Noname manuscript No.
(will be inserted by the editor)

A Second-Order Method for Strongly Convex ℓ_1-Regularization Problems

Kimon Fountoulakis and Jacek Gondzio

the date of receipt and acceptance should be inserted later

Abstract In this paper a robust second-order method is developed for the solution of strongly convex ℓ_1-regularized problems. The main aim is to make the proposed method as inexpensive as possible, while even difficult problems can be efficiently solved. The proposed approach is a primal-dual Newton Conjugate Gradients (pdNCG) method. Convergence properties of pdNCG are studied and worst-case iteration complexity is established. Numerical results are presented on synthetic sparse least-squares problems and real world machine learning problems.

Keywords ℓ_1-regularization · Strongly convex optimization · Second-order methods · Iteration Complexity · Newton Conjugate-Gradients method

Mathematics Subject Classification (2000) 68W40, 65K05, 90C06, 90C25, 90C30, 90C51

1 Introduction

We are concerned with the solution of the following optimization problem

$$\text{minimize } f_\tau(x) := \tau\|x\|_1 + \varphi(x), \tag{1}$$

J. Gondzio is supported by EPSRC Grant EP/I017127/1

Kimon Fountoulakis
School of Mathematics and Maxwell Institute, The University of Edinburgh, Peter Guthrie Tait Road, Edinburgh EH9 3FD, United Kingdom.
E-mail: K.Fountoulakis@sms.ed.ac.uk
Tel.: +44 131 650 5083

Jacek Gondzio
School of Mathematics and Maxwell Institute, The University of Edinburgh, Peter Guthrie Tait Road, Edinburgh EH9 3FD, United Kingdom.
E-mail: J.Gondzio@ed.ac.uk
Tel.: +44 131 650 8574, Fax: +44 131 650 6553

where $x \in \mathbb{R}^m$, $\tau > 0$ and $\|\cdot\|_1$ is the ℓ_1-norm. The following three assumptions are made.

- The function $\varphi(x)$ is twice differentiable, and
- strongly convex everywhere, which implies that at any x its second derivative $\nabla^2\varphi(x)$ is uniformly bounded

$$\lambda_m I_m \preceq \nabla^2\varphi(x) \preceq \lambda_1 I_m, \tag{2}$$

  with $0 < \lambda_m \leq \lambda_1$, where I_m is the $m \times m$ identity matrix.
- The second derivative of $\varphi(x)$ is Lipschitz continuous

$$\|\nabla^2\varphi(y) - \nabla^2\varphi(x)\| \leq L_\varphi \|y - x\|, \tag{3}$$

  for any x, y, where $L_\varphi \geq 0$ is the Lipschitz constant, $\|\cdot\|$ represents the Euclidean distance for vectors and the spectral norm for matrices.

A variety of problems originating from the "new" economy including Big-Data [21], Machine Learning [24] and Regression [25] problems to mention a few can be cast in the form of (1). Such problems usually consist of large-scale data, which frequently impose restrictions on methods that have been so far employed. For instance, the new methods have to be memory efficient and ideally, within seconds they should offer noticeable progress in reducing the objective function. First-order methods meet some of these requirements. They avoid matrix factorizations, which implies low memory requirements, additionally, they sometimes offer fast progress in the initial stages of optimization. Unfortunately, as demonstrated by numerical experiments presented later in this paper, first-order methods miss essential information about the conditioning of the problems, which might result in slow practical convergence. The main advantage of first-order methods, which is to rely only on simple gradient or coordinate updates becomes their essential weakness.

We do not think this inherent weakness of first-order methods can be remedied. For this reason, in this paper, a second-order method is used instead, i.e.,, a primal-dual Newton Conjugate Gradients. The optimization community seems to consider the second-order methods to be rather expensive. The main aim in this paper is to make the proposed method as *inexpensive* as possible, while even complicated problems can be efficiently solved. To accomplish this, pdNCG is used in a matrix-free environment i.e.,, Conjugate Gradients is used to compute inexact Newton directions. No matrix factorization is performed and no excessive memory requirements are needed. Consequently, the main drawbacks of Newton method are removed, while at the same time their fast convergence properties are provably retained. In order to meet this goal, the ℓ_1-norm is approximated by a smooth function, which has derivatives of all degrees. Hence, problem (1) is replaced by

$$\text{minimize } f_\tau^\mu(x) := \tau\psi_\mu(x) + \varphi(x). \tag{4}$$

where $\psi_\mu(x)$ denotes the smooth function, which substitutes the ℓ_1-norm and μ is a parameter that controls the quality of approximation. Smoothing will

allow access to second-order information and essential curvature information will be exploited.

On the theoretical front we show that the analysis of pdNCG can be performed in a variable metric using an important property of CG. The variable metric is the standard Euclidean norm scaled by an approximation of the second-order derivative at every iteration of pdNCG. Based on the variable metric we give a complete analysis of pdNCG, i.e., proof of global convergence, global and local convergence rates, local region of fast convergence rate and worst-case iteration complexity.

In what follows in this section we give a brief introduction of the smoothing technique. In Section 2, necessary basic results are given, which will be used to support theoretical results in Section 4. In Section 3, the proposed pdNCG method is described in details. In Section 4, the convergence analysis and worst-case iteration complexity of pdNCG is studied. In Section 5, numerical results are presented.

1.1 Pseudo-Huber regularization

The non-smoothness of the ℓ_1-norm prevents a straightforward application of the second-order method to problem (1). In this subsection, we focus on approximating the non-smooth ℓ_1-norm by a smooth function. To meet such a goal, the first-order methods community replaces the ℓ_1-norm with the so-called Huber penalty function $\sum_{i=1}^{m} \phi_\mu(x_i)$ [4], where

$$\phi_\mu(x_i) = \begin{cases} \frac{1}{2}\frac{x_i^2}{\mu}, & \text{if } |x_i| \le \mu \\ |x_i| - \frac{1}{2}\mu, & \text{if } |x_i| \ge \mu \end{cases} \quad i = 1, 2, \ldots, m$$

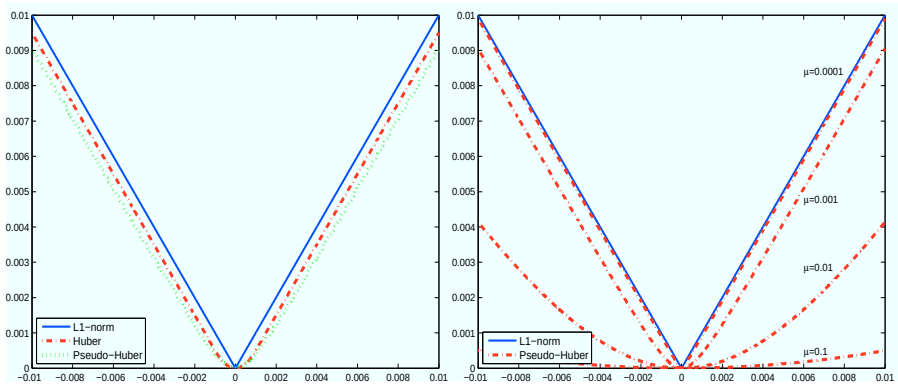
and $\mu > 0$. The smaller the parameter μ of the Huber function is, the better the function approximates the ℓ_1-norm. Observe that the Huber function is only first-order differentiable, therefore, this approximation trick is not applicable to second-order methods. Fortunately, there is a smooth version of the Huber function, the pseudo-Huber function, which has derivatives of all degrees [11]. The pseudo-Huber function parameterized with $\mu > 0$ is

$$\psi_\mu(x) = \sum_{i=1}^{m} \Big((\mu^2 + x_i^2)^{\frac{1}{2}} - \mu \Big). \tag{5}$$

A comparison of the three functions ℓ_1-norm, Huber and Pseudo-Huber function can be seen in Figure 1.

The advantages of such an approach are listed below.

- Availability of second-order information owed to the differentiability of the pseudo-Huber function.
- Opening the door to using iterative methods to compute descent directions, which take into account the curvature of the problem, such as CG.



(a) ℓ_1-norm, Huber, Pseudo-Huber functions **(b)** Pseudo-Huber function for $\mu \to 0$

Fig. 1 Comparison of the approximation functions, Huber and pseudo-Huber, with the ℓ_1-norm in one dimensional space. **Fig.1a** shows the quality of approximation for the Huber and pseudo-Huber functions. **Fig.1b** shows how pseudo-Huber function converges to the ℓ_1-norm as $\mu \to 0$

There is an obvious cost that comes along with the above benefits, and that is the approximate nature of the pseudo-Huber function. There is a concern that in case that a very accurate solution is required, the pseudo-Huber function may be unable to deliver it. In theory, since the quality of the approximation is controlled by parameter μ in (5), see Figure 1, the pseudo-Huber function can recover any level of accuracy under the condition that sufficiently small μ is chosen. The reader is referred to [1] for a perturbation analysis when the ℓ_1-norm is replaced with the Pseudo-Huber function. In practise a very small parameter μ might worsen the conditioning of the linear algebra of the solver. However, we shall provide numerical evidence that even when μ is set to small values, the proposed method behaves well and remains very efficient.

2 Preliminaries

The $\|\cdot\|_\infty$ denotes the infinity norm. The operator $diag(\cdot)$ takes as input a vector and creates a diagonal matrix with the input vector on the diagonal. The operator $[\cdot]_{ij}$ returns the element at row i and column j of the input matrix, similarly, the operator $[\cdot]_i$ returns the element at position i of the input vector.

2.1 Properties of pseudo-Huber function

The gradient of the pseudo-Huber function $\psi_\mu(x)$ in (5) is given by

$$\nabla\psi_\mu(x) = \left[x_1\left(\mu^2 + x_1^2\right)^{-\frac{1}{2}}, \ldots, x_m\left(\mu^2 + x_m^2\right)^{-\frac{1}{2}}\right], \tag{6}$$

and the Hessian is given by

$$\nabla^2\psi_\mu(x) = \mu^2 diag\Big(\Big[(\mu^2+x_1^2)^{-\frac{3}{2}},\ldots,(\mu^2+x_m^2)^{-\frac{3}{2}}\Big]\Big). \tag{7}$$

The next lemma guarantees that the Hessian of the pseudo-Huber function $\psi_\mu(x)$ is bounded.

Lemma 1 *The Hessian matrix $\nabla^2\psi_\mu(x)$ satisfies*

$$0I \prec \nabla^2\psi_\mu(x) \preceq \frac{1}{\mu}I$$

where I is the identity matrix in appropriate dimension.

Proof The result follows easily by observing that $0 < (\mu^2+x_i^2)^{-\frac{3}{2}} \leq 1/\mu^3$ for any x_i, $i = 1, 2\ldots, m$. The proof is complete.

The next lemma shows that the Hessian matrix of the pseudo-Huber function is Lipschitz continuous.

