SIAM J. Sci. Comput.
Vol. 43, No. 6, pp. A4067–A4094

AN ALGORITHM SOLVING COMPRESSIVE SENSING PROBLEM BASED ON MAXIMAL MONOTONE OPERATORS*

YOHANN TENDERO[†], IGOR CIRIL[†], JÉRÔME DARBON[‡], AND SUSANA SERNA[§]

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

Key words. sparse solution recovery, compressive sensing, inverse scale space, ℓ^1 minimization, nonsmooth optimization, maximal monotone operator, phase transition

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

DOI. 10.1137/19M1260670

1. Introduction.

Compressive sensing and sparsity-related paradigms have gained enormous interest in the last decade and can be used for, e.g., data analysis, signal and image processing, inverse problems, or acquisition devices. Indeed, in many cases the unknowns of an underdetermined system can be obtained by finding the sparsest (or simplest) solution to a linear system

$$A\boldsymbol{u} = \boldsymbol{b}. \tag{1.1}$$

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

$$(P_{\ell^1}) \qquad \begin{cases} \inf\limits_{\boldsymbol{u}\in\mathbb{R}^n} & \|\boldsymbol{u}\|_{\ell^1}, \\ \text{s.t.} & A\boldsymbol{u} = \mathbf{b}. \end{cases}$$

*Submitted to the journal's Methods and Algorithms for Scientific Computing section May 8, 2019; accepted for publication (in revised form) August 9, 2021; published electronically December 16, 2021. A preliminary version of the work appears in [13].
https://doi.org/10.1137/19M1260670
Funding: The work of the third author was supported by the National Science Foundation under grant NSF-1820821.

[†]DR2I, Institut Polytechnique des Sciences Avancées, 94200, Ivry-sur-Seine, France (yohann.tendero@ipsa.fr, igor.ciril@ipsa.fr).
[‡]Division of Applied Mathematics, Brown University, Providence, RI 02912 USA (jerome_darbon@brown.edu).
[§]Departament de Matematiques, Universitat Autonoma de Barcelona, Bellaterra 08193, Barcelona, Spain (susana.serna@uab.cat).

It turns out that under various assumptions, the minimizers remain the same if one replaces the ℓ^0 pseudo-norm by the ℓ^1 norm (see, e.g., [11, 12, 16, 17] and the references therein). Problem (P_{ℓ^1}) is a convex albeit nonsmooth optimization problem in high dimension (n can be thought of as the number of pixels of an image, for instance). For these reasons developing efficient algorithmic solutions is still a challenge in many cases. For instance, the CVX system "is not meant for very large problems" [20, sect. 1.3, p. 3] that arise from signal/image processing applications [24, 37]. Hence, many algorithms have been proposed to solve ℓ^1 minimization problems; see, e.g., [34, 21, 2, 8, 9, 44, 45, 43, 42, 18, 40]. In this paper, we propose a simple algorithm that can be employed to solve these ℓ^1 minimization problems up to the machine precision. Indeed, it is only assumed that the matrix A has full row rank. This paper also exhibits a numerical comparison with several classic algorithms in the literature. These comparisons illustrate that our algorithm compares advantageously: the theoretically predicted phase transition (see, e.g., [29, 10]) is empirically observed with a higher accuracy.

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

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

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

Outline of this paper. This paper is organized as follows. Section 2 gives a very compact, yet self-contained, presentation of the numerical computations needed to implement the algorithm proposed in this paper (see Algorithm 2.1 on page A4070). Section 3 on page A4071 proves the mathematical validity of this algorithm. In other words, we shall prove that the solution computed by Algorithm 2.1 is exact (and numerically, up to the machine precision). The convergence (in finite time) of Algorithm 2.1 to a solution to (P_{ℓ^1}) is mathematically guaranteed. Section 4 on page A4075 proposes a numerical evaluation and comparison of Algorithm 2.1 with some state-of-the-art solutions solving (P_{ℓ^1}). We show in this section that our method has a higher probability of success to reconstruct solutions with high precision compared to other state-of-the-art methods, i.e., the phase transition is observed with a high precision. Discussions and conclusions are summarized in section 5 on page A4080. Appendix A on page A4081 contains several proofs used throughout this paper. A glossary containing the notation and basic definitions is in section B on page A4090. In what follows, Latin numerals refer to the glossary of notation on page A4090. Section C on page A4091 contains general results on convex analysis used in this paper.

2. An algorithm solving (P_{ℓ^1}).

This section presents the algorithm proposed in this paper. As usual in the literature on compressive sensing, we shall assume that $A \in \mathcal{M}_{m,n}(\mathbb{R})$ with $m \ll n$. The algorithm we shall develop in this paper begins by computing a solution to the dual problem associated to (P_{ℓ^1}) which then computes a solution to the primal. The first step involves the computation of a finite and piecewise affine trajectory, or more precisely the positions $\boldsymbol{\lambda}_k$ where the trajectory changes slope. The second step relies on the computation of a solution to a constrained least square problem. The construction leads to Algorithm 2.1 (page A4070).

Consider the Lagrangian $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ of (P_{ℓ^1}), namely

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

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

$$\begin{aligned} g(\boldsymbol{\lambda}) &:= - \inf_{\boldsymbol{u} \in \mathbb{R}^n} \mathcal{L}(\boldsymbol{u}, \boldsymbol{\lambda}) = - \inf_{\boldsymbol{u} \in \mathbb{R}^n} \left\{ J(\boldsymbol{u}) - \left\langle -A^T \boldsymbol{\lambda}, \boldsymbol{u} \right\rangle \right\} - \langle \boldsymbol{\lambda}, -\boldsymbol{b} \rangle \\ &= J^* \left(-A^T \boldsymbol{\lambda} \right) + \langle \boldsymbol{\lambda}, \boldsymbol{b} \rangle = \chi_{\mathbf{B}_\infty} \left(-A^T \boldsymbol{\lambda} \right) + \langle \boldsymbol{\lambda}, \boldsymbol{b} \rangle, \end{aligned} \tag{2.2}$$

where J^* denotes the Lengendre–Fenchel transform of J (see (xvi)) and $\chi_{\mathbf{B}_\infty}$ denotes the convex characteristic function of ℓ^∞ (see (vii)) unit ball $B_\infty \subset \mathbb{R}^n$ (see (xi)). (We recall that hereafter Latin numerals refer to the glossary of notation on page A4090.)

Consider further the optimization problem

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

where g is given by (2.2). As we shall see, under classic assumptions, problems (P_{ℓ^1}) and (D_{ℓ^1}) have at least one solution (see Proposition 3.4 on page A4071). We now give a strategy to solve (D_{ℓ^1}). With the trajectory $[0, +\infty) \ni t \mapsto \boldsymbol{\lambda}(t)$ explicitly given, for every $t \geqslant 0$,

$$\begin{cases} \dfrac{\mathrm{d}^+ \boldsymbol{\lambda}}{\mathrm{dt}}(t) = -\Pi_{\partial g(\boldsymbol{\lambda}(t))}(\mathbf{0}), \\ \boldsymbol{\lambda}(0) = \boldsymbol{\lambda}_0 \end{cases} \tag{2.3}$$

converges for some finite time $t_K \in [0, +\infty)$ to a solution to (D_{ℓ^1}). The main idea of (2.3) is that it generalizes the usual steepest Euclidean descent for nonsmooth convex functions. When the function is not differentiable, then (2.3) selects the smallest velocity in the ℓ^2 sense among all possible velocities that corresponds to the subdifferential of the function at a nondifferentiable point. Note that the subdifferential always only contains one element, which is the gradient, when the function is differentiable. Formula (2.3) formalizes an evolution equation governed by the (multivalued) maximal monotone operator ∂g (see, for instance, [1, eq. 2, p. 158]). In (2.3), $\Pi_{\partial g(\boldsymbol{\lambda}(t))}$ denotes the Euclidean projection (xviii) on $\partial g(\boldsymbol{\lambda}(t))$ and $\boldsymbol{\lambda}_0 \in \operatorname{dom} g$ is some initial state. We always set $\boldsymbol{\lambda}_0 = \mathbf{0}$ in our experiments. For any $\boldsymbol{\lambda} \in \operatorname{dom} g$ the multivalued monotone operator ∂g is given by the nonempty convex cone

$$\partial g(\boldsymbol{\lambda}) = \left\{ \boldsymbol{b} + \sum_{i \in \mathrm{S}(\boldsymbol{\lambda})} \eta_i A \tilde{\boldsymbol{e}}_i \; : \; \eta_i \geqslant 0, i \in \mathrm{S}(\boldsymbol{\lambda}) \right\}, \tag{2.4}$$

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

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

In (2.5) and everywhere else, $\boldsymbol{e}_i$ denotes the ith canonical vector of $\mathbb{R}^n$.

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

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

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

Note that (2.6) and (2.7) are easy to compute since these quantities are given explicitly and only involve computations of inner products. Therefore, from (2.3) we observe that it remains to compute the direction $\boldsymbol{d}_k = -\Pi_{\partial g(\boldsymbol{\lambda}_k)}(\boldsymbol{0})$ which corresponds to computing the projection on a nonempty closed convex cone given by $\partial g(\boldsymbol{\lambda}_k)$. Note that this subdifferential has an explicit formula given by (2.4). One can use a constrained least square solver, available in MATLAB, to compute the solution. (See also Remark 2.1 below.) To sum up, to compute a solution to (D_{ℓ^1}) one can compute the limit of the trajectory $\boldsymbol{\lambda}(t)$ given by (2.3) using the update rules (2.6) and (2.7). This limit is attained after finitely many updates (see also Proposition 3.15). It remains to compute a solution to (P_{ℓ^1}) given $\bar{\boldsymbol{\lambda}}$ solution to (D_{ℓ^1}).

Given $\bar{\boldsymbol{\lambda}}$ solution to (D_{ℓ^1}), one can compute a solution $\bar{\boldsymbol{u}}$ to (P_{ℓ^1}) by solving the constrained least square problem

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

We are now in position to state the entire algorithm.

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

```
Input: Matrix $A$, $\boldsymbol{b}$
Output: $\bar{\boldsymbol{u}}$ solution to $(P_{\ell^1})$
Set $k := 0$ and $\boldsymbol{\lambda}_k := \boldsymbol{0} \in \mathbb{R}^m$ repeat
    1. Compute $\mathrm{S}(\boldsymbol{\lambda}_k)$ (see (2.5)).
    2. Compute $\boldsymbol{d}_k$ as $\boldsymbol{d}_k := -\Pi_{\partial g(\boldsymbol{\lambda}_k)}(\boldsymbol{0})$ (see Remark 2.1).
    3. Compute $\mathrm{S}^+(\boldsymbol{d}_k)$ (see (2.6)) then $\overline{\Delta t}_k$ (see (2.7)).
    4. Set $\boldsymbol{\lambda}_{k+1} := \boldsymbol{\lambda}_k + \overline{\Delta t}_k\, \boldsymbol{d}_k$.
    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 $\boldsymbol{d}_k = \boldsymbol{0}$ (see Remark 2.1);
Compute $\bar{\boldsymbol{u}}$ using (2.8).
```

Remark 2.1. To compute $\boldsymbol{d}_k$ we define $G := \left\{\sum_{i\in\mathrm{S}(\boldsymbol{\lambda}_k)} \eta_i A\tilde{\boldsymbol{e}}_i : \eta_i \geqslant 0, i \in \mathrm{S}(\boldsymbol{\lambda}_k)\right\}$. We have that $\boldsymbol{d}_k := -\Pi_{\partial g(\boldsymbol{\lambda}_k)}(\boldsymbol{0}) = -\Pi_G(-\boldsymbol{b}) - \boldsymbol{b}$ (see Lemma A.5 on page A4087) can be computed from a constrained least square problem similar to (2.8). We refer to [15, section 3.2] and the references therein for a detailed review of exact (up to machine precision) numerical algorithms solving the above constrained least square problem. For instance, one can use the *lsqnonneg* MATLAB routine, although we used an implementation based on [31] that is supposedly faster than the MATLAB routine. The termination condition, namely $\boldsymbol{d}_k = \boldsymbol{0}$, was replaced by $\|\boldsymbol{d}_k\|_{\ell^2} \vee \|\overline{\Delta} t_k \boldsymbol{d}_k\|_{\ell^2} < 10^{-10}$ in all of our experiments. The projection in step 5 is unnecessary if the precision of numbers is high enough. However, we empirically observed that it increased the performance of the method for the MATLAB implementation.

3. From maximal monotone operator to ℓ^1 solutions of linear problems.

This section justifies the mathematical validity of Algorithm 2.1 presented in section 2.

We recall that to solve (P_{ℓ^1}), we first solve the dual (D_{ℓ^1}) then compute a solution to the primal problem (P_{ℓ^1}). Hence, we first give the assumptions that justify the existence of solutions to problems (P_{ℓ^1}) and (D_{ℓ^1}) and give a closed formula that allows us to compute the solution to (P_{ℓ^1}) from a solution to (D_{ℓ^1}). This is done in Proposition 3.4. We then briefly justify the fact that the trajectory we used in the previous section converges to a solution to the dual. This is done in Proposition 3.6. This proposition translates into Algorithm 2.1 on page A4070 and is illustrated numerically in section 4 on page A4075.

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

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

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

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

Proof. See Appendix A.2 on page A4084. □

Remark 3.2. The assumptions of Proposition 3.1 allow us to cover the case of compressive sensing problems. Noting that one can relax the assumption that A is full row rank, we just need that $\boldsymbol{b} \in \operatorname{span} A$. For instance, if for some specific application the observed $\boldsymbol{b}$'s belong to a subspace B, then we just need span $A \supset B$.

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

Definition 3.3 (active set). *For any $\boldsymbol{\lambda} \in$ dom g we define*

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

and $\boldsymbol{e}_i$ denotes the ith canonical vector of $\mathbb{R}^n$.

Proposition 3.4 (existence of solutions and computation of a solution to (P_{ℓ^1})). *We posit the same assumptions as in Proposition* 3.1*.*

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

$$\left\{ \boldsymbol{y} : \boldsymbol{y} = \sum_{\substack{\tilde{u}_i \geqslant 0\ \ \forall i \in \mathrm{S}(-\bar{\boldsymbol{\lambda}}) \\ \tilde{u}_i = 0\ \ \forall i \in \{1,\dots,2n\}\backslash S(-\bar{\boldsymbol{\lambda}})}} \tilde{u}_i A\tilde{\boldsymbol{e}}_i \right\}, \tag{3.4}$$

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

We have that the vector $\bar{\boldsymbol{u}}$ *obtained from the above coefficients* $\widetilde{u}_i$,

$$\bar{\boldsymbol{u}} := \sum_{i=1}^{n} u_i \boldsymbol{e}_i \text{ with } u_i := \widetilde{u}_i - \widetilde{u}_{i+n}, \tag{3.5}$$

is a solution to (P_{ℓ^1}).

