
This is the **accepted version** of the journal article:

Tendero, Yohann; Ciril, Igor; Darbon, Jérôme; [et al.]. «An Algorithm Solving Compressive Sensing Problem Based on Maximal Monotone Operators». SIAM Journal on Scientific Computing, Vol. 43, Issue 6 (January 2021), p. A4067-A4094. DOI 10.1137/19M1260670

This version is available at <https://ddd.uab.cat/record/306643>

under the terms of the  license

1 **AN ALGORITHM SOLVING COMPRESSIVE SENSING PROBLEMS**
2 **BASED ON MAXIMAL MONOTONE OPERATORS***

3 YOHANN TENDERO^T, IGOR CIRIL[‡], JÉRÔME DARBON[§], AND SUSANA SERNA[¶]

4 **Abstract.** The need to solve ℓ^1 regularized linear problems can be motivated by various com-
5 pressive sensing and sparsity related techniques for data analysis and signal or image processing.
6 These problems lead to non-smooth convex optimization in high dimensions. Theoretical works pre-
7 dict a sharp phase transition for the exact recovery of compressive sensing problems. Our numerical
8 experiments show that state-of-the-art algorithms are not effective enough to observe this phase tran-
9 sition accurately. This paper proposes a simple formalism that enables us to produce an algorithm
10 that computes an ℓ^1 minimizer under the constraints $A\mathbf{u} = \mathbf{b}$ up to the machine precision. In addi-
11 tion, a numerical comparison with standard algorithms available in the literature is exhibited. The
12 comparison shows that our algorithm compares advantageously with other state-of-the-art methods,
13 both in terms of accuracy and efficiency. With our algorithm, the aforementioned phase transition
14 is observed at high precision.

15 **Key words.** Sparse solution recovery, Compressive sensing, Inverse scale space, ℓ^1 minimization,
16 Non-smooth optimization, Maximal monotone operator, Phase transition.

17 **AMS subject classifications.** 34A60 ,49M29 ,90C06 ,90C25.

18 **1. Introduction.** Compressive sensing and sparsity-related paradigms have gai-
19 ned enormous interest in the last decade and can be used for, e.g., data analysis, signal
20 and image processing, inverse problems or acquisition devices. Indeed, in many cases
21 the unknowns of an under-determined system can be obtained by finding the sparsest
22 (or simplest) solution to a linear system

23 (1.1)
$$A\mathbf{u} = \mathbf{b}.$$

24 With this formulation \mathbf{b} is the observed data, $A \in \mathcal{M}_{m \times n}(\mathbb{R})$, $m \ll n$ and the columns
25 of A represent a suitable frame or dictionary able to sparsely encode or observe $\mathbf{u} \in \mathbb{R}^n$.
26 However, finding a minimizer of the ℓ^0 pseudo-norm under the constraints (1.1) is a
27 highly non-convex and non-smooth optimization problem. Hence, methods [19, 25,
28 30, 35, 39, 3, 14, 28] that aim at tackling ℓ^0 pseudo norm minimization guarantee
29 an optimal solution only with high probability and for a specific class of matrices A .
30 Another class of methods consists in using an ℓ^1 relaxation. The problem therefore
31 becomes

32
$$(P_{\ell^1}) \quad \begin{cases} \inf_{\mathbf{u} \in \mathbb{R}^n} & \|\mathbf{u}\|_{\ell^1} \\ \text{s.t.} & A\mathbf{u} = \mathbf{b}. \end{cases}$$

33 It turns out that under various assumptions, the minimizers remain the same if one
34 replaces the ℓ^0 pseudo-norm by the ℓ^1 norm (see, e.g., [11, 12, 16, 17] and the references

*Submitted to the editors DATE. A preliminary version of the work is [13].

Funding: Research of J. Darbon has been supported by the National Science Foundation under Grant No. NSF-1820821.

^TDR2I, Institut Polytechnique des Sciences Avancées, 94200, Ivry-sur-Seine, France yohann.tendero@ipsa.fr.

[‡]DR2I, Institut Polytechnique des Sciences Avancées, 94200, Ivry-sur-Seine, France igor.ciril@ipsa.fr.

[§]Division of Applied Mathematics, Brown University, Providence, RI 02912, USA

[¶]Departament de Matemàtiques, Universitat Autònoma de Barcelona, Bellaterra, Barcelona, Spain susana.serna@uab.cat

35 therein). Problem (P_{ℓ^1}) is a convex albeit non-smooth optimization problem in high
 36 dimension (n can be thought as the number of pixels of an image for instance). For
 37 these reasons developing efficient algorithmic solutions is still a challenge in many
 38 cases. For instance, the CVX system “is not meant for very large problems” [20,
 39 Sec. 1.3, p.3] that arise from signal/image processing applications [24, 37]. Hence,
 40 many algorithms have been proposed to solve ℓ^1 minimization problems, see, e.g., [34,
 41 21, 2, 8, 9, 44, 45, 43, 42, 18, 40]. In this paper, we propose a simple algorithm
 42 that can be employed to solve these ℓ^1 minimization problems up to the machine
 43 precision. Indeed, it is only assumed that the matrix A has full row rank. This
 44 paper also exhibits a numerical comparison with several classic algorithms in the
 45 literature. These comparisons illustrate that our algorithm compares advantageously:
 46 the theoretically predicted phase transition, see e.g. [29, 10], is empirically observed
 47 with a higher accuracy.

48 To design our algorithm, we required that: i) the method computes a solution
 49 to (P_{ℓ^1}) up to the machine precision, and that ii) the method requires few computa-
 50 tions involving vectors of length n .

51 The first requirement can be thought as guaranteeing the quality of the solution
 52 or the fidelity to the problem. The second requirement can be thought as promoting
 53 the numerical efficiency. Indeed, computations with vectors of length $m \ll n$ require
 54 less memory than the memory needed for vectors of the primal. (We recall that the
 55 unknown \mathbf{u} lives in a high dimensional space, while the observed data \mathbf{b} lives in a
 56 space of dimension $m \ll n$). It seems unrealistic to find a minimizer to (P_{ℓ^1}) up to
 57 the machine precision with a direct method. Consequently, the approach we employ
 58 is iterative and can be summarized as follows.

59 To the best of our knowledge, the most similar approach to the one developed
 60 in this paper is the AISS [7] method. AISS iterates over two variables: a primal one
 61 that belongs to \mathbb{R}^n and a dual one in \mathbb{R}^m . Instead, we compute *one* finite discrete
 62 sequence $\boldsymbol{\lambda}_k$ for $k = 1, \dots, K$ in \mathbb{R}^m . The last iterate, namely $\boldsymbol{\lambda}_K$, is an solution to
 63 the dual problem of (P_{ℓ^1}) up to the machine precision. Given $\boldsymbol{\lambda}_K$ a simple formula
 64 allows us to compute a solution $\bar{\mathbf{u}}$ to (P_{ℓ^1}) up to the machine precision. This last
 65 computation is the only one that requires vectors of the high dimensional space. Our
 66 main assumption throughout this paper is that $\exists \mathbf{u}$ such that $A\mathbf{u} = \mathbf{b}$, i.e., (P_{ℓ^1}) has at
 67 least one solution. This can be guaranteed if one assumes, as we shall do hereinafter,
 68 that A has full rank.

69

70 *Outline of the paper.* The paper is organized as follows. Section 2 gives a very
 71 compact, yet self-contained, presentation of the numerical computations needed to
 72 implement the algorithm proposed in this paper (see algorithm 2.1 on page 4). Sec-
 73 tion 3 on page 5 proves the mathematical validity of this algorithm. In other words,
 74 we shall prove that the solution computed by algorithm 2.1 is exact (and numerically,
 75 up to the machine precision). The convergence (in finite time) of algorithm 2.1 to a
 76 solution to (P_{ℓ^1}) is mathematically guaranteed. Section 4 on page 9 proposes a numer-
 77 ical evaluation and comparison of algorithm 2.1 with some state-of-the-art solutions
 78 solving (P_{ℓ^1}) . We show in this section that our method has a higher probability of
 79 success to reconstruct solutions with high precision compared to other state-of-the-art
 80 methods, i.e., the phase transition is observed with a high precision. Discussions and
 81 conclusions are summarized in section 5 on page 14. The appendix 6 on page 16 con-
 82 tains several proofs used throughout this paper. A glossary containing the notations
 83 and basic definitions is in appendix 7 on page 24. In the sequel, Latin numerals refer
 84 to the glossary of notations on page 24. Appendix 8 on page 25 contains general

85 results on convex analysis used in this paper.

86 **2. An Algorithm Solving (P_{ℓ^1}) .** This section presents the algorithm proposed
 87 in this paper. As usual in the literature on compressive sensing, we shall assume that
 88 $A \in \mathcal{M}_{m,n}(\mathbb{R})$ with $m \ll n$. The algorithm we shall develop in this paper begins by
 89 computing a solution to the dual problem associated to (P_{ℓ^1}) then computes a solution
 90 to the primal. The first step involves the computation of a finite and piecewise affine
 91 trajectory, or more precisely the positions $\boldsymbol{\lambda}_k$ where the trajectory changes of slope.
 92 The second step relies on the computation of a solution to a constrained least square
 93 problem. The construction leads to algorithm 2.1 (page 4).
 94 Consider the Lagrangian $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ of (P_{ℓ^1}) namely

$$95 \quad (2.1) \quad \mathcal{L}(\mathbf{u}, \boldsymbol{\lambda}) := J(\mathbf{u}) + \langle \boldsymbol{\lambda}, A\mathbf{u} \rangle + \langle \boldsymbol{\lambda}, -\mathbf{b} \rangle,$$

96 where $J(\cdot) = \|\cdot\|_{\ell^1}$. Consider also the function $g : \mathbb{R}^m \rightarrow \mathbb{R} \cup \{+\infty\}$ defined by

$$97 \quad g(\boldsymbol{\lambda}) := - \inf_{\mathbf{u} \in \mathbb{R}^n} \mathcal{L}(\mathbf{u}, \boldsymbol{\lambda}) = - \inf_{\mathbf{u} \in \mathbb{R}^n} \{J(\mathbf{u}) - \langle -A^T \boldsymbol{\lambda}, \mathbf{u} \rangle\} - \langle \boldsymbol{\lambda}, -\mathbf{b} \rangle$$

$$98 \quad (2.2) \quad = J^* (-A^T \boldsymbol{\lambda}) + \langle \boldsymbol{\lambda}, \mathbf{b} \rangle = \chi_{B_\infty} (-A^T \boldsymbol{\lambda}) + \langle \boldsymbol{\lambda}, \mathbf{b} \rangle,$$

99 where J^* denotes the Legendre-Fenchel transform of J (see **(xvi)**) and χ_{B_∞} denotes
 100 the convex characteristic function of ℓ^∞ (see **(vii)**) unit ball $B_\infty \subset \mathbb{R}^n$ (see **(xi)**). (We
 101 recall that hereinafter Latin numerals refer to the glossary of notations on page 24.)
 102 Consider further the optimization problem

$$103 \quad (D_{\ell^1}) \quad \inf_{\boldsymbol{\lambda} \in \mathbb{R}^m} g(\boldsymbol{\lambda}),$$

104 where g is given by (2.2). As we shall see, under classic assumptions problems (P_{ℓ^1})
 105 and (D_{ℓ^1}) have at least one solution (see proposition 3.4 on page 5). We now give a
 106 strategy to solve (D_{ℓ^1}) . The trajectory $[0, +\infty) \ni t \mapsto \boldsymbol{\lambda}(t)$ explicitly given, for every
 107 $t \geq 0$, by

$$108 \quad (2.3) \quad \begin{cases} \frac{d^+ \boldsymbol{\lambda}}{dt}(t) = -\Pi_{\partial g(\boldsymbol{\lambda}(t))}(\mathbf{0}) \\ \boldsymbol{\lambda}(0) = \boldsymbol{\lambda}_0 \end{cases}$$

109 converges for some finite time $t_K \in [0, +\infty)$ to a solution to (D_{ℓ^1}) . The main idea
 110 of (2.3) is that it generalizes the usual steepest Euclidean descent for non-smooth con-
 111 vex functions. When the function is not differentiable, then (2.3) selects the smallest
 112 velocity in the ℓ^2 sense among all possible velocities that corresponds to the subdif-
 113 ferential of the function at a non-differentiable point. Note that the subdifferential
 114 always only contains one element, which is the gradient, when the function is differen-
 115 tiable. Formula (2.3) formalizes an evolution equation governed by the (multi-valued)
 116 maximal monotone operator ∂g (see, for instance, [1, Eq. 2, p. 158]). In (2.3),
 117 $\Pi_{\partial g(\boldsymbol{\lambda}(t))}$ denotes the Euclidean projection **(xviii)** on $\partial g(\boldsymbol{\lambda}(t))$ and $\boldsymbol{\lambda}_0 \in \text{dom } g$ is
 118 some initial state. We always set $\boldsymbol{\lambda}_0 = \mathbf{0}$ in our experiments. For any $\boldsymbol{\lambda} \in \text{dom } g$ the
 119 multi-valued monotone operator ∂g is given by the non-empty convex cone

$$120 \quad (2.4) \quad \partial g(\boldsymbol{\lambda}) = \left\{ \mathbf{b} + \sum_{i \in S(\boldsymbol{\lambda})} \eta_i A \tilde{\mathbf{e}}_i : \eta_i \geq 0, i \in S(\boldsymbol{\lambda}) \right\},$$

121 where the set $S(\boldsymbol{\lambda})$ is defined by

$$122 \quad (2.5) \quad S(\boldsymbol{\lambda}) := \{i \in \{1, \dots, 2n\} : \langle \boldsymbol{\lambda}, A\tilde{\mathbf{e}}_i \rangle = 1\} \quad \text{and} \quad \tilde{\mathbf{e}}_i = \begin{cases} \mathbf{e}_i & \text{for } i = \{1, \dots, n\} \\ -\mathbf{e}_{i-n} & \text{for } i = \{n+1, \dots, 2n\}. \end{cases}$$

123 In (2.5) and everywhere else, \mathbf{e}_i denotes the i -th canonical vector of \mathbb{R}^n .

124 In addition, the trajectory given by (2.3) is piecewise affine. This means that the
125 next iterate $\boldsymbol{\lambda}_{k+1}$ produced by the algorithm is computed from the current iterate
126 $\boldsymbol{\lambda}_k$, the scalar $(t_{k+1} - t_k)$ and the direction $\mathbf{d}_k = -\Pi_{\partial g(\boldsymbol{\lambda}_k)}(\mathbf{0})$. We now detail the
127 computation of the scalar $(t_{k+1} - t_k)$. For any $k \in \mathbb{N}$, we define

$$128 \quad (2.6) \quad S^+(\mathbf{d}_k) := \{i \in \{1, \dots, 2n\} : \langle \mathbf{d}_k, A\tilde{\mathbf{e}}_i \rangle > 0\}$$

$$129 \quad (2.7) \quad \text{and we have } \bar{\Delta}t_k := (t_{k+1} - t_k) = \min \left\{ \frac{1 - \langle A\tilde{\mathbf{e}}_i, \boldsymbol{\lambda}_k \rangle}{\langle A\tilde{\mathbf{e}}_i, \mathbf{d}_k \rangle}, i \in S^+(\mathbf{d}_k) \right\}.$$

130 Note that (2.6) and (2.7) are easy to compute since these quantities are given
131 explicitly and only involves computations of inner products. Therefore, from (2.3) we
132 observe that it remains to compute the direction $\mathbf{d}_k = -\Pi_{\partial g(\boldsymbol{\lambda}_k)}(\mathbf{0})$ which corresponds
133 to computing the projection on a non-empty closed convex cone given by $\partial g(\boldsymbol{\lambda}_k)$.
134 Note that this subdifferential has an explicit formula given by (2.4). One can use a
135 constrained least square solver, available in Matlab, to compute the solution. (See also
136 remark 2.1 below.) To sum up, to compute a solution to (D_{ℓ^1}) one can compute the
137 limit of the trajectory $\boldsymbol{\lambda}(t)$ given by (2.3) using the update rules (2.6) and (2.7). This
138 limit is attained after finitely many updates (see also proposition 3.15). It remains to
139 compute a solution to (P_{ℓ^1}) given $\bar{\boldsymbol{\lambda}}$ solution to (D_{ℓ^1}) .
140 Given $\bar{\boldsymbol{\lambda}}$ solution to (D_{ℓ^1}) , one can compute a solution $\bar{\mathbf{u}}$ to (P_{ℓ^1}) by solving the
141 constrained least square problem

$$142 \quad (2.8) \quad \begin{cases} \min_{\mathbf{u} \in \mathbb{R}^n} & \|\mathbf{A}\mathbf{u} - \mathbf{b}\|_{\ell^2} \\ \text{s.t.} & u_i \geq 0 \text{ if } \langle \bar{\boldsymbol{\lambda}}, A\mathbf{e}_i \rangle = -1, u_i \leq 0 \text{ if } \langle \bar{\boldsymbol{\lambda}}, A\mathbf{e}_i \rangle = 1 \text{ and } u_i = 0 \text{ otherwise.} \end{cases}$$

143 We are now in position to state the entire algorithm.

144

Algorithm 2.1 Algorithm computing $\bar{\mathbf{u}}$ solution to (P_{ℓ^1}) .

Input: Matrix A , \mathbf{b}

Output: $\bar{\mathbf{u}}$ solution to (P_{ℓ^1})

Set $k := 0$ and $\boldsymbol{\lambda}_k := \mathbf{0} \in \mathbb{R}^m$ **repeat**

1. Compute $S(\boldsymbol{\lambda}_k)$ (see (2.5));
2. Compute \mathbf{d}_k as $\mathbf{d}_k := -\Pi_{\partial g(\boldsymbol{\lambda}_k)}(\mathbf{0})$ (see remark 2.1)
3. Compute $S^+(\mathbf{d}_k)$ (see (2.6)) then $\bar{\Delta}t_k$ (see (2.7))
4. Set $\boldsymbol{\lambda}_{k+1} := \boldsymbol{\lambda}_k + \bar{\Delta}t_k \mathbf{d}_k$;
5. Set $k = k + 1$ and set $\boldsymbol{\lambda} := \frac{\boldsymbol{\lambda}}{\|A^T \boldsymbol{\lambda}\|_{\ell^\infty}}$ if $\|A^T \boldsymbol{\lambda}\|_{\ell^\infty} > 1$;

until $\mathbf{d}_k = \mathbf{0}$ (see remark 2.1);

Compute $\bar{\mathbf{u}}$ using (2.8).

145 *Remark 2.1.* To compute \mathbf{d}_k we define $G := \left\{ \sum_{i \in S(\boldsymbol{\lambda}_k)} \eta_i A\tilde{\mathbf{e}}_i : \eta_i \geq 0, i \in S(\boldsymbol{\lambda}_k) \right\}$. We
146 have that $\mathbf{d}_k := -\Pi_{\partial g(\boldsymbol{\lambda}_k)}(\mathbf{0}) = -\Pi_G(-\mathbf{b}) - \mathbf{b}$ (see lemma 6.5 on page 21) can be com-
147 puted from a constrained least square problem similar to (2.8). We refer to [15,

148 Section 3.2] and the references therein for a detailed review of exact (up to machine
 149 precision) numerical algorithms solving the above constrained least square problem.
 150 For instance, one can use the *lsqnonneg* Matlab routine albeit we used an implementa-
 151 tion based on [31] that is supposedly faster than the Matlab routine. The termination
 152 condition, namely $\mathbf{d}_k = \mathbf{0}$, was replaced by $\|\mathbf{d}_k\|_{\ell^2} \vee \|\bar{\Delta}_k \mathbf{d}_k\|_{\ell^2} < 10^{-10}$ in all of our
 153 experiments. The projection in step 5 is unnecessary if the precision of numbers is
 154 high enough. However, we empirically observed that it increased the performance of
 155 the method for the Matlab implementation.

