Computational Optimization and Applications
https://doi.org/10.1007/s10589-021-00270-x

A Laplacian approach to ℓ_1-norm minimization

Vincenzo Bonifaci[1]

Received: 9 June 2020 / Accepted: 24 February 2021

Abstract

We propose a novel differentiable reformulation of the linearly-constrained ℓ_1 minimization problem, also known as the basis pursuit problem. The reformulation is inspired by the Laplacian paradigm of network theory and leads to a new family of gradient-based methods for the solution of ℓ_1 minimization problems. We analyze the iteration complexity of a natural solution approach to the reformulation, based on a multiplicative weights update scheme, as well as the iteration complexity of an accelerated gradient scheme. The results can be seen as bounds on the complexity of iteratively reweighted least squares (IRLS) type methods of basis pursuit.

Keywords ℓ_1 regression · Basis pursuit · Iteratively reweighted least squares · Multiplicative weights · Laplacian paradigm · Convex optimization

1 Introduction

An important primitive in the areas of signal processing and statistics is that of finding a minimum ℓ_1-norm solution to an underdetermined system of linear equations. Specifically, for some $n \le m$, let $\hat{s} \in \mathbb{R}^m$ represent an unknown signal, $b \in \mathbb{R}^n$ a measurement vector, and $A \in \mathbb{R}^{n \times m}$ a full-rank matrix such that $A\hat{s} = b$. In some circumstances, the unknown signal $\hat{s}$ can be recovered by computing a minimum ℓ_1-norm solution to the system $As = b$; in other words, solving the following optimization problem:

$$\begin{aligned} &\text{minimize } \|s\|_1 \\ &\text{subject to } As = b, \quad s \in \mathbb{R}^m. \end{aligned} \tag{BP}$$

This ℓ_1-minimization problem is known as *basis pursuit* [20]. It is a central problem in the theory of sparse representation and arises in several applications, such

This work was carried out in association with IASI-Consiglio Nazionale delle Ricerche, Italy.

✉ Vincenzo Bonifaci
vincenzo.bonifaci@uniroma3.it

[1] Dipartimento di Matematica e Fisica, Università degli Studi Roma Tre, Rome, Italy

Published online: 19 March 2021

as compressed sensing [48], phase retrieval [30], imaging [42], and face recognition [47]. Through a standard reduction, it also captures the *ℓ_1-regression* problem used in statistical estimation and learning. Given its central role in the areas of sparse representation and statistics, the literature on the basis pursuit problem and ℓ_1-regression is extensive; see for example [18, 27, 40] and references therein. Several algorithms for basis pursuit are discussed in [27, Chapter 15], [17, 48]; for an experimental comparison, see [47].

The convex optimization problem (BP) can be cast as a linear program and thus could be solved via an interior-point method. Another popular approach to ℓ_1-minimization is the *iteratively reweighted least squares* (IRLS) method, which is based on iteratively solving a series of adaptively weighted ℓ_2-minimization problems. Various versions of IRLS schemes have been studied for a long time [31, 39]. IRLS methods are popular in practice, due to their simplicity, their experimental performance, and the fact that they do not require preprocessing nor special initialization rules [19]. Despite this, theoretical guarantees for IRLS methods in the literature are not common, particularly in terms of global convergence bounds (some examples are [7, 23, 43]). A recent IRLS algorithm stands out in the context of this paper, as it applies to the basis pursuit problem and comes with a worst-case guarantee: a $\tilde{O}(m^{1/3}\epsilon^{-8/3})$ iterations algorithm due to [21, Theorem 5.1], derived by further developing the approach of [22].

The present work contributes to developing the understanding and design of IRLS-type methods for basis pursuit. We propose a novel exact reformulation of (BP) as a differentiable convex problem over the positive orthant, which we call the *dissipation minimization* problem. A distinguishing feature of this approach is that it entails the solution of a single differentiable convex problem. The reformulation leads naturally to a new family of IRLS-type methods solving (BP).

We exemplify this approach by providing global convergence bounds for discrete IRLS-type algorithms for (BP). We explore two possible routes to the solution of the dissipation minimization problem, and thus of (BP), where we use the established framework of first-order optimization methods to derive two provably convergent iterative algorithms. We bound their iteration complexity as $O(m^2/\epsilon^3)$ and $O(m^2/\epsilon^2)$, respectively, where ϵ is the relative error parameter. These methods are in the IRLS family since each iteration can be reduced to the solution of a weighted least squares problem. Both methods are very simple to implement and the first one exhibits a geometric convergence rate in numerical experiments.

Our approach breaks the $\epsilon^{-8/3}$ bound for an IRLS method (at the cost of a worse dependency on m). We nevertheless emphasize that the goal of this work is not to establish the superiority of a specific algorithm, but rather to highlight a new approach that, already when coupled with off-the-shelf optimization methods, offers a principled way to derive IRLS-type algorithms with competitive theoretical performance. Subsequently to the first appearance of our results (on arXiv), an improved bound of $\tilde{O}(m^{1/3}\epsilon^{-2/3} + \epsilon^{-2})$ iterations for a more sophisticated IRLS-type algorithm for (BP) has been derived by [25] (again building on the ideas of [22] and [21]). While this algorithm has a rather more favorable worst-case dependency on the parameters, in practice it requires roughly $1/\epsilon$ iterations [25, Section 4]; in contrast, as we observe in Sect. 6, the experimental convergence rate of our approach is consistent with a *geometric* rate, that is, the iterations required appear to be linear in $\log(1/\epsilon)$, suggesting that a much stronger theoretical bound may hold in our setting.

Our dissipation-based reformulation of (BP) is new and may be of independent interest. It is rooted in the *Laplacian framework* of network theory [22]: it generalizes concepts such as the Laplacian matrix and the transfer matrix, which were originally developed to express the relation between electrical quantities across different terminals of a resistive network. (Many of our formulas have simple interpretations when the constraint matrix A is derived from a network matrix). In particular, the definition of the dissipation function is based on a generalization of the Laplacian potential of a network. This reinforces the idea from [21] that concepts originally developed for network optimization can be fruitful in the context of ℓ_1-regression.

To better motivate our algorithmic approach to the solution of the dissipation reformulation, in Sect. 4 we introduce dissipation-minimizing dynamics which are an application of the *mirror descent* (or *natural gradient*) dynamics [2, 3, 35] to our new objective function. Leveraging this intuition, in Sect. 5.1 we show how the algorithmic framework of [35] (see also [6]) can be applied to the dissipation minimization problem. The improved algorithm discussed in Section 5.2 is instead based on Nesterov's well-known *accelerated gradient method* [38].

The dynamics studied in Sections 4 and 5 bear some formal similarity to the so-called *Physarum dynamics*, studied in the context of natural computing, which are the network dynamics of a slime mold [10, 15, 43–45]. The fact that Physarum dynamics are of IRLS type was first observed in [43]. In this context, our result can be seen as the derivation of a Physarum-like dynamics purely from an optimization principle: dissipation minimization following the natural gradient. A relevant difference is that the specific dynamics we study is a gradient system, while the dynamics studied by [10, 43] is provably not a gradient system. This is precisely what enables us to apply the machinery of first-order convex optimization methods, and acceleration in particular.

We finally note that a different proof of Theorem 3.1 has been independently provided by [26] in the context of the Physarum dynamics.

Notation. For a vector $x \in \mathbb{R}^m$, we use $\text{diag}(x)$ to denote the $m \times m$ diagonal matrix with the coefficients of x along the diagonal. The inner product of two vectors $x, y \in \mathbb{R}^m$ is denoted by $\langle x, y \rangle = x^\top y$. The maximum (respectively, minimum) eigenvalue of a diagonalizable matrix M is denoted by $\lambda_{\max}(M)$ (respectively, $\lambda_{\min}(M)$). For a vector $x \in \mathbb{R}^m$, $\|x\|_p$ denotes the ℓ_p-norm of x ($1 \le p \le \infty$), and $|x|$ denotes the vector y such that $y_i = |x_i|$, $i = 1, \dots, m$. Similarly, x^2 denotes the vector y such that $y_i = x_i^2$, $i = 1, \dots, m$. With a slight overlap of notation, which should nevertheless not cause any confusion (Table 1), we instead reserve x^k with a symbolic index k to denote the vector produced by the kth step of an iterative algorithm.