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

Proof. See Appendix A.3 on page A4084. □

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

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

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

$$\begin{cases} \dfrac{\mathrm{d}^+\boldsymbol{\lambda}(t)}{\mathrm{d}t} = -\Pi_{\partial g(\boldsymbol{\lambda}(t))}(\boldsymbol{0}), \\ \boldsymbol{\lambda}(0) = \boldsymbol{\lambda}_0, \end{cases} \tag{3.6}$$

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

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

Proof. See Appendix A.4 on page A4084. □

The proposition above means that the limit of the trajectory $\boldsymbol{\lambda}(t)$ is a solution to (D_{ℓ^1}). In what follows, we shall prove that the limit is attained for a finite time $t \geqslant 0$. It is worth noticing the similarity between (3.6) and inverse scale space methods (see, e.g., [7, 32]). To compute $\boldsymbol{\lambda}(t)$ one could rely on an Euler scheme to approximate the trajectory, for instance. However, a numerical computation of the trajectory $\boldsymbol{\lambda}(t)$ up to the machine precision is doable. This is the goal of the next paragraph.

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

Proposition 3.13 (page A4074) proves that $\boldsymbol{\lambda}(t)$ defined by (3.6) is piecewise affine. In other words, $\boldsymbol{\lambda}(t)$ is made of pieces of straight lines. Hence, the computation of $\boldsymbol{\lambda}(t)$ boils down to the detection of "kicks," i.e., positions where $\boldsymbol{\lambda}(t)$ changes slope and the computation of these slopes. The computation of these slopes is obtained from (3.6) and Lemma 3.8. Propositions 3.10 and 3.14 yield a direct and optimal numerical method to detect kick times, i.e., times t such that $\boldsymbol{\lambda}(t)$ and $\boldsymbol{\lambda}(t+\varepsilon)$ don't have the same slope for some $\varepsilon > 0$. Propositions 3.11 and 3.15 give the termination condition and prove that $\boldsymbol{\lambda}(t)$ converges to a solution to (D_{ℓ^1}) after finitely many kicks. We recall that Proposition 3.4 (page A4071) directly gives an explicit formula that allows us to compute a solution to (P_{ℓ^1}) given a solution to (D_{ℓ^1}) obtained as the limit of the trajectory $\boldsymbol{\lambda}(t)$.

We recall that one of the two main ingredients to compute the trajectory $\boldsymbol{\lambda}(t)$ is the computation of slopes given by a projection onto the closed convex cone $\partial g\left(\boldsymbol{\lambda}(t)\right)$ (see Proposition 3.6 on page A4072). Hence, a closed formula for ∂g is needed. This is the goal of the next proposition that leads to Lemma 3.8.

Proposition 3.7 (the function g defined by (3.2) is polyhedral). *We posit the same assumptions as in Proposition* 3.1. *The function g defined in* (3.2) *is a polyhedral proper and convex function that satisfies dom $g = C \neq \varnothing$ and we have*

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

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

Proof. See Appendix A.5 on page A4084. □

We now give a formula for the subdifferential of g.

Lemma 3.8 (subdifferential formula for g). *We posit the same assumptions as in Proposition* 3.1. *We have dom ∂g = dom $g = C \neq \varnothing$ and, for any $\boldsymbol{\lambda} \in C$,*

$$\partial g(\boldsymbol{\lambda}) = \{\boldsymbol{b}\} + \operatorname{co}\{A\tilde{\boldsymbol{e}}_i : i \in \mathrm{S}(\boldsymbol{\lambda})\}, \tag{3.8}$$

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

Proof. See Appendix A.6 on page A4084. □

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

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

$$\langle \boldsymbol{d}, A\tilde{\boldsymbol{e}}_i\rangle \leqslant 0 \ \forall i \in \mathrm{S}(\boldsymbol{\lambda}) \ \text{and} \tag{3.9}$$

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

Proof. See Appendix A.7 on page A4085. □

Proposition 3.10 (kick time computation). *We posit the same assumptions as in Proposition* 3.1 *and further assume that $\boldsymbol{\lambda} \in$ dom g and that $\boldsymbol{d}$ is a direction that satisfies* (3.9). *Consider $\tilde{\boldsymbol{e}}_i$ given by* (3.3), *the set $\mathrm{S}^+(\boldsymbol{d})$ defined by*

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

and the scalar $\overline{\Delta}t$ *defined by*

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

We have that $\overline{\Delta}t$ *satisfies* $\overline{\Delta}t > 0$*. In addition,* $\mathrm{S}^+(\boldsymbol{d}) = \varnothing$ *iff* $(\boldsymbol{\lambda} + t\boldsymbol{d}) \in$ *dom g for every* $t \geqslant 0$*. Furthermore, we have*

$$(\boldsymbol{\lambda} + t\boldsymbol{d}) \in \textit{dom } g \textit{ iff } t \in \left[0, \overline{\Delta}t\right], \tag{3.13}$$

$$\forall t \in \left[0, \overline{\Delta}t\right), \quad \mathrm{S}\left(\boldsymbol{\lambda} + t\boldsymbol{d}\right) \subset \mathrm{S}\left(\boldsymbol{\lambda}\right) \quad \textit{and} \quad \partial g\left(\boldsymbol{\lambda} + t\boldsymbol{d}\right) \subset \partial g\left(\boldsymbol{\lambda}\right). \tag{3.14}$$

Proof. See Appendix A.8 on page A4085. □

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

$$\boldsymbol{d} := -\Pi_{\partial g(\boldsymbol{\lambda})}(\mathbf{0}) \tag{3.15}$$

is well defined. Consider $\boldsymbol{d}$ *defined by* (3.15) *and* $\overline{\Delta}t$, $\mathrm{S}^+(\boldsymbol{d})$ *defined in Proposition* 3.10*. We have that the three following conditions are equivalent:*

$$\boldsymbol{d} = \mathbf{0} \quad \Leftrightarrow \quad \overline{\Delta}t = +\infty \quad \Leftrightarrow \quad \mathrm{S}^+(\boldsymbol{d}) = \varnothing. \tag{3.16}$$

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

Proof. See Appendix A.9 on page A4074. □

PROPOSITION 3.12 ($\Pi_{\partial g(\boldsymbol{\lambda})}(\mathbf{0})$ is constant on time intervals). *We posit the same assumptions as in Proposition* 3.1*. Consider any* $\boldsymbol{\lambda} \in$ *dom g,* $\boldsymbol{d}$ *defined by* (3.15)*, and* $\overline{\Delta}t$ *defined in Corollary* 3.10*. We have*

$$\forall t \in \left[0, \overline{\Delta}t\right) \qquad \Pi_{\partial g(\boldsymbol{\lambda})}(\mathbf{0}) \in \partial g(\boldsymbol{\lambda} + t\boldsymbol{d}), \tag{3.17}$$

$$\forall t \in \left[0, \overline{\Delta}t\right) \qquad \Pi_{\partial g(\boldsymbol{\lambda})}(\mathbf{0}) = \Pi_{\partial g(\boldsymbol{\lambda}+t\boldsymbol{d})}(\mathbf{0}). \tag{3.18}$$

Proof. See Appendix A.10 on page A4087. □

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

PROPOSITION 3.13 (and definition: piecewise affine trajectory $\boldsymbol{\lambda}(t)$). *We posit the same assumptions as in Proposition* 3.1*. Consider* $\boldsymbol{\lambda}_0 \in$ *dom g and the sequences* $(t_k)_k \subset [0, +\infty]$*,* $(\boldsymbol{d}_k)_k$*, and* $(\boldsymbol{\lambda}(t_k))_k$ *recursively defined by*

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

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

$$\boldsymbol{\lambda}(t) := \boldsymbol{\lambda}(t_k) + (t - t_k)\boldsymbol{d}_k \textit{ for any } \; t \in [t_k, t_{k+1}), \qquad \boldsymbol{\lambda}(t_0) := \boldsymbol{\lambda}_0. \tag{3.20}$$

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

Proof. See Appendix A.11 on page A4089. □

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

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

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

Proof. See Appendix A.12 on page A4089. □

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

Proof. See Appendix A.13 on page A4089. □

We now briefly justify that the computations in Algorithm 2.1 (page A4070) end with a solution to (P_{ℓ^1}) after finitely many iterations. We obtained that for any $\boldsymbol{\lambda}_0 \in \operatorname{dom} g$ (see Proposition 3.13) the sequence defined in (3.19) converges (see Proposition 3.15) after finitely many kicks to a solution to (D_{ℓ^1}). In Algorithm 2.1, the initialization step namely $\boldsymbol{\lambda}_0 = \mathbf{0}$ is valid since $\mathbf{0} \in \operatorname{dom} g$. In addition, it is easily seen that steps 1–5 implement (3.19). From Proposition 3.15, we deduce the validity of the termination condition. Proposition 3.15 justifies that this termination condition is reached after finitely many iterations. Hence, the while loop ends with some $\bar{\boldsymbol{\lambda}}$ solution to (D_{ℓ^1}). Therefore, the computation of $\bar{\boldsymbol{u}}$ solution to (P_{ℓ^1}) is justified by Proposition 3.4. Therefore, the validity of Algorithm 2.1 is proved.

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

4. Experiments.

This section proposes an empirical evaluation of the following methods to solve (P_{ℓ^1}): AISS [7], LARS [18], SPGL1 [40, 41], SeDuMI [38], and Algorithm 2.1. Two parameters settings are consdered for SeDuMI: the first version version which is called "standard precision" (SP) uses the standard parameters provided in the CVX package, while the second version which is called "high precision" (HP) uses the option "cvx_precision best." The supplementary material gives the same comparisons between OMP [35], CoSamp [33], and GISS [32]. Note that OMP, CoSamp, and GISS are greedy-based numerical algorithms. LARS, SPGL1, AISS, and Algorithm 2.1 are ℓ^1-based numerical algorithms. SeDuMi [38] is a toolbox for linear, second order, and semidefinite problems. These methods are compared in terms of a "probability of success" (defined below) and average number of iterations needed. The criterion will be used to observe a so-called phase transition that separates cases where algorithms successfully recover the sparsest solution and when they fail. Note that solutions with high precision are required to observe an accurate phase transition because if the precision of the computed solutions is too poor, then any estimation can be considered as a solution (i.e., a "success" in our experiments). Numerically, it seems to be hard to know a priori the desired precision on the solutions to observe phase transitions. Therefore, it is of interest to have numerical methods that can achieve reconstructions with high precision, i.e., up to the machine precision.

First, we describe the experimental setup. In these experiments the sensing matrix A always has $1,000$ columns. The entries of A are drawn from i.i.d. realizations of a centered Gaussian distribution. Without loss of generality we may normalize the columns of A to unit Euclidean norm. The number of rows of A, i.e., the dimension of the ambient space m, vary in $M := \{50, \dots, 325\}$ with increments of 25. For each number of rows, we vary the *sparsity level* s between 5% and 40% with increments of 5% and therefore consider the discrete set $S := \{0.05, \dots, 0.4\}$. The sparsity level is related to the ℓ^0 norm of $\boldsymbol{u}$ by "$\|\boldsymbol{u}\|_{\ell^0} = \text{round}\,(s \times m)$" following [10]. The positions of the nonzero entries of $\boldsymbol{u}$ are chosen randomly, with uniform probability. The nonzero entries of $\boldsymbol{u}$ are drawn from a uniform distribution on $[-1, 1]$. To do so, for each parameter (i.e., sparsity level s and dimension of ambient space m) we repeated the experiments $1,000$ times. The implementations of AISS and SPGL1 we used are the ones given by the authors of [7, 32, 41]. For LARS [18], we used the SPAMS 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 parameters have been used for all methods. We now give the criteria used for the numerical comparisons of these numerical algorithms.

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

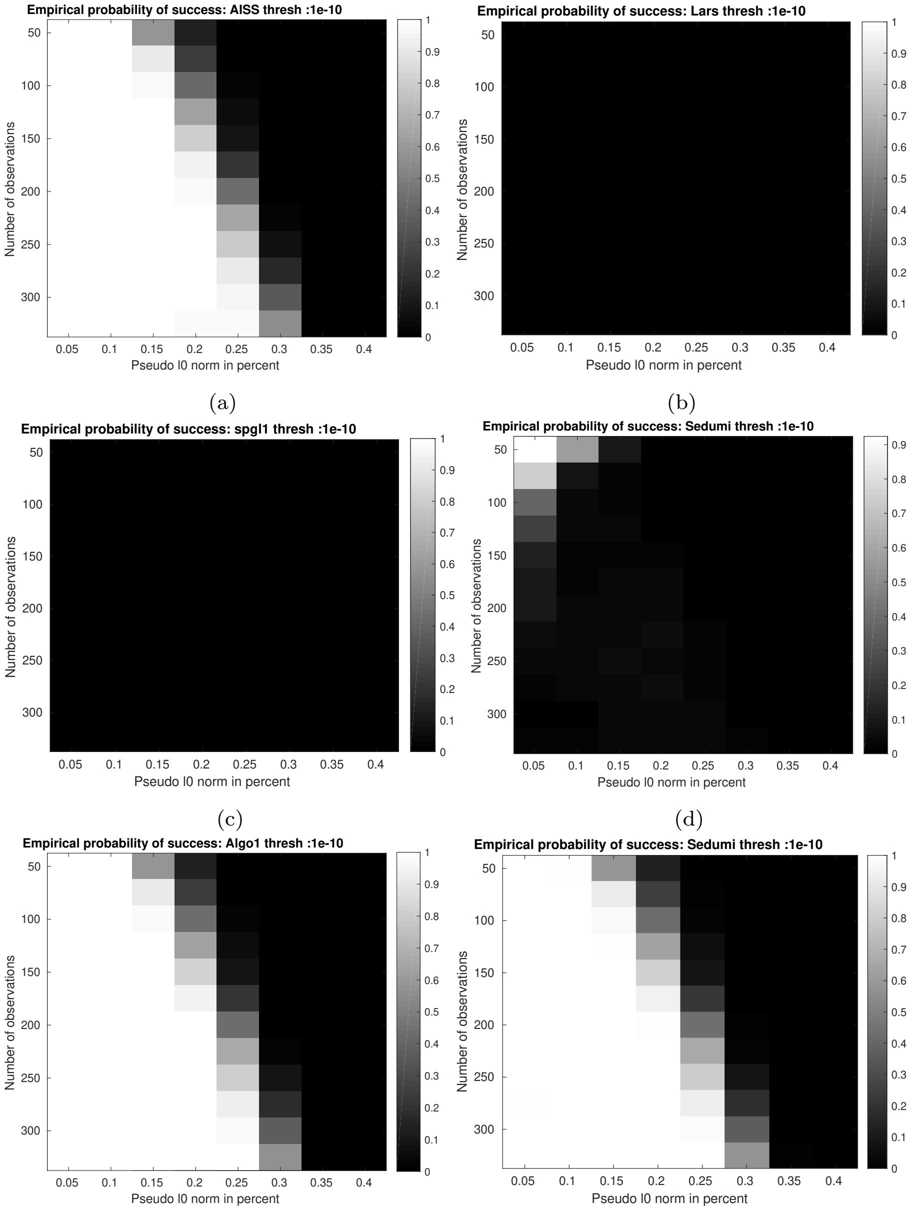
$$P_{(m,s)} := \frac{1}{\#\text{ of tests}} \sum_i \mathbb{1}_{\left\{\frac{\|\boldsymbol{u}^i-\boldsymbol{u}^i_{est}\|_{\ell^2}}{\|\boldsymbol{u}^i\|_{\ell^2}}<\varepsilon\right\}}(i), \tag{4.1}$$