156 **3. From Maximal Monotone Operator to ℓ^1 Solutions of Linear Prob-**
 157 **lems.** This section justifies the mathematical validity of algorithm 2.1 presented in
 158 section 2.

159 We recall that to solve (P_{ℓ^1}) , we first solve the dual (D_{ℓ^1}) then compute a solution
 160 to the primal problem (P_{ℓ^1}) . Hence, we first give the assumptions that justify the
 161 existence of solutions to problems (P_{ℓ^1}) and (D_{ℓ^1}) and give a closed formula that
 162 allows us to compute the solution to (P_{ℓ^1}) from a solution to (D_{ℓ^1}) . This is done in
 163 proposition 3.4. We then briefly justify the fact that the trajectory we used in the
 164 previous section converges to a solution to the dual. This is done in proposition 3.6.
 165 This proposition translates into algorithm 2.1 on page 4 and is illustrated numerically
 166 in section 4 on page 9.

167 **PROPOSITION 3.1** (and definition). *We assume that $A \in \mathcal{M}_{m,n}(\mathbb{R})$ has full row*
 168 *rank and that $J(\cdot) = \|\cdot\|_{\ell^1}$. We consider the functions*

$$169 \quad (3.1) \quad \forall \mathbf{u} \in \mathbb{R}^n, \quad f(\mathbf{u}) := J(\mathbf{u}) + \chi_{\{\mathbf{b}\}}(A\mathbf{u});$$

$$170 \quad (3.2) \quad \forall \boldsymbol{\lambda} \in \mathbb{R}^m, \quad g(\boldsymbol{\lambda}) := J^*(-A^T \boldsymbol{\lambda}) + \langle \boldsymbol{\lambda}, \mathbf{b} \rangle = \chi_{\mathbf{B}_\infty}(-A^T \boldsymbol{\lambda}) + \langle \boldsymbol{\lambda}, \mathbf{b} \rangle.$$

171 *We have $f \in \Gamma_0(\mathbb{R}^n)$ and $g \in \Gamma_0(\mathbb{R}^m)$ (see (x)).*

172 *Proof.* See appendix 16 on page 18. □

173 **Remark 3.2.** The assumptions of proposition 3.1 allow to cover the case of com-
 174 pressive sensing problems. Note that one can relax the assumption that A is full row
 175 rank, we just need that $\mathbf{b} \in \text{span } A$. For instance if, for some specific application, the
 176 observed \mathbf{b} 's belong to a subspace B then we just need $\text{span } A \supset B$.

177 We recall that we wish to solve (P_{ℓ^1}) using a solution to (D_{ℓ^1}) . To this aim the
 178 following definition and proposition are needed.

179 **DEFINITION 3.3** (Active set). *For any $\boldsymbol{\lambda} \in \text{dom } g$ we define*

$$180 \quad (3.3) \quad S(\boldsymbol{\lambda}) := \{i \in \{1, \dots, 2n\} : \langle \boldsymbol{\lambda}, A\tilde{\mathbf{e}}_i \rangle = 1\} \quad \text{and} \quad \tilde{\mathbf{e}}_i = \begin{cases} \mathbf{e}_i & \text{for } i = \{1, \dots, n\} \\ -\mathbf{e}_{i-n} & \text{for } i = \{n+1, \dots, 2n\}, \end{cases}$$

181 *and \mathbf{e}_i denotes the i -th canonical vector of \mathbb{R}^n .*

182 **PROPOSITION 3.4** (Existence of solutions and computation of a solution to (P_{ℓ^1})).
 183 *We posit the same assumptions as in proposition 3.1. We have*

- 184 1. *Problems (P_{ℓ^1}) and (D_{ℓ^1}) have at least one solution.*
- 185 2. *Let $\tilde{\boldsymbol{\lambda}}$ be a solution to (D_{ℓ^1}) . Consider the coefficients \tilde{u}_i such that $\tilde{u}_i = 0 \forall i \in$
 186 $\{1, \dots, 2n\} \setminus S(-\tilde{\boldsymbol{\lambda}})$ and $\tilde{u}_i \geq 0$ for $i \in S(-\tilde{\boldsymbol{\lambda}})$, of the Euclidean projection of*

187 **b** onto

$$188 \quad (3.4) \quad \left\{ \mathbf{y} : \mathbf{y} = \sum_{\substack{\tilde{u}_i \geq 0 \quad \forall i \in S(-\bar{\lambda}) \\ \tilde{u}_i = 0 \quad \forall i \in \{1, \dots, 2n\} \setminus S(-\bar{\lambda})}} \tilde{u}_i A \tilde{\mathbf{e}}_i \right\},$$

189 where $S(-\bar{\lambda})$ and $\tilde{\mathbf{e}}_i$ are defined by (3.3).

190 We have that the vector $\bar{\mathbf{u}}$ obtained from the above coefficients \tilde{u}_i

$$191 \quad (3.5) \quad \bar{\mathbf{u}} := \sum_{i=1}^n u_i \mathbf{e}_i, \text{ with } u_i := \tilde{u}_i - \tilde{u}_{i+n}$$

192 is a solution to (P_{ℓ^1}) .

193 Note that (3.4) is equivalent to formula (2.8) given in section 2. Indeed, in (3.4) at
194 least one of the coefficients \tilde{u}_i or \tilde{u}_{i+n} is zero.

195 *Proof.* See appendix 17 on page 18. \square

196 *Remark 3.5.* The reconstruction formula given by (2.8) is different from the re-
197 construction methods that can sometimes be found in the literature (see, e.g., [32, al-
198 gorithm 6, p. 11]). However, for matrices satisfying compressive sensing assumptions
199 (see, e.g., [12, 16]), the signal can be obtained from an unconstrained least-squares
200 solution to $A\mathbf{u} = \mathbf{b}$. Indeed, the support constraint issued form $\bar{\lambda}$ boils down to
201 solving, in the least squares sense, $B\mathbf{u} = \mathbf{b}$, where B is a sub-matrix formed from A
202 by removing appropriate columns. Note that in this case, there is no sign constraint
203 on u_i contrarily to (2.8). In addition, in many cases, the unconstrained least squares
204 solution can be computed using a Moore-Penrose pseudo inverse formula. However,
205 the least squares solution and (2.8) will, in general, differ: they have same ℓ^0 pseudo
206 norms but different ℓ^1 norms.

207 To solve (D_{ℓ^1}) we rely on a specific trajectory of feasible points for (D_{ℓ^1}) governed
208 by the maximal monotone operator ∂g (see, e.g., [1]). The main properties of this
209 trajectory are summarized in the next proposition.

210 **PROPOSITION 3.6** (Properties of the trajectory $\boldsymbol{\lambda}(t)$ [1, 5]). *We posit the same*
211 *assumptions as in proposition 3.1. Consider the evolution equation explicitly given,*
212 *for every $t \in [0, +\infty)$, by*

$$213 \quad (3.6) \quad \begin{cases} \frac{d^+ \boldsymbol{\lambda}(t)}{dt} = -\Pi_{\partial g(\boldsymbol{\lambda}(t))}(\mathbf{0}) \\ \boldsymbol{\lambda}(0) = \boldsymbol{\lambda}_0, \end{cases}$$

214 where $\boldsymbol{\lambda}_0 \in \text{dom } \partial g$. We have that the solution $\boldsymbol{\lambda} : [0, +\infty) \ni t \mapsto \boldsymbol{\lambda}(t) \in \mathbb{R}^m$ to (3.6)
215 satisfies:

- 216 1. for every $t \in [0, +\infty)$, $\boldsymbol{\lambda}(\cdot)$ is continuous, right-differentiable and belongs to
217 $\text{dom } \partial g$;
- 218 2. the limits of $g(\boldsymbol{\lambda}(t))$ and $\boldsymbol{\lambda}(t)$ when $t \rightarrow +\infty$ exist;
- 219 3. $\lim_{t \rightarrow +\infty} g(\boldsymbol{\lambda}(t)) = \min_{\boldsymbol{\lambda} \in \mathbb{R}^m} g(\boldsymbol{\lambda})$ and $\lim_{t \rightarrow +\infty} \boldsymbol{\lambda}(t) \in \arg \min_{\boldsymbol{\lambda} \in \mathbb{R}^m} g(\boldsymbol{\lambda})$.

220 *Proof.* See appendix 18 on page 18. \square

221 The proposition above means that the limit of the trajectory $\boldsymbol{\lambda}(t)$ is a solution to (D_{ℓ^1}) .
222 In the sequel, we shall prove that the limit is attained for a finite time $t \geq 0$. It is

223 worth noticing the similarity between (3.6) and inverse scale space methods (see,
 224 e.g., [7, 32]). To compute $\lambda(t)$ one could rely on an Euler scheme to approximate the
 225 trajectory for instance. However, a numerical computation of the trajectory $\lambda(t)$ up
 226 to the machine precision is doable. This is the goal of the next paragraph.

227 **Computation of the trajectory $\lambda(t)$ given by (3.6).** We recall that to obtain
 228 an algorithm we need to compute a solution $\bar{\lambda}$ to (D_{ℓ^1}) . To do so, we recall that we
 229 compute the positions where $\lambda(t_k)$ changes of slope. Since $\text{dom } g \neq \mathbb{R}^m$ we cannot
 230 recur to classic textbooks such as, e.g., [22, Chap. VIII]. Thus, some work is needed.

231 Proposition 3.13 (on page 8) proves that $\lambda(t)$ defined by (3.6) is piecewise affine.
 232 In other words, $\lambda(t)$ is made of pieces of straight lines. Hence, the computation of
 233 $\lambda(t)$ boils down to the detection of “kicks”, i.e., positions where $\lambda(t)$ changes slope
 234 and the computation of these slopes. The computation of these slopes is obtained
 235 from (3.6) and lemma 3.8. Propositions 3.10 and 3.14 yield a direct and optimal
 236 numerical method to detect kick times, i.e., times t such that $\lambda(t)$ and $\lambda(t + \varepsilon)$ don't
 237 have the same slope for some $\varepsilon > 0$. Propositions 3.11 and 3.15 give the termination
 238 condition and proves that $\lambda(t)$ converges to a solution to (D_{ℓ^1}) after finitely many
 239 kicks. We recall that proposition 3.4 (page 5) directly gives an explicit formula that
 240 allows us to compute a solution to (P_{ℓ^1}) given a solution to (D_{ℓ^1}) obtained as the
 241 limit of the trajectory $\lambda(t)$.

242 We recall that one of the two main ingredients to compute the trajectory $\lambda(t)$ is
 243 the computation of slopes given by a projection onto the closed convex cone $\partial g(\lambda(t))$
 244 (see proposition 3.6 on page 6). Hence, a closed formula for ∂g is needed. This is the
 245 goal of the next proposition that leads to lemma 3.8.

246 **PROPOSITION 3.7** (The function g defined by (3.2) is polyhedral). *We posit the*
 247 *same assumptions as in proposition 3.1. The function g defined in (3.2) is a polyhedral*
 248 *proper and convex function that satisfies $\text{dom } g = C \neq \emptyset$ and we have*

$$249 \quad (3.7) \quad g(\lambda) = \langle \lambda, \mathbf{b} \rangle + \chi_C(\lambda), \quad \text{where } C := \{\lambda \in \mathbb{R}^m : \langle \lambda, A\tilde{\mathbf{e}}_i \rangle \leq 1, i \in \{1, \dots, 2n\}\}$$

250 and $\tilde{\mathbf{e}}_i$ is defined in (3.3).

251 *Proof.* See appendix 6.5 on page 19. □

252 We now give a formula for the subdifferential of g .

253 **LEMMA 3.8** (Subdifferential formula for g). *We posit the same assumptions as*
 254 *in proposition 3.1. We have $\text{dom } \partial g = \text{dom } g = C \neq \emptyset$ and, for any $\lambda \in C$,*

$$255 \quad (3.8) \quad \partial g(\lambda) = \{\mathbf{b}\} + \text{co} \{A\tilde{\mathbf{e}}_i : i \in S(\lambda)\},$$

256 where $\tilde{\mathbf{e}}_i, S(\lambda)$ are given by (3.3) and co by (v).

257 *Proof.* See appendix 6.6 on page 19. □

258 With the above formula it is easily seen that one can compute the slope of $\lambda(t)$ for
 259 any $t \geq 0$. It remains to compute the kick times, i.e., times t when the slope of the
 260 trajectory $\lambda(t)$ changes. This is the goal of the next three propositions and lemma.

261 **PROPOSITION 3.9** (and definition: descent direction). *We posit the same setup*
 262 *as in proposition 3.1. We say that a direction $\mathbf{d} \in \mathbb{R}^m \setminus \{\mathbf{0}\}$ is a descent direction for g*
 263 *at $\lambda \in \text{dom } g$ iff $(\lambda + t\mathbf{d}) \in \text{dom } g$ and $g(\lambda + t\mathbf{d}) < g(\lambda)$ for some $t > 0$. Moreover,*
 264 *we have that a direction $\mathbf{d} \neq \mathbf{0}$ is a descent direction for g at λ iff \mathbf{d} satisfies*

$$265 \quad (3.9) \quad \langle \mathbf{d}, A\tilde{\mathbf{e}}_i \rangle \leq 0 \quad \forall i \in S(\lambda) \quad \text{and}$$

$$266 \quad (3.10) \quad g'(\lambda, \mathbf{d}) = \langle \mathbf{d}, \mathbf{b} \rangle < 0, \quad \text{where } \tilde{\mathbf{e}}_i \text{ is given by (3.3).}$$

267 *Proof.* See appendix 21 on page 19. □

268 PROPOSITION 3.10 (Kick time computation). *We posit the same assumptions*
 269 *as in proposition 3.1 and further assume that $\lambda \in \text{dom } g$ and that \mathbf{d} is a direction that*
 270 *satisfies (3.9). Consider $\tilde{\mathbf{e}}_i$ given by (3.3), the set $S^+(\mathbf{d})$ defined by*

$$271 \quad (3.11) \quad S^+(\mathbf{d}) := \{i \in \{1, \dots, 2n\} : \langle A\tilde{\mathbf{e}}_i, \mathbf{d} \rangle > 0\}$$

272 *and the scalar $\bar{\Delta}t$ by*

$$273 \quad (3.12) \quad \begin{cases} \bar{\Delta}t := \min \left\{ \frac{1 - \langle A\tilde{\mathbf{e}}_i, \lambda \rangle}{\langle A\tilde{\mathbf{e}}_i, \mathbf{d} \rangle} : i \in S^+(\mathbf{d}) \right\} & \text{if } S^+(\mathbf{d}) \neq \emptyset; \\ \bar{\Delta}t := +\infty & \text{otherwise.} \end{cases}$$

274 *We have that $\bar{\Delta}t$ satisfies $\bar{\Delta}t > 0$. In addition, $S^+(\mathbf{d}) = \emptyset$ iff $(\lambda + t\mathbf{d}) \in \text{dom } g$ for*
 275 *every $t \geq 0$. Furthermore, we have*

$$276 \quad (3.13) \quad (\lambda + t\mathbf{d}) \in \text{dom } g \text{ iff } t \in [0, \bar{\Delta}t];$$

$$277 \quad (3.14) \quad \forall t \in [0, \bar{\Delta}t) \quad S(\lambda + t\mathbf{d}) \subset S(\lambda) \quad \text{and} \quad \partial g(\lambda + t\mathbf{d}) \subset \partial g(\lambda).$$

278 *Proof.* See appendix 23 on page 20. □

279 LEMMA 3.11 (Well posedness of $\mathbf{d} := -\Pi_{\partial g(\lambda)}(\mathbf{0})$, optimality conditions). *We*
 280 *posit the same assumptions as in proposition 3.1. For any $\lambda \in \text{dom } g$, the vector given*
 281 *by*

$$282 \quad (3.15) \quad \mathbf{d} := -\Pi_{\partial g(\lambda)}(\mathbf{0}).$$

283 *is well defined. Consider \mathbf{d} defined by (3.15) and $\bar{\Delta}t$, $S^+(\mathbf{d})$ defined in proposi-*
 284 *tion 3.10. We have that the three following conditions are equivalent*

$$285 \quad (3.16) \quad \mathbf{d} = \mathbf{0} \quad \Leftrightarrow \quad \bar{\Delta}t = +\infty \quad \Leftrightarrow \quad S^+(\mathbf{d}) = \emptyset.$$

286 *In addition, $\lambda \in \text{dom } g$ is a solution to (D_{ℓ^1}) iff the conditions in (3.16) hold true.*

287 *Proof.* See appendix 24 on page 8 □

288 PROPOSITION 3.12 ($\Pi_{\partial g(\lambda)}(\mathbf{0})$ is constant on time intervals). *We posit the same*
 289 *assumptions as in proposition 3.1. Consider any $\lambda \in \text{dom } g$, \mathbf{d} defined by (3.15) and*
 290 *$\bar{\Delta}t$ defined in corollary 3.10. We have*

$$291 \quad (3.17) \quad \forall t \in [0, \bar{\Delta}t) \quad \Pi_{\partial g(\lambda)}(\mathbf{0}) \in \partial g(\lambda + t\mathbf{d});$$

$$292 \quad (3.18) \quad \forall t \in [0, \bar{\Delta}t) \quad \Pi_{\partial g(\lambda)}(\mathbf{0}) = \Pi_{\partial g(\lambda + t\mathbf{d})}(\mathbf{0}).$$

293 *Proof.* See appendix 25 on page 21. □

294 We are now in position to give a mathematical definition of the trajectory computed
 295 by the algorithm.

296 PROPOSITION 3.13 (and definition: piecewise affine trajectory $\lambda(t)$). *We posit*
 297 *the same assumptions as in proposition 3.1. Consider $\lambda_0 \in \text{dom } g$ and the sequences*
 298 *$(t_k)_k \subset [0, +\infty]$, $(\mathbf{d}_k)_k$ and $(\lambda(t_k))_k$ recursively defined by*

$$299 \quad (3.19) \quad \begin{cases} t_0 := 0; & \mathbf{d}_k := -\Pi_{\partial g(\lambda(t_k))}(\mathbf{0}); & t_{k+1} := t_k + \bar{\Delta}t_k; \\ \lambda(t_{k+1}) := \lambda(t_k) + (t_{k+1} - t_k)\mathbf{d}_k & \text{if } t_{k+1} < +\infty \\ \lambda(t_{k+1}) := \lambda(t_k) & \text{otherwise,} \end{cases}$$

300 where $\bar{\Delta}t_k$ is obtained from proposition 3.10 (applied with $\lambda := \lambda(t_k)$ and $\mathbf{d} := \mathbf{d}_k$).
 301 Consider also the affine interpolate (continuous) trajectory $\lambda : [0, +\infty] \ni t \mapsto \mathbb{R}^m$
 302 defined by

$$303 \quad (3.20) \quad \lambda(t) := \lambda(t_k) + (t - t_k)\mathbf{d}_k \text{ for any } t \in [t_k, t_{k+1}), \quad \lambda(t_0) := \lambda_0.$$

304 We have that the trajectory $\lambda(t)$ given in (3.20) coincides for every $t \geq 0$ with the
 305 solution to the evolution equation (3.6). In addition, for every $t \geq 0$ we have $\lambda(t) \in$
 306 $\text{dom } g$.