Organization of the paper. The rest of the paper is organized as follows. In Sect. 2, we present the dissipation minimization reformulation of basis pursuit and some of its structural properties. In Sect. 3 we prove the equivalence between basis pursuit and dissipation minimization. In Sect. 4 we look at the continuous dynamics obtained by applying mirror descent to the dissipation minimization objective and connect them with existing literature. In Sect. 5, we analyze a discretization of these dynamics that yields an iterative IRLS-type method for the solution of the

Table 1 Worst-case iteration complexity of some IRLS methods for ℓ_1-norm minimization

Algorithm	Iteration complexity
Chin et al. [21]	$\tilde{O}(m^{1/3}\epsilon^{-8/3})$
PGS—Theorem 5.2	$\tilde{O}(m^2\epsilon^{-3})$
AGS—Theorem 5.4	$O(m^2\epsilon^{-2})$
Ene and Vladu [25]	$\tilde{O}(m^{1/3}\epsilon^{-2/3} + \epsilon^{-2})$

dissipation minimization problem and, hence, of basis pursuit; this method can be seen as an application of the well-known multiplicative weights update scheme, and its iteration complexity is $O(m^2/\epsilon^3)$. Then, by leveraging Nesterov's accelerated gradient scheme, we present and analyze an improved IRLS-type method with iteration complexity $O(m^2/\epsilon^2)$. In Sect. 6, implementations of the two methods are compared against existing solvers from the `l1benchmark` suite [47].

2 Basis pursuit and the dissipation minimization problem

2.1 Assumptions on the basis pursuit problem

We make the following assumptions on (BP):

(A.1) the matrix A has full rank and $n \leq m$;
(A.2) the system $As = b$ has at least one solution s' such that $s'_j \neq 0$ for each $j = 1, \dots, m$.

Proposition 2.1 *Any (BP) instance satisfying (A.1) can be transformed (in linear time) into an equivalent instance that satisfies both (A.1) and (A.2).*

Proof If a basis pursuit instance (A, b) satisfies (A.1) but not (A.2), form a new instance (A', b) where A' is obtained from A by duplicating every column. Observe the following about the two instances:

- A' has full rank and $n' = n \leq m \leq 2m = m'$ (hence it satisfies (A.1)).
- For any solution to (A, b), there is a solution to (A', b) with the same cost.
- Let $u = A^\top(AA^\top)^{-1}b$ be the least-square solution to $As = b$. There is at least one solution to $A's' = b$ with $s'_j \neq 0$ for each $j = 1, \dots, 2m$, given by

$$s'_{2j-1} = \begin{cases} u_j/2 & \text{if } u_j \neq 0, \\ +1 & \text{if } u_j = 0, \end{cases} \quad s'_{2j} = \begin{cases} u_j/2 & \text{if } u_j \neq 0, \\ -1 & \text{if } u_j = 0, \end{cases} \quad j = 1, \dots, m.$$

  Hence, (A', b) satisfies (A.2).
- No optimal solution to the instance (A', b) is such that $s'_{2j-1} \cdot s'_{2j} < 0$ for some j: if that was the case, one could form a solution of lesser cost by replacing each of

s'_{2j-1} and s'_{2j} with their average. Thus, any optimal solution s' to (A', b) can be transformed back into a solution s to (A, b) by taking $s_j = s'_{2j-1} + s'_{2j}$ for each $j = 1, \dots, m$. Such a solution satisfies $\|s\|_1 = \|s'\|_1$ and thus must be optimal for (A, b). Thus, the two instances have the same optimal value.

- Both the transformation of A into A' and the transformation of an optimal solution of (A', b) into an optimal solution of (A, b) can be carried out in time proportional to the size of the data (that is, in linear time).

□

Remark 2.1 A special case of (BP) is when A is derived from a network matrix. Specifically, consider a connected network with $n + 1$ nodes and m edges, and suppose edge j connects node u to node v. Define $b_j \in \mathbb{R}^m$ as $(b_j)_u = 1$, $(b_j)_v = -1$, and all other entries 0. The matrix $B = [b_1 \cdots b_m] \in \mathbb{R}^{(n+1)\times m}$ is called the *incidence matrix* of the network. For any connected network, the incidence matrix B has rank n and, additionally, any row of B can be expressed as a linear combination of the remaining n rows, because the sum of all rows is a zero vector. Let A be the submatrix of B obtained by deleting an arbitrary row. Then A satisfies assumption (A.1) and thus, without loss of generality, (A.2). A solution s to $As = b$ can be interpreted as an assignment of flow values to each edge such that the net in-flow at every node $v = 1, \dots, n$ matches the prescribed demand b_v.

2.2 The dissipation potential

In this section we introduce the *dissipation potential*, which is the function on which our reformulation of the basis pursuit problem is based.

Definition 2.1 Given $A \in \mathbb{R}^{n\times m}$, the *Laplacian-like matrix* relative to a vector $x \in \mathbb{R}^m_{\geq 0}$ is the matrix $L(x) \overset{\text{def}}{=} AXA^\top$, where $X = \text{diag}(x)$.

Remark 2.2 In the network setting described in Remark 2.1, a vector $x \in \mathbb{R}^m_{>0}$ can be interpreted as a set of weights, or *conductances*, on the edges of the network. Let B be the incidence matrix of the network as defined in Remark 2.1. Then the matrix $BXB^\top$ is the *weighted Laplacian* of the network [29]. The matrix $L(x) = AXA^\top$ is sometimes called the *reduced Laplacian*.

Proposition 2.2 *If $x > 0$, then $L(x)$ is positive definite.*

Proof Since A has full rank, so has $AX^{1/2}$; hence $L(x) = (AX^{1/2})(AX^{1/2})^\top$ is positive definite. □

The following function definition is central to our approach.

Definition 2.2 Let $A \in \mathbb{R}^{n\times m}$, $b \in \mathbb{R}^n$ be such that (A.1)–(A.2) hold. Define $f_0, f : \mathbb{R}^m \to (-\infty, +\infty]$ as

$$f_0(x) \overset{\text{def}}{=} \begin{cases} \mathbf{1}^\top x + b^\top L^{-1}(x) b, & \text{if } x \in \mathbb{R}^m_{>0} \\ +\infty & \text{if } x \notin \mathbb{R}^m_{>0}. \end{cases} \tag{2.1}$$

$$f(x) \overset{\text{def}}{=} \liminf_{x' \to x} f_0(x'), \qquad x \in \mathbb{R}^m. \tag{2.2}$$

We call f the *dissipation potential*. An equivalent definition of f is as the *convex closure* of f_0, which is the function whose epigraph in $\mathbb{R}^{m+1}$ is the closure of the epigraph of f_0 [41, Chapter 7]. The *effective domain* of f is the set

$$\operatorname{dom} f \overset{\text{def}}{=} \{x \in \mathbb{R}^m \ : \ f(x) < +\infty\}.$$

The functions f and f_0 differ only on the boundary of the positive orthant. We will show that f always achieves a minimum on $\mathbb{R}^m_{\geq 0}$, and hence on $\mathbb{R}^m$. One of our main results (Theorem 3.1) is that this minimum equals the minimum of (BP).

Remark 2.3 Consider again the case where the matrix A is derived from a network matrix, as in Remark 2.1. The node of the network corresponding to the row that was removed from the incidence matrix to form A is called the *grounded* node. Now assume that for some $u = 1, \dots, n$ the vector $b \in \mathbb{R}^n$ is such that $b_v = 0$ if $v \neq u$, $b_v = 1$ if $v = u$. Then the Laplacian potential $b^\top L^{-1}(x) b$ yields the *effective resistance* between the grounded node and node u when the conductances of the network are specified by the vector x. A standard result in network theory is that decreasing the conductance of any edge can only increase the effective resistance between any two nodes (see, for example, [28]). Thus, the minimization of the dissipation potential f involves an equilibrium between two opposing tendencies: decreasing any x_j decreases the linear term $\mathbf{1}^\top x$, but increases the Laplacian term $b^\top L^{-1}(x) b$.

2.3 Basic properties of the dissipation potential

We proceed to show that the dissipation potential attains a minimum. We start with some basic properties of f_0.