Lemma 2 *The Hessian matrix $\nabla^2\psi_\mu(x)$ is Lipschitz continuous*

$$\|\nabla^2\psi_\mu(y) - \nabla^2\psi_\mu(x)\| \leq \frac{1}{\mu^2}\|y-x\|.$$

Proof

$$\begin{aligned}\|\nabla^2\psi_\mu(y) - \nabla^2\psi_\mu(x)\| &= \Big\|\int_0^1 \frac{d\nabla^2\psi_\mu(x+s(y-x))}{ds}ds\Big\| \\ &\leq \int_0^1 \Big\|\frac{d\nabla^2\psi_\mu(x+s(y-x))}{ds}\Big\|ds \end{aligned}\tag{8}$$

where $\frac{d\nabla^2\psi_\mu(x+s(y-x))}{ds}$ is a diagonal matrix with each diagonal component, $i = 1, 2, \ldots m$, given by

$$\Big[\frac{d\nabla^2\psi_\mu(x+s(y-x))}{ds}\Big]_{ii} = \frac{-3\mu^2(x_i+s(y_i-x_i))(y_i-x_i)}{(\mu^2+(x_i+s(y_i-x_i))^2)^{\frac{5}{2}}}.$$

Using the previous observation we have that

$$\Big\|\frac{d\nabla^2\psi_\mu(x+s(y-x))}{ds}\Big\| = \max_{i=1,2,\ldots,m}\Big|\Big[\frac{d\nabla^2\psi_\mu(x+s(y-x))}{ds}\Big]_{ii}\Big| \tag{9}$$

Moreover, we have

$$\begin{aligned}\Big|\Big[\frac{d\nabla^2\psi_\mu(x+s(y-x))}{ds}\Big]_{ii}\Big| &= \Big|\frac{-3\mu^2(x_i+s(y_i-x_i))(y_i-x_i)}{(\mu^2+(x_i+s(y_i-x_i))^2)^{\frac{5}{2}}}\Big| \\ &= \Big|\frac{-3\mu^2(x_i+s(y_i-x_i))}{(\mu^2+(x_i+s(y_i-x_i))^2)^{\frac{5}{2}}}\Big||(y_i-x_i)| \end{aligned}\tag{10}$$

where the first absolute value in (10) has a maximum at $\frac{\mu-2x_i}{2(y_i-x_i)}$, which gives

$$\Big|\frac{-3\mu^2(x_i+s(y_i-x_i))}{(\mu^2+(x_i+s(y_i-x_i))^2)^{\frac{5}{2}}}\Big| \leq \frac{48}{25\sqrt{5}\mu^2} < \frac{1}{\mu^2}. \tag{11}$$

Combining (10) and (11) we get

$$\Big|\Big[\frac{d\nabla^2\psi_\mu(x+s(y-x))}{ds}\Big]_{ii}\Big| \leq \frac{1}{\mu^2}|y_i-x_i|. \tag{12}$$

Replacing (12) in (9) and using the fact that $\|\cdot\|_\infty \leq \|\cdot\|$ we get

$$\Big\|\frac{d\nabla^2\psi_\mu(x+s(y-x))}{ds}\Big\| \leq \frac{1}{\mu^2}\|y-x\|.$$

Replacing the above expression in (8) and calculating the integral we arrive at the desired result. The proof is complete.

The next lemma shows that the gradient of the pseudo-Huber function is Lipschitz continuous.

Lemma 3 *The gradient $\nabla\psi_\mu(x)$ is Lipschitz continuous*

$$\|\nabla\psi_\mu(y)-\nabla\psi_\mu(x)\| \leq \frac{1}{\mu}\|y-x\|.$$

Proof Using the fundamental theorem of calculus, like in proof of Lemma 2, and Lemma 1 it is easy to show the result. The proof is complete.

2.2 Properties of function $f_\tau^\mu(x)$

The gradient of $f_\tau^\mu(x)$ is given by

$$\nabla f_\tau^\mu(x) = \tau\nabla\psi_\mu(x) + \nabla\varphi(x)$$

where $\nabla\psi_\mu(x)$ has been defined in (6). The Hessian matrix of $f_\tau^\mu(x)$ is

$$\nabla^2 f_\tau^\mu(x) = \tau\nabla^2\psi_\mu(x) + \nabla^2\varphi(x).$$

where $\nabla^2\psi_\mu(x)$ has been defined in (7). Using (2) and Lemma 1 we get the following bounds on the Hessian matrix of $f_\tau^\mu(x)$

$$\lambda_m I \prec \nabla^2 f_\tau^\mu(x) \preceq \Big(\frac{\tau}{\mu}+\lambda_1\Big)I, \tag{13}$$

where I is the identity matrix in appropriate dimension.

Lemma 4 *For any x and x^*, the minimizer of $f_\tau^\mu(x)$, the following holds*

$$\frac{1}{2\Big(\frac{\tau}{\mu}+\lambda_1\Big)}\|\nabla f_\tau^\mu(x)\|^2 \leq f_\tau^\mu(x) - f_\tau^\mu(x^*) \leq \frac{1}{2\lambda_m}\|\nabla f_\tau^\mu(x)\|^2$$

and

$$\|x-x^*\| \leq \frac{2}{\lambda_m}\|\nabla f_\tau^\mu(x)\|.$$

Proof The right hand side of the first inequality is proved on page 460 of [6]. The left hand side of the first inequality is proved by using strong convexity of $f_\tau^\mu(x)$,

$$f_\tau^\mu(y) \leq f_\tau^\mu(x) + \nabla f_\tau^\mu(x)^\intercal (y-x) + \frac{\frac{\tau}{\mu}+\lambda_1}{2}\|y-x\|^2$$

and defining $\tilde{y} = x - \frac{1}{\frac{\tau}{\mu}+\lambda_1}\nabla f_\tau^\mu(x)$. We get

$$f_\tau^\mu(x) - f_\tau^\mu(x^*) \geq f_\tau^\mu(x) - f_\tau^\mu(\tilde{y}) \geq \frac{1}{2\left(\frac{\tau}{\mu}+\lambda_1\right)}\|\nabla f_\tau^\mu(x)\|^2.$$

The last inequality is proved on page 460 of [6]. The proof is complete.

The following lemma guarantees that the Hessian matrix $\nabla^2 f_\tau^\mu(x)$ is Lipschitz continuous. In this lemma, L_φ is defined in (3).

Lemma 5 *The function $\nabla^2 f_\tau^\mu(x)$ is Lipschitz continuous*

$$\|\nabla^2 f_\tau^\mu(y) - \nabla^2 f_\tau^\mu(x)\| \leq L_{f_\tau^\mu}\|y-x\|,$$

where $L_{f_\tau^\mu} := \frac{\tau}{\mu^2} + L_\varphi$.

Proof Using Lemma 2 and (3) we have

$$\begin{aligned}\|\nabla^2 f_\tau^\mu(y) - \nabla^2 f_\tau^\mu(x)\| &\leq \tau\|\nabla^2\psi_\mu(y) - \nabla^2\psi_\mu(x)\| + \|\nabla^2\varphi(y) - \nabla^2\varphi(x)\| \\ &\leq \left(\frac{\tau}{\mu^2} + L_\varphi\right)\|y-x\|.\end{aligned}$$

The Lipschitz constant of $\nabla^2 f_\tau^\mu(x)$ is therefore $L_{f_\tau^\mu} := \frac{\tau}{\mu^2} + L_\varphi$.

The next lemma shows how well the second-order Taylor expansion of $f_\tau^\mu(x)$ approximates the function $f_\tau^\mu(x)$.

Lemma 6 *If $q_\tau^\mu(y)$ is a quadratic approximation of the function $f_\tau^\mu(x)$ at x*

$$q_\tau^\mu(y) := f_\tau^\mu(x) + \nabla f_\tau^\mu(x)^\intercal (y-x) + \frac{1}{2}(y-x)^\intercal \nabla^2 f_\tau^\mu(x)(y-x),$$

then

$$|f_\tau^\mu(y) - q_\tau^\mu(y)| \leq \frac{1}{6}L_{f_\tau^\mu}\|y-x\|^3.$$

Proof Using corollary 1.5.3 in [19] and Lemma 5 we have

$$\begin{aligned}|f_\tau^\mu(y) - q_\tau^\mu(y)| &\leq \|y-x\|^2 \int_0^1\int_0^t \|\nabla^2 f_\tau^\mu(x+s(y-x)) - \nabla^2 f_\tau^\mu(x)\| ds dt \\ &\leq \|y-x\|^2 \int_0^1\int_0^t sL_{f_\tau^\mu}\|y-x\| ds dt \\ &= \frac{1}{6}L_{f_\tau^\mu}\|y-x\|^3.\end{aligned}$$

The proof is complete.

2.3 Alternative optimality conditions

The first-order optimality conditions of problem (4) are $\nabla f_\tau^\mu(x) = \tau\nabla\psi_\mu(x) + \nabla\varphi(x) = 0$. Therefore, one could simply apply a Newton-CG method in order to find a root of this equation. However, in a series of papers [7,8] it has been noted that the linearization of $\nabla\psi_\mu(x)$ for Newton-CG method might be a poor approximation of $\nabla\psi_\mu(x)$ close to the optimal solution, hence, the method is misbehaving. This argument is supported with numerical experiments in [8]. it is also worth mentioning that our empirical experience confirms the results of the previous paper. To deal with this problem the authors in [8] suggested to solve a reformulation of the optimality conditions which, for the problems of our interest, is

$$\begin{aligned} \tau y + \nabla\varphi(x) &= 0 \\ D^{-1}y - x &= 0 \\ \|y\|_\infty &\leq 1, \end{aligned} \tag{14}$$

where D is a diagonal matrix with components

$$[D]_{ii} = (\mu^2 + x_i^2)^{-\frac{1}{2}} \ \forall i = 1, 2, \ldots, m. \tag{15}$$

The idea behind this reformulation is that the linearization of the second equations in (14), i.e., $y_i/[D]_{ii} - x_i = 0 \ \forall i = 1, 2, \ldots, m$, is of much better quality than the linearization of $[\nabla\psi_\mu(x)]_i$ for $\mu \approx 0$ and $x_i \approx 0$; a scenario that is unavoidable since for small μ the optimal solution of (4) is expected to be approximately sparse. To see why this is true, observe that for small μ and $x_i \approx 0$, the gradient $[\nabla\psi_\mu(x)]_i$ becomes close to singular and its linearization is expected to be inaccurate. On the other hand, $y_i/[D]_{ii} - x_i$ as a function of x_i is not singular for $\mu \approx 0$ and $x_i \approx 0$, hence, its linearization is expected to be more accurate. Empirical justification for the previous is also given in Section 3 in [8].

In this paper, we follow the same reasoning and solve (14) instead.

2.4 Primal-dual reformulation

In this subsection we show that the alternative optimality conditions (14) correspond to a primal-dual formulation of problem (4). This also explains the primal-dual suffix in the name of the proposed method.