307 *Proof.* See appendix 29 on page 23. □

308 To compute $\lambda(t)$ the algorithm relies on the computation of the sequence $(\mathbf{d}_k, t_k)_k$
 309 defined by (3.19). The next two propositions prove that $\lambda(t)$ changes of slope at every
 310 t_k and that the sequences in (3.19) are finite.

311 **PROPOSITION 3.14** (Optimality of the sampling of the trajectory $\lambda(t)$). *We posit*
 312 *the same assumptions as in proposition 3.1 and further assume that $\lambda(t_k) \in \text{dom } g$ is*
 313 *not a solution to (D_{ℓ^1}) . For $\lambda(t_{k+1})$ given by proposition 3.13 we have*

$$314 \quad (3.21) \quad \Pi_{\partial g(\lambda(t_k))}(\mathbf{0}) \neq \Pi_{\partial g(\lambda(t_{k+1}))}(\mathbf{0}) \text{ and } \|\Pi_{\partial g(\lambda(t_{k+1}))}(\mathbf{0})\|_{\ell^2} < \|\Pi_{\partial g(\lambda(t_k))}(\mathbf{0})\|_{\ell^2}.$$

315 *Proof.* See appendix 30 on page 23.

316 **PROPOSITION 3.15** ($\lambda(t)$ converges to a minimizer of (D_{ℓ^1}) after finitely many
 317 kicks). *We posit the same assumptions as in proposition 3.1. Consider the sequences*
 318 *$(t_k)_k$, $(\mathbf{d}_k)_k$ and the trajectory $\lambda(t)$ defined in proposition 3.13. We have that $\exists K \in \mathbb{N}$*
 319 *such that $\lambda(t) = \lambda(t_K)$ for every $t \geq t_K$. In addition, $\lambda(t_K)$ is a solution to (D_{ℓ^1})*
 320 *and \mathbf{d}_K satisfies $\mathbf{d}_K = \mathbf{0}$.*

321 *Proof.* See appendix 31 on page 24. □

322 We now briefly justify that the computations in algorithm 2.1 (page 4) end with a
 323 solution to (P_{ℓ^1}) after finitely many iterations. We obtained that for any $\lambda_0 \in \text{dom } g$
 324 (see proposition 3.13) the sequence defined in (3.19) converges (see proposition 3.15)
 325 after finitely many kicks to a solution to (D_{ℓ^1}) . In algorithm 2.1, the initialization step
 326 namely $\lambda_0 = \mathbf{0}$ is valid since $\mathbf{0} \in \text{dom } g$. In addition, it is easily seen that steps 1-5
 327 implement (3.19). From proposition 3.15, we deduce the validity of the termination
 328 condition. Proposition 3.15 justifies that this termination condition is reached after
 329 finitely many iterations. Hence, the while loop ends with some $\bar{\lambda}$ solution to (D_{ℓ^1}) .
 330 Therefore, the computation of $\bar{\mathbf{u}}$ solution to (P_{ℓ^1}) is justified by proposition 3.4.
 331 Therefore, the validity of algorithm 2.1 is proved.

332 *Remark 3.16.* Supplementary material shows that our proposed approach can be
 333 extended to handle affine inequality constraints. In addition, supplementary material
 334 presents how our proposed algorithm 2.1 can be used to solve the optimization problem
 335 with constraints of the form $\|A\mathbf{u} - \mathbf{b}\|_{\ell^2} \leq \epsilon$, i.e., when there is Gaussian noise. This
 336 approach will be presented in another paper.

337 **4. Experiments.** This section proposes an empirical evaluation of the following
 338 methods to solve (P_{ℓ^1}) : AISS [7], LARS [18], SPGL1 [40, 41], SeDuMI [38] and algo-
 339 rithm 2.1. Two parameters settings are considered for SeDuMI: the first version version
 340 which is called ‘‘standard precision’’ (SP) uses the standard parameters provided in
 341 the CVX package, while the second version which is called ‘‘high precision’’ (HP) uses
 342 the option ‘‘cvx_precision best’’. Supplementary material gives the same comparisons

343 between OMP [35], CoSamp [33] and GISS [32]. Note that OMP, CoSamp and GISS
 344 are greedy-based numerical algorithms. LARS, SPGL1, AISS and algorithm 2.1 are
 345 ℓ^1 -based numerical algorithms. SeDuMi [38] is a toolbox for linear, second order and
 346 semi-definite problems. These methods are compared in terms of a “probability of
 347 success” (defined below) and average number of iterations needed. The criterion will
 348 be used to observe a so called phase transition that separates cases where algorithms
 349 successfully recover the sparsest solution and when they fail. Note that solutions with
 350 high precision are required to observe an accurate phase transition because, if the pre-
 351 cision of the computed solutions is too poor, then any estimation can be considered
 352 as a solution (i.e., a “success” in our experiments). Numerically, it seems to be hard
 353 to know *a priori* the desired precision on the solutions to observe phase transitions.
 354 Therefore, it is of interest to have numerical methods that can achieve reconstructions
 355 with high precision, i.e., up to the machine precision.

356 First we describe the experimental setup. In these experiments the sensing matrix
 357 A always has 1000 columns. The entries of A are drawn from i.i.d. realizations of
 358 a centered Gaussian distribution. Without loss of generality we may normalize the
 359 columns of A to unit Euclidean norm. The number of rows of A , i.e., the dimension
 360 of the ambient space m , vary in $M := \{50, \dots, 325\}$ with increments of 25. For each
 361 number of rows, we vary the *sparsity level* s between 5% and 40% with increments
 362 of 5% and therefore consider the discrete set $S := \{0.05, \dots, 0.4\}$. The sparsity level
 363 is related to the ℓ^0 norm of \mathbf{u} by “ $\|\mathbf{u}\|_{\ell^0} = \text{round}(s \times m)$ ” following [10]. The posi-
 364 tions of the non-zero entries of \mathbf{u} are chosen randomly, with uniform probability. The
 365 non-zero entries of \mathbf{u} are drawn from a uniform distribution on $[-1, 1]$. To do so, for
 366 each parameter (i.e., sparsity level s and dimension of ambient space m) we repeated
 367 the experiments 1,000 times. The implementations of AISS and SPGL1 we used are
 368 the ones given by the authors of [7, 32, 41]. For LARS [18], we used the SPAMS
 369 toolbox [26]. The implementation of SeDuMi [38] we used can be found at https://sedumi.ie.lehigh.edu/sedumi/files/sedumi-downloads/SeDuMi_1.3.zip. Default pa-
 370 rameters have been used for all methods. We now give the criteria used for the
 371 numerical comparisons of these numerical algorithms.
 372

373 We need to define the “*success*” of an algorithm. We choose to define “*success*”
 374 as “the output of an algorithm is equal to the source element \mathbf{u} ”. This choice can
 375 be justified by several theoretical works, see, e.g., [11, 12, 16, 17]. This criterion,
 376 namely *the output is equal to the source element*, is chosen for the numerical experi-
 377 ments proposed thereafter. Note that this criterion seems slightly in favor of methods
 378 specifically designed for the compressive sensing method compared to methods that
 379 propose to solve (P_{ℓ^1}) . Here, this means that the comparisons are slightly biased in
 380 favor of [33, 35]. We also need to deal with the finite numerical precision of computa-
 381 tions. Thus, we define that a reconstruction is a *success* if the relative error satisfies
 382 $\frac{\|\mathbf{u} - \mathbf{u}_{est}\|_{\ell^2}}{\|\mathbf{u}\|_{\ell^2}} < \varepsilon$, where $\varepsilon = 10^{-10}$ or $\varepsilon = 10^{-4}$. Hence, for any $(m, s) \in M \times S$, the
 383 empirical probability of success is given by

$$384 \quad (4.1) \quad P_{(m,s)} := \frac{1}{\# \text{ of tests}} \sum_i \mathbb{1}_{\left\{ \frac{\|\mathbf{u}^i - \mathbf{u}_{est}^i\|_{\ell^2}}{\|\mathbf{u}^i\|_{\ell^2}} < \varepsilon \right\}}(i),$$

385 where \mathbf{u}_{est}^i (resp. \mathbf{u}^i) is the estimated signal (resp. source signal). Each method
 386 is tested on the same data by using the same random seed. Note that this type of
 387 experimental setup has been used before, for instance in [25].

388 Remark that another choice for defining “*success*” could be stated as “the output
 389 of an algorithm is a solution to (P_{ℓ^1}) ”. However, this criterion would be verified

390 for every output of algorithm 2.1. Indeed, algorithm 2.1 ends with some $\bar{\mathbf{u}}$ that
 391 numerically verifies an optimality condition associated with (P_{ℓ^1}) . Thus, this choice
 392 seems uninformative. Therefore, we have decided to not consider this definition of
 393 “success” in this paper. We first consider $\varepsilon = 10^{-10}$. Figure 4.1 depicts the empirical
 394 probability of success (4.1) for AISS, LARS, SPGL1, SeDuMi and algorithm 2.1.
 395 We also consider the difference of probability of success between algorithm 2.1 and
 396 all other methods that is defined as follows

$$397 \quad (4.2) \quad D_{(m,s)} := P_{(m,s)}^{\text{algorithm 2.1}} - P_{(m,s)},$$

398 where $m \in M$, $s \in S$, $P_{(m,s)}^{\text{algo 2.1}}$ (resp. $P_{(m,s)}$) denotes the quantity (4.1) obtained with
 399 algorithm 2.1 (resp. AISS, LARS and SPGL1). Note that a positive (negative) value
 400 in (4.2) means that algorithm 2.1 achieves a higher (lower) probability of success than
 401 the compared algorithm. These differences of probability of success are depicted in
 402 figure 4.2. We deduce from figure 4.2 that algorithm 2.1 always achieves a higher
 403 probability of success than AISS and GISS. We observe that LARS, SeDuMi (stan-
 404 dard precision) and SPGL1 algorithms do not perform well for $\varepsilon = 10^{-10}$ since the
 405 probability of success tends to be low, even for problems with very sparse signals. We
 406 also observe that both SeDuMi (high precision) and our proposed algorithm produce
 407 the best results. Table 4.1 gives the main assumptions on A and \mathbf{b} for LARS [18],
 408 SPGL1 [40, 41], AISS [7], SeDuMi [38] and algorithm 2.1 In this table, we also give
 409 the empirical probability that at least $x\%$ of signals are successfully reconstructed for
 410 each method. This statistical indicator is defined as follows

$$411 \quad (4.3) \quad P_{\geq x} = \frac{\#\{(m,s) \in M \times S : P_{(m,s)} \geq x\}}{\#M \cdot \#S},$$

412 where $P_{(m,s)}$ is defined by (4.1) and $\#$ denotes the cardinality of a set. Supplemen-
 413 tary material presents numerical results in terms of ℓ^1 -norm for ℓ^1 -based methods
 414 namely AISS, LARS, SPGL1, SeDuMi and algorithm 2.1. Up to a probability of 0.95
 415 AISS, SeDuMi (HP) and our algorithm give the same best results. For probability
 416 0.99 SeDuMi and our algorithm give the same best results. For higher probabilities
 417 Algorithm 2.1 gives the best results.

418 Table 4.2 presents the time results for AISS, LARS, SPGL1, SeDuMi and Al-
 419 gorithm 2.1. All experiments are done using a single core of an Intel Core 10600k.
 420 We observe that our proposed algorithm is very competitive compared to the state-
 421 of-the-art competitors. Indeed, our proposed algorithm outperforms the competitors
 422 for sparsity 5/10% and 50/175 rows while the second best algorithm is AISS. The
 423 computational time of our proposed algorithm is similar to AISS for sparsity 15/20%
 424 and 175/300 rows. For sparsity 25/30% and 175/300 rows AISS performs better than
 425 our proposed algorithm. We observe that the runtime of LARS [18], SPGL1 [40, 41],
 426 SeDuMi (SP) [38] remains close to constant when the sparsity is greater or equals
 427 20%: this suggests that for these levels of sparsity LARS [18], SPGL1 [40, 41], Se-
 428 DuMi [38] computed poor solutions as it has been numerically exhibited previously.
 429 Recall that SeDuMi (HP) [38] computes very good results as previously shown but
 430 the computational time is significantly larger than our proposed algorithm 2.1 and
 431 AISS except for the case of 30% sparsity with 300 rows.

432 As noted above the numerical results for LARS and SPGL1 show that these
 433 two numerical methods are not able to produce good results for the above set of
 434 experiments with $\varepsilon = 10^{-10}$. We now present numerical experiments for a higher
 435 threshold in (4.1) where we set $\varepsilon = 10^{-4}$. Figure 4.3 depicts the empirical probability

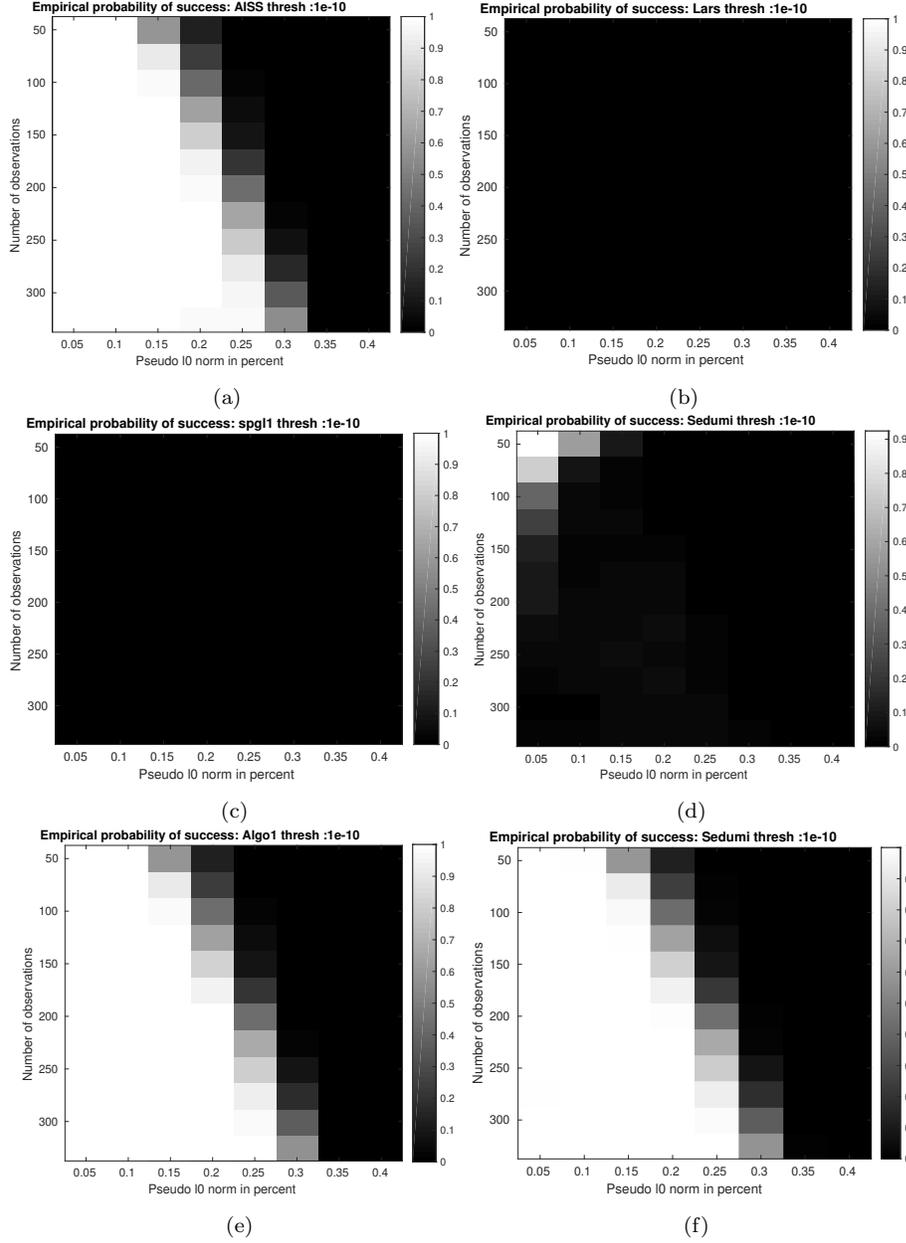


FIG. 4.1. Empirical probability of success (4.1), with $\varepsilon = 10^{-10}$. Panel (a): AISS [7], panel (b): LARS [18], panel (c): SPGL1 [40, 41], panel (d) : SeDuMi (standard precision) [38], panel (e): algorithm 2.1 and panel (f): SeDuMi (high precision) [38]. The non-zero entries of the source element \mathbf{u} are drawn from a uniform distribution on $[-1, 1]$. The entries in A are drawn from i.i.d. realizations of a Gaussian distribution. With their default parameters LARS, SPGL1 and SeDuMi (standard precision) are not able to produce good result for the above set of experiments. However, SeDuMi (high precision) produces good results. We also present results for an higher threshold $\varepsilon = 10^{-4}$, see figure 4.3.