Lemma 2.1 *The function f_0 is positive, convex and differentiable on $\mathbb{R}^m_{>0}$.*

Proof Positivity follows from the positive-definiteness of $L^{-1}(x)$ for $x \in \mathbb{R}^m_{>0}$ (implied by Proposition 2.2). For convexity, it suffices to show that the mapping $x \mapsto b^\top L^{-1}(x) b$ is convex on $\mathbb{R}^m_{>0}$. First observe that $x \mapsto AXA^\top$ is a linear matrix-valued function, i.e., each one of the entries of $AXA^\top$ is a linear function of x, since multiplying X on the left and right with A and $A^\top$ yields linear combinations of the elements of x. Second, the matrix to scalar function $Y \mapsto b^\top Y^{-1} b$ is convex on the cone of positive definite matrices, for any $b \in \mathbb{R}^n$ (see for example [16, Section 3.1.7]). By combining the two facts above, it follows that the composition $x \mapsto b^\top (AXA^\top)^{-1} b$ is convex, and hence so is f_0. Finally, since the entries of $L(x)$

are linear functions of x, the function f_0 is a rational function with no poles in $\mathbb{R}^m_{>0}$, hence differentiable. □

To argue that f attains a minimum, we first recall some notions from convex analysis [8, 41]. An extended real-valued function $f : \mathbb{R}^m \to [-\infty, +\infty]$ is called *proper* if its domain is nonempty and the function never attains the value $-\infty$. It is called *closed* if its epigraph is closed. It is called *coercive* if it is proper and $\lim_{\|x\|\to\infty} f(x) = +\infty$.

Lemma 2.2 *The function f is nonnegative, coercive, proper, closed and convex on $\mathbb{R}^m$.*

Proof By Lemma 2.1, f_0 is convex on $\mathbb{R}^m$, since it is convex on its effective domain. Moreover f_0 is proper, since $L^{-1}(x)$ is positive definite and thus $0 < f_0(x) < +\infty$ for any $x \in \mathbb{R}^m_{>0}$. By construction, f coincides with the closure of f_0 and thus it is a closed proper convex function [41, Theorem 7.4]. Its nonnegativity follows from the positivity of f_0 and from (2.2). To show coerciveness, note that $\lim_{\|x\|\to+\infty} f(x) = +\infty$, because $b^\top(AXA^\top)^{-1}b \geq 0$ for any $x \in \mathrm{dom} f_0$, and $\mathbf{1}^\top x \to +\infty$ as $\|x\| \to +\infty$ with $x \in \mathrm{dom} f_0$. □

Corollary 2.1 *The function f attains a minimum on $\mathbb{R}^m_{\geq 0}$.*

Proof Under the hypotheses of Lemma 2.2, a function attains a minimal value over any nonempty closed set intersecting its domain [8, Theorem 2.14]; in particular, f attains its minimal value over $\mathbb{R}^m_{\geq 0}$. □

Since $f(x) = \liminf_{x'\to x} f_0(x')$, the minimum attained by f over $\mathbb{R}^m_{\geq 0}$ equals $\inf_{x>0} f_0(x)$. Note also that this minimum may be attained on the boundary of $\mathrm{dom} f$.

2.4 Gradient and Hessian

In this section we derive some formulas for the gradient and Hessian of f on the interior of its domain.

Definition 2.3 Let $x \in \mathbb{R}^m_{>0}$, $A \in \mathbb{R}^{n\times m}$, $b \in \mathbb{R}^n$, $L(x) = AXA^\top$. The *voltage vector* at x is $d(x) \overset{\mathrm{def}}{=} A^\top L^{-1}(x) b \in \mathbb{R}^m$.

Remark 2.4 In the network setting described in Remark 2.1, $d_j(x)$ expresses the *voltage along edge j* when an external current b_u enters each node $u = 1, \ldots, n$ (and a balancing current $-\sum_u b_u$ enters the grounded node).

The next lemma relates the gradient $\nabla f(x)$ to the voltage vector at x.

Lemma 2.3 *Let $x \in \mathbb{R}^m_{>0}$. For any $j = 1, \ldots, m$, $\frac{\partial f(x)}{\partial x_j} = 1 - (a_j^\top L^{-1}(x) b)^2 = 1 - d_j^2(x)$, where a_j stands for the jth column of A.*

Proof First observe that $L(x) = AXA^\top = \sum_{j=1}^m x_j a_j a_j^\top$ and thus $\partial L / \partial x_j = a_j a_j^\top$. We apply the following identity for the derivative of a matrix inverse [36, Section 8.4]:

$$\frac{\partial L^{-1}}{\partial x_j} = -L^{-1} \frac{\partial L}{\partial x_j} L^{-1}. \tag{2.3}$$

We obtain

$$\frac{\partial b^\top L^{-1} b}{\partial x_j} = -b^\top L^{-1} \frac{\partial L}{\partial x_j} L^{-1} b = -b^\top L^{-1} a_j a_j^\top L^{-1} b = -(a_j^\top L^{-1} b)^2.$$

The claim follows by the definition of *f*. □

To express the Hessian of *f*, in addition to the voltages we need the notion of transfer matrix.

Definition 2.4 Let $x \in \mathbb{R}^m_{>0}$, $A \in \mathbb{R}^{n \times m}$, $L(x) = AXA^\top$. The *transfer matrix* at x is $T(x) \overset{\text{def}}{=} A^\top L^{-1}(x) A$.

Remark 2.5 In the network setting described in Remark 2.1, the transfer matrix $T(x)$ expresses the relation between input currents and output voltages, when the conductances are given by the vector x. Namely, $T_{ij}(x)$ is the amount of voltage observed along edge i of the network when a unit external current is applied between the endpoints of edge j.

Corollary 2.2 *For any* $x > 0$, $\nabla^2 f(x) = 2\,(d(x)\, d(x)^\top) \odot T(x)$, *where* $\odot$ *denotes the Schur matrix product defined by* $(U \odot V)_{ij} = U_{ij} \cdot V_{ij}$.

Proof For any $i, j = 1, \ldots, m$, by Lemma 2.3 and applying once more (2.3), we get

$$\left[\nabla^2 f(x)\right]_{ij} = 2\Big(b^\top L^{-1} a_i a_i^\top L^{-1} a_j a_j^\top L^{-1} b\Big) = 2\, d_i(x) d_j(x) a_i^\top L^{-1} a_j.$$

The claim follows by Definition 2.4. □

2.5 Bounds on the norms of gradient and Hessian

In this section we derive some norm bounds for the gradient and Hessian of the dissipation potential *f*; they will be used crucially to derive complexity bounds for the algorithms studied in Sect. 5.

Two matrices M, M' are called *congruent* if there is a nonsingular matrix S such that $M' = SMS^\top$. For the proofs in this section, the main tool we rely on is the following algebraic fact relating the eigenvalues of congruent matrices; see for example [33, Theorem 4.5.9] for a proof.

Theorem 2.1 (Ostrowski) *Let $M, S \in \mathbb{R}^{m\times m}$ be two symmetric matrices, with S nonsingular. For $k = 1, \dots, m$, let $\lambda_k(M)$, $\lambda_k(SMS^\top)$ denote the k-th largest eigenvalue of M and $SMS^\top$, respectively. For each $k = 1, \dots, m$ there is a positive real number $\theta_k \in [\lambda_{\min}(SS^\top), \lambda_{\max}(SS^\top)]$ such that*

$$\lambda_k(SMS^\top) = \theta_k \lambda_k(M). \tag{2.4}$$

Lemma 2.4 *Let $x \in \mathbb{R}^m_{>0}$. Each nonzero eigenvalue of $T(x)$ belongs to $[(\max_i x_i)^{-1}, (\min_i x_i)^{-1}]$.*