Every i^{th} term $|x_i|_\mu := (\mu^2 + x_i^2)^{1/2} - \mu$ of the pseudo-Huber function in (5) approximates the absolute value $|x_i|$. The terms $|x_i|_\mu \ \forall i$ can be obtained by regularizing the dual formulation of $|x_i| \ \forall i$, i.e.,

$$|x_i| = \sup_{|y_i|\leq 1} x_i y_i \quad \text{where} \quad y_i \in \mathbb{R},$$

with the term $\mu(1 - y_i^2)^{1/2} - \mu$, to obtain

$$|x_i|_\mu = \sup_{|y_i|\leq 1} x_i y_i + \mu(1 - y_i^2)^{\frac{1}{2}} - \mu.$$

Hence, the pseudo-Huber function (5) has the following dual formulation

$$\psi_\mu(x) = \sum_{i=1}^{m} \sup_{|y_i|\leq 1} x_i y_i + \mu(1-y_i^2)^{\frac{1}{2}} - \mu.$$

Therefore, problem (4) is equivalent to the primal-dual formulation

$$\min_x \sup_{\|y\|_\infty \leq 1} \tau(x^\intercal y + \mu \sum_{i=1}^{m} (1-y_i^2)^{\frac{1}{2}} - m\mu) + \varphi(x), \tag{16}$$

where $y \in \mathbb{R}^m$ denotes the dual variables. The first order optimality conditions of the primal-dual formulation (16) are

$$\begin{aligned} \tau y + \nabla\varphi(x) &= 0 \\ x_i - \mu y_i (1-y_i^2)^{-\frac{1}{2}} &= 0 \ \forall i = 1,2,\ldots,m \\ \|y\|_\infty &\leq 1, \end{aligned} \tag{17}$$

It can be easily shown that the first-order optimality conditions (17) of the primal-dual formulation (16) are equivalent to (14).

2.5 A property of Conjugate Gradients algorithm

The following property of CG is used in the convergence analysis of pdNCG.

Lemma 7 *Let $Ax = b$, where A is a symmetric and positive definite matrix. Furthermore, let us assume that this system is solved using CG approximately; CG is terminated prematurely at the ith iteration. Then if CG is initialized with the zero solution the approximate solution x_i satisfies*

$$x_i^\intercal A x_i = x_i^\intercal b.$$

The same result holds when Preconditioned CG (PCG) is used.

Proof The following property is shown in proof of Lemma 2.4.1 in [15]. If CG algorithm is initialised with the zero solution $p_0 = 0$, then it returns a solution x_i, which satisfies

$$x_i := \arg\min_p \{\frac{1}{2} p^\intercal A p - p^\intercal b \mid p \in \mathcal{E}_i\},$$

where

$$\mathcal{E}_i := \text{span}(b, Ab, \ldots, A^{i-1}b).$$

Therefore for every $p \in \mathcal{E}_i$, at $t = 0$, we get

$$\frac{d(\frac{1}{2}(x_i + tp)^\intercal A(x_i + tp) - (x_i + tp)^\intercal b)}{dt} = (Ax_i - b)^\intercal p = 0.$$

Since, $x_i \in \mathcal{E}_i$, then

$$(Ax_i - b)^\intercal x_i = 0 \Longleftrightarrow x_i^\intercal A x_i = x_i^\intercal b.$$

This completes the first part. In case that PCG is employed with symmetric positive definite preconditioner $P = EE^\intercal$, then PCG is equivalent to solving approximately the system $E^{-1}AE^{-\intercal}\xi = E^{-1}b$ using CG and then calculating $x_i = E^{-\intercal}\xi_i$. Therefore, by applying the previous we get that $\xi_i^\intercal E^{-1}AE^{-\intercal}\xi_i = \xi_i^\intercal E^{-1}b$ and by substituting $\xi_i = E^\intercal x_i$ we prove the second part. The proof is complete.

3 Primal-Dual Newton Conjugate Gradients

In this section we describe a variation of Newton-CG, which we name primal-dual Newton-CG (pdNCG), for the solution of the primal-dual optimality conditions (14). The method is similar to the one in [8] for signal reconstruction problems, although, the two approaches differ in step 3 of pdNCG. Additionally, we make a step further and give complete convergence analysis and worst-case iteration complexity results in Section 4. A detailed pseudo-code of the method is given below.

Algorithm pdNCG

1: **Loop:** For $k = 1, 2, ...,$ until $\|d^k\|_{x^k} \leq \epsilon$, where $\epsilon > 0$.

2: Obtain d^k by solving approximately the system

$$H(x^k, y^k)d = -\nabla f_\tau^\mu(x^k) \tag{18}$$

using CG or PCG, where

$$H(x, y) = \tau D(I - Ddiag(x)diag(y)) + \nabla^2\varphi(x) \tag{19}$$

and matrix D is defined in (15). Obtain Δy^k by calculating

$$\Delta y^k = D(I - Ddiag(x)diag(y))d - (y^k - Dx^k). \tag{20}$$

3: Set $\tilde{y}^{k+1} = y^k + \Delta y^k$ and calculate

$$y^{k+1} := P_{\|\cdot\|_\infty \leq 1}(\tilde{y}^{k+1}),$$

where $P_{\|\cdot\|_\infty \leq 1}(\cdot)$ is the orthogonal projection in the ℓ_∞ ball.

4: Find the least integer $j \geq 0$ such that the function $f_\tau^\mu(x)$ is sufficiently decreased along d^k

$$f_\tau^\mu(x^k + c_3^j d^k) \leq f_\tau^\mu(x^k) - c_2 c_3^j \|d^k\|_{x^k}^2,$$

where $0 < c_2 < 1/2$, $0 < c_3 < 1$, and set $\alpha = c_3^j$.

5: Set $x^{k+1} = x^k + \alpha d^k$.

In algorithm pdNCG we make use of the local norm

$$\|\cdot\|_{x^k} := \sqrt{\langle \cdot, H(x^k, y^k)\cdot\rangle}, \tag{21}$$

where $H(x^k, y^k)$ is a positive definite matrix under the condition that $\|y^k\|_\infty \leq 1$ (Lemma 8). Step 2 of pdNCG is the approximate solution of the linearization of the first two equations in (14). The matrix $H(x, y)$ is obtained by simply eliminating the variables Δy^k in the linearized system. Step 2 is performed by CG or PCG, which is always initialized with the zero solution and it is terminated when

$$\|r_\tau^\mu(x, y)\| \leq \eta \|\nabla f_\tau^\mu(x)\|, \tag{22}$$

where $r_\tau^\mu(x, y) = H(x, y)d + \nabla f_\tau^\mu(x)$ is the residual and $0 \leq \eta < 1$ is a user-defined constant. In practice we have observed that setting η^k =1.0e–1 results in very fast convergence, however, the method will be analyzed for η^k set as in

$$\eta^k = \min\{\frac{1}{2}, \|\nabla f_\tau^\mu(x^k)\|^{c_0}\}, \tag{23}$$

with $c_0 = 1$.

Step 3 is a projection of $\tilde{y}^{k+1}$ to the set $\|y\|_\infty \leq 1$ such that feasibility of the third condition in (14) is always maintained. The projection operator is

$$v := P_{\|\cdot\|_\infty \leq 1}(u) = \text{sign}(u)\min(|u|, 1)$$

and it is applied component-wise. Step 4 is a backtracking line-search technique in order to guarantee that the sequence $\{x^k\}$ generated by pdNCG monotonically decreases the objective function $f_\tau^\mu(x)$.

4 Convergence analysis and worst-case iteration complexity

In this section we analyze the pdNCG method. In particular, we prove global convergence, we study the global and local convergence rates and we explicitly define a region in which pdNCG has fast convergence rate. Additionally, worst-case iteration complexity result of pdNCG is presented. The reader will notice that the results in this section are established when CG is used in step 2 of pdNCG. However, based on Lemma 7 it is trivial to show that the same results hold if PCG is used.

Before we introduce notational conventions for this section, it is necessary to find uniform bounds for matrix $H(x, y)$ in (19). This is shown in the following lemma.

Lemma 8 *If $\|y\|_\infty \leq 1$, then matrix $H(x, y)$ is uniformly bounded by*

$$\lambda_m I \prec H(x, y) \preceq \Big(\frac{\tau}{\mu} + \lambda_1\Big) I,$$

where I is the identity matrix of appropriate dimension.

Proof This result easily follows by using the definition of $H(x, y)$ in (19) and (2). A similar argument, but for signal reconstruction problems, is also claimed in [8], page 1970. The proof is complete.

The equivalence of the Euclidean and the local norm (21) if $\|y\|_\infty \leq 1$, is given by the following inequality

$$\lambda_m^{\frac{1}{2}}\|d\| \leq \|d\|_x \leq \left(\frac{\tau}{\mu} + \lambda_1\right)^{\frac{1}{2}}\|d\|. \tag{24}$$

The upper bound of the largest eigenvalue of $H(x, y)$ if $\|y\|_\infty \leq 1$, will be denoted by $\tilde{\lambda}_1 = (\tau/\mu + \lambda_1)$. An upper bound of the condition number of matrix $H(x, y)$ will be denoted by $\kappa = \tilde{\lambda}_1/\lambda_m$. The Lipschitz constant $L_{f_\tau^\mu}$ defined in Lemma 5, will be denoted by L. Finally, the indices τ and μ from function $f_\tau^\mu(x)$ are dropped.

4.1 Global convergence

First, the minimum decrease of the objective function at every iteration of pdNCG is calculated.

Lemma 9 *Let $x \in \mathbb{R}^m$ be the current iteration of pdNCG, $d \in \mathbb{R}^m$ be the pdNCG direction for the primal variables, which is calculated using CG. The parameter η of the termination criterion* (22) *of CG is set to $0 \leq \eta < 1$. If x is not the minimizer of problem* (4)*, i.e., $\nabla f(x) \neq 0$, then the backtracking line-search algorithm in step* 4 *of pdNCG will calculate a step-size $\bar{\alpha}$ such that*

$$\bar{\alpha} \geq c_3 \frac{\lambda_m}{\tilde{\lambda}_1}.$$

For this step-size $\bar{\alpha}$ the following holds

$$f(x) - f(x(\bar{\alpha})) > c_4\|d\|_x^2,$$

where $c_4 = c_2 c_3 \frac{1}{\kappa}$ and $x(\bar{\alpha}) = x + \bar{\alpha}d$.

Proof For $x(\alpha) = x + \alpha d$ and from smoothness of $f(x)$ we have

$$f(x(\alpha)) \leq f(x) + \alpha\nabla f(x)^\intercal d + \frac{\alpha^2}{2}\tilde{\lambda}_1\|d\|^2.$$

From Lemma 8 we have that $H(x, y)$ is positive definite if $\|y\|_\infty \leq 1$, which is the condition that always satisfied by step 3 of pdNCG. Then, if $\nabla f(x) \neq 0$ the CG algorithm terminated at the ith iteration returns the vector $d_i \neq 0$, which according to Lemma 7 satisfies

$$d_i^\intercal H(x, y)d_i = -d_i^\intercal \nabla f(x).$$

Therefore, by setting $d := d_i$ we get

$$f(x(\alpha)) \leq f(x) - \alpha\|d\|_x^2 + \frac{\alpha^2}{2}\tilde{\lambda}_1\|d\|^2.$$

Using (24) we get

$$f(x(\alpha)) \leq f(x) - \alpha\|d\|_x^2 + \frac{\alpha^2}{2}\frac{\tilde{\lambda}_1}{\lambda_m}\|d\|_x^2.$$

The right hand side of the above inequality is minimized for $\alpha^* = \frac{\lambda_m}{\tilde{\lambda}_1}$, which gives

$$f(x(\bar{\alpha})) \leq f(x) - \frac{1}{2}\frac{\lambda_m}{\tilde{\lambda}_1}\|d\|_x^2.$$

Observe that for this step-size the exit condition of the backtracking line-search algorithm is satisfied, since

$$f(x(\bar{\alpha})) \leq f(x) - \frac{1}{2}\frac{\lambda_m}{\tilde{\lambda}_1}\|d\|_x^2 < f(x) - c_2\frac{\lambda_m}{\tilde{\lambda}_1}\|d\|_x^2.$$

Therefore the step-size $\bar{\alpha}$ returned by the backtracking line-search algorithm is in worst-case bounded by

$$\bar{\alpha} \geq c_3\frac{\lambda_m}{\tilde{\lambda}_1},$$

which results in the following decrease of the objective function

$$f(x) - f(x(\bar{\alpha})) > c_2c_3\frac{\lambda_m}{\tilde{\lambda}_1}\|d\|_x^2 = c_2c_3\frac{1}{\kappa}\|d\|_x^2.$$

The proof is complete.