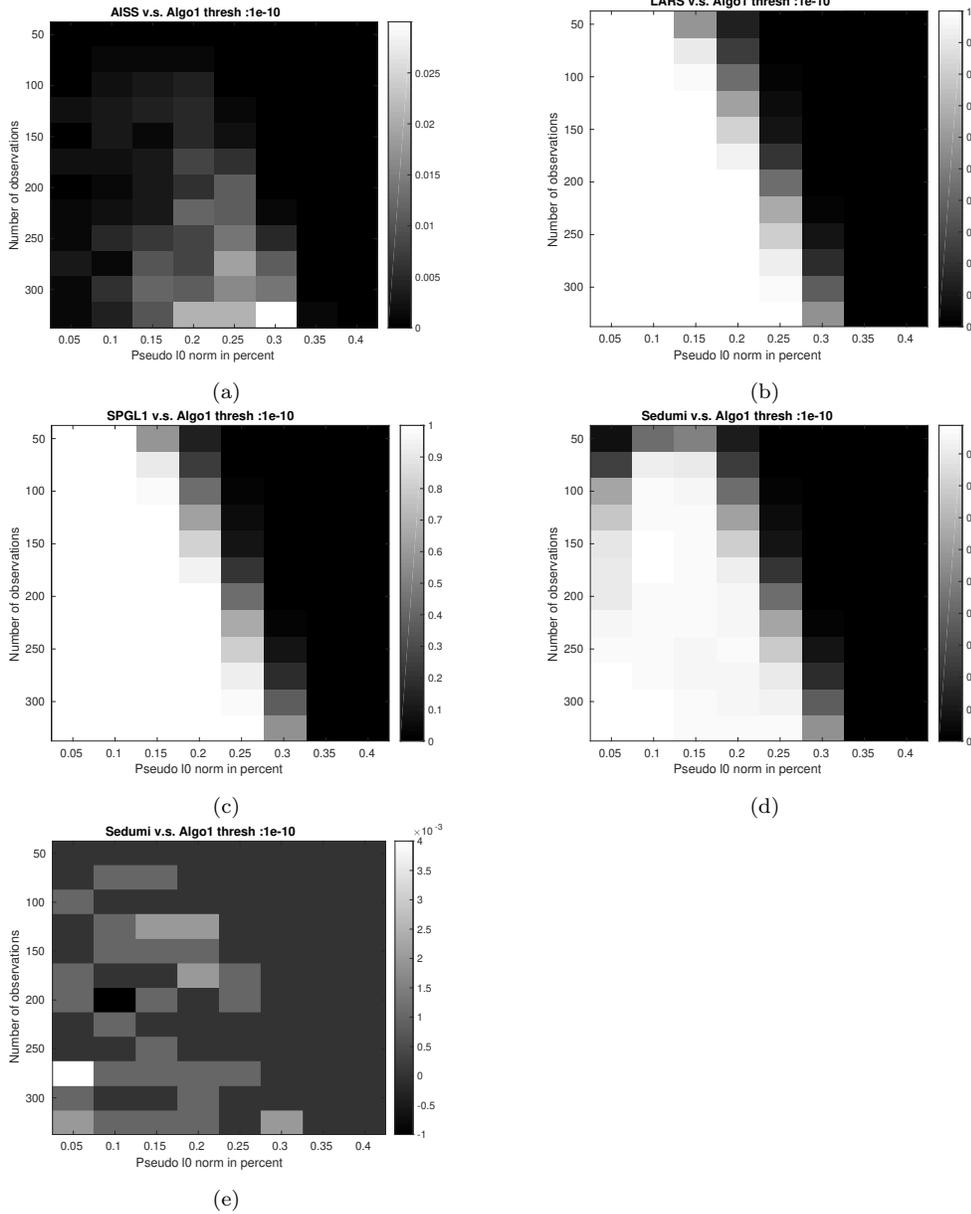


FIG. 4.2. Differences of probability of success (4.2), with $\varepsilon = 10^{-10}$. panel (a): algorithm 2.1-AISS [7] panel panel (b): algorithm 2.1-LARS [18] panel (c): algorithm 2.1-SPGL1 [40, 41], panel (d) algorithm 2.1-SeDuMi (standard precision) [38] and panel (e) algorithm 2.1-SeDuMi (high precision) [38]. A positive value indicates that algorithm 2.1 achieves a higher probability of success than the considered method, a negative value the contrary.

436 of success (4.1) for AISS, LARS, SPGL1, SeDuMi and algorithm 2.1. Figure 4.4
 437 depicts the differences of probability of success. These results for $\varepsilon = 10^{-4}$ show
 438 that all numerical algorithms have a higher empirical probability of success compared
 439 to the results for $\varepsilon = 10^{-10}$. In particular, we note that SPGL1 and LARS that
 440 were performing poorly for $\varepsilon = 10^{-10}$ have dramatically improved their performance.

TABLE 4.1

Main assumption and statistical indicator of “success” for LARS, SPGL1, AISS, SeDuMi and algorithm 2.1. The numbers without parentheses correspond to $\varepsilon = 10^{-10}$ and those between parentheses correspond to $\varepsilon = 10^{-4}$. Below, R.I.C. stands for restricted isometry constant see, e.g., [33] and S.F.P.D. stands for strong feasibility of primal and dual program.

Algorithm	LARS [18]	SPGL1 [40, 41]	AISS [7]	SeDuMi (SP) [38]	SeDuMi (HP) [38]	Algorithm 2.1
Assumption	R.I.C.	$\exists u : Au = b$	$\exists u : Au = b$	S.F.P.D.	S.F.P.D.	full row rank
$P_{\geq 0.9}$ (4.3)	0 (0.4688)	0 (0.0833)	0.4688 (0.4688)	0.104 (0.4688)	0.4688 (0.4688)	0.4688 (0.4688)
$P_{\geq 0.95}$ (4.3)	0 (0.4375)	0 (0.0625)	0.4375 (0.4375)	0 (0.4583)	0.4375 (0.4375)	0.4375 (0.4375)
$P_{\geq 0.99}$ (4.3)	0 (0.4167)	0 (0.0104)	0.3438 (0.4167)	0 (0.4167)	0.4167 (0.4167)	0.4167 (0.4167)
$P_{\geq 0.999}$ (4.3)	0 (0.3750)	0 (0)	0.1250 (0.3646)	0 (0.3333)	0.3229 (0.3437)	0.3646 (0.3750)
$P_{\geq 1}$ (4.3)	0 (0.3646)	0 (0)	0.0521 (0.3646)	0 (0.1875)	0.1562 (0.1979)	0.3333 (0.3646)

441 Also, from figure 4.4, we observe that LARS and our proposed algorithm produce
 442 very similar results. It seems that LARS works for the considered experiments (see
 443 figure 4.4) although it was proved in [6] that LARS may not converge.

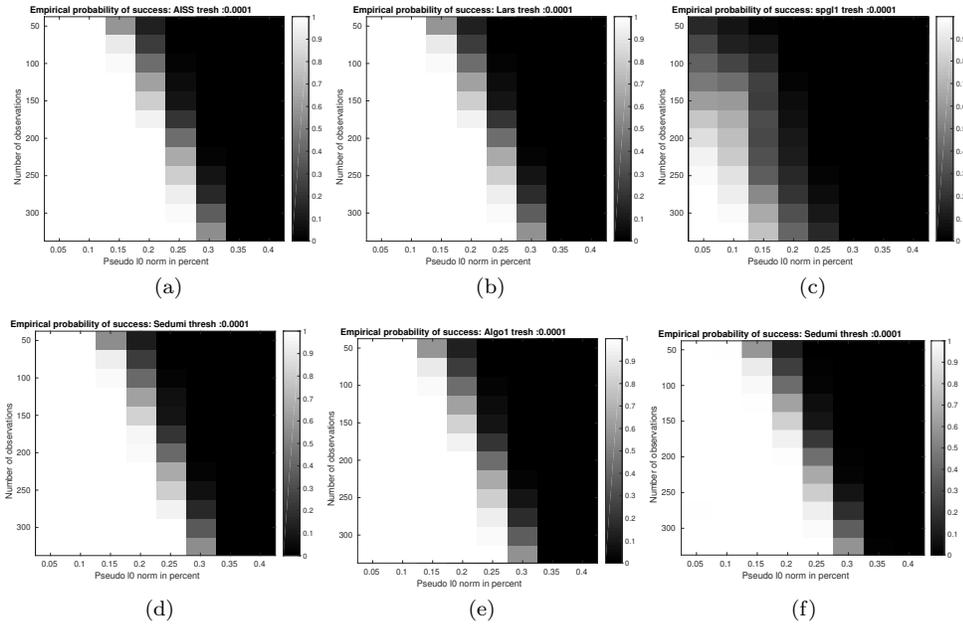


FIG. 4.3. Empirical probability of success (4.1), with $\varepsilon = 10^{-4}$. Panel (a): AISS [7], panel (b): LARS [18], panel (c): SPGL1 [40, 41], panel (d) : SeDuMi (standard precision) [38], panel (e): algorithm 2.1 and panel (f): SeDuMi (high precision) [38]. The non-zero entries of the source element u are drawn from a uniform distribution on $[-1, 1]$. The entries in A are drawn from i.i.d. realizations of a Gaussian distribution.

444 **5. Conclusion.** In this paper, a new algorithm to solve ℓ^1 regularized linear
 445 problems up to the machine precision has been proposed. The method is based on i)
 446 the numerical computation of a finite sequence that converges to a solution the dual
 447 problem and ii) an explicit recovery formula -based on a non-negative least squares-
 448 to compute a solution to the primal problem. The sequence we employed is driven
 449 by an evolution equation ruled by a maximal monotone operator. The numerical
 450 computations of this algorithm involve: the computation of a projection onto a closed
 451 convex cone and the evaluation of inner products. The sequence in the dual space
 452 lives in a low dimensional space compared to the unknown. Hence, most of the
 453 numerical efforts require fewer memory usage than primal-based method. Numerical

TABLE 4.2

Computational time results for the following methods: algorithm 2.1, AISS [7] and SeDuMi (SP) [38], SPGL1 [40, 41], LARS [18], SeDuMi (HP) [38], the number of columns is set to 1,000 as everywhere else in this paper and various number of rows (NR) and several level of sparsity. Time results are given in seconds and corresponds to the average time of 200 experiments. The variance is also given in parenthesis.

NR	algorithm	sparsity		
		5%	10%	15%
50	Algorithm 2.1	6.2437e-04 (1.1274e-08)	9.9688e-04 (4.6010e-07)	0.0056 (1.2178e-05)
	AISS [7]	0.0046 (6.2449e-04)	0.0053 (5.8065e-04)	0.0135 (5.9706e-04)
	SeDuMi (SP) [38]	0.0326 (2.9972e-04)	0.0369 (2.7969e-04)	0.0497 (3.4206e-04)
	SPGL1 [40, 41]	0.0115 (3.5370e-05)	0.0240 (2.2200e-04)	0.0693 (4.0352e-04)
	LARS [18]	0.0059 (1.2202e-06)	0.0069 (3.5498e-06)	0.0100 (7.6192e-06)
	SeDuMi (HP) [38]	0.2029 (5.4966e-04)	0.2130 (5.9799e-04)	0.2330 (0.0010)
175	Algorithm 2.1	0.0033 (1.1617e-07)	0.0061 (5.7905e-07)	0.0161 (2.6142e-05)
	AISS [7]	0.0052 (5.6492e-04)	0.0071 (6.6601e-04)	0.0153 (7.7310e-04)
	SeDuMi (SP) [38]	0.1497 (3.6402e-04)	0.1714 (3.4310e-04)	0.1729 (4.1359e-04)
	SPGL1 [40, 41]	0.0105 (2.3802e-05)	0.0192 (3.6582e-05)	0.0372 (2.3437e-04)
	LARS [18]	0.0097 (7.4096e-06)	0.0159 (2.9454e-05)	0.0198 (2.0914e-05)
	SeDuMi (HP) [38]	0.5813 (0.0027)	0.6406 (0.0038)	0.6787 (0.0028)
300	Algorithm 2.1	0.0081 (1.1997e-07)	0.0162 (1.9130e-06)	0.0411 (6.8016e-05)
	AISS [7]	0.0063 (6.0184e-04)	0.0102 (5.8937e-04)	0.0283 (8.1150e-04)
	SeDuMi (SP) [38]	0.3502 (5.3829e-04)	0.3724 (4.9952e-04)	0.3854 (5.9227e-04)
	SPGL1 [40, 41]	0.0112 (2.1614e-05)	0.0181 (2.6325e-05)	0.0299 (5.3353e-05)
	LARS [18]	0.0244 (1.7508e-05)	0.0300 (3.3939e-05)	0.0369 (4.4780e-05)
	SeDuMi (HP) [38]	1.2914 (0.0152)	1.4947 (0.0179)	1.5797 (0.0143)
NR	algorithm	sparsity		
		20%	25%	30%
50	Algorithm 2.1	0.0085 (3.6050e-06)	0.0091 (1.0906e-06)	0.0092 (7.7791e-07)
	AISS [7]	0.0186 (7.7498e-04)	0.0197 (7.6886e-04)	0.0195 (7.6616e-04)
	SeDuMi (SP) [38]	0.0547 (3.6486e-04)	0.0539 (3.0153e-04)	0.0538 (3.0346e-04)
	SPGL1 [40, 41]	0.0755 (3.2247e-04)	0.0779 (2.6061e-04)	0.0791 (2.8112e-04)
	LARS [18]	0.0056 (1.2995e-06)	0.0059 (1.2474e-06)	0.0601 (1.2694e-06)
	SeDuMi (HP) [38]	0.2407 (9.3084e-04)	0.2496 (0.0010)	0.2537 (0.0012)
175	Algorithm 2.1	0.0997 (0.0165)	0.4943 (0.0270)	0.5521 (0.0028)
	AISS [7]	0.0866 (0.0083)	0.3485 (0.0115)	0.3739 (0.0024)
	SeDuMi (SP) [38]	0.2077 (0.0011)	0.2491 (0.0013)	0.2534 (4.9534e-04)
	SPGL1 [40, 41]	0.1187 (0.0026)	0.1421 (0.0015)	0.1334 (9.7981e-04)
	LARS [18]	0.0199 (1.7296e-05)	0.0173 (3.1513e-06)	0.0209 (1.6831e-05)
	SeDuMi (HP) [38]	0.7787 (0.0170)	.9592 (0.0147)	0.9820 (0.0054)
300	Algorithm 2.1	0.1402 (0.0035)	0.7100 (0.3449)	3.6610 (2.1706)
	AISS [7]	0.1039 (0.0025)	0.4710 (0.1126)	2.0890 (0.5683)
	SeDuMi (SP) [38]	0.4069 (7.2509e-04)	0.4403 (0.0021)	0.5739 (0.0049)
	SPGL1 [40, 41]	0.0591 (4.4726e-04)	0.1639 (0.0038)	0.2013 (0.0019)
	LARS [18]	0.0435 (4.1577e-05)	0.0515 (2.4983e-05)	0.0532 (9.1295e-06)
	SeDuMi (HP) [38]	1.7613 (0.0235)	1.8634 (0.0752)	2.6824 (0.1787)

454 comparisons with other existing state-of-the-art methods is exhibited for noiseless
455 compressive sensing (basis pursuit) problems.

456 The numerical comparisons above showed that our algorithm compares advanta-
457 geously with existing methods: the phase transition is observed with a higher accuracy.
458 The algorithm proposed in this paper is parameter-less once a starting point has been
459 chosen. However, the starting point can be tuned to further speed up the method. A
460 future work could study the impact of this choice in terms of convergence speed.

461 We also leave as future work theoretical and numerical comparisons with approx-
462 imate path-methods (as opposed to piecewise affine paths such as our approach) such
463 as [27] which corresponds to an approximate discretization of trajectories. In partic-

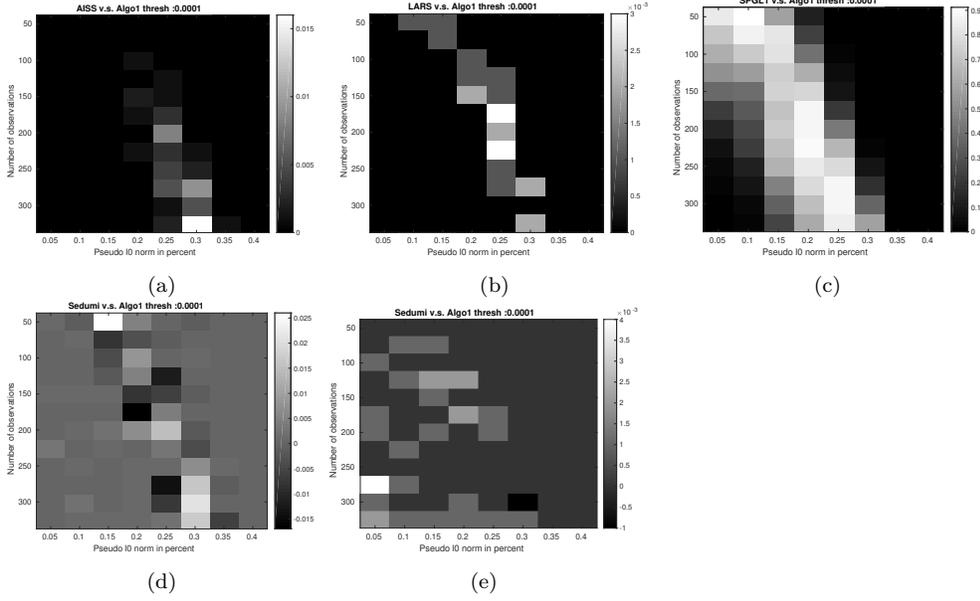


FIG. 4.4. Differences of probability of success (4.2), with $\varepsilon = 10^{-4}$. Panel (a): algorithm 2.1-AISS [7], panel (b): algorithm 2.1-LARS [18], panel (c): algorithm 2.1-SPGL1 [40, 41], panel (d) algorithm 2.1-SeDuMi (standard precision) [38] and panel (e) algorithm 2.1-SeDuMi (high precision) [38]. A positive value indicates that algorithm 2.1 achieves a higher probability of success than the considered method, a negative value the contrary.

ular, it would be interest to know if it is better to compute an exact trajectory versus
 an approximate trajectory from a computational point of view.

6. Appendix. This section contains several proofs used throughout this paper
 and some properties on the projection on polyhedral convex cone.

6.1. Some properties of functions J, J^*, f and \mathcal{J} .

LEMMA 6.1 (Some elementary properties of J and J^*). *We posit the same
 assumptions as in proposition 3.1. We have*

1. $J \in \Gamma_0(\mathbb{R}^n)$, $\text{dom}(J) = \mathbb{R}^n$, $J^* = \chi_{\mathbf{B}_\infty} \in \Gamma_0(\mathbb{R}^n)$ and $\text{dom}(J^*) = B_\infty$;
2. (Primal feasibility)

$$(6.1) \quad \mathbf{0} \in \text{int}(A \text{ dom } J - \{\mathbf{b}\}) = A \mathbb{R}^n - \{\mathbf{b}\} = \mathbb{R}^m \text{ (see (iv))};$$

3. (Dual feasibility)

$$(6.2) \quad \mathbf{0} \in \text{int}\left(A^T \text{dom } \chi_{\{\mathbf{b}\}}^* + \text{dom } J^*\right) = \text{int}\left(\text{span } A^T + B_\infty\right).$$

Proof. We sequentially prove the three assertions.

Note that $\text{dom } J = \mathbb{R}^n$ and that J is convex. It follows that $J \in \Gamma_0(\mathbb{R}^n)$ and, from
 theorem 8.10, that $J^* \in \Gamma_0(\mathbb{R}^n)$. Combining lemma 8.2 with proposition 8.4 we
 obtain that for any $\mathbf{u} \in \mathbb{R}^n$ we have $J^*(\mathbf{u}) = \chi_{\mathbf{B}_\infty}(\mathbf{u})$ and $\text{dom } J^* = B_\infty$.

From $\text{dom } J = \mathbb{R}^n$ and the assumption that A has full row rank, we have $A \text{ dom } J =$
 $\text{span } A = \mathbb{R}^m$ and (6.1) immediately follows.

Applying lemma 8.3 with $C := \{\mathbf{b}\}$ we have $\chi_{\{\mathbf{b}\}}^*(\cdot) = \langle \cdot, \mathbf{b} \rangle \in \Gamma_0(\mathbb{R}^m)$ and also

483 $\text{dom} \left(\chi_{\{\mathbf{b}\}}^*(\cdot) \right) = \mathbb{R}^m$. Since, in addition, $\text{dom} J^* = B_\infty$, we have

$$484 \quad (6.3) \quad A^T \text{dom} \chi_{\{\mathbf{b}\}}^* + \text{dom} J^* = B_\infty + \text{span} A^T.$$

485 We have obviously have $B_\infty \subset B_\infty + \text{span} A^T$, and from (6.3) we deduce (6.2). \square

486 PROPOSITION 6.2 (and definition : function \mathcal{J}). *We posit the same assumptions*
487 *as in proposition 3.1. Consider the function $\mathcal{J} : \mathbb{R}^m \rightarrow \mathbb{R} \cup \{+\infty\}$ defined by*

$$488 \quad (6.4) \quad \forall \boldsymbol{\lambda} \in \mathbb{R}^m, \quad \mathcal{J}(\boldsymbol{\lambda}) := J^*(-A^T \boldsymbol{\lambda}) = \chi_C(\boldsymbol{\lambda})$$

489 where C is defined by (3.7). We have $\mathcal{J} \in \Gamma_0(\mathbb{R}^m)$ and $\text{dom} \mathcal{J} = C \neq \emptyset$.