Proof Consider the matrix $\Pi(x) \overset{\text{def}}{=} X^{1/2}T(x)X^{1/2}$. By Definition 2.4,

$$\Pi(x) = (AX^{1/2})^\top (AXA^\top)^{-1}(AX^{1/2}).$$

Hence, $\Pi(x)$ is the orthogonal projection matrix that projects onto the range of $(AX^{1/2})^\top$. In particular, $\Pi(x)^2 = \Pi(x)$ and each eigenvalue of $\Pi(x)$ equals 0 or 1. Since $T(x) = X^{-1/2}\Pi(x)X^{-1/2}$, the matrices $T(x)$ and $\Pi(x)$ are congruent. By Theorem 2.1, the algebraic multiplicity of the zero eigenvalue of $T(x)$ and $\Pi(x)$ is the same, and each positive eigenvalue of $T(x)$ must lie between the smallest and the largest eigenvalue of X^{-1}. These are $(\max_i x_i)^{-1}$ and $(\min_i x_i)^{-1}$, respectively. □

Lemma 2.5 *Let $x \in \mathbb{R}^m_{>0}$. Then $\|d(x)\|_\infty \le (\min_{i=1,\dots,m} x_i)^{-1} \cdot \|s\|_2$, where s is any solution to $As = b$. In particular, for $c_{A,b} \overset{\text{def}}{=} b^\top(AA^\top)^{-1}b$,*

$$\|d(x)\|_\infty \le (\min_{i=1,\dots,m} x_i)^{-1}\,(c_{A,b})^{1/2}. \tag{2.5}$$

Additionally, if s^* is an optimal solution to (BP),

$$c_{A,b}^{1/2} \le \|s^*\|_1 \le (m \cdot c_{A,b})^{1/2}. \tag{2.6}$$

Proof Note that $d(x) = A^\top L^{-1}(x)b = A^\top L^{-1}(x)As = T(x)s$. Hence

$$\|d(x)\|_\infty = \|T(x)s\|_\infty \le \|T(x)s\|_2. \tag{2.7}$$

Since the largest eigenvalue of $T(x)$ is at most $(\min_i x_i)^{-1}$ by Lemma 2.4, we can bound $\|T(x)s\|_2 \le (\min_i x_i)^{-1}\|s\|_2$, proving the first part of the claim. For the second part, consider the least square solution $u \overset{\text{def}}{=} A^\top(AA^\top)^{-1}b$. Then $\|u\|_2 = c_{A,b}^{1/2}$, and using the optimality of u for the ℓ_2 norm and of s^* for the ℓ_1 norm we derive

$$c_{A,b} = \|u\|_2^2 \le \|s^*\|_2^2 \le \|s^*\|_1^2 \le \|u\|_1^2 \le m\|u\|_2^2 = m \cdot c_{A,b}.$$

□

Corollary 2.3 *If $x \in \mathbb{R}^m_{>0}$, then*

$$\|\nabla f(x)\|_\infty \le 1 + (\min_{i=1,\dots,m} x_i)^{-2}\, c_{A,b}. \tag{2.8}$$

Proof Combine Lemma 2.5 with Lemma 2.3. □

Lemma 2.6 *If $x \in \mathbb{R}^m_{>0}$, then the largest eigenvalue of $\nabla^2 f(x)$ satisfies*

$$\lambda_{\max}(\nabla^2 f(x)) \le 2\,(\min_{i=1,\dots,m} x_i)^{-3} \cdot c_{A,b}. \tag{2.9}$$

Proof We can use the matrix identity $M \odot (zz^\top) = \mathrm{diag}(z) \cdot M \cdot \mathrm{diag}(z)$ to reexpress Corollary 2.2 as $\nabla^2 f(x) = 2D(x)T(x)D(x)$, where $D(x) \overset{\mathrm{def}}{=} \mathrm{diag}(d(x))$. Hence, by Theorem 2.1, the largest eigenvalue of $\nabla^2 f(x)$ satisfies

$$\lambda_{\max}\left(\nabla^2 f(x)\right) = 2\,\theta\,\lambda_{\max}(T(x)) \tag{2.10}$$

for some θ lying between the smallest and largest eigenvalues of $D(x)^2$. Since by Lemma 2.5

$$\theta \le \lambda_{\max}(D(x)^2) = \|d(x)\|^2_\infty \le (\min_i x_i)^{-2} c_{A,b}, \tag{2.11}$$

combining (2.10) and (2.11) with Lemma 2.4 we get

$$\lambda_{\max}(\nabla^2 f(x)) \le 2(\min_i x_i)^{-3} c_{A,b}.$$

□

3 Equivalence between basis pursuit and dissipation minimization

In this section we prove the equivalence between basis pursuit and dissipation minimization.

Theorem 3.1 *The value of the optimization problem*

$$\begin{aligned} &\text{minimize } \|s\|_1 \\ &\text{subject to } As = b, \quad s \in \mathbb{R}^m \end{aligned} \tag{BP}$$

is equal to the value of the optimization problem

$$\begin{aligned} &\text{minimize } \frac{1}{2}\mathbf{1}^\top x + \frac{1}{2} b^\top (AXA^\top)^{-1} b \\ &\text{subject to } x \in \mathbb{R}^m_{>0} \end{aligned} \tag{DM}$$

where $X = \mathrm{diag}(x)$.

We call (DM) the *dissipation minimization problem* associated to A and b. Note that the objective in (DM) is exactly $f_0(x)/2$, hence by (2.2) the minimum of (DM) equals the minimum of $f(x)/2$ over $\mathbb{R}^m_{\ge 0}$; the fact that this minimum is achieved is guaranteed by Corollary 2.1.

Definition 3.1 Let $x > 0$. The *solution induced by x* is the vector $q(x) \overset{\mathrm{def}}{=} XA^\top L^{-1}(x)b$.

The term "solution" is justified by the fact that $Aq(x) = LL^{-1}b = b$. Induced solutions have the following simple characterization.

Lemma 3.1 *Let $x \in \mathbb{R}^m_{>0}$. The solution induced by x, $q(x)$, equals the unique optimal solution to the quadratic optimization problem:*

$$\begin{aligned} &\text{minimize } s^\top X^{-1} s && (\mathrm{QP}_x)\\ &\text{subject to } As = b, \quad s \in \mathbb{R}^m. \end{aligned}$$

Proof This lemma is a straightforward generalization of Thomson's principle [12, Chapter 9] from electrical network theory. We adapt an existing proof [13, Lemma 3] to the notation used in this paper. Since the objective function in (QP_x) is strictly convex, the problem has a unique optimal solution. Consider any solution s, and let $r = s - q(x)$. Then $Ar = b - b = 0$ and hence

$$s^\top X^{-1} s = (q+r)^\top X^{-1}(q+r) = q^\top X^{-1} q + 2r^\top X^{-1} q + r^\top X^{-1} r \geq q^\top X^{-1} q,$$

since $r^\top X^{-1} r \geq 0$ and $r^\top X^{-1} q = r^\top A^\top L^{-1} b = (Ar)^\top L^{-1} b = 0$. Therefore, the objective function value of any solution s to (QP_x) is at least as large as the objective function value of the solution $q(x)$. □

The value of (QP_x) is, in fact, the Laplacian potential $b^\top L^{-1}(x) b$.

Corollary 3.1 *The minimum of (QP_x) equals $q(x)^\top X^{-1} q(x) = b^\top L^{-1}(x) b$.*

Proof We already proved that the minimum of (QP_x) is $q(x)^\top X^{-1} q(x)$. Substituting the definition of $q(x)$,

$$q^\top X^{-1} q = (b^\top L^{-1} A X) X^{-1} (X A^\top L^{-1} b) = b^\top L^{-1} L L^{-1} b = b^\top L^{-1} b.$$

□

Lemma 3.2 *For any $x > 0$, $q(x) \in \mathbb{R}^m$ is such that $Aq = b$ and $\|q(x)\|_1 \leq f(x)/2$. Thus, the value of (BP) is at most that of (DM).*

Proof For any $x \in \mathbb{R}^m_{>0}$, consider its induced solution $q(x) = XA^\top L(x)^{-1} b$. We already observed that $q(x)$ is feasible for (BP). Moreover, we can bound:

$$\begin{aligned} \|q(x)\|_1 &= x^\top X^{-1} |q| \\ &\leq (x^\top X^{-1} x)^{1/2} \cdot (q^\top X^{-1} q)^{1/2} \\ &= (\mathbf{1}^\top x)^{1/2} \cdot (b^\top L^{-1}(x) b)^{1/2} \qquad \text{(by Corollary 3.3)} \\ &\leq \frac{1}{2}\mathbf{1}^\top x + \frac{1}{2} b^\top L^{-1}(x) b \\ &= \frac{1}{2} f(x), \end{aligned}$$

where the first upper bound follows from the Cauchy-Schwarz inequality, and the second from the Arithmetic Mean-Geometric Mean inequality. □

To prove the converse of Lemma 3.2, we develop an intermediate lemma that relates the value of an optimal solution s^* of (BP) to the dissipation value of a vector x such that $x = |s|$ with s sufficiently close to s^*.