Global convergence of pdNCG for the primal variables is established in the following theorem.

Theorem 1 *Let $\{x^k\}$ be a sequence generated by pdNCG. The parameter η of the termination criterion* (22) *of the CG algorithm is set to $0 \leq \eta < 1$. Then the sequence $\{x^k\}$ converges to x^*, which is the minimizer of $f(x)$ in problem (4).*

Proof From Lemma 8 and step 3 of pdNCG we have that matrix $H(x, y)$ is symmetric and positive definite at any x^k, y^k. Moreover, if $0 \leq \eta < 1$ in (22), then CG returns $d^k = 0$ at a point x^k if and only if $\nabla f(x^k) = 0$. Hence, only at optimality CG will return a zero direction. Moreover, from Lemma 9 we get that if $\nabla f(x^k) \neq 0$, then $\bar{\alpha}^k$ is bounded away from zero and the function $f(x)$ is monotonically decreasing when the step $\bar{\alpha}^k d^k$ is applied. The monotonic decrease of the objective function implies that $\{f(x^k)\}$ converges to a limit, thus, $\{f(x^k) - f(x^{k+1})\} \to 0$. Since $f(x^0) < \infty$ and $f(x)$ is monotonically decreased, where x^0 is a finite first guess given as an input to pdNCG, then the sequence $\{x^k\}$ belongs in a closed, bounded and therefore, compact sublevel set. Hence, the sequence $\{x^k\}$ must have a subsequence, which converges to a point x^* and this implies that $\{x^k\}$ also converges to x^*. Using Lemma 9 and $\{f(x^k) - f(x^{k+1})\} \to 0$ we get that $\|d^k\|_x \to 0$, hence, due to positive definiteness of $H(x, y)$, $\|d^k\| \to 0$, which implies that $\|\nabla f(x^k)\| \to 0$. Therefore, x^* is a stationary point of function $f(x)$. Strong convexity of $f(x)$ guarantees that a stationary point must be a minimizer. The proof is complete.

Convergence of the dual variables is shown in the following theorem.

Theorem 2 *Let the assumptions of Theorem 1 hold. Then we have that the sequences of dual variables produced by pdNCG satisfy* $\{y^k\} \to Dx^*$*, where* x^* *is the optimal solution of problem* (4)*. Furthermore, the previous implies that the primal-dual iterates of pdNCG converge to the solution of system* (14)*.*

Proof From Theorem 1 we have that $d^k \to 0$ and $x^k \to x^*$. Hence, from (20) we get that $\Delta y^k \to -y^k + Dx^*$. Moreover, we have that the iterates at step 3 of pdNCG $\tilde{y}^k \to Dx^*$ and consequently

$$y^k = P_{\|\cdot\|_\infty \leq 1}(\tilde{y}) \to P_{\|\cdot\|_\infty \leq 1}(Dx^*) = Dx^*.$$

It is easy to check that these values of y^* with the optimal variable x^* satisfy the system (14). The proof is complete.

4.2 Region of fast convergence rate

In this subsection we define a region based on $\|d\|_x$, in which by setting parameter η as in (23) with $c_0 = 1$, pdNCG converges with fast rate. The lemma below shows the behaviour of the function $f(x)$ when a step along the primal pdNCG direction is made.

Lemma 10 *Let* $x \in \mathbb{R}^m$ *be the current iteration of pdNCG,* $d \in \mathbb{R}^m$ *be the pdNCG direction for primal variables calculated by CG, which is terminated according to criterion (22) with* $0 \leq \eta < 1$*. Then*

$$f(x) - f(x(\alpha)) \geq \alpha\|d\|_x^2 - \frac{\alpha^2}{2}\|d\|_x^2 - \frac{\alpha^3}{6}\frac{L}{\lambda_m^{\frac{3}{2}}}\|d\|_x^3,$$

where $x(\alpha) = x + \alpha d$ *and* $\alpha > 0$*.*

Proof Using Lemma 6 and setting $y = x(\alpha) = x + \alpha d$ we get

$$f(x(\alpha)) \leq f(x) + \alpha\nabla f(x)^\intercal d + \frac{\alpha^2}{2}d^\intercal\nabla^2 f(x)d + \frac{\alpha^3}{6}L\|d\|^3.$$

From Lemma 8 and step 3 of pdNCG we have that (24) holds. Hence, using (24) and Lemma 7 we get

$$f(x(\alpha)) \leq f(x) - \alpha\|d\|_x^2 + \frac{\alpha^2}{2}\|d\|_x^2 + \frac{\alpha^3}{6}\frac{L}{\lambda_m^{\frac{3}{2}}}\|d\|_x^3.$$

The result is obtained by rearrangement of terms. The proof is complete.

The next lemma determines bounds on the norm of the primal direction d^k as a function of $\|\nabla f_\tau^\mu(x^k)\|$.

Lemma 11 *Let $d \in \mathbb{R}^m$ be the pdNCG primal direction calculated by CG, which is terminated according to criterion (22) with $0 \leq \eta < 1$. Then the following holds*

$$\frac{1-\eta^2}{2\tilde{\lambda}_1^{\frac{1}{2}}}\|\nabla f(x)\| \leq \|d\|_x \leq \frac{1}{\lambda_m^{\frac{1}{2}}}\|\nabla f(x)\|$$

Proof By squaring (22) and making simple rearrangements of it we get

$$d^\intercal H(x,y)^2 d + 2\nabla f(x)^\intercal H(x,y)d + (1-\eta^2)\|\nabla f(x)\|^2 \leq 0. \tag{25}$$

From step 3 of pdNCG we have that the condition of Lemma 8 is satisfied. Therefore by using Lemma 8 and Cauchy-Schwarz inequality in (25) we get

$$\lambda_m^2\|d\|^2 - 2\tilde{\lambda}_1^{\frac{1}{2}}\|\nabla f(x)\|\|d\|_x + (1-\eta^2)\|\nabla f(x)\|^2 \leq 0.$$

By dropping the quadratic term $\lambda_m^2\|d\|^2$ from the previous inequality and dividing by $\|\nabla f(x)\|$, after making appropriate rearrangements we get

$$\|d\|_x \geq \frac{1-\eta^2}{2\tilde{\lambda}_1^{\frac{1}{2}}}\|\nabla f(x)\|.$$

This proves the left hand side of the result. For the right hand side, we simply use Lemma 7 and (24)

$$d^\intercal H(x,y)d = \|d\|_x^2 = -d^\intercal \nabla f(x) \leq \|d\|\|\nabla f(x)\| \leq \frac{1}{\lambda_m^{\frac{1}{2}}}\|d\|_x\|\nabla f(x)\|.$$

By dividing with $\|d\|_x$ we obtain the right hand side of our claim. The proof is complete.

The following lemma will be used to prove local fast convergence rate of pdNCG for the primal variables.

Lemma 12 *Let the iterates x^k and y^k be produced by pdNCG, then the following holds*

$$\|\nabla^2 f(x^k) - H(x^k,y^k)\| \leq \gamma\|d^k\|_{x^k},$$

where

$$\gamma = \Big(\frac{8\tilde{\lambda}_1^{\frac{1}{2}}}{\lambda_m}(L+M+\frac{M}{\mu}) + \frac{M}{\lambda_m^{\frac{1}{2}}\mu}\Big),$$

M is a positive constant.

Proof Let x^* be the optimal solution of problem (4). We rewrite

$$\nabla^2 f(x^k) - H(x^k,y^k) = \nabla^2 f(x^k) - \nabla^2 f(x^*) + \nabla^2 f(x^*) - H(x^k,y^k).$$

Moreover, let y^* be the optimal dual variable, which according to Theorem 2 satisfies $y^* = D(x^*)x^*$. Notice that matrix D in (15) is dependent on variable x; for the purposes of this proof we will explicitly denote this dependence.

From the definition of $H(x,y)$ in (19) we have that $H(x^*,y^*) = \nabla^2 f(x^*)$. The following holds

$$\|\nabla^2 f(x^k) - H(x^k,y^k)\| \leq \|\nabla^2 f(x^k) - \nabla^2 f(x^*)\| + \|H(x^*,y^*) - H(x^k,y^k)\|.$$

By Lipschitz continuity of $\nabla^2 f(x)$ in Lemma 5 we get that

$$\|\nabla^2 f(x^k) - H(x^k,y^k)\| \leq L\|x^* - x^k\| + \|H(x^*,y^*) - H(x^k,y^k)\|. \tag{26}$$

We now focus on bounding $\|H(x^*,y^*) - H(x^k,y^k)\|$. Using the fundamental theorem of calculus we have

$$H(x^*,y^*) - H(x^k,y^k) = \int_0^1 \frac{dH(x^*(s),y^*(s))}{d(x^*(s),y^*(s))}[x^* - x^k; y^* - y^k]ds,$$

where $x^*(s) = x^* + s(x^* - x^k)$ and $y^*(s) = y^* + s(y^* - y^k)$. Hence,

$$\|H(x^*,y^*) - H(x^k,y^k)\| \leq (\|x^* - x^k\| + \|y^* - y^k\|) \int_0^1 \Big\| \frac{dH(x^*(s),y^*(s))}{d(x^*(s),y^*(s))} \Big\| ds.$$