490 *Proof.* From item 1 of lemma 6.1 we have $J^* \in \Gamma_0(\mathbb{R}^n)$. Note that (6.2) in
491 lemma 6.1 implies that $\text{span} A^T \cap \text{dom} J^* \neq \emptyset$. Then, from theorem 8.5 we obtain
492 that $\mathcal{J}(\cdot) := J^*(-A^T \cdot) \in \Gamma_0(\mathbb{R}^m)$. Moreover, for any $\boldsymbol{\lambda} \in \mathbb{R}^m$, we have

$$493 \quad (6.5) \quad \mathcal{J}(\boldsymbol{\lambda}) = \chi_{B_\infty}(A^T \boldsymbol{\lambda}) = \chi_C(\boldsymbol{\lambda}).$$

494 The first equality in (6.5) is justified by combining item 1 ($J^* = \chi_{B_\infty}$) of lemma 6.1
495 and that $-A^T \boldsymbol{\lambda} \in B_\infty \Leftrightarrow A^T \boldsymbol{\lambda} \in B_\infty$. The second equality in (6.5) is justified by the
496 fact that $\boldsymbol{\lambda} \in C \Leftrightarrow A^T \boldsymbol{\lambda} \in B_\infty$. Indeed, we have

$$497 \quad (6.6) \quad \boldsymbol{\lambda} \in C \Leftrightarrow \langle \boldsymbol{\lambda}, A\tilde{\mathbf{e}}_i \rangle \leq 1 \quad \forall i = \{1, \dots, 2n\} \Leftrightarrow A^T \boldsymbol{\lambda} \in B_\infty,$$

498 where $\tilde{\mathbf{e}}_i$ is defined by (3.3). The first equivalence in (6.6) is obvious from the definition
499 of C given by (3.7). The last equivalence follows from the definition of the $\ell^\infty(\mathbb{R}^n)$
500 unit ball (see (vii)). From (6.5) we can verify that $\text{dom} \mathcal{J} = C$ and that $\mathbf{0} \in C \neq \emptyset$. \square

501 LEMMA 6.3 (Subdifferential formulas for f , J^* and \mathcal{J}). *We posit the same*
502 *assumptions as in proposition 3.1. We have*

503 1. (Subdifferential formula for f)

$$504 \quad (6.7) \quad \forall \mathbf{u} \in \text{dom} J \cap \text{dom} \chi_{\{\mathbf{b}\}}(A \cdot) \quad \partial f(\mathbf{u}) = \partial J(\mathbf{u}) + A^T \partial \chi_{\{\mathbf{b}\}}(A\mathbf{u});$$

505 2. (subdifferential formula for J^*)

$$506 \quad (6.8) \quad \forall \boldsymbol{\lambda} \in \text{dom} g \quad \partial J^*(-A^T \boldsymbol{\lambda}) = N_{B_\infty}(-A^T \boldsymbol{\lambda}) = \text{co}\{\tilde{\mathbf{e}}_i : i \in S(-\boldsymbol{\lambda})\};$$

507 3. (Subdifferential formula for \mathcal{J})

$$508 \quad (6.9) \quad \forall \boldsymbol{\lambda} \in C \quad \partial \mathcal{J}(\boldsymbol{\lambda}) = AN_{B_\infty}(A^T \boldsymbol{\lambda}) = \text{co}\{A\tilde{\mathbf{e}}_i : i \in S(\boldsymbol{\lambda})\},$$

509 where $N_{B_\infty}(A^T \boldsymbol{\lambda})$ is the normal cone to B_∞ at $A^T \boldsymbol{\lambda} \in \mathbb{R}^n$ (see (vi)), the
510 set $S(\boldsymbol{\lambda})$ is defined by (3.3) and the $2n$ vectors $\tilde{\mathbf{e}}_i$ of \mathbb{R}^n are defined by (3.3)
511 (page 5).

512 *Proof.* We sequentially justify (6.7)-(6.9). Combining (6.1) in lemma 6.1 and
513 theorem 8.16 (with “ $U = J$ ” and “ $V = \chi_{\{\mathbf{b}\}}$ ”) we immediately obtain (6.7). The
514 first equality in (6.8) is justified by lemma 8.6. The second equality in (6.8) follows
515 from lemma 8.7 applied with $p := 2n$, $\mathbf{s}_i := \mathbf{e}_i$ for $i = 1, \dots, n$, $\mathbf{s}_i := -\mathbf{e}_{i-n}$ for $i =$
516 $n+1, \dots, 2n$, $r_i := 1$ for $i = 1, \dots, 2n$ and $W(-A^T \boldsymbol{\lambda}) = S(-\boldsymbol{\lambda})$. We now justify (6.9).
517 From (6.4) in proposition 6.2 we have $\mathcal{J}(\cdot) = J^*(A^T \cdot)$. To prove the first equality
518 in (6.9), we need to justify that

$$519 \quad (6.10) \quad \text{int}(\text{dom} J^*) \cap \text{span} A^T \neq \emptyset.$$

520 Assuming that (6.10) holds true, combining item 1 in lemma 6.1 ($J^* \in \Gamma_0(\mathbb{R}^n)$) and
 521 theorem 8.9 (with “ $f = J^*$ ”) we obtain that $\partial\mathcal{J}(\boldsymbol{\lambda}) = -A\partial J^*(-A^T\boldsymbol{\lambda})$. We notice
 522 that

$$523 \quad (6.11) \quad -\partial J^*(-A^T\boldsymbol{\lambda}) = \text{co}\{-\tilde{\boldsymbol{e}}, i \in S(-\boldsymbol{\lambda})\} = \text{co}\{\tilde{\boldsymbol{e}}, i \in S(\boldsymbol{\lambda})\} = \partial J^*(A^T\boldsymbol{\lambda}).$$

524 Indeed, the first equality in (6.11) is justified by (6.8). The second equality is obvious
 525 from the definition of $S(\boldsymbol{\lambda})$ in definition 3.3 and the last equality follows. From (6.11)
 526 we immediately obtain (6.9). We now justify (6.10). From, again, item 1 in lemma 6.1
 527 we have $\text{dom } J^* = B_\infty$ and therefore deduce that $\mathbf{0} \in \text{int}(\text{dom } J^*) \cap \text{span } A^T$ which
 528 justifies (6.10). This concludes our proof. \square

529 **6.2. Proof of proposition 3.1 on page 5.**

530 *Proof.* We first prove that $f \in \Gamma_0(\mathbb{R}^n)$ then that $g \in \Gamma_0(\mathbb{R}^m)$ and that (3.2) hold
 531 true. From the assumption that A has a full row rank, it follows that $\text{span } A \cap$
 532 $\text{dom } \chi_{\{\mathbf{b}\}} = \text{span } A \cap \{\mathbf{b}\} \neq \emptyset$. In addition, $\chi_{\{\mathbf{b}\}} \in \Gamma_0(\mathbb{R}^m)$ as the characteris-
 533 tic function of the closed convex set $\{\mathbf{b}\}$. Therefore, from proposition 8.5 we de-
 534 duce that $\chi_{\{\mathbf{b}\}}(A\cdot) \in \Gamma_0(\mathbb{R}^n)$. Combining lemma 6.1 and proposition 8.10 we have
 535 that $f \in \Gamma_0(\mathbb{R}^n)$ as the sum of the finite valued convex function $J \in \Gamma_0(\mathbb{R}^n)$ and
 536 $\chi_{\{\mathbf{b}\}}(A\cdot) \in \Gamma_0(\mathbb{R}^n)$.

537 Now we prove that $g \in \Gamma_0(\mathbb{R}^m)$. From proposition 6.2 the function $\mathcal{J}(\cdot) = J^*(-A^T\cdot) \in$
 538 $\Gamma_0(\mathbb{R}^m)$. From proposition 8.10 we obtain that $g \in \Gamma_0(\mathbb{R}^m)$ as the sum of the finite
 539 valued convex function $\langle \mathbf{b}, \cdot \rangle$ and $J^*(-A^T\cdot) \in \Gamma_0(\mathbb{R}^m)$. The second equality in (3.2)
 540 follows from item 1 ($J^* = \chi_{B_\infty}$) of lemma 6.1. This concludes our proof. \square

541 **6.3. Proof of proposition 3.4 on page 5.**

542 *Proof.* The proof is in two steps. Step 1 proves that problems (P_{ℓ^1}) and (D_{ℓ^1})
 543 have at least one solution. Step 2 justifies (3.4) and (3.5).

544 **Step 1. Problems (P_{ℓ^1}) and (D_{ℓ^1}) have at least one solution and .** Com-
 545 bining the definitions of function f and g given in proposition 3.1, lemma 6.1, propo-
 546 sition 8.17 and theorem 8.18 with $U := J$ and $V := \chi_{\{\mathbf{b}\}}$ we conclude that prob-
 547 lems (P_{ℓ^1}) and (D_{ℓ^1}) have at least one solution. This concludes step1. We now turn
 548 to step2.

549 **Step2. Formulas (3.4) and (3.5) hold true.** From [1, p. 166-167] applied
 550 with “ $U := J$ and $V := \chi_{\{\mathbf{b}\}}$ ” we have that any point $\bar{\mathbf{u}}$ in the non-empty closed
 551 convex set $\mathcal{S}(\bar{\boldsymbol{\lambda}}) = \partial J^*(-A^T\bar{\boldsymbol{\lambda}}) \cap \{\mathbf{u} : A\mathbf{u} = \mathbf{b}\}$ is a solution to (P_{ℓ^1}) . The set $\mathcal{S}(\bar{\boldsymbol{\lambda}})$ is
 552 non-empty and from step 1 the primal has a solution. Consider $\bar{\boldsymbol{\lambda}}$ solution to (D_{ℓ^1}) .
 553 Combining theorem 8.11 and lemma 3.8 we obtain that $\mathbf{b} \in \text{co}\{-A\tilde{\boldsymbol{e}}, i \in S(\bar{\boldsymbol{\lambda}})\} =$
 554 $\text{co}\{A\tilde{\boldsymbol{e}}, i \in S(-\bar{\boldsymbol{\lambda}})\}$. This means that \mathbf{b} can be written as $\mathbf{b} = \sum_{i=1}^{2n} \tilde{u}_i A\tilde{\boldsymbol{e}}_i$, where
 555 $\tilde{u}_i \geq 0, \forall i \in (-\bar{\boldsymbol{\lambda}})$ and $\tilde{u}_i = 0 \forall i \in \{1, \dots, 2n\} \setminus S(-\bar{\boldsymbol{\lambda}})$. Consider $\bar{\mathbf{u}}$ defined by (3.5).
 556 It is easy to see that $\bar{\mathbf{u}} \in \mathcal{S}(\bar{\boldsymbol{\lambda}})$ and therefore $\bar{\mathbf{u}}$ is a solution to (P_{ℓ^1}) . This concludes
 557 our proof.

558 **6.4. Proof of proposition 3.6 on page 6.**

559 *Proof.* From proposition 3.1, we have $g \in \Gamma_0(\mathbb{R}^m)$. Hence, from proposition 8.12
 560 we immediately obtain that $\partial g(\cdot)$ is a maximal monotone operator. Item 1 of propo-
 561 sition 3.6 follows from theorem 8.14. Items 2 and 3 of proposition 3.6 follows from
 562 theorem 8.15. This concludes our proof. \square

563 **6.5. Proof of proposition 3.7 on page 7.**

564 *Proof.* Proposition 3.7 is obvious combining proposition 6.2 and (3.2) in proposi-
 565 tion 3.1. This concludes our proof. \square

566 **6.6. Proof of lemma 3.8 on page 7.**

567 *Proof.* Combining propositions 3.1 and 6.2, we have that g can be written as

$$568 \quad (6.12) \quad \forall \boldsymbol{\lambda} \in \mathbb{R}^m, \quad g(\boldsymbol{\lambda}) = \mathcal{J}(\boldsymbol{\lambda}) + \langle \boldsymbol{\lambda}, \mathbf{b} \rangle.$$

569 From, again, proposition 6.2, we deduce that $\text{int}(\text{dom } \mathcal{J}) \cap \text{int}(\text{dom } \langle \cdot, \mathbf{b} \rangle) \neq \emptyset$.
 570 Hence, combining theorem 8.13 and lemma 6.3 we obtain (3.8) and that $\forall \boldsymbol{\lambda} \in \text{dom } g =$
 571 C we have $\mathbf{b} \in \partial g(\boldsymbol{\lambda}) \neq \emptyset$. This concludes our proof. \square

572 **6.7. Proof of proposition 3.9 on page 7.**

573 *Proof.* We first establish the following lemma

574 LEMMA 6.4 (Directional derivative and Taylor formula for g).

575 *We posit the same assumptions as in proposition 3.1. For every $\boldsymbol{\lambda} \in \text{dom } g$ and any*
 576 *\mathbf{d} that satisfies (3.9) we have*

$$577 \quad (6.13) \quad g'(\boldsymbol{\lambda}, \mathbf{d}) = \langle \mathbf{d}, \mathbf{b} \rangle;$$

$$578 \quad (6.14) \quad g(\boldsymbol{\lambda} + t\mathbf{d}) = g(\boldsymbol{\lambda}) + tg'(\boldsymbol{\lambda}, \mathbf{d}) \quad \text{for some } t > 0 \text{ small enough.}$$

579 *Proof.* Let $\boldsymbol{\lambda} \in \text{dom } g$. By assumption we have that \mathbf{d} satisfies (3.9). Combining
 580 proposition 3.7 and (3.9) we immediately obtain that $(\boldsymbol{\lambda} + t\mathbf{d}) \in \text{dom } g$ for some $t > 0$
 581 small enough. Hence from the definition of g (3.7), we obtain that, for some small
 582 enough $t > 0$,

$$583 \quad (6.15) \quad g(\boldsymbol{\lambda} + t\mathbf{d}) - g(\boldsymbol{\lambda}) = \langle \boldsymbol{\lambda} + t\mathbf{d}, \mathbf{b} \rangle - \langle \boldsymbol{\lambda}, \mathbf{b} \rangle = t\langle \mathbf{d}, \mathbf{b} \rangle.$$

584 Formula (6.13) follows (xiii). Combining (6.13) and (6.15) we deduce (6.14). This
 585 concludes our proof. \square

586 We first prove that the conditions (3.9)-(3.10) are necessary then that they are suffi-
 587 cient.

588 If \mathbf{d} is a descent direction for g at $\boldsymbol{\lambda} \in \text{dom } g$ then, from definition 3.9, $(\boldsymbol{\lambda} + t\mathbf{d}) \in \text{dom } g$
 589 for some $t > 0$ small enough. From the definition of C (3.7) it follows that $\boldsymbol{\lambda} + t\mathbf{d}$ sat-
 590 isfies, in particular, $\langle \boldsymbol{\lambda} + t\mathbf{d}, A\tilde{\mathbf{e}}_i \rangle \leq 1$ for every $i \in S(\boldsymbol{\lambda})$. The definition of $S(\boldsymbol{\lambda})$ (3.3)
 591 and the fact that $\boldsymbol{\lambda} \in \text{dom } g$ imply that necessarily $\langle \mathbf{d}, A\tilde{\mathbf{e}}_i \rangle \leq 0$ for every $i \in S(\boldsymbol{\lambda})$
 592 and (3.9) holds true. In addition, from definition 3.9 we have $g(\boldsymbol{\lambda} + t\mathbf{d}) < g(\boldsymbol{\lambda})$ for
 593 some $t > 0$ enough small and combining (6.13)-(6.14) we obtain that (3.10) holds
 594 true. Hence, (3.9)-(3.10) are necessary conditions. We now turn to the sufficiency.

595 Conversely, consider $\mathbf{d} \in \mathbb{R}^m \setminus \{\mathbf{0}\}$ satisfying (3.9)-(3.10). From proposition 3.7 we
 596 have that $\boldsymbol{\lambda} \in \text{dom } g$ satisfies $\langle \boldsymbol{\lambda}, A\tilde{\mathbf{e}}_i \rangle \leq 1$ for every $i \in \{1, \dots, 2n\}$. On the one hand,
 597 from (3.9), for any $i \in S(\boldsymbol{\lambda})$ we have $\langle \mathbf{d}, A\tilde{\mathbf{e}}_i \rangle \leq 0$ and therefore $\langle \boldsymbol{\lambda} + t\mathbf{d}, A\tilde{\mathbf{e}}_i \rangle \leq 1$
 598 $\forall t > 0$. On the other hand, from $\boldsymbol{\lambda} \in \text{dom } g$ and the definition of $S(\boldsymbol{\lambda})$ we deduce that
 599 for any $i \in \{1, \dots, 2n\} \setminus S(\boldsymbol{\lambda})$ we have $\langle \boldsymbol{\lambda}, A\tilde{\mathbf{e}}_i \rangle < 1$ and therefore that $\langle \boldsymbol{\lambda} + t\mathbf{d}, A\tilde{\mathbf{e}}_i \rangle \leq 1$
 600 for $t > 0$ small enough. Thus, $\langle \boldsymbol{\lambda} + t\mathbf{d}, -A\tilde{\mathbf{e}}_i \rangle \leq 1$ for every $i \in \{1, \dots, 2n\}$ and
 601 $t > 0$ small enough. It follows that $(\boldsymbol{\lambda} + t\mathbf{d}) \in \text{dom } g$ for some $t > 0$ small enough.
 602 Combining (3.10) and (6.14) we obtain $g(\boldsymbol{\lambda} + t\mathbf{d}) < g(\boldsymbol{\lambda})$ for some $t > 0$ small enough.
 603 It follows that \mathbf{d} is a direction descent for g at $\boldsymbol{\lambda}$. This concludes our proof. \square

604 **6.8. Proof of proposition 3.10 on page 8.**

605 *Proof.* We recall that, in the sequel, $\lambda \in \text{dom } g$ and that \mathbf{d} is a descent direction
 606 for g at $\lambda \in \text{dom } g$. We sequentially consider the three following complementary cases
 607 **Case 1.** The case of indexes i such that $i \in S(\lambda)$.

608 **Case 2.** The case of $i \in \{1, \dots, 2n\} \setminus S(\lambda)$ and $\langle A\tilde{\mathbf{e}}_i, \mathbf{d} \rangle \leq 0$.

609 **Case 3.** The case of $i \in \{1, \dots, 2n\} \setminus S(\lambda)$ and $\langle A\tilde{\mathbf{e}}_i, \mathbf{d} \rangle > 0$.

610

611 **Case 1.** From proposition 3.9 equation (3.9), we have that any descent direction
 612 \mathbf{d} for g at $\lambda \in \text{dom } g$ satisfies $\langle \mathbf{d}, A\tilde{\mathbf{e}}_i \rangle \leq 0$ for every $i \in S(\lambda)$. From definition 3.3
 613 (page 5), for every $i \in S(\lambda)$, we have $\langle \lambda, A\tilde{\mathbf{e}}_i \rangle = 1$ and therefore deduce that

$$614 \quad (6.16) \quad i \in S(\lambda) \Rightarrow \langle \lambda + t\mathbf{d}, A\tilde{\mathbf{e}}_i \rangle \leq 1 \quad \forall t \geq 0.$$

615 **Case 2.** For any $i \in \{1, \dots, 2n\} \setminus S(\lambda)$, from $\lambda \in \text{dom } g$ we deduce that $\langle \lambda, A\tilde{\mathbf{e}}_i \rangle < 1$.