Lemma 3.3 *Let $s \in \mathbb{R}^m$, $\epsilon \in (0,1)$ be such that $As = b$, $s_j \neq 0$ and $(1-\epsilon)\left|s_j^*\right| \leq \left|s_j\right| \leq \left|s_j^*\right| + \epsilon/m$ for some s^* such that $As^* = b$ and each $j = 1, \dots, m$. Then for $x = |s|$,*

$$\frac{1}{2}f(x) \leq \frac{\epsilon}{2} + \frac{1}{2}\left(1 + \frac{1}{1-\epsilon}\right)\|s^*\|_1. \tag{3.1}$$

Proof On one hand, by the assumed upper bound $\left|s_j\right| \leq |s_j^*| + \epsilon/m$, trivially

$$\mathbf{1}^\top x = \|s\|_1 \leq \|s^*\|_1 + \epsilon. \tag{3.2}$$

On the other hand, consider the solution $q(x)$ induced by x and recall that $q(x)$ is feasible for (BP), since $Aq = b$, and optimal for (QP$_x$). By the assumed lower bound $\left|s_j\right| \geq (1-\epsilon)\left|s_j^*\right|$, and by Lemma 3.1,

$$\begin{aligned} b^\top L^{-1}(x) b &= q^\top X^{-1} q \\ &\leq s^{*\top} X^{-1} s^* = \sum_{j=1}^{m} \frac{1}{\left|s_j\right|}(s_j^*)^2 \\ &\leq (1-\epsilon)^{-1} \sum_j \left|s_j^*\right| = (1-\epsilon)^{-1}\|s^*\|_1, \end{aligned} \tag{3.3}$$

where the first upper bound follows from the fact that s^* is a feasible point of(QP$_x$) by assumption, and the second follows from the hypothesis. Combining (3.3) and (3.3), we get

$$\frac{1}{2}f(x) \leq \frac{1}{2}\|s^*\|_1 + \frac{\epsilon}{2} + \frac{1}{2}(1-\epsilon)^{-1}\|s^*\|_1.$$

□

Lemma 3.4 *The value of (DM) is at most that of (BP).*

Proof Consider an optimal solution $s^* \in \mathbb{R}^m$ to (BP). Let $s' \in \mathbb{R}^m$ be a solution to $As = b$ such that $s'_j \neq 0$ for all $j = 1, \dots, m$ (such an s' exists by assumption (A.2)). For any $\delta \in (0,1)$, let $s(\delta) \overset{\text{def}}{=} (1-\delta)s^* + \delta s'$ and $x(\delta) \overset{\text{def}}{=} |s(\delta)| > 0$. For any $\epsilon \in (0,1)$ we can ensure that the hypotheses of Lemma 3.3 are satisfied by choosing a small enough $\delta > 0$. For such a value of δ, Lemma 3.3 yields

$$\frac{1}{2}f(x(\delta)) \leq \frac{\epsilon}{2} + \frac{1}{2}\left(1 + \frac{1}{1-\epsilon}\right)\|s^*\|_1. \tag{3.4}$$

As ϵ can be chosen arbitrarily small, and the right-hand side of (3.4) approaches $\|s^*\|_1$ as $\epsilon \to 0$, we obtain the claim. □

This concludes the proof of Theorem 3.1. Not only are the optimal values of (BP) and (DM) the same, but one can bound the suboptimality of any feasible point of (BP) in terms of the dissipation value of a corresponding vector.

Theorem 3.2 *Let $s \in \mathbb{R}^m$ be a feasible point of (BP) such that $s_j \neq 0$ for all $j = 1, \ldots, m$, and let $x = |s|$, $\rho(x) \overset{\text{def}}{=} \|d(x)\|_\infty$. The quantity $\left(1 + \rho^{-1}(x)\right)\|s\|_1 - \rho^{-1}(x) \cdot f(x)$ is an upper bound on the suboptimality of s.*

Proof Consider the following linear formulation of (BP) (left) and its dual (right):

$$\begin{array}{ll}
\text{minimize } \mathbf{1}^\top x & \text{maximize } b^\top v \\
\text{subject to } x + s \geq 0 & \text{subject to } \lambda + \mu = \mathbf{1} \\
x - s \geq 0 & \qquad \lambda - \mu + A^\top v = 0 \\
As = b & \qquad \lambda, \mu \geq 0 \\
x, s \in \mathbb{R}^m. & \qquad \lambda, \mu \in \mathbb{R}^m, v \in \mathbb{R}^n.
\end{array}$$

Given any solution s to (BP) such that $x = |s| > 0$, let us take

$$\begin{aligned}
v &= \rho^{-1}(x)(AXA^\top)^{-1}b, \\
\lambda &= (\mathbf{1} - A^\top v)/2, \\
\mu &= (\mathbf{1} + A^\top v)/2.
\end{aligned}$$

Then $\|A^\top v\|_\infty \leq 1$ by definition of $\rho(x)$; moreover, $\lambda + \mu = \mathbf{1}$, $\lambda - \mu + A^\top v = 0$, and $\lambda, \mu \geq 0$. Thus, (x, s) is a primal feasible solution, (λ, μ, v) is a dual feasible solution, and by weak duality

$$\mathbf{1}^\top x \geq b^\top v = \rho^{-1}(x) b^\top (AXA^\top)^{-1} b.$$

This implies a duality gap of

$$\begin{aligned}
\mathbf{1}^\top x - \rho^{-1}(x) b^\top L^{-1}(x) b &= \mathbf{1}^\top x - \rho^{-1}(x)(f(x) - \mathbf{1}^\top x) \\
&= \left(1 + \rho^{-1}(x)\right)\|s\|_1 - 2\rho^{-1}(x) \cdot \frac{1}{2} f(x).
\end{aligned}$$

□

We close this section by observing that a simpler proof of Theorem 3.1 can be obtained by the following quadratic variational formulation of the ℓ_1-norm: for any $s \in \mathbb{R}^m$,

$$\|s\|_1 = \inf_{x \in \mathbb{R}^m_{>0}} \frac{1}{2} \sum_{j=1}^{m} \left(\frac{s_j^2}{x_j} + x_j \right),$$

see, for example, [4, Sect. 1.4.2]. Therefore

$$\begin{aligned}
\min_{\substack{s \in \mathbb{R}^m \\ As = b}} \|s\|_1 &= \min_{\substack{s \in \mathbb{R}^m \\ As = b}} \inf_{x \in \mathbb{R}^m_{>0}} \frac{1}{2} \left(\frac{s_j^2}{x_j} + x_j \right) \\
&= \inf_{x \in \mathbb{R}^m_{>0}} \left(\frac{1}{2} \left(\min_{\substack{s \in \mathbb{R}^m \\ As = b}} s^\top X^{-1} s \right) + \frac{1}{2} \mathbf{1}^\top x \right) \\
&= \inf_{x \in \mathbb{R}^m_{>0}} \left(\frac{1}{2}\, b^\top L^{-1}(x) b + \frac{1}{2}\, \mathbf{1}^\top x \right),
\end{aligned}$$

where the last identity follows from Corollary 3.1. However, the full strength of Lemma 3.2 and Lemma 3.4 is crucial to be able to constructively transform feasible points for (DM) into feasible points for (BP) and vice versa.

4 Continuous dynamics for dissipation minimization

Theorem 3.1 readily suggests an approach to the solution of the basis pursuit problem. Namely, the solution of the non-smooth, equality constrained formulation (BP) is reduced to the solution of the differentiable formulation (DM) on the positive orthant. In this section, we describe a continuous dynamics whose trajectories provably converge to the optimal solution of (DM); the resulting dynamical system inspires the discrete algorithms for (DM) that we rigorously analyze in Section 5. We also clarify the relation with a related but distinct dynamics that has already been studied in the literature.