We now prove that $dH(x^*(s),y^*(s))/d(x^*(s),y^*(s))$ is bounded in the set $\mathbb{R}^m \times \{y \in \mathbb{R}^m \mid \|y\|_\infty \leq 1\} \subset \mathbb{R}^{2m}$. Observe that the partial derivatives $H(x,y)$ with respect to x or y are continuous. Therefore, $dH(x^*(s),y^*(s))/d(x^*(s),y^*(s))$ is a continuous tensor. In this case, the only candidates of unboundedness are the limits $x \to \pm\infty$. It is easy to show that at the limits all partial derivatives are finite and this implies that every component of the tensor is bounded in $\mathbb{R}^m \times \{y \in \mathbb{R}^m \mid \|y\|_\infty \leq 1\}$. We will denote the bound by a positive constant M, hence,

$$\|H(x^*,y^*) - H(x^k,y^k)\| \leq M(\|x^* - x^k\| + \|y^* - y^k\|). \tag{27}$$

It remains to find a bound for $\|y^* - y^k\|$. From step 3 of pdNCG we have

$$\begin{aligned}
\|y^* - y^k\| &\leq \|P_{\|\cdot\|_\infty \leq 1}(y^*) - P_{\|\cdot\|_\infty \leq 1}(\tilde{y}^k)\| \leq \|y^* - \tilde{y}^k\| \\
&\leq \|D(x^*)x^* - D(x^k)x^k\| \\
&+ \|D(x^k)(I - D(x^k)diag(x^k)diag(y^k))\|\|d^k\| \\
&= \|\nabla\psi_\mu(x^*) + \nabla\psi_\mu(x^k)\| \\
&+ \|D(x^k)(I - D(x^k)diag(x^k)diag(y^k))\|\|d^k\|.
\end{aligned}$$

Using Lemma 3 and

$$D(x^k)(I - D(x^k)diag(x^k)diag(y^k)) \preceq D(x^k) \preceq \frac{1}{\mu}I,$$

which holds for $\|y\|_\infty \leq 1$, we have that

$$\|y^* - y^k\| \leq \frac{1}{\mu}\|x^* - x^k\| + \frac{1}{\mu}\|d^k\|. \tag{28}$$

By combining inequalities (27) and (28) in (26) we get

$$\|\nabla^2 f(x^k) - H(x^k, y^k)\| \leq (L + M + \frac{M}{\mu})\|x^* - x^k\| + \frac{M}{\mu}\|d^k\|.$$

Combining Lemmas 4, 11 and (23) for a bound on $\|x^* - x^k\|$ we get

$$\|\nabla^2 f(x^k) - H(x^k, y^k)\| \leq \frac{8\tilde{\lambda}_1^{\frac{1}{2}}}{\lambda_m}(L + M + \frac{M}{\mu})\|d^k\|_{x^k} + \frac{M}{\mu}\|d^k\|.$$

Using (24) we get the result. The proof is complete.

Based on Lemmas 10, 11 and 12, a region is defined in the following lemma, in which unit-step sizes are calculated by the backtracking line-search algorithm. Additionally, for this region, $\|d^{k+1}\|_{x^{k+1}}$ is bounded as a function of $\|d^k\|_{x^k}$. In this lemma the constants c_2 and c_3 have been defined in step 4 of pdNCG, moreover, $x^{k+1} = x^k + d^k$.

Lemma 13 *If $\|d^k\|_{x^k} \leq 3(1-2c_2)\frac{\lambda_m^{\frac{3}{2}}}{L}$, then the backtracking line-search algorithm in step 4 of pdNCG calculates unit step-sizes. Moreover, if the parameter η^k of the termination criterion (22) of CG is set as in (23) with $c_0 = 1$, then for two consequent primal directions d^k, d^{k+1} and points x^k, x^{k+1}, the following holds*

$$\frac{1}{2}\frac{16\tilde{\lambda}_1\lambda_m + 2\gamma\lambda_m^{\frac{1}{2}} + L}{\lambda_m^{\frac{3}{2}}}\|d^{k+1}\|_{x^{k+1}} \leq \Big(\frac{1}{2}\frac{16\tilde{\lambda}_1\lambda_m + 2\gamma\lambda_m^{\frac{1}{2}} + L}{\lambda_m^{\frac{3}{2}}}\|d^k\|_{x^k}\Big)^2.$$

Proof By setting $\bar{\alpha} = 1$ in Lemma 10 we get

$$f(x^k) - f(x^{k+1}) \geq \frac{1}{2}\|d^k\|_{x^k}^2 - \frac{1}{6}\frac{L}{\lambda_m^{\frac{3}{2}}}\|d^k\|_{x^k}^3 = \Big(\frac{1}{2} - \frac{1}{6}\frac{L}{\lambda_m^{\frac{3}{2}}}\|d^k\|_{x^k}\Big)\|d^k\|_{x^k}^2.$$

if $\|d^k\|_{x^k} \leq 3(1-2c_2)\frac{\lambda_m^{\frac{3}{2}}}{L}$ we get

$$f(x^k) - f(x^{k+1}) \geq c_2\|d^k\|_{x^k}^2,$$

which implies that $\bar{\alpha} = 1$ satisfies the exit condition of the backtracking line-search algorithm. Let us define the quantities $\nabla f(x(t))^\intercal h$, where $h \in \mathbb{R}^m$,

$x(t) = x^k + td^k$ and $x(\delta) = x^k + \delta d^k$ then we have

$$\begin{aligned}
\nabla f(x(t))^\intercal h &= \nabla f(x^k)^\intercal h + t(d^k)^\intercal \nabla^2 f(x^k) h \\
&+ \int_0^t \int_0^u \nabla^3 f(x(\delta))[d^k, d^k, h] d\delta du \\
&\leq \nabla f(x^k)^\intercal h + t(d^k)^\intercal \nabla^2 f(x^k) h \\
&+ \int_0^t \int_0^u \Big| \nabla^3 f(x(\delta))[d^k, d^k, h] \Big| d\delta du \\
&= \nabla f(x^k)^\intercal h + t(d^k)^\intercal \nabla^2 f(x^k) h \\
&+ \int_0^t \int_0^u \lim_{\delta \to 0} \Big| \frac{(d^k)^\intercal (\nabla^2 f(x(\delta)) - \nabla^2 f(x^k)) h}{\delta} \Big| d\delta du \\
&\leq \nabla f(x^k)^\intercal h + t(d^k)^\intercal \nabla^2 f(x^k) h \\
&+ \|d^k\| \|h\| \int_0^t \int_0^u \lim_{\delta \to 0} \Big\| \frac{1}{\delta} (\nabla^2 f(x(\delta)) - \nabla^2 f(x^k)) \Big\| d\delta du \\
&\leq \nabla f(x^k)^\intercal h + t(d^k)^\intercal \nabla^2 f(x^k) h + \|d^k\| \|h\| \int_0^t \int_0^u L \|d^k\| d\delta du \\
&= \nabla f(x^k)^\intercal h + t(d^k)^\intercal \nabla^2 f(x^k) h + \frac{t^2}{2} L \|d^k\|^2 \|h\|.
\end{aligned}$$

By taking absolute values and setting $t = 1$ we get

$$\begin{aligned}
|\nabla f(x^{k+1})^\intercal h| &\leq |\nabla f(x^k)^\intercal h + (d^k)^\intercal \nabla^2 f(x^k) h| + \frac{1}{2} L \|d^k\|^2 \|h\| \\
&\leq \|\nabla f(x^k) + \nabla^2 f(x^k) d^k\| \|h\| + \frac{1}{2} L \|d^k\|^2 \|h\| \\
&\leq \|\nabla f(x^k) + H(x^k, y^k) d^k\| \|h\| \\
&+ \|\nabla^2 f(x^k) - H(x^k, y^k)\| \|d^k\| \|h\| + \frac{1}{2} L \|d^k\|^2 \|h\| \qquad (29)
\end{aligned}$$

Observe that from (23) with $c^0 = 1$ we have that $\eta^k \leq \|\nabla f(x^k)\|$. Hence, combining the previous with Lemma 11 and (22) in (29) we have that

$$\begin{aligned}
|\nabla f(x^{k+1})^\intercal h| &\leq 8\tilde{\lambda}_1 \|d^k\|_{x^k}^2 \|h\| + \|\nabla^2 f(x^k) - H(x^k, y^k)\| \|d^k\| \|h\| \\
&+ \frac{1}{2} L \|d^k\|^2 \|h\|.
\end{aligned}$$

Using Lemma 12 we have

$$|\nabla f(x^{k+1})^\intercal h| \leq 8\tilde{\lambda}_1 \|d^k\|_{x^k}^2 \|h\| + \gamma \|d^k\|_{x^k} \|d^k\| \|h\| + \frac{1}{2} L \|d^k\|^2 \|h\|.$$

From the equivalence of norms (24) we get

$$|\nabla f(x^{k+1})^\intercal h| \leq \frac{1}{2} \frac{16\tilde{\lambda}_1 \lambda_m + 2\gamma \lambda_m^{\frac{1}{2}} + L}{\lambda_m^{\frac{3}{2}}} \|d^k\|_{x^k}^2 \|h\|_{x^{k+1}}.$$

The previous result holds for every $h \in \mathbb{R}^m$, hence, by setting $h = d^{k+1}$ and by using Lemma 7 we prove the second part of this lemma. The proof is complete.

The following corollary states the region of fast convergence rate of Newton-CG. By fast rate it is meant that if pdNCG is initialized in this region, then the worst-case iteration complexity result for convergence to x^* is of the form $\log_2 \log_2 \frac{\text{constant}}{\text{required accuracy}}$. This statement is proved in Subsection 4.3 in Theorem 4.

Corollary 1 *If the parameter η^k in the termination criterion (22) of CG is set as in* (23) *with $c_0 = 1$ and $\|d^k\|_{x^k} < \varpi$, $0 < \varpi \leq c_5$, where*

$$c_5 = \min\Big\{3(1-2c_2)\frac{\lambda_m^{\frac{3}{2}}}{L}, \frac{\lambda_m^{\frac{3}{2}}}{16\tilde{\lambda}_1\lambda_m + 2\gamma\lambda_m^{\frac{1}{2}} + L}\Big\},$$

then according to Lemma 13 pdNCG convergences with fast rate.

4.3 Worst-case iteration complexity

The following theorem shows the worst-case iteration complexity of pdNCG in order to enter the region of fast convergence rate, i.e., $\|d\|_x < \varpi$, where $0 < \varpi \leq c_5$ and c_5 has been defined in Corollary 1. In this theorem the constant c_4 has been defined in Lemma 9, c_2 and c_3 are constants of the backtracking line-search algorithm in step 4 of pdNCG. Moreover, x^* denotes the minimizer of problem (4).