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

Remark that another choice for defining "*success*" could be stated as "the output of an algorithm is a solution to (P_{ℓ^1})." However, this criterion would be verified for every output of Algorithm 2.1. Indeed, Algorithm 2.1 ends with some $\bar{\boldsymbol{u}}$ that numerically verifies an optimality condition associated with (P_{ℓ^1}). Thus, this choice seems uninformative. Therefore, we have decided to not consider this definition of "success" in this paper. We first consider $\varepsilon = 10^{-10}$. Figure 1 depicts the empirical probability of success (4.1) for AISS, LARS, SPGL1, SeDuMi, and Algorithm 2.1. We also consider the difference of probability of success between Algorithm 2.1 and all other methods that is defined as follows:

$$D_{(m,s)} := P^{\text{algorithm 2.1}}_{(m,s)} - P_{(m,s)}, \tag{4.2}$$

where $m \in M,\ s \in S,\ P^{\text{algo 2.1}}_{(m,s)}$ (resp., $P_{(m,s)}$) denotes the quantity (4.1) obtained



(a) (b)

(c) (d)

(e) (f)

Fig. 1. *Empirical probability of success* (4.1), *with* $\varepsilon = 10^{-10}$. *Panel* (a): *AISS* [7]. *Panel* (b): *LARS* [18]. *Panel* (c): *SPGL*1 [40, 41]. *Panel* (d): *SeDuMi (standard precision)* [38]. *Panel* (e): *Algorithm* 2.1. *Panel* (f): *SeDuMi (high precision)* [38]. *The nonzero entries of the source element* $\boldsymbol{u}$ *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, SPGL*1*, and SeDuMi (standard precision) are not able to produce good results for the above set of experiments. However, SeDuMi (high precision) produces good results. We also present results for a higher threshold* $\varepsilon = 10^{-4}$*; see Figure* 3.

with Algorithm 2.1 (resp., AISS, LARS, and SPGL1). Note that a positive (negative) value in (4.2) means that Algorithm 2.1 achieves a higher (lower) probability of

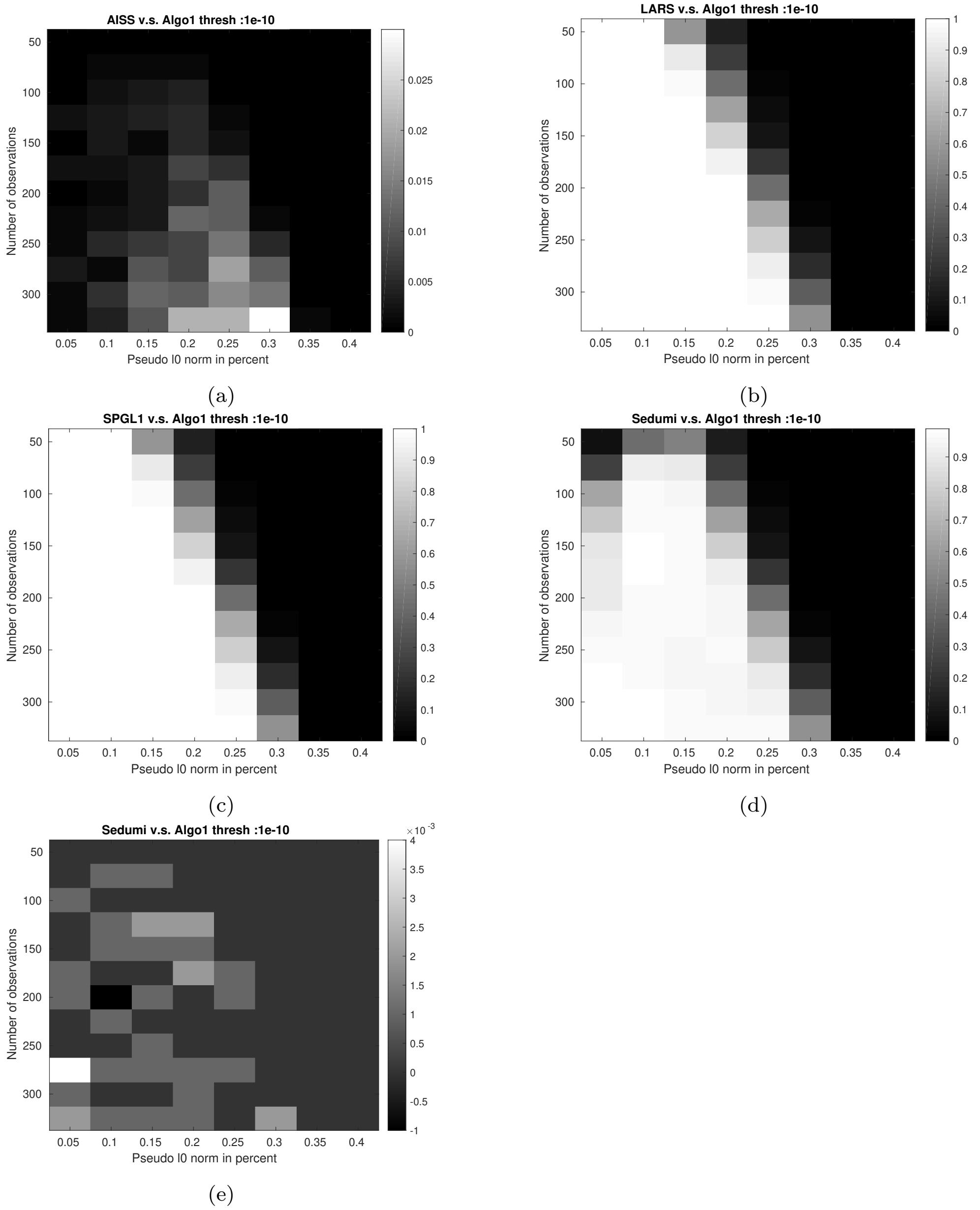


FIG. 2. *Differences of probability of success* (4.2), *with* $\varepsilon = 10^{-10}$. *Panel* (a): *Algorithm* 2.1*–AISS* [7]. *Panel* (b): *Algorithm* 2.1*–LARS* [18]. *Panel* (c): *Algorithm* 2.1*–SPGL*1 [40, 41]. *Panel* (d): *Algorithm* 2.1*–SeDuMi (standard precision)* [38]. *Panel* (e): *Algorithm* 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.*

success than the compared algorithm. These differences of probability of success are depicted in Figure 2. We deduce from Figure 2 that Algorithm 2.1 always achieves a higher probability of success than AISS and GISS. We observe that LARS, SeDuMi (standard precision), and SPGL1 algorithms do not perform well for $\varepsilon = 10^{-10}$ since the probability of success tends to be low, even for problems with very sparse signals.

Table 1

*Main assumption and statistical indicator of "success" for LARS, SPGL*1*, 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.	$\exists \boldsymbol{u} : A\boldsymbol{u} = \boldsymbol{b}$	$\exists \boldsymbol{u} : A\boldsymbol{u} = \boldsymbol{b}$	S.F.P.D.	S.F.P.D.	full row rank
$P_{\geqslant 0.9}$ (4.3)	0 (0.4688)	0 (0.0833)	**0.4688** (0.4688)	0.104 (0.4688)	**0.4688** (0.4688)	**0.4688** (0.4688)
$P_{\geqslant 0.95}$ (4.3)	0 (0.4375)	0 (0.0625)	**0.4375** (0.4375)	0 (0.4583)	**0.4375** (0.4375)	**0.4375** (0.4375)
$P_{\geqslant 0.99}$ (4.3)	0 (0.4167)	0 (0.0104)	0.3438 (0.4167)	0 (0.4167)	**0.4167** (0.4167)	**0.4167** (0.4167)
$P_{\geqslant 0.999}$ (4.3)	0 (0.3750)	0 (0)	0.1250 (0.3646)	0 (0.3333)	0.3229 (0.3437)	**0.3646** (0.3750)
$P_{\geqslant 1}$ (4.3)	0 (0.3646)	0 (0)	0.0521 (0.3646)	0 (0.1875)	0.1562 (0.1979)	**0.3333** (0.3646)

We also observe that both SeDuMi (high precision) and our proposed algorithm produce the best results. Table 1 gives the main assumptions on A and $\boldsymbol{b}$ for LARS [18], SPGL1 [40, 41], AISS [7], SeDuMI [38], and Algorithm 2.1. In this table, we also give the empirical probability that at least x% of signals are successfully reconstructed for each method. This statistical indicator is defined as follows:

$$P_{\geqslant x} = \frac{\#\left\{(m,s) \in M \times S \ : \ P_{(m,s)} \geqslant x\right\}}{\#M \cdot \#S}, \tag{4.3}$$

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

Table 2 presents the time results for AISS, LARS, SPGL1, SeDuMi, and Algorithm 2.1. All experiments are done using a single core of an Intel Core 10600k. We observe that our proposed algorithm is very competitive compared to the state-of-the-art competitors. Indeed, our proposed algorithm outperforms the competitors for sparsity 5/10% and 50/175 rows while the second best algorithm is AISS. The computational time of our proposed algorithm is similar to AISS for sparsity 15/20% and 175/300 rows. For sparsity 25/30% and 175/300 rows AISS performs better than our proposed algorithm. We observe that the runtime of LARS [18], SPGL1 [40, 41], SeDuMi (SP) [38] remains close to constant when the sparsity is greater than or equal to 20%: this suggests that for these levels of sparsity LARS [18], SPGL1 [40, 41], SeDuMi [38] computed poor solutions, as numerically exhibited previously. Recall that SeDuMi (HP) [38] computes very good results, as previously shown, but the computational time is significantly larger than our proposed Algorithm 2.1 and AISS except for the case of 30% sparsity with 300 rows.

As noted above the numerical results for LARS and SPGL1 show that these two numerical methods are not able to produce good results for the above set of experiments with $\varepsilon = 10^{-10}$. We now present numerical experiments for a higher threshold in (4.1) where we set $\varepsilon = 10^{-4}$. Figure 3 depicts the empirical probability of success (4.1) for AISS, LARS, SPGL1, SeDuMi, and Algorithm 2.1. Figure 4 depicts the differences of probability of success. These results for $\varepsilon = 10^{-4}$ show that all numerical algorithms have a higher empirical probability of success compared to the results for $\varepsilon = 10^{-10}$. In particular, we note that SPGL1 and LARS that were performing poorly for $\varepsilon = 10^{-10}$ have dramatically improved their performance. Also, from Figure 4, we observe that LARS and our proposed algorithm produce very similar results. It seems that LARS works for the considered experiments (see

Table 2

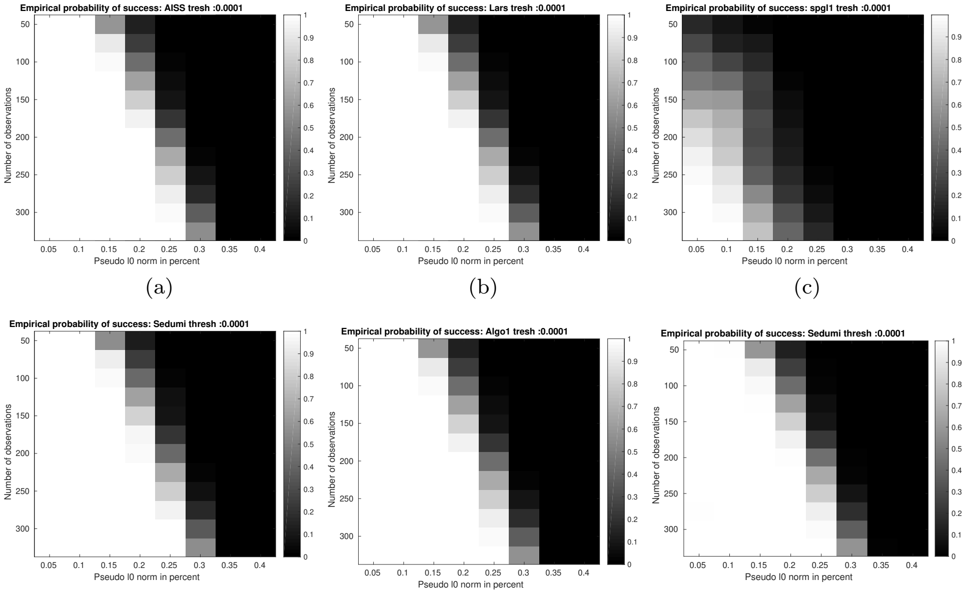
Computational time results for the following methods: Algorithm 2.1*, AISS* [7]*, and SeDuMi (SP)* [38]*, SPGL*1 [40, 41]*, LARS* [18]*, SeDuMi (HP)* [38]*. The number of columns is set to* 1,000 *as everywhere else in this paper and there are various numbers of rows (NR) and several levels of sparsity.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

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)

Figure 4) although it was proved in [6] that LARS may not converge.

5. Conclusion.

In this paper, a new algorithm to solve ℓ^1 regularized linear problems up to the machine precision has been proposed. The method is based on (i) the numerical computation of a finite sequence that converges to a solution to the dual problem and (ii) an explicit recovery formula—based on a nonnegative least square—to compute a solution to the primal problem. The sequence we employed is driven by an evolution equation ruled by a maximal monotone operator. The numerical computations of this algorithm involve the computation of a projection onto a closed convex cone and the evaluation of inner products. The sequence in the dual space lives in a low dimensional space compared to the unknown. Hence, most of the numerical efforts require less memory usage than the primal-based method.



(a) (b) (c)

(d) (e) (f)

FIG. 3. *Empirical probability of success* (4.1)*, with* $\varepsilon = 10^{-4}$*. Panel* (a)*: AISS* [7]*. Panel* (b)*: LARS* [18]*. Panel* (c)*: SPGL*1 [40, 41]*. Panel* (d)*: SeDuMi (standard precision)* [38]*. Panel* (e)*: Algorithm* 2.1*. Panel* (f)*: SeDuMi (high precision)* [38]*. The nonzero entries of the source element* $\boldsymbol{u}$ *are drawn from a uniform distribution on* $[-1, 1]$*. The entries in* A *are drawn from i.i.d. realizations of a Gaussian distribution.*

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

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

We also leave as future work theoretical and numerical comparisons with approximate path-methods (as opposed to piecewise affine paths such as in our approach) such as [27] which corresponds to an approximate discetrization of trajectories. In particular, it would be of interest from a computational point of view to know whether it is better to compute an exact trajectory versus an approximate trajectory.

Appendix A. Proofs.