Mirror descent dynamics. To solve (DM), it is natural to adopt methods for differentiable constrained optimization that are designed for simple constraints. Suppose we want to minimize a generic convex function f over the positive orthant. To this end, consider the following set of ordinary differential equations, aimed at solving $\inf \{f(x) \mid x > 0\}$:

$$\dot{x}_j = -x_j \frac{\partial f(x)}{\partial x_j}, \qquad j = 1, \ldots, m, \tag{4.1}$$

with initial condition $x(0) = x^0$ for some $x^0 > 0$. The intuition behind (4.1) is simple: to approach a global minimum, one should follow the (negative) gradient of f, but one should slow down the rate of change of the j-th component the smaller x_j is, in order not to violate the constraint $x_j > 0$.

When f is the dissipation potential, by Lemma 2.3 this yields the explicit dynamics

$$\dot{x}_j = x_j(d_j^2(x) - 1) = x_j((a_j^\top (AXA^\top)^{-1} b)^2 - 1), \qquad j = 1, \dots, m. \tag{4.2}$$

The dynamical system (4.1) is a nonlinear Lotka-Volterra type system of differential equations, of a kind that is common in population dynamics [32], where the rate of growth of a population is proportional to the size of the population. It is also an example of a *Hessian gradient flow* [1]: it can be expressed in the form

$$\dot{x} = -H^{-1}(x)\nabla f(x) \tag{4.3}$$

where $H(x) = \nabla^2 h(x)$ is the Hessian of a convex function h; namely, here $H(x) = X^{-1}$, and $h : \mathbb{R}^m_{>0} \to \mathbb{R}$ is the *negative entropy* function

$$h(x) \overset{\text{def}}{=} \sum_{j=1}^m x_j \ln x_j - \sum_{j=1}^m x_j. \tag{4.4}$$

System (4.3) can also be expressed as $\frac{d}{dt}\frac{\partial h(x)}{\partial x_j} = -\frac{\partial f(x)}{\partial x_j}, j = 1, \dots, m$, or more succinctly,

$$\frac{d}{dt}\nabla h(x) = -\nabla f(x), \tag{4.5}$$

which is known as the *mirror descent dynamics* or *natural gradient flow* [2]. The well-posedness of (4.3) has been considered, for example, in [1]. The mirror descent dynamics is well-studied and, in particular, convergence results are available, as we next recall. In fact, due to its generality, the mirror descent approach has been proposed as a very useful "meta-algorithm" for optimization and learning [3].

Convergence of the dynamics. The fact that the solution of the mirror descent dynamics (4.3) converges to a minimizer of a convex function f with rate $1/t$ is a well-known result; see, for example, [1, 46]. We include a streamlined proof for completeness, beginning with a straightforward lemma showing that f is monotonically nonincreasing along the trajectories of the dynamical system.

Lemma 4.1 *The values $f(x(t))$ with $x(t)$ given by (4.1) are nonincreasing in t.*

Proof We compute

$$\frac{d}{dt}f(x(t)) = \sum_{j=1}^m \frac{\partial f}{\partial x_j}(x)\frac{dx_j}{dt}(x) = -\sum_{j=1}^m x_j\left(\frac{\partial f}{\partial x_j}(x)\right)^2 \leq 0.$$

□

A key role in the analysis of the mirror descent dynamics is played by the *Bregman divergence* of the function h. This measures the difference between the true value of the function at a point x, and the approximate value at x predicted by a linear model of the function constructed at another point y.

Definition 4.1 The *Bregman divergence* of a convex function $h : \mathbb{R}^m \to (-\infty, +\infty]$ is defined by $D_h(x, y) \overset{\text{def}}{=} h(x) - h(y) - \langle \nabla h(y), x - y \rangle$.

Convexity of h implies the nonnegativity of $D_h(x, y)$. When h is the negative entropy, D_h is the *relative entropy* function (also known as Kullback-Leibler divergence), for which $D_h(x, y) = 0$ if and only if $x = y$.

Using the notion of Bregman divergence, one can prove that the trajectories of the mirror descent dynamics converge to a minimizer of the function.

Theorem 4.1 [1, 46] *Let $x^* \in \mathbb{R}^m_{\geq 0}$ be a minimizer of f. As $t \to \infty$, the values $f(x(t))$ with $x(t)$ given by (4.1) converge to $f(x^*)$.*

In particular,

$$f(x(t)) - f(x^*) \leq \frac{1}{t} D_h(x^*, x(0)) = O\left(\frac{1}{t}\right).$$

Proof In the following, to shorten notation we often write x in place of $x(t)$. Since $(d/dt)\nabla h(x) + \nabla f(x) = 0$ by (4.3), for any y we have $\langle (d/dt)\nabla h(x) + \nabla f(x), x - y \rangle = 0$. This is equivalent to

$$\langle \frac{d}{dt} \nabla h(x), x - y \rangle + \langle \nabla f(x), x - y \rangle = 0. \tag{4.6}$$

On the other hand, since $(d/dt)h(x) = \langle \nabla h(x), \dot{x} \rangle$, a simple calculation shows

$$\frac{d}{dt} D_h(y, x) = \langle \frac{d}{dt} \nabla h(x), x - y \rangle. \tag{4.7}$$

Combining (4.6) and (4.7), and plugging in $y = x^*$,

$$\frac{d}{dt} D_h(x^*, x) = -\langle \nabla f(x), x - x^* \rangle. \tag{4.8}$$

The proof is concluded by a potential function argument [5, 46]. Consider the function

$$\mathcal{E}(t) \overset{\text{def}}{=} D_h(x^*, x) + t(f(x) - f(x^*)).$$

Its time derivative is, by (4.8),

$$\frac{d}{dt} \mathcal{E}(t) = -\langle \nabla f(x), x - x^* \rangle + f(x) - f(x^*) + t \frac{d}{dt} f(x),$$

where the last summand is nonpositive by Lemma 4.1 and the other terms equal, by definition, $-D_f(x^*, x) \leq 0$. Hence, $\mathcal{E}(t) \leq \mathcal{E}(0)$ for all $t \geq 0$, which is equivalent to

$$D_h(x^*, x) + t(f(x) - f(x^*)) \leq D_h(x^*, x(0)),$$

proving the claim. □

Physarum dynamics. A previously studied dynamics that is formally similar to (4.2) is the *Physarum dynamics* [10, 15, 43–45], namely,

$$\dot{x}_j = x_j(\left|d_j(x)\right| - 1) = x_j(\left|a_j^\top(AXA^\top)^{-1}b\right| - 1), \qquad j = 1, \dots, m. \tag{4.9}$$

Differently from (4.2), the dynamics (4.9) is *not* a gradient flow, that is, there is no function f that allows to write the dynamics in the form (4.3) or (4.5) (with h the negative entropy). Still, from a qualitative point of view, the behavior of (4.9) appears to be rather similar to that of (4.2): namely, the trajectories still converge to an optimal solution of the associated (BP) problem (see, for example, [10, Theorem 2.9]).

5 Algorithms for dissipation minimization

We now turn to the problem of designing IRLS-type algorithms for (DM) (and thus (BP)) with provably bounded iteration complexity. Two technical obstacles in the setup of a first-order method for formulation (DM) are: 1) that the positive orthant is not a closed set, and 2) that the gradients of f may not be uniformly bounded on the positive orthant. There is a way to deal with both issues at once: instead of solving $\inf_{x>0} f(x)$, for an appropriately small $\delta > 0$ one can minimize f over

$$\Omega_\delta \overset{\text{def}}{=} \{x \in \mathbb{R}^m \ : \ \delta\mathbf{1} \leq x\}.$$

This is established by the next lemma.