Theorem 3 *Starting from an initial point x^0, such that $\|d^0\|_{x^0} \geq \varpi$ and setting $0 \leq \eta < 1$ in the termination criterion* (22) *of CG, then pdNCG requires at most*

$$K_1 = c_6 \log\Big(\frac{f(x^0) - f(x^*)}{c_7\varpi^2}\Big),$$

iterations to obtain a solution x^k, $k > 0$, such that $\|d^k\|_{x^k} < \varpi$, where

$$c_6 = \frac{2\kappa^2}{(1-\eta^2)^2 c_2 c_3} \quad \textit{and} \quad c_7 = \frac{1}{2\kappa}.$$

Proof Let us assume an iteration index $k > 0$, then from Lemmas 4 and 11 we get

$$f(x^k) - f(x^*) \geq \frac{1}{2\kappa}\|d^k\|_{x^k}^2, \tag{30}$$

and

$$f(x^{k-1}) - f(x^*) \leq \frac{2\kappa}{(1-\eta^2)^2}\|d^{k-1}\|_{x^{k-1}}^2. \tag{31}$$

From Lemma 9 we have

$$f(x^k) < f(x^{k-1}) - c_4\|d^{k-1}\|_{x^{k-1}}^2. \tag{32}$$

Combining (31), (32) and subtracting $f(x^*)$ from both sides we get

$$\begin{aligned} f(x^k) - f(x^*) &< \Big(1 - \frac{(1-\eta^2)^2 c_4}{2\kappa}\Big)(f(x^{k-1}) - f(x^*)) \\ &< \Big(1 - \frac{(1-\eta^2)^2 c_4}{2\kappa}\Big)^k (f(x^0) - f(x^*)) \\ &= \Big(1 - \frac{(1-\eta^2)^2 c_2 c_3}{2\kappa^2}\Big)^k (f(x^0) - f(x^*)) \end{aligned}$$

From the last inequality and (30) we get

$$\frac{1}{2\kappa}\|d^k\|^2_{x^k} < \Big(1 - \frac{(1-\eta^2)^2 c_2 c_3}{2\kappa^2}\Big)^k (f(x^0) - f(x^*)).$$

Using the definitions of constants c_6 and c_7 we have

$$\|d^k\|^2_{x^k} < \Big(1 - \frac{1}{c_6}\Big)^k \frac{1}{c_7}(f(x^0) - f(x^*)).$$

Hence, we conclude that after at most K_1 iterations as defined in the preamble of this theorem, the algorithm produces $\|d^k\|_{x^k} < \varpi$. The proof is complete.

It is worth pointing out that a worst-case iteration complexity result for the global phase (before fast local convergence) of standard Newton method can be obtained by slightly modifying the proof of Theorem 3. In particular, in proof of Theorem 3, one simply has to replace matrix H with $\nabla^2 f(x)$ and set $\eta = 0$ (exact Newton directions), the constant κ remains unchanged since the matrices H and $\nabla^2 f(x)$ have the same uniform bounds, see (13) and Lemma 8. Using the previous adjustments, it is easy to show that the dominant term κ^2 in the result of Theorem 3 is preserved for standard Newton method. To the best of our knowledge, the result of $\mathcal{O}(\kappa^2)$ is the tightest that has been obtained for standard Newton method, see Subsection 9.5 in [6].

The following theorem presents the worst-case iteration complexity result of pdNCG to obtain a solution x^l, of accuracy $f(x^l) - f(x^*) < \epsilon$, when initialized at a point inside the region of fast convergence.

Theorem 4 *Suppose that there is an iteration index k of pdNCG, such that $\|d^k\|_{x^k} < \varpi$. If η in (22) is set as in (23) with $c_0 = 1$, then pdNCG needs at most*

$$K_2 = \log_2 \log_2 \Big(\frac{c_8}{\epsilon}\Big)$$

additional iterations to obtain a solution x^l, $l > k$, such that $f(x^l) - f(x^*) < \epsilon$, where

$$c_8 = \frac{16\kappa\lambda_m^3}{(16\tilde{\lambda}_1\lambda_m + 2\gamma\lambda_m^{\frac{1}{2}} + L)^2}.$$

Proof Suppose that there is an iteration index k such that $\|d^k\|_{x^k} < \varpi$, then for an index $l > k$, by applying Lemma 13 recursively we get

$$\frac{1}{2}\frac{16\tilde{\lambda}_1\lambda_m + 2\gamma\lambda_m^{\frac{1}{2}} + L}{\lambda_m^{\frac{3}{2}}}\|d^l\|_{x^l} \leq \Big(\frac{1}{2}\frac{16\tilde{\lambda}_1\lambda_m + 2\gamma\lambda_m^{\frac{1}{2}} + L}{\lambda_m^{\frac{3}{2}}}\|d^k\|_{x^k}\Big)^{2^{l-k}} < \Big(\frac{1}{2}\Big)^{2^{l-k}}. \tag{33}$$

From Lemmas 4, 11 and η^k in (23) we get

$$f(x^l) - f(x^*) \leq 4\kappa\|d^l\|_{x^l}^2,$$

By replacing (33) in the above inequality we get

$$f(x^l) - f(x^*) < \frac{16\kappa\lambda_m^3}{(16\tilde{\lambda}_1\lambda_m + 2\gamma\lambda_m^{\frac{1}{2}} + L)^2}\Big(\frac{1}{2}\Big)^{2^{l-k+1}}.$$

Hence, in order to obtain a solution x^l, such that $f(x^l) - f(x^*) < \epsilon$, pdNCG requires at most as many iterations as in the preamble of this theorem. The proof is complete.

The following theorem summarizes the complexity result of pdNCG. The constants c_6, c_7 and c_8 in this theorem are defined in Theorems 3 and 4, respectively.

Theorem 5 *Starting from an initial point x^0, such that $\|d^0\|_{x^0} \geq \varpi$, pdNCG requires at most*

$$K_3 = c_6 \log\Big(\frac{f(x^0) - f(x^*)}{c_7\varpi^2}\Big) + \log_2\log_2\Big(\frac{c_8}{\epsilon}\Big)$$

iterations to converge to a solution x^k, $k > 0$, of accuracy

$$f(x^k) - f(x^*) < \epsilon.$$

5 Numerical Experiments

We illustrate the robustness and efficiency of pdNCG on synthetic ℓ_1-regularized Sparse Least-Squares (S-LS) problems and real world ℓ_1-regularized Logistic Regression (LR) problems.

5.1 State-of-the-art first-order methods

A number of efficient first-order methods [2,10,12,20–22,26–28,30,31] have been developed for the solution of problem (1). The most efficient first-order methods, for example [2,21], rely on properties of the ℓ_1-norm to obtain the new direction at each iteration. In particular, at every iteration they require the exact minimization of a subproblem

$$\min_{x^+} \quad \tau\|x^+\|_1 + \varphi(x) + \nabla\varphi(x)^\intercal(x^+ - x) + \frac{L_\varphi}{2}\|x^+ - x\|^2, \tag{34}$$

where x is a given point. Other first-order methods use the decomposability of the former problem and solve it only for some chosen coordinates [21]. In this case, the Lipschitz constant is replaced by partial Lipschitz constants for each chosen coordinate.

In this section we compare pdNCG with two such state-of-the-art first-order methods.

- FISTA (Fast Iterative Shrinkage-Thresholding Algorithm) [2] is an optimal first-order method for problem (1). An efficient implementation of this algorithm can be found as part of the TFOCS (Templates for First-Order Conic Solvers) package [5] under the name N83.
- PCDM (Parallel Coordinate Descent Method) [21]. The published implementation performs parallel coordinate updates asynchronously based on (34), where the coordinates are chosen uniformly at random. This method is well-known for exploiting separability of the problems.

5.2 Implementation details

Solver PCDM is a C++ implementation, while FISTA and pdNCG are implemented in MATLAB. We expect that the programming language will not be an obstacle for FISTA and pdNCG. This is because these methods rely only on basic linear algebra operations, such as the dot product, which are implemented in C++ in MATLAB by default. All experiments are performed on a Dell PowerEdge R920 running Redhat Enterprise Linux with four Intel Xeon E7-4830 v2 2.2GHz processors, 20M Cache, 7.2 GT/s QPI, Turbo (4x10Cores). PCDM as a parallel method exploits 40 cores. Whilst, FISTA and pdNCG, which are MATLAB implementations, exploit multicore systems by performing in parallel simple linear algebra tasks by default. Finally, for pdNCG a simple diagonal preconditioner is used for all experiments. The preconditioner is set to be the inverse of the diagonal of matrix H.

5.3 Parameter tuning

For pdNCG, the smoothing parameter μ is set to 1.0e-4 and the parameter η in (23) is set to 1.0e-1. The backtracking step-size of pdNCG is set to $c_3 = 1/2$

and the parameter of sufficient decrease is set to $c_2 = 1.0e\text{-}3$. For PCDM, the parameter σ is set to $1 + 23(\nu - 1)/(m - 1)$ like it is proposed in [21], where ν is the partial separability degree of the problem that is solved; we will define ν later in this section. Moreover, for PCDM making functions evaluations is prohibited, because it is considered as a very expensive operation, hence, we do not include the running time of making such operations in the total running time. For FISTA we use the default parameter setting. All solvers are initialized to the zero solution.

We run pdNCG for sufficient time such that the problems are adequately solved. Then, FISTA and PCDM are terminated when the objective function $f_\tau(x)$ in (1) is below the one obtained by pdNCG or when a predefined maximum number of iterations limit is reached. All comparisons are presented in figures that show the progress of the objective function against the wall clock time. This way, the reader can compare the performance of the solvers for various levels of accuracy.

5.4 ℓ_1-Regularized Sparse Least-Squares

In this subsection we compare pdNCG with FISTA and PCDM. The comparison is made on a problem for which

$$\varphi(x) = \frac{1}{2}\|Ax - b\|^2$$

in (1), where $x \in \mathbb{R}^m$, $b \in \mathbb{R}^n$, $A \in \mathbb{R}^{n\times m}$ with $n \geq m$. We are interested in problems of this form that are sparse, with ill-conditioned $A^\intercal A$ and *partially* or *highly* separable. The definition of separability that is employed is the same as in [21], which for these problems is measured with the following constant

$$\beta_{LS} := \max_{j\in[1,2,\ldots,n]} \|A_j\|_0, \tag{35}$$

where A_j is the jth row of matrix A. Obviously, the following holds $1 \leq \beta_{LS} \leq m$. Notice that the larger β_{LS} is the less separable the problem becomes. However, observe that β_{LS} captures separability based only on the most dense row of matrix A. This implies that there might exist a matrix A that is very sparse but there is a single row of A that is relatively dense and this will result in large β_{LS}. In the examples that will be presented in this subsection β_{LS} is a small fraction of m.

5.4.1 Benchmark Generator

A generator for non-trivial sparse S-LS problems is given in the following simple process. First, a full-rank matrix $A \in \mathbb{R}^{n\times m}$ with $n \geq m$ is generated.

Second, the eigenvalue decomposition of $A^\intercal A = Q\Lambda Q^\intercal$ is computed. Third, the optimal solution is generated by approximately solving

$$\begin{aligned} x^* := \underset{x\in\mathbb{R}^n}{\arg\min} \quad & \|Q^\intercal x - \Lambda^{-1}e\|^2 \\ \text{subject to: } & \|x\|_0 \leq s, \end{aligned} \tag{36}$$

where e is a vector of ones, $\|\cdot\|_0$ is the zero norm, which counts the number of nonzero components of the input argument and s is a positive integer. To solve the above problem one can use an Orthogonal Matching Pursuit (OMP) [18] solver implemented in [3]. The aim of this approach is to find a sparse x^*, which can be expressed as $x^* = Qv$, where the coefficients v of the linear combination are close to the inverse of the eigenvalues of matrix $A^\intercal A$. Intuitively, this technique will create an x^*, which has strong dependence on subspaces that correspond to the smaller eigenvalues of $A^\intercal A$. It is well known that such an x^* makes the problem difficult to solve, see for example the analysis of Steepest Descent for LS in [23]. Finally, b can be generated such that the following holds

$$x^* := \underset{x\in\mathbb{R}^n}{\arg\min} \|x\|_1 + \|Ax - b\|^2.$$

This is achieved by substituting in the optimality conditions of the previous problem x^* and then choosing b such that the optimality conditions are satisfied. It is easy to extend the generator and consider a minimization of $\min_{x\in\mathbb{R}^n} \tau\|x\|_1 + \|Ax - b\|^2$ instead.