616 Hence, if $\langle A\tilde{\mathbf{e}}_i, \mathbf{d} \rangle \leq 0$ then

$$617 \quad (6.17) \quad i \in \{1, \dots, 2n\} \setminus S(\lambda) \text{ and } \langle A\tilde{\mathbf{e}}_i, \mathbf{d} \rangle \leq 0 \Rightarrow \langle \lambda + t\mathbf{d}, A\tilde{\mathbf{e}}_i \rangle < 1 \quad \forall t \geq 0.$$

618 From (6.17) it is easy to deduce that

$$619 \quad (6.18) \quad i \in \{1, \dots, 2n\} \setminus S(\lambda) \text{ and } \langle A\tilde{\mathbf{e}}_i, \mathbf{d} \rangle \leq 0 \Rightarrow i \in \{1, \dots, 2n\} \setminus S(\lambda + t\mathbf{d}) \quad \forall t \geq 0.$$

620 **Case 3.** We begin by noticing that

$$621 \quad (6.19) \quad \{i \in \{1, \dots, 2n\} \setminus S(\lambda) : \langle A\tilde{\mathbf{e}}_i, \mathbf{d} \rangle > 0\} = \{i \in \{1, \dots, 2n\} : \langle A\tilde{\mathbf{e}}_i, \mathbf{d} \rangle > 0\}.$$

622 Indeed, we recall that from proposition 3.9 any descent direction \mathbf{d} for g at λ implies
 623 that for every $i \in S(\lambda)$ we have $\langle \mathbf{d}, A\tilde{\mathbf{e}}_i \rangle \leq 0$. Hence, if $\langle A\tilde{\mathbf{e}}_i, \mathbf{d} \rangle > 0$ for some $i \in$
 624 $\{1, \dots, 2n\}$ then $i \notin S(\lambda)$. This means that $\{i \in \{1, \dots, 2n\} \setminus S(\lambda) : \langle A\tilde{\mathbf{e}}_i, \mathbf{d} \rangle > 0\} \subset$
 625 $\{i \in \{1, \dots, 2n\} : \langle A\tilde{\mathbf{e}}_i, \mathbf{d} \rangle > 0\}$ and therefore proves (6.19). The converse inclusion
 626 is trivial. Hence, *Case 3* is, from (3.11), the case defined by $S^+(\mathbf{d})$. For any $i \in S^+(\mathbf{d})$,
 627 it is easy to see that

$$628 \quad (6.20) \quad i \in S^+(\mathbf{d}) \Rightarrow \langle \lambda + t\mathbf{d}, A\tilde{\mathbf{e}}_i \rangle \leq 1 \text{ iff } t \in \left[0, \frac{1 - \langle A\tilde{\mathbf{e}}_i, \lambda \rangle}{\langle A\tilde{\mathbf{e}}_i, \mathbf{d} \rangle}\right].$$

629 From (6.20) we obviously deduce

$$630 \quad i \in S^+(\mathbf{d}) \Rightarrow \langle \lambda + t\mathbf{d}, A\tilde{\mathbf{e}}_i \rangle < 1 \text{ iff } t \in \left[0, \frac{1 - \langle A\tilde{\mathbf{e}}_i, \lambda \rangle}{\langle A\tilde{\mathbf{e}}_i, \mathbf{d} \rangle}\right)$$

631 and therefore that

$$632 \quad (6.21) \quad i \in S^+(\mathbf{d}) \Rightarrow i \in \{1, \dots, 2n\} \setminus S(\lambda + t\mathbf{d}) \text{ iff } t \in \left[0, \frac{1 - \langle A\tilde{\mathbf{e}}_i, \lambda \rangle}{\langle A\tilde{\mathbf{e}}_i, \mathbf{d} \rangle}\right).$$

633 In addition, for any $i \in S^+(\mathbf{d})$ we have that $\langle A\tilde{\mathbf{e}}_i, \lambda \rangle < 1$ and therefore that $1 -$
 634 $\langle A\tilde{\mathbf{e}}_i, \lambda \rangle > 0$. Since, for any $i \in S^+(\mathbf{d})$, we also have $\langle A\tilde{\mathbf{e}}_i, \mathbf{d} \rangle > 0$ and we deduce

$$635 \quad (6.22) \quad \forall i \in S^+(\mathbf{d}) \quad \text{we have} \quad \frac{1 - \langle A\tilde{\mathbf{e}}_i, \lambda \rangle}{\langle A\tilde{\mathbf{e}}_i, \mathbf{d} \rangle} > 0.$$

636 From (6.16), (6.17) and (6.20) we deduce that $S^+(\mathbf{d}) = \emptyset$ iff $(\lambda + t\mathbf{d}) \in \text{dom } g$ for
 637 every $t \geq 0$. In addition, from (6.16), (6.17) and (6.20) we deduce that if $S^+(\mathbf{d}) \neq \emptyset$

638 then $(\boldsymbol{\lambda} + t\mathbf{d}) \in \text{dom } g$ for every $t \in [0, \overline{\Delta t}]$, where $\overline{\Delta t}$ is defined by (3.12). The
 639 fact that $\overline{\Delta t} > 0$ follows from (6.22) and, again (3.12). It remains to prove that for
 640 any $t \in [0, \overline{\Delta t})$ we have $S(\boldsymbol{\lambda} + t\mathbf{d}) \subset S(\boldsymbol{\lambda})$. To this aim we consider an arbitrary
 641 $i \in \{1, \dots, 2n\} \setminus S(\boldsymbol{\lambda})$. Combining (6.18), (6.21) and the definition of $\overline{\Delta t}$ as a mini-
 642 mum (3.12), we deduce that $i \in \{1, \dots, 2n\} \setminus S(\boldsymbol{\lambda} + t\mathbf{d})$ for any $t \in [0, \overline{\Delta t})$. Hence,
 643 by considering the complementary set we obtain that for any $t \in [0, \overline{\Delta t})$ we have
 644 $S(\boldsymbol{\lambda} + t\mathbf{d}) \subset S(\boldsymbol{\lambda})$. Furthermore, the fact that $\partial g(\boldsymbol{\lambda} + t\mathbf{d}) \subset \partial g(\boldsymbol{\lambda})$ for all $t \in [0, \overline{\Delta t})$
 645 immediately follows from lemma 3.8. It is easy to see that for every $t \in [0, \overline{\Delta t}]$ we
 646 have $(\boldsymbol{\lambda} + t\mathbf{d}) \in \text{dom } g$. \square

647 6.9. Proof of lemma 3.11 on page 8.

648 *Proof.* The proof is in three steps. The first step justifies the well-posedness
 649 of (3.15). The second step proves that the conditions in (3.16) are equivalent. The
 650 last step proves that $\boldsymbol{\lambda} \in \text{dom } g$ is a solution to (D_{ℓ_1}) iff the conditions in (3.16) hold
 651 true.

652 From lemma 3.8, for any $\boldsymbol{\lambda} \in \text{dom } g$ we have that $\partial g(\boldsymbol{\lambda}) \neq \emptyset$ and obviously closed,
 653 convex. Therefore, for any $\boldsymbol{\lambda} \in \text{dom } g$ (3.15) is well posed.

654 Combining the definitions of \mathbf{d} (3.15) and of $S^+(\mathbf{d})$ (3.11) we have that $\mathbf{d} = \mathbf{0}$ implies
 655 $S^+(\mathbf{d}) = \emptyset$. Conversely, if $S^+(\mathbf{d}) = \emptyset$ then we obtain that $\langle \mathbf{d}, A\mathbf{e}_i \rangle = 0$ for every
 656 canonical vector \mathbf{e}_i of \mathbb{R}^m . From proposition 3.1 we have that A has full row rank
 657 and therefore deduce $\mathbf{d} = \mathbf{0}$. Thus, $S^+(\mathbf{d}) = \emptyset$ is equivalent to $\mathbf{d} = \mathbf{0}$. From the
 658 definitions of $S^+(\mathbf{d})$ (3.11) and of $\overline{\Delta t}$ (3.12) it is obvious that $S^+(\mathbf{d}) = \emptyset$ is equivalent
 659 to $\overline{\Delta t} = +\infty$. Thus, the three conditions in (3.16) are equivalent.

660 From theorem 8.11 we have that $\boldsymbol{\lambda} \in \text{dom } g$ is a solution to (D_{ℓ_1}) iff $\mathbf{0} \in \partial g(\boldsymbol{\lambda})$. Hence,
 661 $\boldsymbol{\lambda} \in \text{dom } g$ is a solution to (D_{ℓ_1}) iff \mathbf{d} defined by (3.15) satisfies $\mathbf{d} = \mathbf{0}$. It follows that
 662 $\boldsymbol{\lambda} \in \text{dom } g$ is a solution to (D_{ℓ_1}) iff the conditions in (3.16) hold true. \square

663 6.10. Proof of proposition 3.12 on page 8.

664 *Proof.* We begin to establish the following lemmas that will be useful for the proof
 665 of propositions 3.12.

666 LEMMA 6.5 (Technical lemma). *Consider a convex set $\emptyset \neq K \subset \mathbb{R}^m$. For any*
 667 *\mathbf{x} we have $\Pi_{K+\mathbf{x}}(\mathbf{0}) = \Pi_K(-\mathbf{x}) + \mathbf{x}$.*

668 *Proof.* From theorem 8.8, we have that a vector $\mathbf{y}_\mathbf{x}$ is the projection of some \mathbf{x}
 669 on K iff $\langle \mathbf{x} - \mathbf{y}_\mathbf{x}, \mathbf{y} - \mathbf{y}_\mathbf{x} \rangle \leq 0 \forall \mathbf{y} \in K$. Hence, the projection $\mathbf{y}_{-\mathbf{x}} := \Pi_K(-\mathbf{x})$ of $-\mathbf{x}$
 670 onto K satisfies $\langle -\mathbf{x} - \mathbf{y}_{-\mathbf{x}}, \mathbf{y} - \mathbf{y}_{-\mathbf{x}} \rangle \leq 0$ for all $\mathbf{y} \in K$. Therefore we obtain, for all
 671 $\mathbf{y} \in K - \mathbf{x}$, that

$$672 \langle -\mathbf{x} - \mathbf{y}_{-\mathbf{x}}, (\mathbf{y} - \mathbf{x}) - \mathbf{y}_{-\mathbf{x}} \rangle \leq 0 \forall \mathbf{y} \in K - \mathbf{x} \Leftrightarrow \langle \mathbf{0} - (\mathbf{x} + \mathbf{y}_{-\mathbf{x}}), \mathbf{y} - (\mathbf{y}_{-\mathbf{x}} + \mathbf{x}) \rangle \leq 0$$

673 and, from theorem 8.8 we obtain that $\mathbf{x} + \mathbf{y}_{-\mathbf{x}}$ is the projection of $\mathbf{0}$ on $K - \mathbf{x}$. In
 674 other words, $\Pi_K(-\mathbf{x}) + \mathbf{x} = \Pi_{K-\mathbf{x}}(\mathbf{0})$ and the formula is proved. \square

675 LEMMA 6.6 ($-\Pi_{\partial g(\boldsymbol{\lambda})}(\mathbf{0})$ satisfies (3.9)). *We posit the same assumptions as in*
 676 *proposition 3.1. For any $\boldsymbol{\lambda} \in \text{dom } g$ consider \mathbf{d} defined by (3.15) in lemma 3.11. We*
 677 *have that \mathbf{d} satisfies (3.9).*

678 *Proof.* Consider $\boldsymbol{\lambda} \in \text{dom } g$ and $S(\boldsymbol{\lambda})$ defined by (3.3) (page 5). We wish to
 679 prove that $\mathbf{d} := -\Pi_{\partial g(\boldsymbol{\lambda})}(\mathbf{0})$ satisfies (3.9). From lemma 6.5 applied with $\mathbf{x} = \mathbf{b}$ and
 680 $K := \text{co}\{A\tilde{\mathbf{e}}_i, i \in S(\boldsymbol{\lambda})\}$ we obtain $\mathbf{d} = -\Pi_{\partial g(\boldsymbol{\lambda})}(\mathbf{0}) = -\mathbf{b} - \Pi_K(-\mathbf{b})$. Thus, from
 681 theorem 8.8 we have that $\Pi_K(-\mathbf{b})$ satisfies

$$682 \langle -\mathbf{b} - \Pi_K(-\mathbf{b}), \mathbf{y} - \Pi_K(-\mathbf{b}) \rangle \leq 0 \quad \forall \mathbf{y} \in K := \text{co}\{A\tilde{\mathbf{e}}_i, i \in S(\boldsymbol{\lambda})\}$$

683 and therefore, since $\mathbf{d} = -\mathbf{b} - \Pi_K(-\mathbf{b})$, we obtain

$$684 \quad (6.23) \quad \langle \mathbf{d}, \mathbf{y} - \Pi_K(-\mathbf{b}) \rangle \leq 0 \quad \forall \mathbf{y} \in K := \text{co} \{A\tilde{\mathbf{e}}_i, i \in S(\boldsymbol{\lambda})\}.$$

685 Any coefficients μ_i of $\Pi_K(-\mathbf{b})$ onto K satisfy

$$686 \quad \Pi_K(-\mathbf{b}) = \sum_{\substack{\mu_i \geq 0 \quad \forall i \in S(\boldsymbol{\lambda}) \\ \mu_i = 0 \quad \text{otherwise}}} \mu_i A\tilde{\mathbf{e}}_i.$$

687 Consider any $j \in S(\boldsymbol{\lambda})$ and the coefficients α_i given by $\alpha_j = 1 + \mu_j$ where $\alpha_i = \mu_i$
688 for $i \neq j \in S(\boldsymbol{\lambda})$ and $\alpha_i = 0$ otherwise. Note that the vector $\sum_i \alpha_i A\tilde{\mathbf{e}}_i \in K$ and that
689 $\sum_i \alpha_i A\tilde{\mathbf{e}}_i - \Pi_K(-\mathbf{b}) = A\tilde{\mathbf{e}}_j$. Thus, from (6.23) we obtain that \mathbf{d} satisfies (3.9). \square

690 LEMMA 6.7 (Descent direction condition). *We posit the same assumptions as*
691 *in proposition 3.1 and consider \mathbf{d} defined by (3.15) in lemma 3.11. We have that if*
692 *$\mathbf{d} \neq \mathbf{0}$ then \mathbf{d} is a descent direction for g at $\boldsymbol{\lambda} \in \text{dom } g$. In addition, for all $\boldsymbol{\lambda} \in \text{dom } g$*
693 *we have*

$$694 \quad (6.24) \quad g'(\boldsymbol{\lambda}, -\Pi_{\partial g(\boldsymbol{\lambda})}(\mathbf{0})) = -\|\Pi_{\partial g(\boldsymbol{\lambda})}(\mathbf{0})\|_{\ell^2}^2.$$

695 *Proof.* We consider \mathbf{d} defined by (3.15) in lemma 3.11. We first establish (6.24)
696 then justify that if $\mathbf{d} \neq \mathbf{0}$ then \mathbf{d} is a descent direction for g at $\boldsymbol{\lambda} \in \text{dom } g$. From
697 theorem 8.8, we have that

$$698 \quad (6.25) \quad \langle \mathbf{d}, \mathbf{s} + \mathbf{d} \rangle \leq 0 \quad \forall \mathbf{s} \in \partial g(\boldsymbol{\lambda}) \quad \Leftrightarrow \quad \langle \mathbf{d}, \mathbf{s} \rangle \leq -\|\mathbf{d}\|_{\ell^2}^2 \quad \forall \mathbf{s} \in \partial g(\boldsymbol{\lambda}).$$

699 We have

$$700 \quad (6.26) \quad -\|\mathbf{d}\|_{\ell^2}^2 \geq \langle \mathbf{d}, \mathbf{b} \rangle = g'(\boldsymbol{\lambda}, \mathbf{d}) = \sup \{ \langle \mathbf{s}, \mathbf{d} \rangle : \mathbf{s} \in \partial g(\boldsymbol{\lambda}) \} \geq \langle -\mathbf{d}, \mathbf{d} \rangle = -\|\mathbf{d}\|_{\ell^2}^2.$$

701 Indeed, in (6.26) the first inequality is obtained by choosing $\mathbf{s} = \mathbf{b} \in \partial g(\boldsymbol{\lambda})$ in (6.25).
702 Combining lemma 6.6 and proposition 3.9 we obtain first equality in (6.26). The
703 second equality is justified by the definition of the subdifferential (see (xvii)). The
704 second inequality follows from $-\mathbf{d} \in \partial g(\boldsymbol{\lambda})$. The last equality is obvious. Thus, we
705 obtain (6.24). From (6.24), it follows that if $\mathbf{d} \neq \mathbf{0}$ we have that \mathbf{d} satisfies (3.9)-(3.10).
706 Hence, from proposition 3.9 we obtain that \mathbf{d} is a descent direction. \square

707 Consider \mathbf{d} defined by (3.15) in lemma 3.11. If $\mathbf{d} = \mathbf{0}$ then (3.17)-(3.18) hold true.
708 From now on, we assume that $\mathbf{d} \neq \mathbf{0}$. Let $t \in [0, \overline{\Delta}t)$, where $\overline{\Delta}t$ is defined in proposi-
709 tion 3.10 (page 8). For any $\boldsymbol{\lambda}' \in \mathbb{R}^m$ we have

$$710 \quad (6.27) \quad g(\boldsymbol{\lambda}') \geq g(\boldsymbol{\lambda}) + \langle \Pi_{\partial g(\boldsymbol{\lambda})}(\mathbf{0}), \boldsymbol{\lambda}' - \boldsymbol{\lambda} \rangle$$

$$711 \quad (6.28) \quad = g(\boldsymbol{\lambda}) + \langle \Pi_{\partial g(\boldsymbol{\lambda})}(\mathbf{0}), \boldsymbol{\lambda}' - \boldsymbol{\lambda} - t\mathbf{d} \rangle - t \|\Pi_{\partial g(\boldsymbol{\lambda})}(\mathbf{0})\|_{\ell^2}^2$$

$$712 \quad (6.29) \quad = g(\boldsymbol{\lambda}) + \langle \Pi_{\partial g(\boldsymbol{\lambda})}(\mathbf{0}), \boldsymbol{\lambda}' - \boldsymbol{\lambda} - t\mathbf{d} \rangle + tg'(\boldsymbol{\lambda}, \mathbf{d})$$

$$713 \quad (6.30) \quad = g(\boldsymbol{\lambda}) + \langle \Pi_{\partial g(\boldsymbol{\lambda})}(\mathbf{0}), \boldsymbol{\lambda}' - (\boldsymbol{\lambda} + t\mathbf{d}) \rangle + t\langle \mathbf{d}, \mathbf{b} \rangle$$

$$714 \quad (6.31) \quad = g(\boldsymbol{\lambda} + t\mathbf{d}) + \langle \Pi_{\partial g(\boldsymbol{\lambda})}(\mathbf{0}), \boldsymbol{\lambda}' - (\boldsymbol{\lambda} + t\mathbf{d}) \rangle.$$

715 The inequality in (6.27) is nothing but the definition of $\Pi_{\partial g(\boldsymbol{\lambda})}(\mathbf{0}) \in \partial g(\boldsymbol{\lambda})$ (see (xvii))
716 and (6.28) follows. Lemma 6.7 (we assumed $\mathbf{d} \neq \mathbf{0}$) justifies (6.29). From proposi-
717 tion 3.10 (page 8), for any $t \in [0, \overline{\Delta}t)$ we have $(\boldsymbol{\lambda} + t\mathbf{d}) \in \text{dom } g$ and, from lemma 6.4
718 we obtain (6.30). Equation (6.31) immediately follows from (3.7) in proposition 3.7.