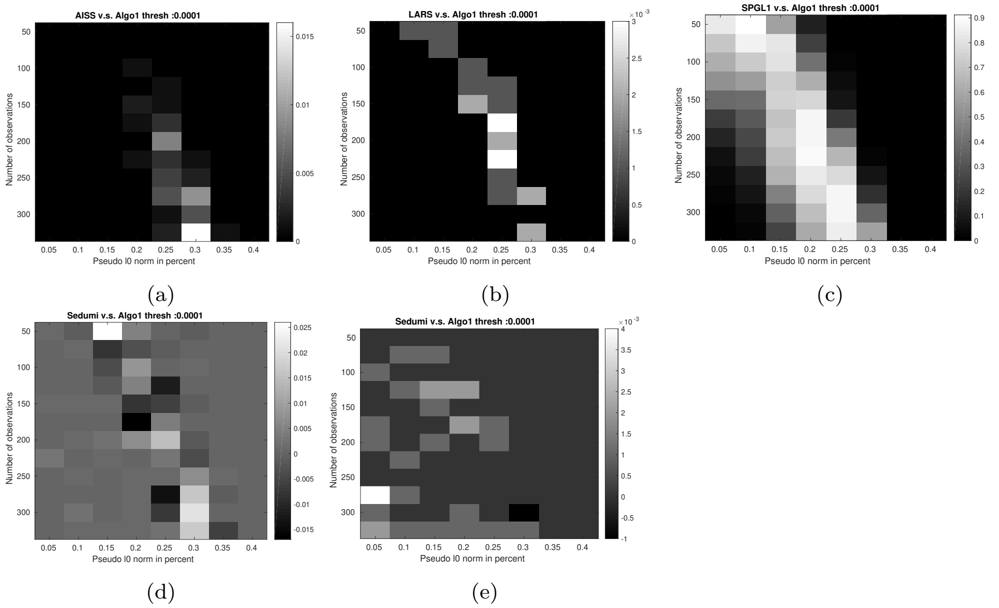
This section contains several proofs used throughout this paper and some properties on the projection on a polyhedral convex cone.

A.1. Some properties of functions J, J^*, f, and $\mathcal{J}$.

LEMMA A.1 (some elementary properties of J and J^*). *We posit the same assumptions as in Proposition* 3.1*. We have the following:*

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

$$\text{(A.1)} \qquad \mathbf{0} \in int(A\ dom\ J - \{\boldsymbol{b}\}) = A\ \mathbb{R}^n - \{\boldsymbol{b}\} = \mathbb{R}^m \ (see\ \text{(iv)}).$$



(a) (b) (c)

(d) (e)

FIG. 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*–SPGL*1 [40, 41]*. Panel* (d)*: Algorithm* 2.1*–SeDuMi (standard precision)* [38]*. 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.*

3. *Dual feasibility:*

$$\mathbf{0} \in int\left(A^T dom\ \chi^*_{\{\boldsymbol{b}\}} + dom\ J^*\right) = int\left(span\ A^T + B_\infty\right). \tag{A.2}$$

Proof. We sequentially prove the three assertions.

Note that dom $J = \mathbb{R}^n$ and that J is convex. It follows that $J \in \Gamma_0(\mathbb{R}^n)$ and, from Proposition C.10, that $J^* \in \Gamma_0(\mathbb{R}^n)$. Combining Lemma C.2 with Proposition C.4 we obtain that for any $\boldsymbol{u} \in \mathbb{R}^n$ we have $J^*(\boldsymbol{u}) = \chi_{\mathbf{B}_\infty}(\boldsymbol{u})$ and dom $J^* = B_\infty$.

From dom $J = \mathbb{R}^n$ and the assumption that A has full row rank, we have A dom $J =$ span $A = \mathbb{R}^m$ and (A.1) immediately follows.

Applying Lemma C.3 with $C := \{\boldsymbol{b}\}$ we have $\chi^*_{\{\boldsymbol{b}\}}(\cdot) = \langle \cdot, \boldsymbol{b}\rangle \in \Gamma_0(\mathbb{R}^m)$ and also dom $\left(\chi^*_{\{\boldsymbol{b}\}}(\cdot)\right) = \mathbb{R}^m$. Since, in addition, dom $J^* = B_\infty$, we have

$$A^T \mathrm{dom}\ \chi^*_{\{\boldsymbol{b}\}} + \mathrm{dom}\ J^* = B_\infty + \mathrm{span}\ A^T. \tag{A.3}$$

We obviously have $B_\infty \subset B_\infty + \mathrm{span}\ A^T$, and from (A.3) we deduce (A.2). □

PROPOSITION A.2 (and definition: function $\mathcal{J}$). *We posit the same assumptions as in Proposition* 3.1*. Consider the function* $\mathcal{J} : \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ *defined by*

$$\forall \boldsymbol{\lambda} \in \mathbb{R}^m, \quad \mathcal{J}(\boldsymbol{\lambda}) := J^*\left(-A^T\boldsymbol{\lambda}\right) = \chi_{\mathbf{C}}(\boldsymbol{\lambda}), \tag{A.4}$$

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

Proof. From item 1 of Lemma A.1, we have $J^* \in \Gamma_0(\mathbb{R}^n)$. Note that (A.2) in Lemma A.1 implies that span $A^T \cap$ dom $J^* \neq \varnothing$. Then, from Theorem C.5 we obtain that $\mathcal{J}(\cdot) := J^*\left(-A^T\cdot\right) \in \Gamma_0(\mathbb{R}^m)$. Moreover, for any $\boldsymbol{\lambda} \in \mathbb{R}^m$, we have

$$\mathcal{J}(\boldsymbol{\lambda}) = \chi_{\mathbf{B}_\infty}\left(A^T\boldsymbol{\lambda}\right) = \chi_{\mathbf{C}}(\boldsymbol{\lambda}). \tag{A.5}$$

The first equality in (A.5) is justified by combining item 1 ($J^* = \chi_{\mathbf{B}_\infty}$) of Lemma A.1 and that $-A^T\boldsymbol{\lambda} \in B_\infty \Leftrightarrow A^T\boldsymbol{\lambda} \in B_\infty$. The second equality in (A.5) is justified by the fact that $\boldsymbol{\lambda} \in C \Leftrightarrow A^T\boldsymbol{\lambda} \in B_\infty$. Indeed, we have

$$\boldsymbol{\lambda} \in C \Leftrightarrow \langle \boldsymbol{\lambda}, A\tilde{\boldsymbol{e}}_i\rangle \leqslant 1 \ \forall i = \{1, \ldots, 2n\} \Leftrightarrow A^T\boldsymbol{\lambda} \in B_\infty, \tag{A.6}$$

where $\tilde{\boldsymbol{e}}_i$ is defined by (3.3). The first equivalence in (A.6) is obvious from the definition of C given by (3.7). The last equivalence follows from the definition of the $\ell^\infty(\mathbb{R}^n)$ unit ball (see (vii)). From (A.5) we can verify that dom $\mathcal{J} = C$ and that $\mathbf{0} \in C \neq \varnothing$. ∎

Lemma A.3 (subdifferential formulas for f, J^*, and $\mathcal{J}$). *We posit the same assumptions as in Proposition* 3.1. *We have the following:*

1. *Subdifferential formula for f:*

$$\forall \boldsymbol{u} \in dom\ J \cap dom\ \chi_{\{\boldsymbol{b}\}}(A\cdot), \qquad \partial f(\boldsymbol{u}) = \partial J(\boldsymbol{u}) + A^T\partial\chi_{\{\boldsymbol{b}\}}(A\boldsymbol{u}). \tag{A.7}$$

2. *Subdifferential formula for J^*:*

$$\forall \boldsymbol{\lambda} \in dom\ g, \qquad \partial J^*\left(-A^T\boldsymbol{\lambda}\right) = N_{B_\infty}\left(-A^T\boldsymbol{\lambda}\right) = \mathrm{co}\{\tilde{\boldsymbol{e}}_i :\ i \in \mathrm{S}(-\boldsymbol{\lambda})\}. \tag{A.8}$$

3. *Subdifferential formula for $\mathcal{J}$:*

$$\forall \boldsymbol{\lambda} \in C \qquad \partial\mathcal{J}(\boldsymbol{\lambda}) = AN_{B_\infty}(A^T\boldsymbol{\lambda}) = \mathrm{co}\{A\tilde{\boldsymbol{e}}_i\ :\ i \in \mathrm{S}(\boldsymbol{\lambda})\}, \tag{A.9}$$

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 set* $\mathrm{S}(\boldsymbol{\lambda})$ *is defined by* (3.3)*, and the $2n$ vectors $\tilde{\boldsymbol{e}}_i$ of $\mathbb{R}^n$ are defined by* (3.3) *(page* A4071*).*

Proof. We sequentially justify (A.7)–(A.9). Combining (A.1) in Lemma A.1 and Theorem C.16 (with "$U = J$" and "$V = \chi_{\{\boldsymbol{b}\}}$") we immediately obtain (A.7). The first equality in (A.8) is justified by Lemma C.6. The second equality in (A.8) follows from Lemma C.7 applied with $p := 2n, \boldsymbol{s}_i := \boldsymbol{e}_i$ for $i = 1, \ldots, n, \boldsymbol{s}_i := -\boldsymbol{e}_{i-n}$ for $i = n+1, \ldots, 2n, r_i := 1$ for $i = 1, \ldots, 2n$, and $W\left(-A^T\boldsymbol{\lambda}\right) = \mathrm{S}\left(-\boldsymbol{\lambda}\right)$. We now justify (A.9). From (A.4) in Proposition A.2 we have $\mathcal{J}(\cdot) = J^*\left(A^T\cdot\right)$. To prove the first equality in (A.9), we need to justify that

$$\mathrm{int}(\mathrm{dom}\ J^*) \cap \mathrm{span}\ A^T \neq \varnothing. \tag{A.10}$$

Assuming that (A.10) holds true, combining item 1 in Lemma A.1 ($J^* \in \Gamma_0(\mathbb{R}^n)$) and Theorem C.9 (with "$f = J^*$") we obtain that $\partial\mathcal{J}(\boldsymbol{\lambda}) = -A\partial J^*\left(-A^T\boldsymbol{\lambda}\right)$. We notice that

$$-\partial J^*\left(-A^T\boldsymbol{\lambda}\right) = \mathrm{co}\{-\tilde{\boldsymbol{e}}, i \in \mathrm{S}(-\boldsymbol{\lambda})\} = \mathrm{co}\left\{\tilde{\boldsymbol{e}}, i \in \mathrm{S}(\boldsymbol{\lambda})\right\} = \partial J^*\left(A^T\boldsymbol{\lambda}\right). \tag{A.11}$$

Indeed, the first equality in (A.11) is justified by (A.8). The second equality is obvious from the definition of $\mathrm{S}(\boldsymbol{\lambda})$ in Definition 3.3 and the last equality follows. From (A.11) we immediately obtain (A.9). We now justify (A.10). From, again, item 1 in Lemma A.1 we have dom $J^* = B_\infty$ and, therefore, deduce that $\mathbf{0} \in \mathrm{int}(\mathrm{dom}\ J^*) \cap \mathrm{span}\ A^T$ which justifies (A.10). This concludes our proof. ∎

A.2. Proof of Proposition 3.1 on page A4071.

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) holds true. From the assumption that A has a full row rank, it follows that span $A \cap$ dom $\chi_{\{\boldsymbol{b}\}}$ = span $A \cap \{\boldsymbol{b}\} \neq \varnothing$. In addition, $\chi_{\{\boldsymbol{b}\}} \in \Gamma_0(\mathbb{R}^m)$ as the characteristic function of the closed convex set $\{\boldsymbol{b}\}$. Therefore, from Theorem C.5 we deduce that $\chi_{\{\boldsymbol{b}\}}(A\cdot) \in \Gamma_0(\mathbb{R}^n)$. Combining Lemma A.1 and Proposition C.10 we have 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 $\chi_{\{\boldsymbol{b}\}}(A\cdot) \in \Gamma_0(\mathbb{R}^n)$.

Now we prove that $g \in \Gamma_0(\mathbb{R}^m)$. From Proposition A.2 the function $\mathcal{J}(\cdot) = J^*\left(-A^T\cdot\right) \in \Gamma_0(\mathbb{R}^m)$. From Proposition C.10 we obtain that $g \in \Gamma_0(\mathbb{R}^m)$ as the sum of the finite valued convex function $\langle \boldsymbol{b}, \cdot\rangle$ and $J^*\left(-A^T\cdot\right) \in \Gamma_0(\mathbb{R}^m)$. The second equality in (3.2) follows from item 1 ($J^* = \chi_{\mathbf{B}_\infty}$) of Lemma A.1. This concludes our proof. □

A.3. Proof of Proposition 3.4 on page A4071.

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

Step 1. Problems $(\boldsymbol{P_{\ell^1}})$ and $(\boldsymbol{D_{\ell^1}})$ have at least one solution. Combining the definitions of function f and g given in Proposition 3.1, Lemma A.1, Proposition C.17, and Theorem C.18 with $U := J$ and $V := \chi_{\{\boldsymbol{b}\}}$, we conclude that problems (P_{ℓ^1}) and (D_{ℓ^1}) have at least one solution. This concludes Step 1. We now turn to Step 2.

Step 2. Formulas (3.4) and (3.5) hold true. From [1, pp. 166–167] applied with " $U := J$ and $V := \chi_{\{\boldsymbol{b}\}}$," we have that any point $\bar{\boldsymbol{u}}$ in the nonempty closed convex set $\mathcal{S}(\bar{\boldsymbol{\lambda}}) = \partial J^*(-A^T\bar{\boldsymbol{\lambda}}) \cap \{\boldsymbol{u} \ : \ A\boldsymbol{u} = \boldsymbol{b}\}$ is a solution to (P_{ℓ^1}). The set $\mathcal{S}(\bar{\boldsymbol{\lambda}})$ is nonempty, and from Step 1 the primal has a solution. Consider $\bar{\boldsymbol{\lambda}}$ solution to (D_{ℓ^1}). Combining Theorem C.11 and Lemma 3.8 we obtain that $\boldsymbol{b} \in \mathrm{co}\left\{-A\tilde{\boldsymbol{e}}, i \in \mathrm{S}\left(\bar{\boldsymbol{\lambda}}\right)\right\} = \mathrm{co}\left\{A\tilde{\boldsymbol{e}}, i \in \mathrm{S}\left(-\bar{\boldsymbol{\lambda}}\right)\right\}$. This means that $\boldsymbol{b}$ can be written as $\boldsymbol{b} = \sum_{i=1}^{2n} \widetilde{u}_i A\tilde{\boldsymbol{e}}_i$, where $\widetilde{u}_i \geqslant 0 \ \forall i \in \left(-\bar{\boldsymbol{\lambda}}\right)$ and $\widetilde{u}_i = 0 \ \forall i \in \{1, \dots, 2n\}\backslash\mathrm{S}\left(-\bar{\boldsymbol{\lambda}}\right)$. Consider $\bar{\boldsymbol{u}}$ defined by (3.5). It is easy to see that $\bar{\boldsymbol{u}} \in \mathcal{S}\left(\bar{\boldsymbol{\lambda}}\right)$ and, therefore, $\bar{\boldsymbol{u}}$ is a solution to (P_{ℓ^1}). This concludes our proof. □

A.4. Proof of Proposition 3.6 on page A4072.