5.4.2 Synthetic sparse least-squares example: increasing conditioning

We now present the performance of pdNCG, FISTA and PCDM for increasing condition number of matrix A^TA. We generate six instances (A, b, x^*), where the condition number of $A^\intercal A$ takes values 1.00e+02, 1.00e+04, 1.00e+06, 1.00e+08, 1.00e+10 and 1.00e+12. Matrix A has $m = 2^{22}$ columns, $n = 2m$ rows and rank m. Moreover, matrix A is sparse, i.e., $nnz(A)/(mn) \approx 3.00e\text{-}07$ and $\beta_{LS} = 2$. The optimal solution x^* has approximately $s \approx 8.0e\text{-}03m$ non-zero components.

The results of this experiment are shown in Figure 2. In this figure the objective function $f_\tau(x)$ is presented against the wall clock time for each solver. Observe the log-scale used for both axes. The wall clock time of the solvers is shown after their first iteration takes place. PCDM was the fastest method for condition number less than or equal to 1.00e+04, while pdNCG was the second fastest method. For condition number 1.00e+06 pdNCG converged in comparable time with PCDM, which was the fastest. For condition number larger than or equal to 1.00e+08 pdNCG was clearly the fastest method. Moreover, for condition number larger than or equal to 1.00e+10 pdNCG was the only method that solved the problems to sufficient accuracy within reasonable time. Notice that despite the problems being sparse and highly separable, PCDM and FISTA did not scale well for the problems with condition number larger

than or equal to 1.0e+10. In particular, when the condition number of $A^\intercal A$ is 1.00e+12, PCDM and FISTA did not converge in competitive time; they were terminated after more than 27 hours of wall clock time.

5.4.3 Synthetic sparse least-squares example: increasing dimensions

In this experiment we present the performance of pdNCG, FISTA and PCDM as the number of variables m increases. We generate three instances (A, b, x^*), where m takes values 2^{20}, 2^{22} and 2^{24}. Matrix A has $n = 2m$ rows, rank m and the condition number of $A^\intercal A$ is 1.00e+08. Moreover, matrix A is sparse, i.e., $\beta_{LS} = 2$, $nnz(A)/(mn) \approx$ 1.00e-06, 3.00e-07 and 5.00e-08, respectively for each m. The optimal solution x^* has again approximately $s \approx$ 8.0e-03m non-zero components.

The results of this experiment are presented in Figure 3. Observe that the required wall-clock time for pdNCG scaled similarly to the first-order methods FISTA and PCDM, despite being a second-order method.

5.5 ℓ_1-Regularized Logistic Regression

In this subsection we compare pdNCG with FISTA and PCDM on six real world ℓ_1-regularized LR problems. For ℓ_1-regularized LR the function $\varphi(x)$ in (1) is set to

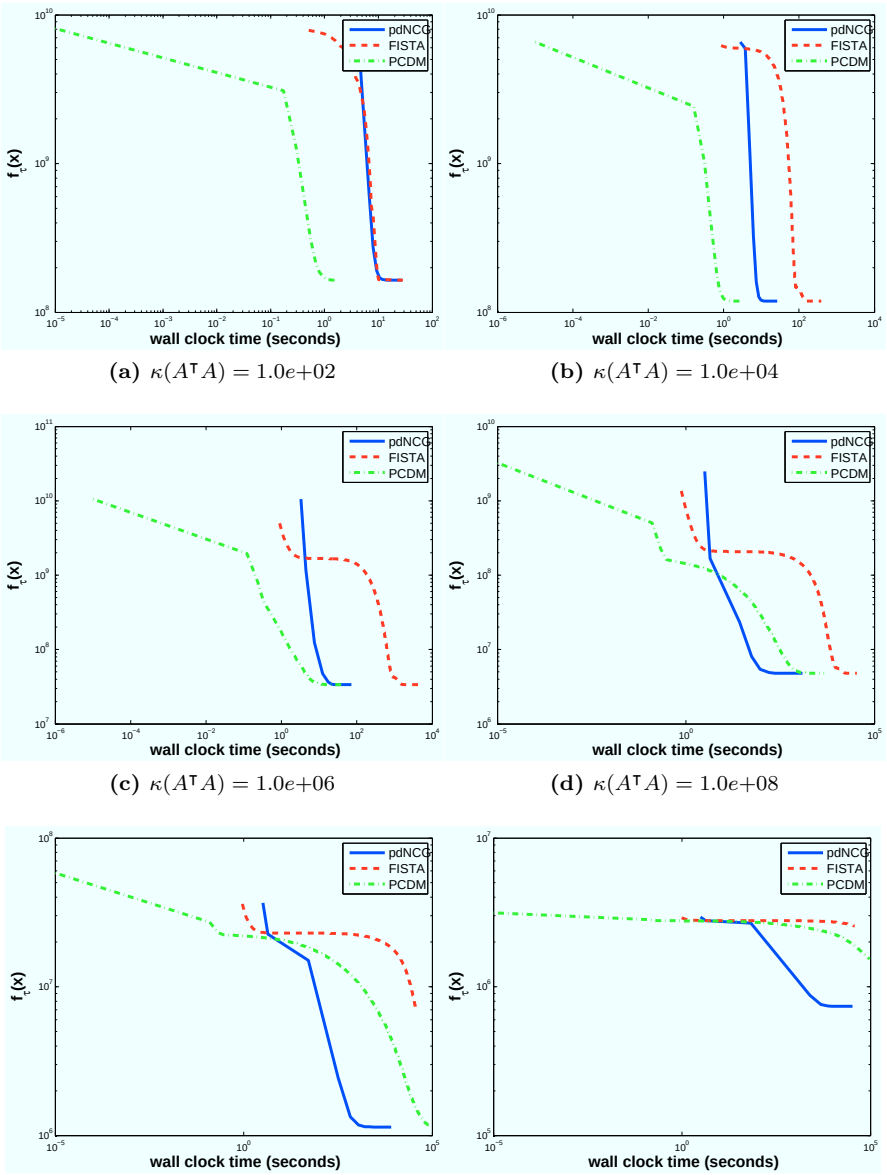
$$\varphi(x) = \sum_{i=1}^{n} \log(1 + e^{-y_i w^\intercal x_i}),$$

where $x_i \in \mathbb{R}^m$ $\forall i = 1, 2, \ldots, n$ are the training samples and $y_i \in \{-1, +1\}$ are the corresponding labels. Such problems are used for training a linear classifier $w \in \mathbb{R}^m$. Although in Linear Support Vector Machine (LSVM) literature there are more alternatives for function $\varphi(x)$, in this section we choose LR because it is second-order differentiable. For more details about support vector machine problems we refer the reader to [33].

We present six ℓ_1-regularized LR problems that are of large scale, sparse and partially or highly separable. Exact information for these problems is given in Table 1. In this table, matrix $X \in \mathbb{R}^{n\times m}$ has the training samples in its rows, the fourth column shows the sparsity of matrix X, where $nnz(X)$ is the number of non-zero components in X. The fifth column shows the degree of partial separability, which is defined as

$$\beta_{LR} := \max_{j\in[1,2,\ldots,n]} \|X_j\|_0, \tag{37}$$

where X_j is the jth row of matrix X. The last column shows the τ that gave the classification with the highest accuracy after performing a fivefold cross validation over various τ values, as proposed in [13]. The calculated values τ resulted for all problems in more than 90% classification accuracy. All problems in Table 1 can be downloaded from the collection of LSVM problems

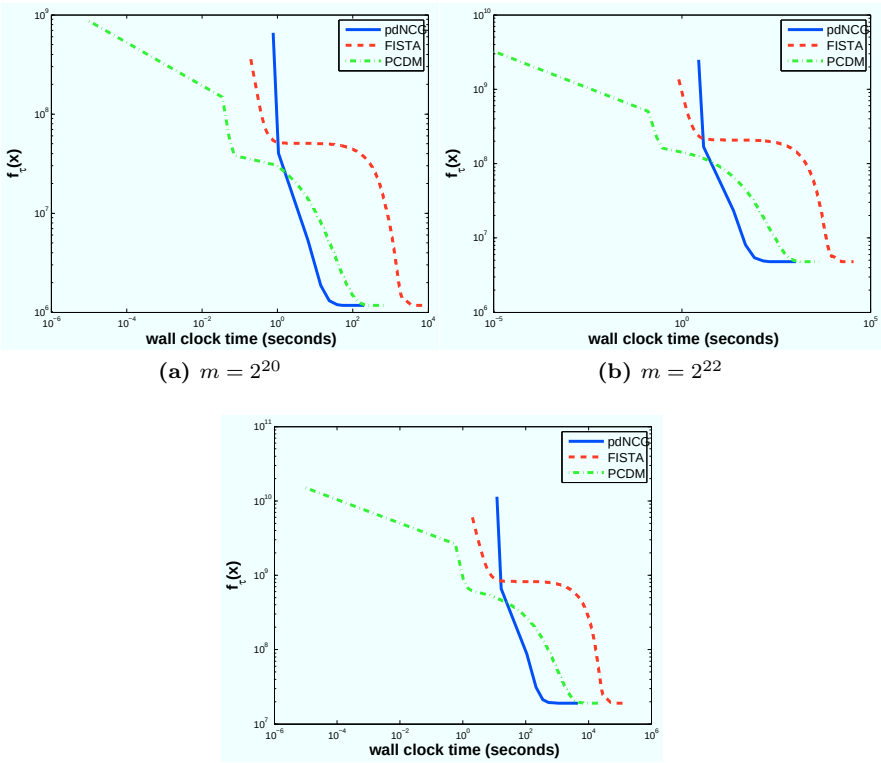


(a) $\kappa(A^\intercal A) = 1.0e{+}02$ **(b)** $\kappa(A^\intercal A) = 1.0e{+}04$

(c) $\kappa(A^\intercal A) = 1.0e{+}06$ **(d)** $\kappa(A^\intercal A) = 1.0e{+}08$

(e) $\kappa(A^\intercal A) = 1.0e{+}10$ **(f)** $\kappa(A^\intercal A) = 1.0e{+}12$

Fig. 2 Performance of pdNCG, FISTA and PCDM on a synthetic sparse S-LS problem for increasing condition number of matrix $A^\intercal A$, denoted as $\kappa(A^\intercal A)$. The axis are in log-scale. In this figure $f_\tau(x)$ denotes the objective value that was obtained by each solver