719 From (6.27)-(6.31) and (xvii) we obtain (3.17). From theorem 8.8, we have that
 720 $\Pi_{\partial g(\boldsymbol{\lambda})}(\mathbf{0})$ satisfies

$$721 \quad \forall \mathbf{s} \in \partial g(\boldsymbol{\lambda}), \quad \langle -\Pi_{\partial g(\boldsymbol{\lambda})}(\mathbf{0}), \mathbf{s} - \Pi_{\partial g(\boldsymbol{\lambda})}(\mathbf{0}) \rangle \leq 0$$

722 which is equivalent to

$$723 \quad \forall \mathbf{s} \in \partial g(\boldsymbol{\lambda}) \quad \|\Pi_{\partial g(\boldsymbol{\lambda})}(\mathbf{0})\|_{\ell^2}^2 \leq \langle \mathbf{s}, \Pi_{\partial g(\boldsymbol{\lambda})}(\mathbf{0}) \rangle.$$

724 Hence, from (3.14) in proposition 3.10 (page 8) we deduce that $\Pi_{\partial g(\boldsymbol{\lambda})}(\mathbf{0})$ satisfies

$$725 \quad (6.32) \quad \forall t \in [0, \bar{\Delta}t) \quad \forall \mathbf{s} \in \partial g(\boldsymbol{\lambda} + t\mathbf{d}) \quad \|\Pi_{\partial g(\boldsymbol{\lambda})}(\mathbf{0})\|_{\ell^2}^2 \leq \langle \mathbf{s}, \Pi_{\partial g(\boldsymbol{\lambda})}(\mathbf{0}) \rangle.$$

726 Combining (3.17), (6.32) and, again, theorem 8.8 we obtain (3.18). \square

727 **6.11. Proof of proposition 3.13 on page 8.**

728 *Proof.* The proof is in two steps. We first justify the well-posedness of (3.19) then
 729 justify that (3.20) coincides with the evolution equation (3.6) (see proposition 3.6 on
 730 page 6).

731 **Step 1.** Let $k = 0$. By assumption, $\boldsymbol{\lambda}(t_k) \in \text{dom } g$. From lemma 3.11 we have that \mathbf{d}_k
 732 is well defined. From proposition 3.10 (page 8) this implies that t_{k+1} is well defined.
 733 In addition, from, again, proposition 3.10 and the definition of t_{k+1} it is easy to see
 734 that $\boldsymbol{\lambda}(t) \in \text{dom } g$ for every $t \in [t_k, t_{k+1}]$. The rest of the recursion follows. Thus,
 735 we obtain that the trajectory $\boldsymbol{\lambda}(t)$ given in (3.20) is mathematically well-posed. It
 736 remains to show that (3.20) coincides with the trajectory given by (3.6).

737 **Step 2.** From proposition 3.12, the vector $-\Pi_{\partial g(\boldsymbol{\lambda})}(\mathbf{0})$ that appears in (3.6) is piece-
 738 wise constant on every intervals $[t_k, t_{k+1})$. In addition, it is easy that the trajectory
 739 given by (3.20) coincides by construction with the solution to the evolution equa-
 740 tion (3.6) for every $t \geq 0$. The fact that for every $t \geq 0$, $\boldsymbol{\lambda}(t) \in \text{dom } g$ follows
 741 combining proposition 3.6 and lemma 3.8. This concludes our proof. \square

742 **6.12. Proof of proposition 3.14 on page 9.**

743 *Proof.* From (3.17) and the lower semi-continuity of $g \in \Gamma_0(\mathbb{R}^m)$, we obtain
 744 $\Pi_{\partial g(\boldsymbol{\lambda}(t_k))}(\mathbf{0}) \in \partial g(\boldsymbol{\lambda}(t_{k+1}))$ and therefore that

$$745 \quad (6.33) \quad \|\Pi_{\partial g(\boldsymbol{\lambda}(t_{k+1}))}(\mathbf{0})\|_{\ell^2} \leq \|\Pi_{\partial g(\boldsymbol{\lambda}(t_k))}(\mathbf{0})\|_{\ell^2}.$$

746 From proposition/definition 3.13 (page 8), we have that $\boldsymbol{\lambda}(t_{k+1}) \in \text{dom } g$ and there-
 747 fore, from lemma 3.8 we have $\partial g(\boldsymbol{\lambda}(t_{k+1})) \neq \emptyset$. The uniqueness of the projection of
 748 $\mathbf{0}$ onto the non-empty closed convex set $\partial g(\boldsymbol{\lambda}(t_{k+1}))$ and (3.17) imply that

$$749 \quad (6.34) \quad \|\Pi_{\partial g(\boldsymbol{\lambda}(t_k))}(\mathbf{0})\|_{\ell^2}^2 = \|\Pi_{\partial g(\boldsymbol{\lambda}(t_{k+1}))}(\mathbf{0})\|_{\ell^2}^2 \Leftrightarrow \Pi_{\partial g(\boldsymbol{\lambda}(t_k))}(\mathbf{0}) = \Pi_{\partial g(\boldsymbol{\lambda}(t_{k+1}))}(\mathbf{0}).$$

750 We wish to prove that $\Pi_{\partial g(\boldsymbol{\lambda}(t_k))}(\mathbf{0}) \neq \Pi_{\partial g(\boldsymbol{\lambda}(t_{k+1}))}(\mathbf{0})$. To do so, we set $\mathbf{d}_k :=$
 751 $-\Pi_{\partial g(\boldsymbol{\lambda}(t_k))}(\mathbf{0})$ and $\mathbf{d}_{k+1} := -\Pi_{\partial g(\boldsymbol{\lambda}(t_{k+1}))}(\mathbf{0})$ and denote by $\bar{\Delta}t_k$ (resp. $\bar{\Delta}t_{k+1}$) the
 752 positive kick times computed, from proposition 3.10, at $\boldsymbol{\lambda}(t_k)$ (resp. $\boldsymbol{\lambda}(t_{k+1})$).

753 By assumption, we have that $\boldsymbol{\lambda}(t_k)$ is not a solution to (D_{ℓ^1}) . From lemma 3.11
 754 we have that $\mathbf{d}_k \neq \mathbf{0}$. Assume, for the sake of contradiction, that $\mathbf{d}_k = \mathbf{d}_{k+1}$. From,
 755 again, proposition 3.10, we would have $(\boldsymbol{\lambda}(t_k) + \bar{\Delta}t_k \mathbf{d}_k) + t\mathbf{d}_k = \boldsymbol{\lambda}(t_{k+1}) + t\mathbf{d}_k \in \text{dom } g$,
 756 for some positive $t \in (0, \bar{\Delta}t_{k+1})$. This is impossible. Indeed, proposition 3.10 applied
 757 at $\boldsymbol{\lambda}(t_k)$ with the direction \mathbf{d}_k implies that $(\boldsymbol{\lambda}(t_k) + t\mathbf{d}_k) \in \text{dom } g$ iff $t \in [0, \bar{\Delta}t_k]$.
 758 Thus, we obtain that $\mathbf{d}_k \neq \mathbf{d}_{k+1}$ and combining (6.33)-(6.34) we obtain (3.21). \square

759 **6.13. Proof of proposition 3.15 on page 9.**

760 *Proof.* The proof is in two steps. The first step justifies the existence of $K \in \mathbb{N}$
 761 and of t_K such that $\boldsymbol{\lambda}(t) = \boldsymbol{\lambda}(t_K)$ for every $t \geq t_K$. The second step justifies that
 762 $\boldsymbol{\lambda}(t_K)$ is a solution to (D_{ℓ^1}) and \mathbf{d}_K satisfies $\mathbf{d}_K = \mathbf{0}$.

763 **Step 1.** This part of the proof follows a classic approach that can be found in,
 764 e.g., [22, Thm. 3.4.8, p. 382]. From lemma 3.8 (page 7) there are 4^n possible sets
 765 $\partial g(\boldsymbol{\lambda})$ for $\boldsymbol{\lambda} \in \text{dom } g$. Each of them is uniquely associated with $\mathbf{d}_k = -\Pi_{\partial g(\boldsymbol{\lambda})}(\mathbf{0})$.
 766 From proposition 3.14 they are all different from each other. This implies that the
 767 sequence $(\mathbf{d}_k)_k$ has a finite number of terms (is finite) and therefore converges for a
 768 finite index K . From proposition 3.10 (page 8) it is easy to deduce that the sequence
 769 $(t_k)_k$ is also finite. From proposition 3.13 we deduce the existence of K . In other
 770 words, we obtained that the trajectory $\boldsymbol{\lambda}(t)$ given in (3.20) satisfies, for some $K \in \mathbb{N}$,
 771 $\boldsymbol{\lambda}(t) = \boldsymbol{\lambda}(t_K)$ for every $t \geq t_K$. We now turn to the second step of the proof.

772 **Step 2.** From proposition 3.13, we have that the trajectory $\boldsymbol{\lambda}(t)$ given in (3.20)
 773 coincides with the trajectory given by (3.6). Thus, the limit of $\boldsymbol{\lambda}(t)$ when $t \rightarrow +\infty$
 774 is a solution to (D_{ℓ^1}) . (The fact that (D_{ℓ^1}) has a solution is justified by lemma 3.4
 775 page 5). From step1, we have that the limit of $\boldsymbol{\lambda}(t)$ is attained for $t = t_K$. Hence,
 776 $\boldsymbol{\lambda}(t_K)$ is a solution to (3.4). From lemma 3.11 we immediately obtain $\mathbf{d}_K = \mathbf{0}$. \square

777 **7. Glossary of notations, definitions.**

- 778 (i) (Vectors) Throughout the paper the vectors of, e.g., \mathbb{R}^n are denoted in bold
 779 typeface, e.g., \mathbf{x} . Other objects like scalars or functions are denoted in non-
 780 bold typeface.
- 781 (ii) (Canonical vectors) Throughout the paper the i -th canonical vectors of, e.g.,
 782 \mathbb{R}^n are denoted by \mathbf{e}_i .
- 783 (iii) (Inner product) For $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ we denote by $\langle \mathbf{x}, \mathbf{y} \rangle$ the Euclidean inner product
 784 in \mathbb{R}^n .
- 785 (iv) (Interior) $\text{int}(E)$: interior of a set E .
- 786 (v) (Conical hull) $\text{co}\{\mathbf{a}_1, \dots, \mathbf{a}_p\} := \{\sum_{i=1}^p \mu_i \mathbf{a}_i : \mu_i \geq 0, \forall i = 1, \dots, p\}$
- 787 (vi) Normal cone to a convex set $C \neq \emptyset$ at $\boldsymbol{\lambda} \in C$:
 788 $N_C(\boldsymbol{\lambda}) = \{\mathbf{s} : \langle \mathbf{s}, \mathbf{s}' - \boldsymbol{\lambda} \rangle \leq 0 \forall \mathbf{s}' \in C\}$ (See, e.g., [22, Def. 5.2.3, p. 136])
- 789 (vii) (the $\ell^\infty(\mathbb{R}^n)$ unit ball) $B_\infty = \{\mathbf{u} : \langle \mathbf{u}, \tilde{\mathbf{e}}_i \rangle \leq 1, i = 1, \dots, 2n\}$ where $\tilde{\mathbf{e}}_i$ is given
 790 by (3.3).
- 791 (viii) (Effective domain) $\text{dom } f$: The domain of a convex function f is the (convex,
 792 possibly empty) set $\text{dom } f = \{\mathbf{x} \in \mathbb{R}^n : f(\mathbf{x}) \in \mathbb{R}\}$
- 793 (ix) (Convex function) A function $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ is said to be convex if
 794 $\forall (\mathbf{x}, \mathbf{y}) \in \mathbb{R}^n \times \mathbb{R}^n$ and $\forall \alpha \in (0, 1)$ $f(\alpha \mathbf{x} + (1 - \alpha)\mathbf{y}) \leq \alpha f(\mathbf{x}) + (1 - \alpha)f(\mathbf{y})$
 795 holds true (in $\mathbb{R} \cup \{+\infty\}$).
- 796 (x) (Set $\Gamma_0(\mathbb{R}^n)$) The set of lower semi-continuous, convex functions with
 797 $\text{dom } f \neq \emptyset$ is denoted $\Gamma_0(\mathbb{R}^n)$
- 798 (xi) (Characteristic function of a set) $\chi_E(\mathbf{x}) = 0$ if $\mathbf{x} \in E$ and $\chi_E(\mathbf{x}) = +\infty$
 799 otherwise
- 800 (xii) (Polyhedral convex function) f is a polyhedral convex function if $f(\mathbf{u}) =$
 801 $h(\mathbf{u}) + \chi_C(\mathbf{u})$ $h(\mathbf{u}) = \max_{i=1, \dots, p} (\langle \mathbf{u}, \mathbf{a}_i \rangle - r_i)$ and
 802 $C = \{\mathbf{u} \in \mathbb{R}^n : \langle \mathbf{u}, \boldsymbol{\alpha}_i \rangle \leq \rho_i, i = 1, \dots, q\}$.
- 803 (xiii) (Directional derivative) For $f \in \Gamma_0(\mathbb{R}^n)$ at $\mathbf{a} \in \text{dom } f$ in the direction \mathbf{d} is
 804 $f'(\mathbf{a}, \mathbf{d}) := \lim_{t \rightarrow 0^+} \frac{f(\mathbf{a} + t\mathbf{d}) - f(\mathbf{a})}{t}$
- 805 (xiv) (Right derivative) $\frac{d^+ \boldsymbol{\lambda}(t)}{dt} := \lim_{h \rightarrow 0^+} \frac{\boldsymbol{\lambda}(t+h) - \boldsymbol{\lambda}(t)}{h}$
- 806 (xv) (Descent direction) $\mathbf{d} \neq \mathbf{0}$ is a descent direction for f at \mathbf{x} if $\exists t > 0$ such that

807 $\mathbf{x} + t\mathbf{d} \in \text{dom } f$ and $f(\mathbf{x} + t\mathbf{d}) < f(\mathbf{x})$ (See, e.g., [22, Def. 1.1.1, p. 343]).
 808 (xvi) (Convex conjugate) For any f convex that satisfies $\text{dom } f \neq \emptyset$, the function
 809 f^* defined by $\mathbb{R}^n \ni \mathbf{s} \mapsto f^*(\mathbf{s}) := \sup_{\mathbf{x} \in \text{dom } f} \{\langle \mathbf{s}, \mathbf{x} \rangle - f(\mathbf{x})\}$. (See, e.g., [23,
 810 Def. 1.1.1, p. 37]). For any $f \in \Gamma_0(\mathbb{R}^n)$ we have $f^* \in \Gamma_0(\mathbb{R}^n)$ (See, e.g., [22,
 811 Thm. 1.1.2., p. 38])

812 (xvii) (Sub-differential) For $f \in \Gamma_0(\mathbb{R}^n)$ and $\mathbf{x} \in \text{dom } f$ the vector $\mathbf{s} \in \mathbb{R}^n$ is a
 813 subgradient of f at \mathbf{x} if one of the following equivalent assertions is satisfied

$$814 \quad (7.1) \quad \forall \mathbf{y} \in \mathbb{R}^n, \quad f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \mathbf{s}, \mathbf{y} - \mathbf{x} \rangle; \quad \text{or} \quad \forall \mathbf{d} \in \mathbb{R}^n, \quad \langle \mathbf{s}, \mathbf{d} \rangle \leq f'(\mathbf{x}, \mathbf{d}).$$

815 We denote by $\partial f(\mathbf{x})$ the closed convex set of vectors $\mathbf{s} \in \mathbb{R}^n$ that satisfy (7.1).

816 For $\mathbf{x} \notin \text{dom } f$ we set $\partial f(\mathbf{x}) := \emptyset$.

817 (xviii) (Euclidean projection) $\Pi_C(\mathbf{x}) = \arg \min_{\mathbf{y} \in C} \|\mathbf{y} - \mathbf{x}\|_{\ell^2}$ for $C \neq \emptyset$ closed and
 818 convex.

819 **8. Mathematical background.** This section contains several propositions and
 820 theorems used throughout proofs given in section 6.

