Cabot, Z. Chbani, and H. Riahi. Inertial forward-backward algorithms with perturbations: application to Tikhonov regularization. *J. Optim. Theory Appl.*, 179(1):1–36, 2018.

[8] H. Attouch and J. Peypouquet. Convergence of inertial dynamics and proximal algorithms governed by maximally monotone operators. *Math. Program.*, 174(1-2, Ser. B):391–432, 2019.

[9] H. Attouch and M. Soueycatt. Augmented Lagrangian and proximal alternating direction methods of multipliers in Hilbert spaces. Applications to games, PDE's and control. *Pac. J. Optim.*, 5(1):17–37, 2009.

[10] H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC. Springer, New York, 2011.

[11] M. Benning, F. Knoll, C.-B. Schönlieb, and T. Valkonen. Preconditioned ADMM with nonlinear operator constraint. In Lorena Bociu, Jean-Antoine Désidéri, and Abderrahmane Habbal, editors, *System Modeling and Optimization*, pages 117–126, Cham, 2016. Springer International Publishing.

[12] R. I. Boţ and E. R. Csetnek. ADMM for monotone operators: convergence analysis and rates. *Adv. Comput. Math.*, 45(1):327–359, 2019.

[13] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, 2011.

[14] K. Bredies and H. Sun. A proximal point analysis of the preconditioned alternating direction method of multipliers. *J. Optim. Theory Appl.*, 173(3):878–907, 2017.

[15] L. Chen, X. Li, D. Sun, and K.-C. Toh. On the equivalence of inexact proximal ALM and ADMM for a class of convex composite programming. *Math. Program.*, 185(1-2, Ser. A):111–161, 2021.

[16] P. L. Combettes and L. E. Glaudin. Quasi-nonexpansive iterations on the affine hull of orbits: from Mann's mean value algorithm to inertial methods. *SIAM J. Optim.*, 27(4):2356–2380, 2017.

[17] W. Deng and W. Yin. On the global and linear convergence of the generalized alternating direction method of multipliers. *J. Sci. Comput.*, 66(3):889–916, 2016.

[18] M. Dettling and P. Bühlmann. Finding predictive gene groups from microarray data. *J. Multivariate Anal.*, 90(1):106–131, 2004.

[19] D. Dua and C. Graff. UCI machine learning repository, 2017.

[20] M. F. Duarte, M. A. Davenport, Dharmpal T., J. N. Laska, Ting S., K. F. Kelly, and R. G. Baraniuk. Single-pixel imaging via compressive sampling: Building simpler, smaller, and less-expensive digital cameras. *IEEE Signal Processing Magazine*, 25(2):83–91, 2008.

[21] J. Eckstein and D. P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Program.*, 55(3):293–318, 1992.

[22] J. Eckstein and W. Yao. Relative-error approximate versions of Douglas-Rachford splitting and special cases of the ADMM. *Math. Program.*, 170(2, Ser. A):417–444, 2018.

[23] J. Eckstein and W. Yao. Relative-error approximate versions of Douglas-Rachford splitting and special cases of the ADMM. *Math. Program.*, 170(2):417–444, 2018.

[24] R.-E. Fan, P.-H. Chen, and C.-J. Lin. Working set selection using second order information for training support vector machines. *Journal of Machine Learning Research*, 6:1889–1918, 12 2005.

[25] M. Fortin and R. Glowinski. On decomposition-coordination methods using an augmented Lagrangian. In M. Fortin and R. Glowinski, editors, *Augmented Lagrangian methods: Applications to the numerical solution of boundary-value problems*, volume 15 of *Studies in Mathematics and its Applications*, pages 97–146. North-Holland, Amsterdam, 1983.

[26] J. Franklin. The elements of statistical learning: data mining, inference and prediction. *Math. Intelligencer*, 27(2):83–85, 2005.

[27] D. Gabay. Applications of the method of multipliers to variational inequalities. In M. Fortin and R. Glowinski, editors, *Augmented Lagrangian methods: Applications to the numerical solution of boundary-value problems*, volume 15 of *Studies in Mathematics and its Applications*, pages 299–331. North-Holland, Amsterdam, 1983.

[28] R. Glowinski and A. Marroco. Sur l'approximation, par éléments finis d'ordre 1, et la résolution, par pénalisation-dualité, d'une classe de problèmes de Dirichlet non linéaires. *C. R. Acad. Sci. Paris Sér. A*, 278:1649–1652, 1974.

[29] R. Glowinski, S. J. Osher, and W. Yin, editors. *Splitting methods in communication, imaging, science, and engineering*. Scientific Computation. Springer, Cham, 2016.

[30] I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror. Result analysis of the nips 2003 feature selection challenge. volume 17, 01 2004.

[31] W. W. Hager and H. Zhang. Convergence rates for an inexact ADMM applied to separable convex optimization. *Comput. Optim. Appl.*, 77(3):729–754, 2020.

[32] B. He and X. Yuan. On the $O(1/n)$ convergence rate of the Douglas-Rachford alternating direction method. *SIAM J. Numer. Anal.*, 50(2):700–709, 2012.

[33] D. A. Lorenz and Q. Tran-Dinh. Non-stationary Douglas-Rachford and alternating direction method of multipliers: adaptive step-sizes and convergence. *Comput. Optim. Appl.*, 74(1):67–92, 2019.

[34] R. D. C. Monteiro and B. F. Svaiter. On the complexity of the hybrid proximal extragradient method for the iterates and the ergodic mean. *SIAM J. Optim.*, 20(6):2755–2787, 2010.

[35] R. D. C. Monteiro and B. F. Svaiter. Iteration-complexity of block-decomposition algorithms and the alternating direction method of multipliers. *SIAM J. Optim.*, 23(1):475–507, 2013.

[36] A. Y. Ng. Feature selection, L_1 vs. L_2 regularization, and rotational invariance. In *Proceedings, Twenty-First International Conference on Machine Learning, ICML*, pages 615–622, 2004.

[37] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, second edition, 2006.

[38] Z. Opial. Weak convergence of the sequence of successive approximations for nonexpansive mappings. *Bull. Amer. Math. Soc.*, 73:591–597, 1967.

[39] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, NJ, 1970.

[40] 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.*, 24(1):269–297, 2014.

[41] M. V. Solodov and B. F. Svaiter. A hybrid approximate extragradient-proximal point algorithm using the enlargement of a maximal monotone operator. *Set-Valued Anal.*, 7(4):323–345, 1999.

[42] M. V. Solodov and B. F. Svaiter. A hybrid projection-proximal point algorithm. *J. Convex Anal.*, 6(1):59–70, 1999.

[43] M. V. Solodov and B. F. Svaiter. A unified framework for some inexact proximal point algorithms. *Numer. Funct. Anal. Optim.*, 22(7-8):1013–1035, 2001.

[44] B. F. Svaiter. A partially inexact ADMM with $o(1/n)$ asymptotic convergence rate, $\mathcal{O}(1/n)$ complexity, and immediate relative error tolerance. *Optimization*, 70(10):2061–2080, 2021.

[45] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B*, 58(1):267–288, 1996.

[46] J. Xie. On inexact ADMMs with relative error criteria. *Comput. Optim. Appl.*, 71(3):743–765, 2018.

[47] J. Xie, A. Liao, and X. Yang. An inexact alternating direction method of multipliers with relative error criteria. *Optim. Lett.*, 11(3):583–596, 2017.