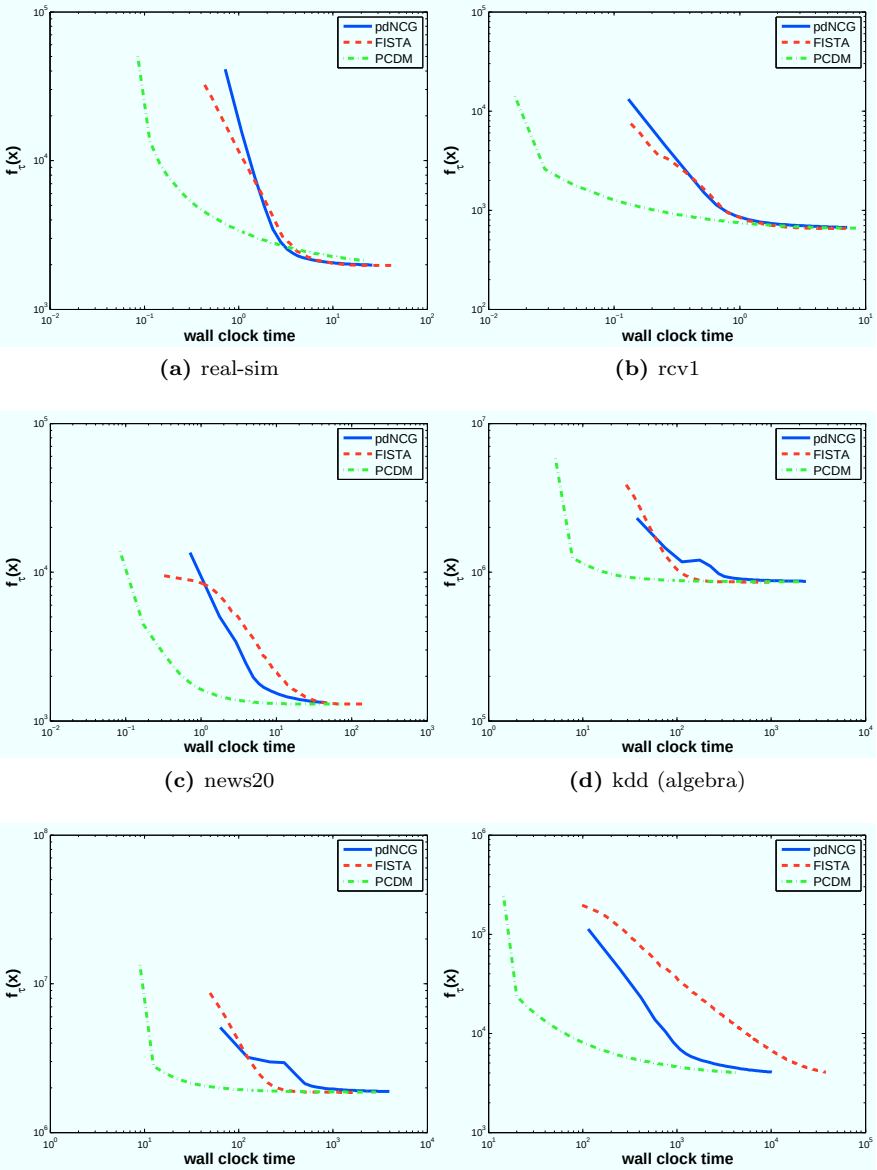
(a) $m = 2^{20}$ **(b)** $m = 2^{22}$

(c) $m = 2^{24}$

Fig. 3 Performance of pdNCG, FISTA and PCDM on a synthetic sparse S-LS problem for increasing number of variables m. The axis are in log-scale. In this figure $f_\tau(x)$ denotes the objective value that was obtained by each solver

in [9]. Notice that for most of the problems in Table 1, $m < n$, which means that the problems are not strongly-convex everywhere as assumed in (2). However, the problems in Table 1 have a unique solution, which implies that they are strongly-convex locally to the optimal solution. We choose to solve these instances because the problems for which $m > n$ in collection [9] are small scale, hence, possible numerical experiments might not provide significant insight into the behaviour of the methods. It is important to mention that all implementations of the compared methods can handle such cases without any modification.

The results of the comparison among the solvers pdNCG, FISTA and PCDM are shown in Figure 4. Notice in Subfigure 4d that for pdNCG the objective function $f_\tau(x)$ seems not to decrease always monotonically. This behaviour might have occurred because backtracking line-search can terminate before the condition in Step 4 of pdNCG is satisfied, if the maximum number of backtracking iterations is exceeded.



(a) real-sim **(b)** rcv1

(c) news20 **(d)** kdd (algebra)

(e) kdd (bridge to algebra) **(f)** webspam

Fig. 4 Comparison on ℓ_1-regularized LR problems for pdNCG, FISTA and PCDM.

Table 1 Properties of six ℓ_1-regularized LR problems, which are used as benchmarks in this paper. The second and third columns show the number of training samples and features, respectively. The fourth and fifth columns show the sparsity of matrix X and the degree of partial separability β_{LR} in (37), respectively. The last column is the τ found using fivefold cross-validation

Problem	$\boldsymbol{n}$	$\boldsymbol{m}$	$\mathbf{nnz(X)/(mn)}$	$\boldsymbol{\beta_{LR}}$	$\boldsymbol{\tau}$
[17] real-sim	72,309	20,958	2.40e-02	3484	4.00e-02
[16] rcv1	20,242	47,236	1.60e-02	980	4.00e-02
[14] news20	19,996	1,355,191	3.35e-04	16423	4.00e-02
[32] kdd (algebra)	8,407,752	20,216,830	1.79e-06	85	2.00e-00
[32] kdd (br. to alg.)	19,264,097	29,890,095	9.83e-07	75	2.00e-00
[29] webspam	350,000	16,609,143	2.24e-04	46991	4.00e-02

6 Conlcusion

In this paper we have studied an inexpensive but still robust primal-dual Newton-CG (pdNCG) method. The proposed method is developed for the solution of ℓ_1-regularized problems, which might display some degree of ill-conditioning; that is, display noticeable differences of the magnitude of eigenvalues. For such problems it is crucial that the methods capture information from the second-order derivative. We have given synthetic sparse least-squares examples and six real world machine learning problems that satisfy the previous criteria and we provided computational evidence that on these problems the proposed method is efficient. An implementation of pdNCG and scripts that reproduce the numerical experiments can be downloaded from `http://www.maths.ed.ac.uk/ERGO/pdNCG/`.

Finally, we have shown that by using the property of CG described in Lemma 7, the convergence analysis of pdNCG can be performed in a variable metric, which is defined based on approximate second-order derivatives. The variable metric opens the door for a tight convergence analysis of pdNCG, which includes global and local convergence rates, explicit definition of fast local convergence region and worst-case iteration complexity.

References

1. R. Acar and C. R. Vogel. Analysis of bounded variation penalty methods for ill-posed problems. *Inverse Problems*, 10:1217–1229, 1994.
2. A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.
3. S. Becker. CoSaMP and OMP for sparse recovery. `http://www.mathworks.co.uk/matlabcentral/fileexchange/32402-cosamp-and-omp-for-sparse-recovery`, 2012.
4. S. R. Becker, J. Bobin, and E. J. Candès. Nesta: A fast and accurate first-order method for sparse recovery. *SIAM J. Imaging Sciences*, 4(1):1–39, 2011.
5. S. R. Becker, E. J. Candés, and M. C. Grant. Templates for convex cone problems with applications to sparse signal recovery. *Mathematical Programming Computation*, 3(3):165–218, 2011. Software available at `http://tfocs.stanford.edu`.
6. S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press New York, NY, USA, 2004.

7. R. H. Chan, T. F. Chan, and H. M. Zhou. Advanced signal processing algorithms. *in Proceedings of the International Society of Photo-Optical Instrumentation Engineers, F. T. Luk, ed., SPIE*, pages 314–325, 1995.
8. T. F. Chan, G. H. Golub, and P. Mulet. A nonlinear primal-dual method for total variation-based image restoration. *SIAM J. Sci. Comput.*, 20(6):1964–1977, 1999.
9. Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.
10. K.-W. Chang, C.-J. Hsieh, and C.-J. Lin. Coordinate descent method for large-scale ℓ_2-loss linear support vector machines. *Journal of Machine Learning Research*, 9:1369–1398, 2008.
11. R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
12. C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan. A dual coordinate descent method for large-scale linear SVM. *Proceedings of the 25th international conference on Machine Learning, ICML 2008*, pages 408–415, 2008.
13. C.-W. Hsu, C.-C. Chang, and C.-J. Lin. A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University, 2010.
14. S. S. Keerthi and D. DeCoste. A modified finite newton method for fast solution of large scale linear svms. *Journal of Machine Learning Research*, 6:341–361, 2005.
15. C. T. Kelly. *Iterative Methods for Linear and Nonlinear Equations*. SIAM, Philadelphia, PA., 1995.
16. D. D. Lewis, Yiming Yang, T. G. Rose, and F. Li. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.
17. A. McCallum. Real-sim: Real vs. Simulated data for binary classification problem. `http://www.cs.umass.edu/~mccallum/code-data.html`.
18. D. Needell and J. A. Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321, 2009.
19. J. Renegar. *A Mathematical View of Interior-Point Methods in Convex Optimization*. MOS-SIAM Series on Optimization, Cornell University, Ithaca, New York, 2001.
20. P. Richtárik and M. Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 2012.
21. P. Richtárik and M. Takáč. Parallel coordinate descent methods for big data optimization. Technical report, School of Mathematics, Edinburgh University, 2012. Software available at `https://code.google.com/p/ac-dc/`.
22. S. Shalev-Shwartz and A. Tewari. Stochastic methods for ℓ_1-regularized loss minimization. *Journal of Machine Learning Research*, 12(4):1865–1892, 2011.
23. J. R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Technical report, Carnegie Mellon University Pittsburgh, PA, USA, 1994.
24. S. Sra, S. Nowozin, and S. J. Wright. *Optimization for Machine Learning*. MIT Press, 2011.
25. R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Roy. Statist. Soc.*, 58(1):267–288, 1996.
26. P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109(3):475–494, 2001.
27. P. Tseng. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM J. Optim.*, 22:341–362, 2012.
28. P. Tseng and S. Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Math. Program., Ser. B*, 117:387–423, 2009.
29. S. Webb, J. Caverlee, and C. Pu. Introducing the webb spam corpus: Using email spam to identify web spam automatically. *In Proceedings of the Third Conference on Email and Anti-Spam (CEAS)*, 2006.
30. S. J. Wright. Accelerated block-coordinate relaxation for regularized optimization. *SIAM Journal on Optimization*, 22(1):159–186, 2012.
31. T. T. Wu and K. Lange. Coordinate descent algorithms for lasso penalized regression. *The Annals of Applied Statistics*, 2(1):224–244, 2008.

32. H.-F. Yu, H.-Y. Lo, H.-P. Hsieh, J.-K. Lou, T. G. McKenzie, J.-W. Chou, P.-H. Chung, C.-H. Ho, C.-F. Chang, Y.-H. Wei, J.-Y. Weng, E.-S. Yan, C.-W. Chang, T.-T. Kuo, Y.-C. Lo, P.-T. Chang, C. Po, C.-Y. Wang, Y.-H. Huang, C.-W. Hung, Y.-X. Ruan, Y.-S. Lin, S.-D. Lin, H.-T. Lin, and C.-J. Lin. Feature engineering and classifier ensemble for kdd cup 2010. *In JMLR Workshop and Conference Proceedings*, 2011.
33. G. X. Yuan, C. H. Ho, and C. J. Lin. Recent advances of large-scale linear classification. *Proceedings of the IEEE*, 100(9):2584–2603, 2012.