Proof. From Proposition 3.1, we have $g \in \Gamma_0(\mathbb{R}^m)$. Hence, from Proposition C.12 we immediately obtain that $\partial g(\cdot)$ is a maximal monotone operator. Item 1 of Proposition 3.6 follows from Theorem C.14. Items 2 and 3 of Proposition 3.6 follow from Theorem C.15. This concludes our proof. □

A.5. Proof of Proposition 3.7 on page A4073.

Proof. Proposition 3.7 is obvious combining Proposition A.2 and (3.2) in Proposition 3.1. This concludes our proof. □

A.6. Proof of Lemma 3.8 on page A4073.

Proof. Combining Propositions 3.1 and A.2, we have that g can be written as

$$\forall \boldsymbol{\lambda} \in \mathbb{R}^m, \quad g\left(\boldsymbol{\lambda}\right) = \mathcal{J}\left(\boldsymbol{\lambda}\right) + \left\langle \boldsymbol{\lambda}, \boldsymbol{b}\right\rangle. \tag{A.12}$$

From, again, Proposition A.2, we deduce that $\mathrm{int}\left(\mathrm{dom}\ \mathcal{J}\right) \cap \mathrm{int}\left(\mathrm{dom}\ \langle\cdot, \boldsymbol{b}\rangle\right) \neq \varnothing$. Hence, combining Theorem C.13 and Lemma A.3 we obtain (3.8) and that $\forall\boldsymbol{\lambda} \in$ dom $g = C$ we have $\boldsymbol{b} \in \partial g(\boldsymbol{\lambda}) \neq \varnothing$. This concludes our proof. □

A.7. Proof of Proposition 3.9 on page A4073.

Proof. We first establish the following lemma.

LEMMA A.4 (directional derivative and Taylor formula for g). *We posit the same assumptions as in Proposition* 3.1. *For every* $\boldsymbol{\lambda} \in$ *dom* g *and any* $\boldsymbol{d}$ *that satisfies* (3.9), *we have*

$$g'(\boldsymbol{\lambda}, \boldsymbol{d}) = \langle \boldsymbol{d}, \boldsymbol{b} \rangle, \tag{A.13}$$

$$g(\boldsymbol{\lambda} + t\boldsymbol{d}) = g(\boldsymbol{\lambda}) + tg'(\boldsymbol{\lambda}, \boldsymbol{d}) \quad \textit{for some } t > 0 \textit{ small enough.} \tag{A.14}$$

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

$$g(\boldsymbol{\lambda} + t\boldsymbol{d}) - g(\boldsymbol{\lambda}) = \langle \boldsymbol{\lambda} + t\boldsymbol{d}, \boldsymbol{b} \rangle - \langle \boldsymbol{d}, \boldsymbol{b} \rangle = t\langle \boldsymbol{d}, \boldsymbol{b} \rangle. \tag{A.15}$$

Formula (A.13) follows (xiii). Combining (A.13) and (A.15) we deduce (A.14). This concludes our proof. ∎

We first prove that the conditions (3.9)–(3.10) are more necessary than they are sufficient.

If $\boldsymbol{d}$ is a descent direction for g at $\boldsymbol{\lambda} \in \text{dom } g$, then, from Proposition 3.9, $(\boldsymbol{\lambda}+t\boldsymbol{d}) \in \text{dom } g$ for some $t > 0$ small enough. From the definition of C (3.7), it follows that $\boldsymbol{\lambda} + t\boldsymbol{d}$ satisfies, in particular, $\langle \boldsymbol{\lambda} + t\boldsymbol{d}, A\tilde{\boldsymbol{e}}_i \rangle \leqslant 1$ for every $i \in \mathrm{S}(\boldsymbol{\lambda})$. The definition of $\mathrm{S}(\boldsymbol{\lambda})$ (3.3) and the fact that $\boldsymbol{\lambda} \in \text{dom } g$ imply that necessarily $\langle \boldsymbol{d}, A\tilde{\boldsymbol{e}}_i \rangle \leqslant 0$ for every $i \in \mathrm{S}(\boldsymbol{\lambda})$ and (3.9) holds true. In addition, from Proposition 3.9 we have $g(\boldsymbol{\lambda} + t\boldsymbol{d}) < g(\boldsymbol{\lambda})$ for some $t > 0$ enough small, and combining (A.13)–(A.14) we obtain that (3.10) holds true. Hence, (3.9)–(3.10) are necessary conditions. We now turn to the sufficiency.

Conversely, consider $\boldsymbol{d} \in \mathbb{R}^m \backslash \{\boldsymbol{0}\}$ satisfying (3.9)–(3.10). From Proposition 3.7 we have that $\boldsymbol{\lambda} \in \text{dom } g$ satisfies $\langle \boldsymbol{\lambda}, A\tilde{\boldsymbol{e}}_i \rangle \leqslant 1$ for every $i \in \{1, \ldots, 2n\}$. On the one hand, from (3.9), for any $i \in \mathrm{S}(\boldsymbol{\lambda})$ we have $\langle \boldsymbol{d}, A\tilde{\boldsymbol{e}}_i \rangle \leqslant 0$ and, therefore, $\langle \boldsymbol{\lambda} + t\boldsymbol{d}, A\tilde{\boldsymbol{e}}_i \rangle \leqslant 1$ $\forall t > 0$. On the other hand, from $\boldsymbol{\lambda} \in \text{dom } g$ and the definition of $\mathrm{S}(\boldsymbol{\lambda})$ we deduce that for any $i \in \{1, \ldots, 2n\} \backslash \mathrm{S}(\boldsymbol{\lambda})$ we have $\langle \boldsymbol{\lambda}, A\tilde{\boldsymbol{e}}_i \rangle < 1$ and, therefore, that $\langle \boldsymbol{\lambda}+t\boldsymbol{d}, A\tilde{\boldsymbol{e}}_i \rangle \leqslant 1$ for $t > 0$ small enough. Thus, $\langle \boldsymbol{\lambda} + t\boldsymbol{d}, -A\tilde{\boldsymbol{e}}_i \rangle \leqslant 1$ for every $i \in \{1, \ldots, 2n\}$ and $t > 0$ small enough. It follows that $(\boldsymbol{\lambda} + t\boldsymbol{d}) \in \text{dom } g$ for some $t > 0$ small enough. Combining (3.10) and (A.14) we obtain $g(\boldsymbol{\lambda} + t\boldsymbol{d}) < g(\boldsymbol{\lambda})$ for some $t > 0$ small enough. It follows that $\boldsymbol{d}$ is a direction descent for g at $\boldsymbol{\lambda}$. This concludes our proof. ∎

A.8. Proof of Proposition 3.10 on page A4073.

Proof. We recall that, in what follows, $\boldsymbol{\lambda} \in \text{dom } g$ and $\boldsymbol{d}$ is a descent direction for g at $\boldsymbol{\lambda} \in \text{dom } g$. We sequentially consider the three following complementary cases:
Case 1. The case of indexes i such that $i \in \mathrm{S}(\boldsymbol{\lambda})$.
Case 2. The case of $i \in \{1, \ldots, 2n\} \backslash \mathrm{S}(\boldsymbol{\lambda})$ and $\langle A\tilde{\boldsymbol{e}}_i, \boldsymbol{d} \rangle \leqslant 0$.
Case 3. The case of $i \in \{1, \ldots, 2n\} \backslash \mathrm{S}(\boldsymbol{\lambda})$ and $\langle A\tilde{\boldsymbol{e}}_i, \boldsymbol{d} \rangle > 0$.

Case 1. From Proposition 3.9 and (3.9), we have that any descent direction $\boldsymbol{d}$ for g at $\boldsymbol{\lambda} \in \text{dom } g$ satisfies $\langle \boldsymbol{d}, A\tilde{\boldsymbol{e}}_i \rangle \leqslant 0$ for every $i \in \mathrm{S}(\boldsymbol{\lambda})$. From Definition 3.3 (page A4071), for every $i \in \mathrm{S}(\boldsymbol{\lambda})$, we have $\langle \boldsymbol{\lambda}, A\tilde{\boldsymbol{e}}_i \rangle = 1$ and, therefore, deduce that

$$i \in \mathrm{S}(\boldsymbol{\lambda}) \;\Rightarrow\; \langle \boldsymbol{\lambda} + t\boldsymbol{d}, A\tilde{\boldsymbol{e}}_i \rangle \leqslant 1 \quad \forall t \geqslant 0. \tag{A.16}$$

Case 2. For any $i \in \{1,\ldots,2n\}\backslash \mathrm{S}(\boldsymbol{\lambda})$, from $\boldsymbol{\lambda} \in \operatorname{dom} g$ we deduce that $\langle \boldsymbol{\lambda}, A\tilde{\boldsymbol{e}}_i\rangle < 1$. Hence, if $\langle A\tilde{\boldsymbol{e}}_i, \boldsymbol{d}\rangle \leqslant 0$, then

$$i \in \{1,\ldots,2n\}\backslash \mathrm{S}(\boldsymbol{\lambda}) \quad \text{and} \quad \langle A\tilde{\boldsymbol{e}}_i, \boldsymbol{d}\rangle \leqslant 0 \;\Rightarrow\; \langle \boldsymbol{\lambda}+t\boldsymbol{d}, A\tilde{\boldsymbol{e}}_i\rangle < 1 \quad \forall t \geqslant 0. \tag{A.17}$$

From (A.17) it is easy to deduce that

$$i \in \{1,\ldots,2n\}\backslash \mathrm{S}(\boldsymbol{\lambda}) \quad \text{and} \quad \langle A\tilde{\boldsymbol{e}}_i, \boldsymbol{d}\rangle \leqslant 0 \;\Rightarrow\; i \in \{1,\ldots,2n\}\backslash \mathrm{S}(\boldsymbol{\lambda}+t\boldsymbol{d}) \quad \forall t \geqslant 0. \tag{A.18}$$

Case 3. We begin by noticing that

$$\{i \in \{1,\ldots,2n\}\backslash \mathrm{S}(\boldsymbol{\lambda}) \;:\; \langle A\tilde{\boldsymbol{e}}_i, \boldsymbol{d}\rangle > 0\} = \{i \in \{1,\ldots,2n\} \;:\; \langle A\tilde{\boldsymbol{e}}_i, \boldsymbol{d}\rangle > 0\}. \tag{A.19}$$

Indeed, we recall that from Proposition 3.9 that any descent direction $\boldsymbol{d}$ for g at $\boldsymbol{\lambda}$ implies that for every $i \in \mathrm{S}(\boldsymbol{\lambda})$ we have $\langle \boldsymbol{d}, A\tilde{\boldsymbol{e}}_i\rangle \leqslant 0$. Hence, if $\langle A\tilde{\boldsymbol{e}}_i, \boldsymbol{d}\rangle > 0$ for some $i \in \{1,\ldots,2n\}$, then $i \notin \mathrm{S}(\boldsymbol{\lambda})$. This means that $\{i \in \{1,\ldots,2n\}\backslash \mathrm{S}(\boldsymbol{\lambda}) \;:\; \langle A\tilde{\boldsymbol{e}}_i, \boldsymbol{d}\rangle > 0\} \supset \{i \in \{1,\ldots,2n\} \;:\; \langle A\tilde{\boldsymbol{e}}_i, \boldsymbol{d}\rangle > 0\}$ and, therefore, proves (A.19). The converse inclusion is trivial. Hence, Case 3 is, from (3.11), the case defined by $\mathrm{S}^+(\boldsymbol{d})$. For any $i \in \mathrm{S}^+(\boldsymbol{d})$, it is easy to see that

$$i \in \mathrm{S}^+(\boldsymbol{d}) \;\Rightarrow\; \langle \boldsymbol{\lambda}+t\boldsymbol{d}, A\tilde{\boldsymbol{e}}_i\rangle \leqslant 1 \text{ iff } t \in \left[0, \frac{1-\langle A\tilde{\boldsymbol{e}}_i, \boldsymbol{\lambda}\rangle}{\langle A\tilde{\boldsymbol{e}}_i, \boldsymbol{d}\rangle}\right]. \tag{A.20}$$

From (A.20) we obviously deduce that

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

and, therefore, that

$$i \in \mathrm{S}^+(\boldsymbol{d}) \;\Rightarrow\; i \in \{1,\ldots,2n\}\backslash \mathrm{S}(\boldsymbol{\lambda}+t\boldsymbol{d}) \text{ iff } t \in \left[0, \frac{1-\langle A\tilde{\boldsymbol{e}}_i, \boldsymbol{\lambda}\rangle}{\langle A\tilde{\boldsymbol{e}}_i, \boldsymbol{d}\rangle}\right). \tag{A.21}$$

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

$$\forall i \in \mathrm{S}^+(\boldsymbol{d}) \quad \text{we have} \quad \frac{1-\langle A\tilde{\boldsymbol{e}}_i, \boldsymbol{\lambda}\rangle}{\langle A\tilde{\boldsymbol{e}}_i, \boldsymbol{d}\rangle} > 0. \tag{A.22}$$

From (A.16), (A.17), and (A.20) we deduce that $\mathrm{S}^+(\boldsymbol{d}) = \varnothing$ iff $(\boldsymbol{\lambda}+t\boldsymbol{d}) \in \operatorname{dom} g$ for every $t \geqslant 0$. In addition, from (A.16), (A.17), and (A.20) we deduce that if $\mathrm{S}^+(\boldsymbol{d}) \neq \varnothing$, then $(\boldsymbol{\lambda}+t\boldsymbol{d}) \in \operatorname{dom} g$ for every $t \in [0, \overline{\Delta t}]$, where $\overline{\Delta t}$ is defined by (3.12). The fact follows that $\overline{\Delta t} > 0$ from (A.22) and, again, (3.12). It remains to prove that for any $t \in [0, \overline{\Delta t})$ we have $\mathrm{S}(\boldsymbol{\lambda}+t\boldsymbol{d}) \subset \mathrm{S}(\boldsymbol{\lambda})$. To this aim we consider an arbitrary $i \in \{1,\ldots,2n\}\backslash \mathrm{S}(\boldsymbol{\lambda})$. Combining (A.18), (A.21), and the definition of $\overline{\Delta t}$ as a minimum (3.12), we deduce that $i \in \{1,\ldots,2n\}\backslash \mathrm{S}(\boldsymbol{\lambda}+t\boldsymbol{d})$ for any $t \in [0, \overline{\Delta t})$. Hence, by considering the complementary set we obtain that for any $t \in [0, \overline{\Delta t})$ we have $\mathrm{S}(\boldsymbol{\lambda}+t\boldsymbol{d}) \subset \mathrm{S}(\boldsymbol{\lambda})$. Furthermore, the fact that $\partial g(\boldsymbol{\lambda}+t\boldsymbol{d}) \subset \partial g(\boldsymbol{\lambda})$ for all $t \in \left[0, \overline{\Delta t}\right)$ immediately follows from Lemma 3.8. It is easy to see that for every $t \in \left[0, \overline{\Delta t}\right]$ we have $(\boldsymbol{\lambda}+t\boldsymbol{d}) \in \operatorname{dom} g$. ∎

A.9. Proof of Lemma 3.11 on page A4074.