Lemma 5.1 *Let x^* be a minimizer of f. Then $f(x^*) \leq \min_{x\in\Omega_\delta} f(x) \leq f(x^*) + \delta m$.*

Proof The first inequality is trivial. As for the second, recall that $f(x) = \mathbf{1}^\top x + b^\top L^{-1}(x)b$ for any $x > 0$, and that in the latter sum, the second term is non-increasing with x (by Lemma 2.3). Thus, for any $x > 0$,

$$f(x + \delta\mathbf{1}) = \mathbf{1}^\top(x + \delta\mathbf{1}) + b^\top L^{-1}(x + \delta\mathbf{1})b \leq \delta m + f(x).$$

In other words, for any $x > 0$, there is $y \geq \delta\mathbf{1}$ (namely, $y = x + \delta\mathbf{1}$) such that $f(y) \leq f(x) + \delta m$. □

In the following, we let $\delta \overset{\text{def}}{=} \epsilon\, c_{A,b}^{1/2}/(2m)$, where ϵ is the desired error factor and $c_{A,b}$ is as defined in Lemma 2.5; this, by Lemma 2.5 and Theorem 3.1, ensures that the additional error incurred by restricting solutions to Ω_δ is at most $(\epsilon/2)\|s^*\|_1 = (\epsilon/4)f(x^*)$.

5.1 Primal gradient scheme

Guided by (4.5), we might consider its forward Euler discretization

$$\nabla h(x^{k+1}) - \nabla h(x^k) = -\eta \nabla f(x^k), \tag{5.1}$$

where $x^k \in \Omega_\delta$ denotes the kth iterate, and $\eta \in \mathbb{R}_{>0}$ an appropriate step size. Indeed, the update (5.1) falls within a well-studied methodology for first-order convex optimization [9, 35]. We adapt this framework to the solution of (DM).

The *primal gradient scheme* is a first-order method for minimizing a differentiable convex function f over a closed convex set Q. This scheme, which is defined with respect to a reference function h, proceeds as follows [6, 35]:

1. Initialize $x^0 \in Q$. Let $\beta > 0$ be a parameter.
2. At iteration $k = 0, 1, \ldots$, compute $\nabla f(x^k)$ and set

$$x^{k+1} \leftarrow \operatorname{argmin}_{x \in Q} \{ \langle \nabla f(x^k), x - x^k \rangle + \beta D_h(x, x^k) \}. \tag{5.2}$$

We apply the scheme with h as defined in (4.4) and with $Q = \Omega_\delta$. Then, the minimization in (5.2) can be carried out analytically; it reduces to

$$x_j^{k+1} = \max\{\delta, x_j^k \cdot \exp(-\beta^{-1}[\nabla f(x^k)]_j)\}, \qquad j = 1, \ldots, m. \tag{5.3}$$

Update (5.3) is straightforward to implement as long as one can compute $\nabla f(x^k)$. This computation is discussed in Section 5.3.

Convergence of the primal gradient scheme. As shown in [35], the primal gradient scheme achieves an absolute error bounded by $O(\beta/k)$ after k iterations provided that the function f is *β-smooth relative to h*. In our case, where both f and h are twice-differentiable on Q, relative β-smoothness is defined as

$$\lambda_{\max}(\nabla^2 f(x)) \le \beta \cdot \lambda_{\max}(\nabla^2 h(x)) \qquad \text{for all } x \in Q. \tag{5.4}$$

Theorem 5.1 [35] *If f is β-smooth relative to h, then for all $k \ge 1$, the updates* (5.2) *satisfy*

$$f(x^k) - f(x^*|_Q) \le \frac{\beta}{k} D_h(x^*|_Q, x^0).$$

where $x^*|_Q \overset{\text{def}}{=} \operatorname{argmin}_{x \in Q} f(x)$.

To apply Theorem 5.1 in our setting, we need to bound the smoothness parameter β. We do this by leveraging the bounds derived in Section 2.5.

Lemma 5.2 *Equation* (5.4) *holds for* $\beta = 8m^2/\epsilon^2$.

Proof Condition (5.4) is equivalent to the condition that the largest eigenvalue of the matrix

$$\left[\nabla^2 h(x)\right]^{-1}\nabla^2 f(x) = X\nabla^2 f(x)$$

be at most β (see [33, Theorem 7.7.3]). The matrix $X\nabla^2 f(x)$ is similar to $X^{1/2}\nabla^2 f(x)X^{1/2}$, hence it suffices to bound the eigenvalues of the latter. Since $\nabla^2 f(x) = 2D(x)T(x)D(x)$ with $D(x) = \mathrm{diag}(d(x))$,

$$X^{1/2}\nabla^2 f(x)X^{1/2} = 2X^{1/2}DTDX^{1/2} = 2DX^{1/2}TX^{1/2}D = 2D\Pi D,$$

where we used the fact that X and $D(x)$ are diagonal. By the proof of Lemma 2.4, the eigenvalues of $\Pi(x)$ are all 0 or 1. Hence, using again the relation between the eigenvalues of congruent matrices (Theorem 2.1), we conclude that the largest eigenvalue of $X^{1/2}\nabla^2 f(x)X^{1/2}$ is bounded by that of $2D(x)^2$. Since $D(x) = \mathrm{diag}(d(x))$, the latter equals $2\|d(x)\|_\infty^2$, which is $2c_{A,b}/\delta^2 = 8m^2/\epsilon^2$ by Lemma 2.5 and the definitions of Ω_δ and δ. □

Theorem 5.2 *The primal gradient scheme* (5.3) *applied to the dissipation minimization problem (DM) achieves relative error at most* ϵ *after* $96m^2\log(m/\epsilon)/\epsilon^3 = \tilde{O}(m^2/\epsilon^3)$ *iterations.*

Proof By Theorem 5.1 and Lemma 5.2, after k iterations it holds that

$$f(x^k) - f(x^*|_Q) \le 8Rm^2/(k\epsilon^2), \tag{5.5}$$

where $R \overset{\text{def}}{=} D_h(x^*|_Q, x^0)$. Since $f(x^*|_Q) \le (1+\epsilon/4)f(x^*)$ (by Lemma 5.1, since $Q = \Omega_\delta$), this implies

$$f(x^k) - f(x^*) \le 8Rm^2/(k\epsilon^2) + \frac{\epsilon}{4}f(x^*). \tag{5.6}$$

Thus, $f(x^k) - f(x^*) \le \epsilon f(x^*)$ if we take $k = \lceil 32Rm^2/(3\epsilon^3 f(x^*))\rceil$. We complete the proof by bounding $R/f(x^*)$ in terms of $\log(m/\epsilon)$. Let

$$\mu \overset{\text{def}}{=} \max_{j=1,\dots,m} \frac{x_j^*|_Q}{x_j^0}$$

Observe that since $x^0 \in Q$,

$$\mu \le \frac{1}{\delta}\max_j x_j^*|_Q \le \frac{2m}{\epsilon c_{A,b}^{1/2}} f(x^*|_Q) \le \frac{2m^{3/2}}{\epsilon(mc_{A,b})^{1/2}}(1+\epsilon/4)f(x^*) \le \frac{8m^{3/2}}{\epsilon f(x^*)}f(x^*)$$

with the last inequality following from (2.6). Thus,

$$\begin{aligned} R &= \sum_{j=1}^m x_j^*|_Q \log\frac{x_j^*|_Q}{x_j^0} \le (\log\mu)f(x^*|_Q) \le (\log\mu)(1+\epsilon/4)f(x^*) \\ &\le 2\log\left(\frac{8m^{3/2}}{\epsilon}\right)f(x^*) \le 9\log\left(\frac{m}{\epsilon}\right)f(x^*). \end{aligned}$$

Hence, $k = \lceil 96m^2 \log(m/\epsilon)/\epsilon^3 \rceil$ iterations suffice to achieve relative error ϵ. □

5.2 Accelerated gradient scheme

The second optimization scheme that we consider is the *accelerated gradient method* of [38]. This can be summarized as follows:

1. Initialize $x^0 \in Q$. Let $\beta > 0$ be a parameter.
2. At iteration $k = 0, 1, \ldots$, compute $\nabla f(x^k)$ and set $\alpha_k = 1/2(k+1)$, $\tau_k = 2/(k+3)$ and

$$y^k \leftarrow \operatorname{argmin}_{x \in Q} \left\{ \frac{\beta}{2} \|x - x^k\|_2^2 + \langle \nabla f(x^k), x - x^k \rangle \right\} \tag{5.7}$$

$$z^k \leftarrow \operatorname{argmin}_{x \in Q} \left\{ \frac{\beta}{2} \|x - x^0\|_2^2 + \sum_{i=0}^{k} \alpha_i \langle \nabla f(x^i), x - x^i \rangle \right\} \tag{5.8}$$

$$x^{k+1} \leftarrow \tau_k z^k + (1 - \tau_k) y^k. \tag{5.9}$$

In our application of the scheme, $Q = \Omega_\delta$ and the minimization in (5.7) and (5.8) can be carried out analytically; explicitly, they become

$$y_j^k = \max\{\delta, x_j^k - \beta^{-1}[\nabla f(x^k)]_j\}, \quad j = 1, \ldots, m \tag{5.10}$$

$$z_j^k = \max\{\delta, x_j^0 - \beta^{-1}[\sum_{i=0}^{k} \alpha_i \nabla f(x^i)]_j\}, \quad j = 1, \ldots, m. \tag{5.11}$$

To implement (5.10)–(5.11), it is enough to be able to access the gradient $\nabla f(x^k)$ and the cumulative gradient $\sum_i \alpha_i \nabla f(x^i)$; the latter can be maintained with one additional update at each iteration.