821 **THEOREM 8.1.** *For any $f \in \Gamma_0(\mathbb{R}^n)$ we have $f^* \in \Gamma_0(\mathbb{R}^n)$.*

822 *Proof.* From $f \in \Gamma_0(\mathbb{R}^n)$ we have $f \neq 0$, and from [22, Pro. 1.2.1, p. 147] there is
 823 an affine function minorizing f on \mathbb{R}^n . Applying [23, Thm. 1.1.2, p. 38] we conclude
 824 that $f^* \in \Gamma_0(\mathbb{R}^n)$. \square

825 **LEMMA 8.2** (Conjugate of absolute value). [4, Table 3.1, p. 76] *Let $f : x \in \mathbb{R} \mapsto$
 826 $f(x) := |x|$. We have for all $v \in \mathbb{R}$, $f^*(v) = \chi_{-1,1}(v) = \begin{cases} 0 & \text{if } v \in [-1, 1] \\ +\infty & \text{otherwise.} \end{cases}$*

827 **LEMMA 8.3** (Conjugate of characteristic function). [23, Ex. 1.1.5, p. 39] *The*
 828 *conjugate of the characteristic function of the nonempty convex set set C (see (xi))*
 829 *is for all $\mathbf{v} \in \mathbb{R}^n$, $\chi_C^*(\mathbf{v}) = \sup_{\mathbf{x} \in C} \langle \mathbf{v}, \mathbf{x} \rangle$.*

830 **PROPOSITION 8.4** (Conjugation in Product Spaces). [36, Prop. 11.22, p. 493]
 831 *Let f_1, \dots, f_n be in $\Gamma_0(\mathbb{R})$, and $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ given by: $\forall (x_1, \dots, x_n) \in$
 832 \mathbb{R}^n , $f(x_1, \dots, x_n) = f_1(x_1) + \dots + f_n(x_n)$. Then $f^*(v_1, \dots, v_n) = f_1^*(v_1) + \dots + f_n^*(v_n)$.*

833 **THEOREM 8.5** (Pre-composition With a Matrix). [22, Prop. 2.1.5, p. 159] *Let*
 834 *$f \in \Gamma_0(\mathbb{R}^m)$ and $A \in \mathcal{M}_{m \times n}(\mathbb{R})$, and assume that $\text{span } A \cap \text{dom } f \neq \emptyset$. We have*
 835 *$f(A \cdot) \in \Gamma_0(\mathbb{R}^n)$.*

836 **LEMMA 8.6** (Subdifferential of Normal Cone to a Closed Convex Set). [23, Def.
 837 1.1.3, p. 93] *The set of normal directions to a closed convex set $C \subset \mathbb{R}^m$ at $\boldsymbol{\lambda} \in C$,*
 838 *is the subdifferential of the characteristic function χ_C at $\boldsymbol{\lambda}$: $N_C(\boldsymbol{\lambda}) := \partial \chi_C$.*

839 **LEMMA 8.7.** [22, Ex. 5.2.6 b), p.138] *Let a closed convex polyhedron defined by*
 840 *$C := \{\mathbf{u} \in \mathbb{R}^n : \langle \mathbf{s}_i; \mathbf{x} \rangle \leq r_i \text{ for } i = 1, \dots, p\}$ where $\mathbf{s}_i \in \mathbb{R}^n$ and $r_i \in \mathbb{R}$ for all $i =$
 841 $1, \dots, p$. The set of active constraints at $\mathbf{u} \in C$ by $W(\mathbf{u}) = \{i \in 1, \dots, p : \langle \mathbf{s}_i; \mathbf{x} \rangle = r_i\}$.
 842 Then we have $N_C(\mathbf{u}) = \text{co}\{\mathbf{s}_i : i \in W(\mathbf{u})\}$.*

843 **PROPOSITION 8.8.** [22, Thm 3.1.1, p. 117] *Let C be a nonempty closed convex*
 844 *set of \mathbb{R}^n . We have that $\mathbf{y}_x \in C$ is the Euclidean projection of some \mathbf{x} onto C if only*
 845 *if $\langle \mathbf{x} - \mathbf{y}_x, \mathbf{y} - \mathbf{y}_x \rangle \leq 0$ for all $\mathbf{y} \in C$.*

846 **THEOREM 8.9** (Subdifferential of Pre-composition with a matrix). *Let $f \in$
 847 $\Gamma_0(\mathbb{R}^n)$ such that $\text{int}(\text{dom } f) \neq \emptyset$ and $A \in \mathcal{M}_{m \times n}(\mathbb{R})$. Assume that $\text{int}(\text{dom } f) \cap$
 848 $\text{span } A \neq \emptyset$. Then, any $\mathbf{u} \in \mathbb{R}^n$ such that $A\mathbf{u} \in \text{dom } f$ we have $\partial(f(A \cdot))(\mathbf{u}) =$
 849 $A^T \partial f(A\mathbf{u})$.*

850 *Proof.* Since we assumed that $\text{int}(\text{dom } f) \neq \emptyset$ we have $\text{ri}(\text{dom } f) \cap \text{span } A =$
 851 $\text{int}(\text{dom } f) \cap \text{span } A = \emptyset$. Thus, from [23, Thm. 3.2.1, p. 117] applied with $\varepsilon = 0$
 852 and $g := f$ to conclude. \square

853 PROPOSITION 8.10. [22, Prop. 2.1.1, p. 158] Let $f_1 \in \Gamma_0(\mathbb{R}^n), \dots, f_p \in \Gamma_0(\mathbb{R}^n)$
 854 and t_1, \dots, t_p be positive numbers. We assume that there is a point where all the f_j are
 855 finite. Then the function $\sum_{i=1}^p t_i f_i \in \Gamma_0(\mathbb{R}^n)$.

856 THEOREM 8.11 (Fermat's rule). [36, Thm. 10.1, p. 422] Let $f \in \Gamma_0(\mathbb{R}^n)$. Then
 857 f has a global minimum at $\bar{\mathbf{u}}$ if and only if $\mathbf{0} \in \partial f(\bar{\mathbf{u}})$.

858 PROPOSITION 8.12. [1, Prop. 1, p. 159] Let $f \in \Gamma_0(\mathbb{R}^n)$. Then the set-valued
 859 map $\mathbb{R}^n \ni \mathbf{u} \mapsto \partial f(\mathbf{u})$ is maximal monotone.

860 THEOREM 8.13 (Subdifferential of sum of Γ_0 -functions). Let $f_1, f_2 \in \Gamma_0(\mathbb{R}^n)$.
 861 We assume that $\text{int}(\text{dom } f_1) \cap \text{int}(\text{dom } f_2) \neq \emptyset$. Then for all $\mathbf{u} \in \text{dom}(f_1 + f_2)$ we
 862 have $\partial(f_1 + f_2)(\mathbf{u}) = \partial f_1(\mathbf{u}) + \partial f_2(\mathbf{u})$.

863 *Proof.* Since $\text{int}(\text{dom } f_1) \cap \text{int}(\text{dom } f_2) \neq \emptyset$ we deduce that $\text{ri}(\text{dom } f_1) \cap$
 864 $\text{ri}(\text{dom } f_2) = \text{int}(\text{dom } f_1) \cap \text{int}(\text{dom } f_2) \neq \emptyset$. Thus, from [23, Cor. 3.1.2, p. 114]
 865 applied with $\varepsilon = 0$ to conclude. \square

866 THEOREM 8.14. [5, Thm. 3.1, p. 54] Let T be a maximal monotone operator
 867 from \mathbb{R}^m to \mathbb{R}^m and $\text{dom}(T)$ be its domain. Consider the problem $\frac{d\boldsymbol{\lambda}(t)}{dt} \in -T(\boldsymbol{\lambda}(t))$
 868 with $\boldsymbol{\lambda}(0) = \boldsymbol{\lambda}_0$. For all $\boldsymbol{\lambda}_0 \in \text{dom}(T)$, there exists a unique solution $\boldsymbol{\lambda}(\cdot) : [0, +\infty) \rightarrow$
 869 \mathbb{R}^m such that :

- 870 1. $\boldsymbol{\lambda}(t) \in \text{dom}(T)$ for all $t > 0$, and $\boldsymbol{\lambda}(0) = \boldsymbol{\lambda}_0$;
- 871 2. the function $\boldsymbol{\lambda}(\cdot)$ is continuous on $[0, +\infty)$;
- 872 3. the function $\boldsymbol{\lambda}(\cdot)$ admits a right derivative $\frac{d^+\boldsymbol{\lambda}(t)}{dt}$ at all $t \geq 0$, given by
 873 $\frac{d^+\boldsymbol{\lambda}(t)}{dt} = -\Pi_{T(\boldsymbol{\lambda}(t))}(\mathbf{0})$ for all $t \in [0, +\infty)$;
- 874 4. the function $\frac{d^+\boldsymbol{\lambda}(\cdot)}{dt}$ is continuous from the right on $[0, +\infty)$.

875 THEOREM 8.15. [1, Thm. 2, p. 160] Let $g \in \Gamma_0(\mathbb{R}^m)$, and assume that g achieves
 876 its minimum at some point. Then, for all $\boldsymbol{\lambda}_0 \in \text{dom}(\partial g)$, the trajectory given by
 877 $\frac{d^+\boldsymbol{\lambda}(t)}{dt} = -\Pi_{\partial g(\boldsymbol{\lambda}(t))}(\mathbf{0})$ with $\boldsymbol{\lambda}(0) = \boldsymbol{\lambda}_0$ converges to a point which minimizes g when
 878 $t \rightarrow +\infty$.

879 THEOREM 8.16. [1, Thm. 4, Eq (28), p. 35-36] Let $A \in \mathcal{M}_{m \times n}(\mathbb{R})$ and $U \in$
 880 $\Gamma_0(\mathbb{R}^n), V \in \Gamma_0(\mathbb{R}^m)$. Assume that $\mathbf{0} \in \text{int}(A \text{ dom } U - \text{dom } V)$. Then, for all $\mathbf{u} \in$
 881 $\text{dom } U \cap \text{dom } V(A \cdot)$ we have $\partial(U + V(A \cdot))(\mathbf{u}) = \partial U(\mathbf{u}) + A^T \partial V(A\mathbf{u})$.

882 PROPOSITION 8.17. [1, Prop. 1, p.163] Let $A \in \mathcal{M}_{m \times n}(\mathbb{R})$ and $U \in \Gamma_0(\mathbb{R}^n), V \in$
 883 $\Gamma_0(\mathbb{R}^m)$. Assume that $\mathbf{0} \in \text{int}(A^T \text{ dom } V^* + \text{dom } U^*)$. Then, $\inf_{\mathbf{u} \in \mathbb{R}^n} (U(\mathbf{u}) + V(A\mathbf{u}))$
 884 has a solution.

885 THEOREM 8.18. [1, Thm. 2, p. 167] Let $A \in \mathcal{M}_{m \times n}(\mathbb{R}), U \in \Gamma_0(\mathbb{R}^n)$ and
 886 $V \in \Gamma_0(\mathbb{R}^m)$. Assume that assumptions of theorem 8.16 and proposition 8.17 hold.
 887 Then, $\inf_{\boldsymbol{\lambda} \in \mathbb{R}^m} (U^*(-A^T \boldsymbol{\lambda}) + V^*(\boldsymbol{\lambda}))$ has a solution.

888 **Acknowledgments.** The Authors thank S. Ladjal and K. Trabelsi for their
 889 colorful comments and suggestions. The Authors thank M. Möller for providing the
 890 implementation of the GISS algorithm.

- 892 [1] J.-P. AUBIN AND A. CELLINA, *Differential Inclusions: Set-Valued Maps and Viability Theory*,
893 Grundlehren der mathematischen Wissenschaften, Springer Berlin Heidelberg, 2012.
- 894 [2] A. BECK AND M. TEBoulLE, *A fast iterative shrinkage-thresholding algorithm for linear inverse*
895 *problems*, SIAM Journal on Imaging Sciences, 2 (2009), pp. 183–202.
- 896 [3] T. BLUMENSATH AND M. E. DAVIES, *Iterative hard thresholding for compressed sensing*, Applied
897 and Computational Harmonic Analysis, 27 (2009), pp. 265–274.
- 898 [4] J. M. BORWEIN AND A. S. LEWIS, *Convex Analysis and Nonlinear Optimization : Theory and*
899 *Examples*, CMS books in mathematics, Springer, New York, 2000.
- 900 [5] H. BRÉZIS, *Opérateurs Maximaux Monotones et Semi-Groupes de Contractions Dans Les Es-*
901 *paces de Hilbert*, North-Holland Publishing Co., American Elsevier Publishing Co., 1973.
- 902 [6] B. BRINGMANN, D. CREMERS, F. KRAHMER, AND M. MÖLLER, *The homotopy method revis-*
903 *ited: Computing solution paths of ℓ_1 -regularized problems*, Mathematics of Computation,
904 87 (2018), pp. 2343–2364, <https://doi.org/10.1090/mcom/3287>, <https://doi.org/10.1090/mcom/3287>.
- 905 [7] M. BURGER, M. MÖLLER, M. BENNING, AND S. OSHER, *An adaptive inverse scale space method*
906 *for compressed sensing*, Mathematics of Computation, 82 (2013), pp. 269–299.
- 907 [8] J.-F. CAI, S. OSHER, AND Z. SHEN, *Convergence of the linearized bregman iteration for ℓ_1 -norm*
908 *minimization*, Mathematics of Computation, 78 (2009), pp. 2127–2136.
- 909 [9] J.-F. CAI, S. OSHER, AND Z. SHEN, *Linearized bregman iterations for compressed sensing*,
910 Mathematics of Computation, 78 (2009), pp. 1515–1536.
- 911 [10] E. J. CANDÈS, J. ROMBERG, AND T. TAO, *Robust uncertainty principles: Exact signal recon-*
912 *struction from highly incomplete frequency information*, IEEE Transactions on information
913 theory, 52 (2006), pp. 489–509.
- 914 [11] E. J. CANDÈS AND T. TAO, *Decoding by linear programming*, Information Theory, IEEE Trans-
915 actions on, 51 (2005), pp. 4203–4215.
- 916 [12] E. J. CANDÈS AND T. TAO, *Near-optimal signal recovery from random projections: Universal*
917 *encoding strategies?*, Information Theory, IEEE Transactions on, 52 (2006), pp. 5406–5425.
- 918 [13] I. CIRIL, J. DARBON, AND Y. TENDERO, *A simple and exact algorithm to solve l_1 linear problems*
919 *- application to the compressive sensing method*, in Proceedings of the 13th International
920 Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Ap-
921 plications (VISIGRAPP 2018) - Volume 4: VISAPP, Funchal, Madeira, Portugal, January
922 27-29, 2018., 2018, pp. 54–62.
- 923 [14] W. DAI AND O. MILENKOVIC, *Subspace pursuit for compressive sensing signal reconstruction*,
924 Information Theory, IEEE Transactions on, 55 (2009), pp. 2230–2249.
- 925 [15] M. DIMICCOLI, *Fundamentals of cone regression*, Statistics Surveys, 10 (2016), pp. 53 – 99,
926 <https://doi.org/10.1214/16-SS114>, <https://doi.org/10.1214/16-SS114>.
- 927 [16] D. L. DONOHO, *Compressed sensing*, Information Theory, IEEE Transactions on, 52 (2006),
928 pp. 1289–1306.
- 929 [17] D. L. DONOHO AND M. ELAD, *Optimally sparse representation in general (nonorthogonal)*
930 *dictionaries via ℓ_1 minimization*, Proceedings of the National Academy of Sciences, 100
931 (2003), pp. 2197–2202.
- 932 [18] B. EFRON, T. HASTIE, I. JOHNSTONE, AND R. TIBSHIRANI, *Least angle regression*, Ann. Statist.,
933 32 (2004), pp. 407–499, <https://doi.org/10.1214/009053604000000067>, <https://doi.org/10.1214/009053604000000067>.
- 934 [19] S. FOUcART, *Stability and robustness of weak orthogonal matching pursuits*, in Recent advances
935 in harmonic analysis and applications, Springer, 2013, pp. 395–405.
- 936 [20] M. C. GRANT AND S. P. BOYD, *The CVX Users' Guide, Release 2.1*.
- 937 [21] E. T. HALE, W. YIN, AND Y. ZHANG, *A Fixed-Point Continuation Method for l_1 -Regularized*
938 *Minimization with Applications to Compressed Sensing*, CAAM Technical Report TR07-
939 07, (2007).
- 940 [22] J.-B. HIRIART-URRUTY AND C. LEMARECHAL, *Convex Analysis and Minimization Algorithms*
941 *I: Fundamentals*, Grundlehren der mathematischen Wissenschaften, Springer Berlin Hei-
942 delberg, 1996.
- 943 [23] J.-B. HIRIART-URRUTY AND C. LEMARECHAL, *Convex Analysis and Minimization Algo-*
944 *rithms II: Advanced Theory and Bundle Methods*, Grundlehren der mathematischen Wis-
945 senschaften, Springer Berlin Heidelberg, 1996.
- 946 [24] J. HUANG, Y. JIAO, X. LU, AND L. ZHU, *Robust decoding from 1-bit compressive sampling with*
947 *ordinary and regularized least squares*, SIAM Journal on Scientific Computing, 40 (2018),
948 pp. A2062–A2086.
- 949 [25] P. JAIN, A. TEWARI, AND I. S. DHILLON, *Orthogonal matching pursuit with replacement*, in
950 Advances in Neural Information Processing Systems, 2011, pp. 1215–1223.
- 951 [26] J. MAIRAL, F. BACH, J. PONCE, G. SAPIRO, R. JENATTON, AND G. OBOZINSKI, *Spams: A*
952

- 954 *sparse modeling software, v2. 3*, URL <http://spams-devel.gforge.inria.fr/downloads.html>,
 955 (2014).
- 956 [27] J. MAIRAL AND B. YU, *Complexity analysis of the lasso regularization path*, in Proceedings
 957 of the 29th International Conference on International Conference on Machine Learning,
 958 Omnipress, 2012, pp. 1835–1842.
- 959 [28] A. MALEKI, *Coherence analysis of iterative thresholding algorithms*, in Communication, Con-
 960 trol, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on, IEEE,
 961 2009, pp. 236–243.
- 962 [29] A. MALEKI AND D. L. DONOHO, *Optimally tuned iterative reconstruction algorithms for com-
 963 pressed sensing*, IEEE Journal of Selected Topics in Signal Processing, 4 (2010), pp. 330–
 964 341.
- 965 [30] S. MALLAT AND Z. ZHANG, *Matching pursuits with time-frequency dictionaries*, Signal Process-
 966 ing, IEEE Transactions on, 41 (1993), pp. 3397–3415.
- 967 [31] M. C. MEYER, *A simple new algorithm for quadratic programming with applications in statis-
 968 tics*, Communications in Statistics-Simulation and Computation, 42 (2013), pp. 1126–1139.
- 969 [32] M. MOELLER AND X. ZHANG, *Fast Sparse Reconstruction: Greedy Inverse Scale Space Flows*,
 970 Mathematics of Computation, 85 (2016), pp. 179–208.
- 971 [33] D. NEEDELL AND J. A. TROPP, *CoSaMP: Iterative signal recovery from incomplete and inac-
 972 curate samples*, Applied and Computational Harmonic Analysis, 26 (2009), pp. 301–321.
- 973 [34] S. OSHER, M. BURGER, D. GOLDFARB, J. XU, AND W. YIN, *An iterative regularization method
 974 for total variation-based image restoration*, Multiscale Modeling & Simulation, 4 (2005),
 975 pp. 460–489.
- 976 [35] Y. C. PATI, R. REZAIIFAR, AND P. KRISHNAPRASAD, *Orthogonal matching pursuit: Recursive
 977 function approximation with applications to wavelet decomposition*, in Conference Record
 978 of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers, IEEE,
 979 1993, pp. 40–44.
- 980 [36] R. ROCKAFELLAR, M. WETS, AND R. WETS, *Variational Analysis*, Grundlehren der mathema-
 981 tischen Wissenschaften, Springer Berlin Heidelberg, 2009.
- 982 [37] H. SCHAEFFER, Y. YANG, H. ZHAO, AND S. OSHER, *Real-time adaptive video compression*,
 983 SIAM Journal on Scientific Computing, 37 (2015), pp. 980–1001.
- 984 [38] J. F. STURM, *Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones*,
 985 Optimization methods and software, 11 (1999), pp. 625–653.
- 986 [39] J. A. TROPP AND A. C. GILBERT, *Signal recovery from random measurements via orthogonal
 987 matching pursuit*, Information Theory, IEEE Transactions on, 53 (2007), pp. 4655–4666.
- 988 [40] E. VAN DEN BERG AND M. P. FRIEDLANDER, *Probing the Pareto frontier for basis pursuit
 989 solutions*, SIAM Journal on Scientific Computing, 31 (2008), pp. 890–912.
- 990 [41] E. VAN DEN BERG AND M. P. FRIEDLANDER, *SPGL1: A solver for large-scale sparse recon-
 991 struction*, December 2019. <https://friedlander.io/spgl1>.
- 992 [42] Z. WEN, W. YIN, D. GOLDFARB, AND Y. ZHANG, *A fast algorithm for sparse reconstruction
 993 based on shrinkage, subspace optimization, and continuation*, SIAM Journal on Scientific
 994 Computing, 32 (2010), pp. 1832–1857.
- 995 [43] J. YANG AND Y. ZHANG, *Alternating direction algorithms for ℓ_1 -problems in compressive sens-
 996 ing*, SIAM Journal on Scientific Computing, 33 (2011), pp. 250–278.
- 997 [44] W. YIN, S. OSHER, D. GOLDFARB, AND J. DARBON, *Bregman iterative algorithms for ℓ_1 -
 998 minimization with applications to compressed sensing*, SIAM Journal on Imaging Sciences,
 999 1 (2008), pp. 143–168.
- 1000 [45] X. ZHANG, M. BURGER, AND S. OSHER, *A unified primal-dual algorithm framework based on
 1001 bregman iteration*, Journal of Scientific Computing, 46 (2011), pp. 20–46.