Proof. The proof is in three steps. The first step justifies the well posedness of (3.15). The second step proves that the conditions in (3.16) are equivalent. The last step proves that $\boldsymbol{\lambda} \in \operatorname{dom} g$ is a solution to (D_{ℓ^1}) iff the conditions in (3.16) hold true.

From Lemma 3.8, for any $\boldsymbol{\lambda} \in \operatorname{dom} g$ we have that $\partial g(\boldsymbol{\lambda}) \neq \varnothing$ and obviously closed, convex. Therefore, for any $\boldsymbol{\lambda} \in \operatorname{dom} g$ (3.15) is well posed.

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

From Theorem C.11 we have that $\boldsymbol{\lambda} \in \operatorname{dom} g$ is a solution to (D_{ℓ^1}) iff $\mathbf{0} \in \partial g(\boldsymbol{\lambda})$. Hence, $\boldsymbol{\lambda} \in \operatorname{dom} g$ is a solution to (D_{ℓ^1}) iff $\boldsymbol{d}$ defined by (3.15) satisfies $\boldsymbol{d} = \mathbf{0}$. It follows that $\boldsymbol{\lambda} \in \operatorname{dom} g$ is a solution to (D_{ℓ^1}) iff the conditions in (3.16) hold true. □

A.10. Proof of Proposition 3.12 on page A4074.

Proof. We begin to establish the following lemmas that will be useful for the proof of Proposition 3.12.

LEMMA A.5 (technical lemma). *Consider a convex set $\varnothing \neq K \subset \mathbb{R}^m$. For any $\boldsymbol{x}$ we have* $\Pi_{K+\boldsymbol{x}}(\mathbf{0}) = \Pi_K(-\boldsymbol{x}) + \boldsymbol{x}$.

Proof. From Proposition C.8, we have that a vector $\boldsymbol{y}_{\boldsymbol{x}}$ is the projection of some $\boldsymbol{x}$ on K iff $\langle \boldsymbol{x} - \boldsymbol{y}_{\boldsymbol{x}}, \boldsymbol{y} - \boldsymbol{y}_{\boldsymbol{x}}\rangle \leqslant 0\ \forall \boldsymbol{y} \in K$. Hence, the projection $\boldsymbol{y}_{-\boldsymbol{x}} := \Pi_K(-\boldsymbol{x})$ of $-\boldsymbol{x}$ onto K satisfies $\langle -\boldsymbol{x} - \boldsymbol{y}_{-\boldsymbol{x}}, \boldsymbol{y} - \boldsymbol{y}_{-\boldsymbol{x}}\rangle \leqslant 0$ for all $\boldsymbol{y} \in K$. Therefore we obtain, for all $\boldsymbol{y} \in K - \boldsymbol{x}$, that

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

and from Proposition C.8 we obtain that $\boldsymbol{x} + \boldsymbol{y}_{-\boldsymbol{x}}$ is the projection of $\mathbf{0}$ on $K - \boldsymbol{x}$. In other words, $\Pi_K(-\boldsymbol{x}) + \boldsymbol{x} = \Pi_{K-\boldsymbol{x}}(\mathbf{0})$ and the formula is proved. □

LEMMA A.6 ($-\Pi_{\partial g(\boldsymbol{\lambda})}(\mathbf{0})$ satisfies (3.9)). *We posit the same assumptions as in Proposition* 3.1. *For any $\boldsymbol{\lambda} \in$ dom g consider $\boldsymbol{d}$ defined by* (3.15) *in Lemma* 3.11. *We have that $\boldsymbol{d}$ satisfies* (3.9).

Proof. Consider $\boldsymbol{\lambda} \in \operatorname{dom} g$ and $\mathrm{S}(\boldsymbol{\lambda})$ defined by (3.3) (page A4071). We wish to prove that $\boldsymbol{d} := -\Pi_{\partial g(\boldsymbol{\lambda})}(\mathbf{0})$ satisfies (3.9). From Lemma A.5 applied with $\boldsymbol{x} = \boldsymbol{b}$ and $K := \operatorname{co}\{A\tilde{\boldsymbol{e}}_i,\ i \in \mathrm{S}(\boldsymbol{\lambda})\}$ we obtain $\boldsymbol{d} = -\Pi_{\partial g(\boldsymbol{\lambda})}(\mathbf{0}) = -\boldsymbol{b} - \Pi_K(-\boldsymbol{b})$. Thus, from Proposition C.8 we have that $\Pi_K(-\boldsymbol{b})$ satisfies

$$\langle -\boldsymbol{b} - \Pi_K(-\boldsymbol{b}), \boldsymbol{y} - \Pi_K(-\boldsymbol{b})\rangle \leqslant 0 \qquad \forall \boldsymbol{y} \in K := \operatorname{co}\{A\tilde{\boldsymbol{e}}_i,\ i \in \mathrm{S}(\boldsymbol{\lambda})\}$$

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

$$\text{(A.23)} \qquad \langle \boldsymbol{d}, \boldsymbol{y} - \Pi_K(-\boldsymbol{b})\rangle \leqslant 0 \qquad \forall \boldsymbol{y} \in K := \operatorname{co}\{A\tilde{\boldsymbol{e}}_i,\ i \in \mathrm{S}(\boldsymbol{\lambda})\}.$$

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

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

Consider any $j \in \mathrm{S}(\boldsymbol{\lambda})$ and the coefficients α_i given by $\alpha_j = 1 + \mu_j$, where $\alpha_i = \mu_i$ for $i \neq j \in \mathrm{S}(\boldsymbol{\lambda})$ and $\alpha_i = 0$ otherwise. Note that the vector $\sum_i \alpha_i A\tilde{\boldsymbol{e}}_i \in K$ and that $\sum_i \alpha_i A\tilde{\boldsymbol{e}}_i - \Pi_K(-\boldsymbol{b}) = A\tilde{\boldsymbol{e}}_j$. Thus, from (A.23) we obtain that $\boldsymbol{d}$ satisfies (3.9). □

LEMMA A.7 (descent direction condition). *We posit the same assumptions as in Proposition* 3.1 *and consider* $\boldsymbol{d}$ *defined by* (3.15) *in Lemma* 3.11*. We have that if* $\boldsymbol{d} \neq \boldsymbol{0}$*, then* $\boldsymbol{d}$ *is a descent direction for* g *at* $\boldsymbol{\lambda} \in$ *dom* g*. In addition, for all* $\boldsymbol{\lambda} \in$ *dom* g *we have*

$$g'\left(\boldsymbol{\lambda}, -\Pi_{\partial g(\boldsymbol{\lambda})}(\boldsymbol{0})\right) = -\left\|\Pi_{\partial g(\boldsymbol{\lambda})}(\boldsymbol{0})\right\|_{\ell^2}^2. \tag{A.24}$$

Proof. We consider $\boldsymbol{d}$ defined by (3.15) in Lemma 3.11. We first establish (A.24) then justify that if $\boldsymbol{d} \neq \boldsymbol{0}$, then $\boldsymbol{d}$ is a descent direction for g at $\boldsymbol{\lambda} \in$ dom g. From Proposition C.8, we have that

$$\langle \boldsymbol{d}, \boldsymbol{s} + \boldsymbol{d}\rangle \leqslant 0 \quad \forall \boldsymbol{s} \in \partial g(\boldsymbol{\lambda}) \quad \Leftrightarrow \quad \langle \boldsymbol{d}, \boldsymbol{s}\rangle \leqslant -\|\boldsymbol{d}\|_{\ell^2}^2 \quad \forall \boldsymbol{s} \in \partial g(\boldsymbol{\lambda}). \tag{A.25}$$

We have

$$-\|\boldsymbol{d}\|_{\ell^2}^2 \geqslant \langle \boldsymbol{d}, \boldsymbol{b}\rangle = g'(\boldsymbol{\lambda}, \boldsymbol{d}) = \sup\{\langle \boldsymbol{s}, \boldsymbol{d}\rangle : \boldsymbol{s} \in \partial g(\boldsymbol{\lambda})\} \geqslant \langle -\boldsymbol{d}, \boldsymbol{d}\rangle = -\|\boldsymbol{d}\|_{\ell^2}^2. \tag{A.26}$$

Indeed, in (A.26) the first inequality is obtained by choosing $\boldsymbol{s} = \boldsymbol{b} \in \partial g(\boldsymbol{\lambda})$ in (A.25). Combining Lemma A.6 and Proposition 3.9 we obtain the first equality in (A.26). The second equality is justified by the definition of the subdifferential (see (xvii)). The second inequality follows from $-\boldsymbol{d} \in \partial g(\boldsymbol{\lambda})$. The last equality is obvious. Thus, we obtain (A.24). From (A.24), it follows that if $\boldsymbol{d} \neq 0$, we have that $\boldsymbol{d}$ satisfies (3.9)–(3.10). Hence, from Proposition 3.9 we obtain that $\boldsymbol{d}$ is a descent direction. □

Consider $\boldsymbol{d}$ defined by (3.15) in Lemma 3.11. If $\boldsymbol{d} = \boldsymbol{0}$, then (3.17)–(3.18) hold true. From now on, we assume that $\boldsymbol{d} \neq \boldsymbol{0}$. Let $t \in \left[0, \overline{\Delta t}\right)$, where $\overline{\Delta t}$ is defined in Proposition 3.10 (page A4073). For any $\boldsymbol{\lambda}' \in \mathbb{R}^m$ we have

$$\begin{aligned}
&g(\boldsymbol{\lambda}') \geqslant g(\boldsymbol{\lambda}) + \left\langle \Pi_{\partial g(\boldsymbol{\lambda})}(\boldsymbol{0}), \boldsymbol{\lambda}' - \boldsymbol{\lambda}\right\rangle && \text{(A.27)}\\
&\qquad = g(\boldsymbol{\lambda}) + \left\langle \Pi_{\partial g(\boldsymbol{\lambda})}(\boldsymbol{0}), \boldsymbol{\lambda}' - \boldsymbol{\lambda} - t\boldsymbol{d}\right\rangle - t\left\|\Pi_{\partial g(\boldsymbol{\lambda})}(\boldsymbol{0})\right\|_{\ell^2}^2 && \text{(A.28)}\\
&\qquad = g(\boldsymbol{\lambda}) + \left\langle \Pi_{\partial g(\boldsymbol{\lambda})}(\boldsymbol{0}), \boldsymbol{\lambda}' - \boldsymbol{\lambda} - t\boldsymbol{d}\right\rangle + tg'(\boldsymbol{\lambda}, \boldsymbol{d}) && \text{(A.29)}\\
&\qquad = g(\boldsymbol{\lambda}) + \left\langle \Pi_{\partial g(\boldsymbol{\lambda})}(\boldsymbol{0}), \boldsymbol{\lambda}' - (\boldsymbol{\lambda} + t\boldsymbol{d})\right\rangle + t\langle \boldsymbol{d}, \boldsymbol{b}\rangle && \text{(A.30)}\\
&\qquad = g(\boldsymbol{\lambda} + t\boldsymbol{d}) + \left\langle \Pi_{\partial g(\boldsymbol{\lambda})}(\boldsymbol{0}), \boldsymbol{\lambda}' - (\boldsymbol{\lambda} + t\boldsymbol{d})\right\rangle. && \text{(A.31)}
\end{aligned}$$

The inequality in (A.27) is nothing but the definition of $\Pi_{\partial g(\boldsymbol{\lambda})}(\boldsymbol{0}) \in \partial g(\boldsymbol{\lambda})$ (see (xvii)) and (A.28) follows. Lemma A.7 (we assumed $\boldsymbol{d} \neq \boldsymbol{0}$) justifies (A.29). From Proposition 3.10 (page A4073), for any $t \in \left[0, \overline{\Delta t}\right)$ we have $(\boldsymbol{\lambda} + t\boldsymbol{d}) \in$ dom g, and from Lemma A.4 we obtain (A.30). Equation (A.31) immediately follows from (3.7) in Proposition 3.7. From (A.27)–(A.31) and (xvii) we obtain (3.17). From Proposition C.8, we have that $\Pi_{\partial g(\boldsymbol{\lambda})}(\boldsymbol{0})$ satisfies

$$\forall \boldsymbol{s} \in \partial g(\boldsymbol{\lambda}), \qquad \left\langle -\Pi_{\partial g(\boldsymbol{\lambda})}(\boldsymbol{0}), \boldsymbol{s} - \Pi_{\partial g(\boldsymbol{\lambda})}(\boldsymbol{0})\right\rangle \leqslant 0,$$

which is equivalent to

$$\forall \boldsymbol{s} \in \partial g(\boldsymbol{\lambda}) \qquad \left\|\Pi_{\partial g(\boldsymbol{\lambda})}(\boldsymbol{0})\right\|_{\ell^2}^2 \leqslant \left\langle \boldsymbol{s}, \Pi_{\partial g(\boldsymbol{\lambda})}(\boldsymbol{0})\right\rangle.$$

Hence, from (3.14) in Proposition 3.10 (page A4073) we deduce that $\Pi_{\partial g(\boldsymbol{\lambda})}(\boldsymbol{0})$ satisfies

$$\forall t \in \left[0, \overline{\Delta t}\right) \qquad \forall \boldsymbol{s} \in \partial g(\boldsymbol{\lambda} + t\boldsymbol{d}), \qquad \left\|\Pi_{\partial g(\boldsymbol{\lambda})}(\boldsymbol{0})\right\|_{\ell^2}^2 \leqslant \left\langle \boldsymbol{s}, \Pi_{\partial g(\boldsymbol{\lambda})}(\boldsymbol{0})\right\rangle. \tag{A.32}$$

Combining (3.17), (A.32), and, again, Proposition C.8, we obtain (3.18). □

A.11. Proof of Proposition 3.13 on page A4074.

Proof. The proof is in two steps. We first justify the well posedness of (3.19), then justify that (3.20) coincides with the evolution equation (3.6) (see Proposition 3.6 on page A4072).

Step 1. Let $k = 0$. By assumption, $\boldsymbol{\lambda}(t_k) \in \operatorname{dom} g$. From Lemma 3.11 we have that $\boldsymbol{d}_k$ is well defined. From Proposition 3.10 (page A4073) this implies that t_{k+1} is well defined. In addition, from, again, Proposition 3.10 and the definition of t_{k+1}, it is easy to see that $\boldsymbol{\lambda}(t) \in \operatorname{dom} g$ for every $t \in [t_k, t_{k+1}]$. The rest of the recursion follows. Thus, we obtain that the trajectory $\boldsymbol{\lambda}(t)$ given in (3.20) is mathematically well posed. It remains to show that (3.20) coincides with the trajectory given by (3.6).

Step 2. From Proposition 3.12, the vector $-\Pi_{\partial g(\boldsymbol{\lambda})}(\mathbf{0})$ that appears in (3.6) is piecewise constant on every interval $[t_k, t_{k+1})$. In addition, it is easy that the trajectory given by (3.20) coincides by construction with the solution to the evolution equation (3.6) for every $t \geqslant 0$. The fact remains that for every $t \geqslant 0$, $\boldsymbol{\lambda}(t) \in \operatorname{dom} g$ follows combining Proposition 3.6 and Lemma 3.8. This concludes our proof. □

A.12. Proof of Proposition 3.14 on page A4075.

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

$$\left\|\Pi_{\partial g(\boldsymbol{\lambda}(t_{k+1}))}(\mathbf{0})\right\|_{\ell^2} \leqslant \left\|\Pi_{\partial g(\boldsymbol{\lambda}(t_k))}(\mathbf{0})\right\|_{\ell^2}. \tag{A.33}$$

From Proposition 3.13 (page A4074), we have that $\boldsymbol{\lambda}(t_{k+1}) \in \operatorname{dom} g$ and, therefore, from Lemma 3.8 we have $\partial g(\boldsymbol{\lambda}(t_{k+1})) \neq \varnothing$. The uniqueness of the projection of $\mathbf{0}$ onto the nonempty closed convex set $\partial g(\boldsymbol{\lambda}(t_{k+1}))$ and (3.17) imply that

$$\left\|\Pi_{\partial g(\boldsymbol{\lambda}(t_k))}(\mathbf{0})\right\|_{\ell^2}^2 = \left\|\Pi_{\partial g(\boldsymbol{\lambda}(t_{k+1}))}(\mathbf{0})\right\|_{\ell^2}^2 \Leftrightarrow \Pi_{\partial g(\boldsymbol{\lambda}(t_k))}(\mathbf{0}) = \Pi_{\partial g(\boldsymbol{\lambda}(t_{k+1}))}(\mathbf{0}). \tag{A.34}$$

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 $\boldsymbol{d}_k := -\Pi_{\partial g(\boldsymbol{\lambda}(t_k))}(\mathbf{0})$ and $\boldsymbol{d}_{k+1} := -\Pi_{\partial g(\boldsymbol{\lambda}(t_{k+1}))}(\mathbf{0})$ and denote by $\overline{\Delta} t_k$ (resp., $\overline{\Delta} t_{k+1}$) the positive kick times computed, from Proposition 3.10, at $\boldsymbol{\lambda}(t_k)$ (resp., $\boldsymbol{\lambda}(t_{k+1})$).

By assumption, we have that $\boldsymbol{\lambda}(t_k)$ is not a solution to (D_{ℓ^1}). From Lemma 3.11 we have that $\boldsymbol{d}_k \neq \mathbf{0}$. Assume, for the sake of contradiction, that $\boldsymbol{d}_k = \boldsymbol{d}_{k+1}$. From, again, Proposition 3.10, we would have $(\boldsymbol{\lambda}(t_k) + \overline{\Delta} t_k \boldsymbol{d}_k) + t\boldsymbol{d}_k = \boldsymbol{\lambda}(t_{k+1}) + t\boldsymbol{d}_k \in \operatorname{dom} g$, for some positive $t \in \left(0, \overline{\Delta} t_{k+1}\right)$. This is impossible. Indeed, Proposition 3.10 applied at $\boldsymbol{\lambda}(t_k)$ with the direction $\boldsymbol{d}_k$ implies that $(\boldsymbol{\lambda}(t_k) + t\boldsymbol{d}_k) \in \operatorname{dom} g$ iff $t \in \left[0, \overline{\Delta} t_k\right]$. Thus, we obtain that $\boldsymbol{d}_k \neq \boldsymbol{d}_{k+1}$ and combining (A.33)–(A.34) we obtain (3.21). □

A.13. Proof of Proposition 3.15 on page A4075.

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

Step 1. This part of the proof follows a classic approach that can be found in, e.g., [22, Thm. 3.4.8, p. 382]. From Lemma 3.8 (page A4073) there are 4^n possible sets $\partial g(\boldsymbol{\lambda})$ for $\boldsymbol{\lambda} \in \operatorname{dom} g$. Each of them is uniquely associated with $\boldsymbol{d}_k = -\Pi_{\partial g(\boldsymbol{\lambda})}(\mathbf{0})$. From Proposition 3.14 they are all different from each other. This implies that the sequence $(\boldsymbol{d}_k)_k$ has a finite number of terms (is finite) and, therefore, converges for a finite index K. From Proposition 3.10 (page A4073) it is easy to deduce that the sequence $(t_k)_k$ is also finite. From Proposition 3.13 we deduce the existence of K. In other words, we obtained that the trajectory $\boldsymbol{\lambda}(t)$ given in (3.20) satisfies, for some $K \in \mathbb{N}$, $\boldsymbol{\lambda}(t) = \boldsymbol{\lambda}(t_K)$ for every $t \geqslant t_K$. We now turn to the second step of the proof.

Step 2. From Proposition 3.13, we have that the trajectory $\boldsymbol{\lambda}(t)$ given in (3.20) coincides with the trajectory given by (3.6). Thus, the limit of $\boldsymbol{\lambda}(t)$ when $t \to +\infty$ is a solution to (D_{ℓ^1}). The fact that (D_{ℓ^1}) has a solution is justified by Proposition 3.4 page A4071). From step 1, we have that the limit of $\boldsymbol{\lambda}(t)$ is attained for $t = t_K$. Hence, $\boldsymbol{\lambda}(t_K)$ is a solution to (3.4). From Lemma 3.11 we immediately obtain $\boldsymbol{d}_K = \mathbf{0}$. □

Appendix B. Glossary of notation and definitions.

(i) (Vectors) Throughout this paper the vectors of, e.g., $\mathbb{R}^n$ are denoted in bold typeface, e.g., $\boldsymbol{x}$. Other objects like scalars or functions are denoted in non-bold typeface.

(ii) (Canonical vectors) Throughout the paper the ith canonical vectors of, e.g., $\mathbb{R}^n$ are denoted by $\boldsymbol{e}_i$.

(iii) (Inner product) For $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ we denote by $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ the Euclidean inner product in $\mathbb{R}^n$.

(iv) (Interior) $\operatorname{int}(E)$: interior of a set E.

(v) (Conical hull) $\operatorname{co}\{\boldsymbol{a}_1, \dots, \boldsymbol{a}_p\} := \{\sum_{i=1}^{p} \mu_i \boldsymbol{a}_i \ : \ \mu_i \geqslant 0 \ \forall i = 1, \dots, p\}$.

(vi) Normal cone to a convex set $C \neq \varnothing$ at $\boldsymbol{\lambda} \in C$:
$N_C(\boldsymbol{\lambda}) = \{\boldsymbol{s} \ : \ \langle \boldsymbol{s}, \boldsymbol{s}' - \boldsymbol{\lambda} \rangle \leqslant 0 \ \forall \boldsymbol{s}' \in C\}$ (see, e.g., [22, Def. 5.2.3, p. 136]).

(vii) ($\ell^\infty(\mathbb{R}^n)$ unit ball) $B_\infty = \{\boldsymbol{u} : \langle \boldsymbol{u}, \tilde{\boldsymbol{e}}_i \rangle \leqslant 1, \, i = 1, \dots, 2n\}$, where $\tilde{\boldsymbol{e}}_i$ is given by (3.3).

(viii) (Effective domain) dom f: The domain of a convex function f is the (convex, possibly empty) set $\operatorname{dom} f = \{\boldsymbol{x} \in \mathbb{R}^n \ : \ f(\boldsymbol{x}) \in \mathbb{R}\}$.

(ix) (Convex function) A function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is said to be convex if $\forall (\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^n \times \mathbb{R}^n$ and $\forall \alpha \in (0,1)$ $f(\alpha \boldsymbol{x} + (1-\alpha)\boldsymbol{y}) \leqslant \alpha f(\boldsymbol{x}) + (1-\alpha) f(\boldsymbol{y})$ holds true (in $\mathbb{R} \cup \{+\infty\}$).

(x) (Set $\Gamma_0(\mathbb{R}^n)$) The set of lower semicontinuous, convex functions with dom $f \neq \varnothing$ is denoted $\Gamma_0(\mathbb{R}^n)$.

(xi) (Characteristic function of a set) $\chi_E(\boldsymbol{x}) = 0$ if $\boldsymbol{x} \in E$ and $\chi_E(\boldsymbol{x}) = +\infty$ otherwise.

(xii) (Polyhedral convex function) f is a polyhedral convex function if $f(\boldsymbol{u}) = h(\boldsymbol{u}) + \chi_{\mathbf{C}}(\boldsymbol{u})$ $h(\boldsymbol{u}) = \max_{i=1,\dots,p} (\langle \boldsymbol{u}, \boldsymbol{a}_i \rangle - r_i)$ and
$C = \{\boldsymbol{u} \in \mathbb{R}^n \langle \boldsymbol{u}, \boldsymbol{\alpha}_i \rangle \leqslant \rho_i, \ i = 1, \dots, q\}$.

(xiii) (Directional derivative) The directional derivative of $f \in \Gamma_0(\mathbb{R})$ at $\boldsymbol{a} \in \operatorname{dom} f$ in the direction $\boldsymbol{d}$ is $f'(\boldsymbol{a}, \boldsymbol{d}) := \lim_{t \to 0^+} \frac{f(\boldsymbol{a} + t\boldsymbol{d}) - f(\boldsymbol{a})}{t}$.

(xiv) (Right derivative) $\frac{\mathrm{d}^+ \boldsymbol{\lambda}(t)}{\mathrm{d}t} := \lim_{h \to 0^+} \frac{\boldsymbol{\lambda}(t+h) - \boldsymbol{\lambda}(t)}{h}$.

(xv) (Descent direction) $\boldsymbol{d} \neq \mathbf{0}$ is a descent direction for f at $\boldsymbol{x}$ if $\exists t > 0$ such that $\boldsymbol{x} + t\boldsymbol{d} \in \operatorname{dom} f$ and $f(\boldsymbol{x} + t\boldsymbol{d}) < f(\boldsymbol{x})$ (see, e.g., [22, Def. 1.1.1, p. 343]).

(xvi) (Convex conjugate) For any f convex that satisfies dom $f \neq \varnothing$, the function f^* defined by $\mathbb{R}^n \ni \boldsymbol{s} \mapsto f^*(\boldsymbol{s}) := \sup_{\boldsymbol{x} \in \operatorname{dom} f} \{\langle \boldsymbol{s}, \boldsymbol{x} \rangle - f(\boldsymbol{x})\}$ (see, e.g., [23, 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, Thm. 1.1.2., p. 38]).

(xvii) (Subdifferential) For $f \in \Gamma_0(\mathbb{R}^n)$ and $\boldsymbol{x} \in \operatorname{dom} f$ the vector $\boldsymbol{s} \in \mathbb{R}^n$ is a subgradient of f at $\boldsymbol{x}$ if one of the following equivalent assertions is satisfied:

$$\text{(B.1)} \quad \forall \boldsymbol{y} \in \mathbb{R}^n, \ f(\boldsymbol{y}) \geqslant f(\boldsymbol{x}) + \langle \boldsymbol{s}, \boldsymbol{y} - \boldsymbol{x} \rangle, \ \text{or} \ \ \forall \boldsymbol{d} \in \mathbb{R}^n, \ \langle \boldsymbol{s}, \boldsymbol{d} \rangle \leqslant f'(\boldsymbol{x}, \boldsymbol{d}).$$

We denote by $\partial f(\boldsymbol{x})$ the closed convex set of vectors $\boldsymbol{s} \in \mathbb{R}^n$ that satisfy (B.1). For $\boldsymbol{x} \notin \operatorname{dom} f$ we set $\partial f(\boldsymbol{x}) := \varnothing$.

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

Appendix C. Mathematical background. This section contains several propositions and theorems used throughout proofs given in Appendix A.

THEOREM C.1. *For any $f \in \Gamma_0(\mathbb{R}^n)$ we have $f^* \in \Gamma_0(\mathbb{R}^n)$.*

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

LEMMA C.2 (conjugate of absolute value [4, Table 3.1, p. 76]). *Let $f : x \in \mathbb{R} \mapsto f(x) := |x|$. We have for all $v \in \mathbb{R}$,*

$$f^*(v) = \chi_{\mathbf{-1},\mathbf{1}}(v) = \begin{cases} 0 & \text{if } v \in [-1,1], \\ +\infty & \text{otherwise.} \end{cases}$$

LEMMA C.3 (conjugate of characteristic function [23, Ex. 1.1.5, p. 39]). *The conjugate of the characteristic function of the nonempty convex set set C (see* (xi)*) is for all $\boldsymbol{v} \in \mathbb{R}^n$, $\chi^*_{\mathbf{C}}(\boldsymbol{v}) = \sup_{x \in C}\langle \boldsymbol{v}, \boldsymbol{x}\rangle$.*

PROPOSITION C.4 (conjugation in product spaces [36, Prop. 11.22, p. 493]). *Let $f_1, \ldots, f_n$ be in $\Gamma_0(\mathbb{R})$, and $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be given by $\forall (x_1, \ldots, x_n) \in \mathbb{R}^n, f(x_1, \ldots, x_n) = f_1(x_1) + \cdots + f_n(x_n)$. Then $f^*(v_1, \ldots, v_n) = f_1^*(v_1) + \cdots + f_n^*(v_n)$.*

THEOREM C.5 (precomposition with a matrix [22, Prop. 2.1.5, p. 159]). *Let $f \in \Gamma_0(\mathbb{R}^m)$ and $A \in \mathcal{M}_{m\times n}(\mathbb{R})$, and assume that span $A \cap$ dom $f \neq \varnothing$. We have $f(A\cdot) \in \Gamma_0(\mathbb{R}^n)$.*

LEMMA C.6 (subdifferential of normal cone to a closed convex set [23, Def. 1.1.3, p. 93]). *The set of normal directions to a closed convex set $C \subset \mathbb{R}^m$ at $\boldsymbol{\lambda} \in C$ is the subdifferential of the characteristic function $\chi_{\mathbf{C}}$ at $\boldsymbol{\lambda}$: $N_C(\boldsymbol{\lambda}) := \partial \chi_{\mathbf{C}}$.*

LEMMA C.7 (see [22, Ex. 5.2.6 b), p. 138]). *Let a closed convex polyhedron be defined by $C := \{\boldsymbol{u} \in \mathbb{R}^n : \langle \boldsymbol{s}_i; \boldsymbol{x}\rangle \leqslant r_i \text{ for } i = 1, \ldots, p\}$, where $\boldsymbol{s}_i \in \mathbb{R}^n$ and $r_i \in \mathbb{R}$ for all $i = 1, \ldots, p$. Let the set of active constraints at $\boldsymbol{u} \in C$ be defined by $W(\boldsymbol{u}) = \{i \in 1, \ldots, p : \langle \boldsymbol{s}_i; \boldsymbol{x}\rangle = r_i\}$. Then we have $N_C(\boldsymbol{u}) = \text{co}\{\boldsymbol{s}_i : i \in W(\boldsymbol{u})\}$.*

PROPOSITION C.8 (see [22, Thm 3.1.1, p. 117]). *Let C be a nonempty closed convex set of $\mathbb{R}^n$. We have that $\boldsymbol{y}_x \in C$ is the Euclidean projection of some $\boldsymbol{x}$ onto C if only if $\langle \boldsymbol{x} - \boldsymbol{y}_x, \boldsymbol{y} - \boldsymbol{y}_x\rangle \leqslant 0$ for all $\boldsymbol{y} \in C$.*

THEOREM C.9 (subdifferential of precomposition with a matrix). *Let $f \in \Gamma_0(\mathbb{R}^n)$ such that int(dom f) $\neq \varnothing$ and $A \in \mathcal{M}_{m\times n}(\mathbb{R})$. Assume that int(dom f) $\cap$ span $A \neq \varnothing$. Then, from any $\boldsymbol{u} \in \mathbb{R}^n$ such that $A\boldsymbol{u} \in$ dom f, we have $\partial\left(f(A\cdot)\right)(\boldsymbol{u}) = A^T \partial f\left(A\boldsymbol{u}\right)$.*

Proof. Since we assumed that int (dom f) $\neq \varnothing$, we have ri (dom f) $\cap$ span A = int (dom f) $\cap$ span $A = \varnothing$. Thus, from [23, Thm. 3.2.1, p. 117] applied with $\varepsilon = 0$ and $g := f$ we conclude the proof. □

PROPOSITION C.10 (see [22, Prop. 2.1.1, p. 158]). *Let $f_1 \in \Gamma_0(\mathbb{R}^n), \ldots, f_p \in \Gamma_0(\mathbb{R}^n)$ and $t_l, \ldots, t_p$ be positive numbers. We assume that there is a point where all the f_j are finite. Then the function $\sum_{i=1}^p t_i f_i \in \Gamma_0(\mathbb{R}^n)$.*

THEOREM C.11 (Fermat's rule [36, Thm. 10.1, p. 422]). *Let $f \in \Gamma_0(\mathbb{R}^n)$. Then f has a global minimum at $\bar{\boldsymbol{u}}$ iff $\mathbf{0} \in \partial f\left(\bar{\boldsymbol{u}}\right)$.*

PROPOSITION C.12 (see [1, Prop. 1, p. 159]). *Let $f \in \Gamma_0(\mathbb{R}^n)$. Then the set-valued map $\mathbb{R}^n \ni \boldsymbol{u} \mapsto \partial f(\boldsymbol{u})$ is maximal monotone.*

THEOREM C.13 (subdifferential of sum of Γ_0-functions). *Let $f_1, f_2 \in \Gamma_0(\mathbb{R}^n)$. We assume that int (dom f_1) $\cap$ int (dom f_2) $\neq \varnothing$. Then for all $\boldsymbol{u} \in$ dom $(f_1 + f_2)$ we have $\partial \left(f_1 + f_2\right)(\boldsymbol{u}) = \partial f_1(\boldsymbol{u}) + f_2(\boldsymbol{u})$.*

Proof. Since int (dom f_1) $\cap$ int (dom f_2) $\neq \varnothing$ we deduce that ri (dom f_1) $\cap$ ri (dom f_2) $=$ int (dom f_1) $\cap$ int (dom f_2) $\neq \varnothing$. Thus, from [23, Cor. 3.1.2, p. 114] applied with $\varepsilon = 0$ we conclude the proof. ◻

THEOREM C.14 (see [5, Thm. 3.1, p. 54]). *Let T be a maximal monotone operator from $\mathbb{R}^m$ to $\mathbb{R}^m$ and let dom (T) be its domain. Consider the problem $\frac{\mathrm{d}\boldsymbol{\lambda}(t)}{\mathrm{d}t} \in -T(\boldsymbol{\lambda}(t))$ with $\boldsymbol{\lambda}(0) = \boldsymbol{\lambda}_0$. For all $\boldsymbol{\lambda}_0 \in$ dom (T), there exists a unique solution $\boldsymbol{\lambda}(\cdot) : [0, +\infty) \to \mathbb{R}^m$ such that*

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

THEOREM C.15 (see [1, Thm. 2, p. 160]). *Let $g \in \Gamma_0(\mathbb{R}^m)$, and assume that g achieves its minimum at some point. Then, for all $\boldsymbol{\lambda}_0 \in$ dom (∂g), the trajectory given by $\frac{\mathrm{d}^+\boldsymbol{\lambda}(t)}{\mathrm{d}t} = -\Pi_{\partial g(\boldsymbol{\lambda}(t))}(\mathbf{0})$ with $\boldsymbol{\lambda}(0) = \boldsymbol{\lambda}_0$ converges to a point which minimizes g when $t \to +\infty$.*

THEOREM C.16 (see [1, Thm. 4, eq. (28), pp. 35–36]). *Let $A \in \mathcal{M}_{m\times n}(\mathbb{R})$ and $U \in \Gamma_0(\mathbb{R}^n), V \in \Gamma_0(\mathbb{R}^m)$. Assume that $0 \in$ int$(A$ dom $U -$ dom $V)$. Then, for all $\boldsymbol{u} \in$ dom $U \cap$ dom $V(A\cdot)$ we have $\partial \left(U + V(A\cdot)\right)(\boldsymbol{u}) = \partial U(\boldsymbol{u}) + A^T \partial V\left(A\boldsymbol{u}\right)$.*

PROPOSITION C.17 (see [1, Prop. 1, p.163]). *Let $A \in \mathcal{M}_{m\times n}(\mathbb{R})$ and $U \in \Gamma_0(\mathbb{R}^n), V \in \Gamma_0(\mathbb{R}^m)$. Assume that $\mathbf{0} \in$ int$\left(A^T\right.$ dom $V^* +$ dom $\left.U^*\right)$. Then, $\inf_{\boldsymbol{u}\in\mathbb{R}^n}$ $(U(\boldsymbol{u}) + V(A\boldsymbol{u}))$ has a solution.*

THEOREM C.18 (see [1, Thm. 2, p. 167]). *Let $A \in \mathcal{M}_{m\times n}(\mathbb{R})$, $U \in \Gamma_0(\mathbb{R}^n)$ and $V \in \Gamma_0(\mathbb{R}^m)$. Assume that the assumptions of Theorem* C.16 *and Proposition* C.17 *hold. Then, $\inf_{\boldsymbol{\lambda}\in\mathbb{R}^m} \left(U^*(-A^T\boldsymbol{\lambda}) + V^*(\boldsymbol{\lambda})\right)$ has a solution.*

Acknowledgments. The authors thank S. Ladjal and K. Trabelsi for their colorful comments and suggestions. The authors also thank M. Möller for providing the implementation of the GISS algorithm.

REFERENCES

[1] J.-P. AUBIN AND A. CELLINA, *Differential Inclusions: Set-Valued Maps and Viability Theory*, Grundlehren Math. Wiss., Springer, Berlin, Heidelberg, 2012.

[2] A. BECK AND M. TEBOULLE, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., 2 (2009), pp. 183–202, https://doi.org/10.1137/080716542.

[3] T. BLUMENSATH AND M. E. DAVIES, *Iterative hard thresholding for compressed sensing*, Appl. Comput. Harmon. Anal., 27 (2009), pp. 265–274.

[4] J. M. BORWEIN AND A. S. LEWIS, *Convex Analysis and Nonlinear Optimization: Theory and Examples*, CMS Books Math., Springer, New York, 2000.

[5] H. BRÉZIS, *Opérateurs Maximaux Monotones et Semi-Groupes de Contractions Dans Les Espaces de Hilbert*, North-Holland Publishing, Amsterdam, American Elsevier Publishing, New York, 1973.

[6] B. BRINGMANN, D. CREMERS, F. KRAHMER, AND M. MÖLLER, *The homotopy method revisited: Computing solution paths of ℓ_1-regularized problems*, Math. Comp., 87 (2018), pp. 2343–2364, https://doi.org/10.1090/mcom/3287.

[7] M. Burger, M. Möller, M. Benning, and S. Osher, *An adaptive inverse scale space method for compressed sensing*, Math. Comp., 82 (2013), pp. 269–299.

[8] J.-F. Cai, S. Osher, and Z. Shen, *Convergence of the linearized Bregman iteration for ℓ_1-norm minimization*, Math. Comp., 78 (2009), pp. 2127–2136.

[9] J.-F. Cai, S. Osher, and Z. Shen, *Linearized Bregman iterations for compressed sensing*, Math. Comp., 78 (2009), pp. 1515–1536.

[10] E. J. Candès, J. Romberg, and T. Tao, *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information*, IEEE Trans. Inform. Theory, 52 (2006), pp. 489–509.

[11] E. J. Candes and T. Tao, *Decoding by linear programming*, IEEE Trans. Inform. Theory, 51 (2005), pp. 4203–4215.

[12] E. J. Candes and T. Tao, *Near-optimal signal recovery from random projections: Universal encoding strategies?*, IEEE Trans. Inform. Theory, 52 (2006), pp. 5406–5425.

[13] I. Ciril, J. Darbon, and Y. Tendero, *A simple and exact algorithm to solve l1 linear problems – Application to the compressive sensing method*, in Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2018) – Volume 4: VISAPP, Funchal, Madeira, Portugal, 2018, pp. 54–62.

[14] W. Dai and O. Milenkovic, *Subspace pursuit for compressive sensing signal reconstruction*, IEEE Trans. Inform. Theory, 55 (2009), pp. 2230–2249.

[15] M. Dimiccoli, *Fundamentals of cone regression*, Statist. Surv., 10 (2016), pp. 53–99, https://doi.org/10.1214/16-SS114.

[16] D. L. Donoho, *Compressed sensing*, IEEE Trans. Inform. Theory, 52 (2006), pp. 1289–1306.

[17] D. L. Donoho and M. Elad, *Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ1 minimization*, Proc. Natl. Acad. Sci. USA, 100 (2003), pp. 2197–2202.

[18] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, *Least angle regression*, Ann. Statist., 32 (2004), pp. 407–499, https://doi.org/10.1214/009053604000000067.

[19] S. Foucart, *Stability and robustness of weak orthogonal matching pursuits*, in Recent Advances in Harmonic Analysis and Applications, Springer Proc. Math. Stat. 25, Springer, New York, 2013, pp. 395–405.

[20] M. C. Grant and S. P. Boyd, *The CVX Users' Guide, Release* 2.1.

[21] E. T. Hale, W. Yin, and Y. Zhang, *A Fixed-Point Continuation Method for l1-Regularized Minimization with Applications to Compressed Sensing*, CAAM Technical Report TR07-07, Rice University, Houston, TX, 2007.

[22] J.-B. Hiriart-Urruty and C. Lemarechal, *Convex Analysis and Minimization Algorithms* I: *Fundamentals*, Grundlehren Math. Wiss., Springer, Berlin, Heidelberg, 1996.

[23] J.-B. Hiriart-Urruty and C. Lemarechal, *Convex Analysis and Minimization Algorithms* II: *Advanced Theory and Bundle Methods*, Grundlehren Math. Wiss., Springer, Berlin, Heidelberg, 1996.

[24] J. Huang, Y. Jiao, X. Lu, and L. Zhu, *Robust decoding from 1-bit compressive sampling with ordinary and regularized least squares*, SIAM J. Sci. Comput., 40 (2018), pp. A2062–A2086, https://doi.org/10.1137/17M1154102.

[25] P. Jain, A. Tewari, and I. S. Dhillon, *Orthogonal matching pursuit with replacement*, in Advances in Neural Information Processing Systems, 2011, pp. 1215–1223.

[26] J. Mairal, F. Bach, J. Ponce, G. Sapiro, R. Jenatton, and G. Obozinski, *SPAMS: A Sparse Modeling Software,* v2.3, http://spams-devel.gforge.inria.fr/downloads.html, 2014.

[27] J. Mairal and B. Yu, *Complexity analysis of the lasso regularization path*, in Proceedings of the 29th International Conference on International Conference on Machine Learning, Omnipress, 2012, pp. 1835–1842.

[28] A. Maleki, *Coherence analysis of iterative thresholding algorithms*, in Proceedings of the 47th Annual Allerton Conference on Communication, Control, and Computing, IEEE, 2009, pp. 236–243.

[29] A. Maleki and D. L. Donoho, *Optimally tuned iterative reconstruction algorithms for compressed sensing*, IEEE J. Sel. Topics Signal Process., 4 (2010), pp. 330–341.

[30] S. Mallat and Z. Zhang, *Matching pursuits with time-frequency dictionaries*, IEEE Trans. Signal Process., 41 (1993), pp. 3397–3415.

[31] M. C. Meyer, *A simple new algorithm for quadratic programming with applications in statistics*, Comm. Statist. Simulation Comput., 42 (2013), pp. 1126–1139.

[32] M. Moeller and X. Zhang, *Fast sparse reconstruction: Greedy inverse scale space flows*, Math. Comp., 85 (2016), pp. 179–208.

[33] D. Needell and J. A. Tropp, *CoSaMP: Iterative signal recovery from incomplete and inaccurate samples*, Appl. Comput. Harmon. Anal., 26 (2009), pp. 301–321.

[34] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, *An iterative regularization method for total variation-based image restoration*, Multiscale Model. Simul., 4 (2005), pp. 460–489, https://doi.org/10.1137/040605412.

[35] Y. C. Pati, R. Rezaiifar, and P. Krishnaprasad, *Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition*, in Proceedings of the Twenty-Seventh Asilomar Conference on Signals, Systems and Computers, IEEE, 1993, pp. 40–44.

[36] R. Rockafellar, M. Wets, and R. Wets, *Variational Analysis*, Grundlehren Math. Wiss., Springer, Berlin, Heidelberg, 2009.

[37] H. Schaeffer, Y. Yang, H. Zhao, and S. Osher, *Real-time adaptive video compression*, SIAM J. Sci. Comput., 37 (2015), pp. B980–B1001, https://doi.org/10.1137/130937792.

[38] J. F. Sturm, *Using SeDuMi* 1.02*, a MATLAB toolbox for optimization over symmetric cones*, Optim. Methods Softw. 11/12 (1999), pp. 625–653.

[39] J. A. Tropp and A. C. Gilbert, *Signal recovery from random measurements via orthogonal matching pursuit*, IEEE Trans. Inform. Theory, 53 (2007), pp. 4655–4666.

[40] E. Van Den Berg and M. P. Friedlander, *Probing the Pareto frontier for basis pursuit solutions*, SIAM J. Sci. Comput., 31 (2008), pp. 890–912, https://doi.org/10.1137/080714488.

[41] E. van den Berg and M. P. Friedlander, *SPGL*1*: A Solver for Large-Scale Sparse Reconstruction*, December 2019, https://friedlander.io/spgl1.

[42] Z. Wen, W. Yin, D. Goldfarb, and Y. Zhang, *A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization, and continuation*, SIAM J. Sci. Comput., 32 (2010), pp. 1832–1857, https://doi.org/10.1137/090747695.

[43] J. Yang and Y. Zhang, *Alternating direction algorithms for ℓ_1-problems in compressive sensing*, SIAM J. Sci. Comput., 33 (2011), pp. 250–278, https://doi.org/10.1137/090777761.

[44] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, *Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing*, SIAM J. Imaging Sci., 1 (2008), pp. 143–168, https://doi.org/10.1137/070703983.

[45] X. Zhang, M. Burger, and S. Osher, *A unified primal-dual algorithm framework based on Bregman iteration*, J. Sci. Comput., 46 (2011), pp. 20–46.