Convergence of the accelerated gradient scheme. The well-known result by [38] shows that the accelerated gradient scheme achieves an absolute error bounded by $O(\beta/k^2)$ after k iterations provided that the gradient of the function f is *β-Lipschitz-continuous* over Q. In our case, where f is twice-differentiable on Q, this means

$$\lambda_{\max}(\nabla^2 f(x)) \le \beta \qquad \text{for all } x \in Q. \tag{5.12}$$

Theorem 5.3 [38] *If ∇f is β-Lipschitz-continuous over Q, then for all $k \ge 1$, the updates* (5.7)–(5.9) *satisfy*

$$f(y^k) - f(x^*|_Q) \le \frac{2\beta}{(k+1)^2} \|x^*|_Q - x^0\|_2^2$$

where $x^*|_Q \overset{\text{def}}{=} \operatorname{argmin}_{x\in Q} f(x)$.

Again, to apply Theorem 5.3 in our setting, we need to bound the smoothness parameter β. We do this by exploiting Lemma 2.6.

Lemma 5.3 *Equation* (5.12) *holds for* $\beta = 16m^3/(\epsilon^3 c_{A,b}^{1/2})$.

Proof Immediate from Lemma 2.6, the fact that $Q = \Omega_\delta$ and the definition of Ω_δ. Recall that $\delta = \epsilon c_{A,b}^{1/2}/(2m)$. □

Theorem 5.4 *If* $x^0 = |u|$ *where* $u \overset{\text{def}}{=} A^\top(AA^\top)^{-1}b$ *is the least square solution to* $As = b$, *the accelerated gradient scheme* (5.7)–(5.9) *applied to the dissipation minimization problem (DM) achieves relative error at most* ϵ *after* $24m^2/\epsilon^2$ *iterations.*

Proof By Theorem 5.3 and Lemma 5.3, after k iterations it holds that

$$f(y^k) - f(x^*|_Q) \leq 32Rm^3/(k^2\epsilon^3 c_{A,b}^{1/2}). \tag{5.13}$$

Since $f(x^*|_Q) \leq (1+\epsilon/4)f(x^*)$ by Lemma 5.1, this implies

$$f(y^k) - f(x^*) \leq (\epsilon/4)f(x^*) + 32Rm^3/(k^2\epsilon^3 c_{A,b}^{1/2}). \tag{5.14}$$

Thus, $f(y^k) - f(x^*) \leq (\epsilon/4)f(x^*) + (\epsilon/2)c_{A,b}^{1/2} < \epsilon f(x^*)$ if the number of iterations k is at least

$$\frac{8m^{3/2}}{\epsilon^2}\left(\frac{R}{c_{A,b}}\right)^{1/2}. \tag{5.15}$$

We complete the proof by bounding $(R/c_{A,b})^{1/2} = \|x^*|_Q - x^0\|_2/c_{A,b}^{1/2}$ in terms of m. Observe that $R^{1/2} = \|x^*|_Q - x^0\|_2 \leq \|x^*|_Q\|_2 + \|x^0\|_2$. By the assumption that $x^0 = |u|$ where u is the least square solution to $As = b$, $\|x^0\|_2 = \|u\|_2 = c_{A,b}^{1/2}$ (recall the definition of $c_{A,b}$ in Lemma 2.5). Moreover,

$$\begin{aligned}\|x^*|_Q\|_2 \leq \|x^*|_Q\|_1 &\leq \frac{1}{2}f(x^*|_Q) \leq \frac{1}{2}f(x^*) + \frac{\epsilon}{2}c_{A,b}^{1/2}\\ &\leq (m\, c_{A,b})^{1/2} + c_{A,b}^{1/2} < 2m^{1/2}c_{A,b}^{1/2}.\end{aligned}$$

Hence $(R/c_{A,b})^{1/2} \leq 3m^{1/2}$ and substitution in (5.15) yields the theorem. □

5.3 Implementing the iterations

We conclude this section by commenting on a few implementations details and in particular on how each iteration of (5.3) and (5.7)–(5.9) could be implemented. A notable point is that each iteration can be reduced to a series of operations that access the matrix A only through the solution of a system of the form

$AWA^\top p = b$, for some diagonal matrix W, or through matrix-vector multiplications of the form Av or $A^\top v$.

Main computational steps. The major computational steps required to implement an iteration of (5.3) and (5.7)–(5.9) are:

1. Computation of the Laplacian-like matrix $L(x) = AXA^\top$: since $X = \mathrm{diag}(x)$ is diagonal, $L(x)$ can be computed via the matrix-matrix product $(AX)A^\top$.
2. Solution of the symmetric linear system $L(x)p = b$ for p, i.e. 1 linear system solve; note that since $L(x) = AXA^\top$, the system $L(x)p = b$ is a symmetric linear system with a positive definite constraint matrix.
3. Computation of the gradient: by Lemma 2.3, computing the vector $d(x) = A^\top L^{-1}(x)b$ is enough to compute the gradient at x, since $\nabla f(x) = \mathbf{1} - d^2(x)$. To compute $d(x)$, it is enough to premultiply the solution p of the linear system $L(x)p = b$ with $A^\top$.

Hence, the computational cost is of 1 matrix-matrix product, 1 linear system solve, and 1 matrix-vector product. The remaining operations involve single-vector operations, and thus are of lower order cost.

Warm start. Heuristically, the solution of the system $L(x^{k+1})p = b$, which is required to compute the gradient at iteration $k + 1$, can be expected to be close to that of the system $L(x^k)p = b$ when x^{k+1} is close to x^k. Hence, one possibility in practice is to use the solution obtained at step k to warm-start the linear equation solver at step $k + 1$, with a possible substantial reduction in the computational cost of each iteration.

Initial point and exit criterion. We assumed the starting point is the least square solution in Theorem 5.4, but this was only to optimize the worst-case iteration bound. In fact, Theorem 5.2 and Eq. (5.15) always apply and the schemes we discussed do not require a special initialization apart from membership into Ω_δ; hence, any point that is not too close to the boundary of the positive orthant is a suitable starting point. We can stop the schemes after the number of iterations k is large enough to ensure the error guarantees of Theorems 5.2 and 5.4 (or Eq. (5.15)), or when the condition number of the linear system $L(x)p = b$ becomes too large. Alternatively, a natural exit criterion in practice can be based on the duality gap provided by Theorem 3.2.

Obtaining feasible iterates for (BP). The algorithms as described above produce iterates in the positive orthant, that is, iterates that are feasible for (DM), but after all, our goal was to obtain feasible iterates of (BP). By using the ideas of Lemma 3.2, we can easily associate with any iterate $x^k \in \mathbb{R}^m_{>0}$ an iterate s^k that is feasible for (BP), and the cost of which is not larger than the dissipation cost of x^k: namely, take $s^k = q(x^k) = X^k A^\top L(x^k)^{-1} b$. By the proof of Lemma 3.2, we know that $\|s^k\|_1 \le f(x^k)/2$. Thus, the error bounds for $f(x^k)$ can be directly translated into error bounds for $\|s^k\|_1$. Note that s^k can be computed essentially for free, since $s^k = X^k d(x^k)$ and $d(x^k)$ is a byproduct of the gradient computation at iteration k.

6 Numerical comparison with other algorithms for ℓ_1-minimization

We include in this section a numerical comparison of our schemes to other well-known algorithms for ℓ_1-minimization. The results suggest that both the primal scheme and a slightly revised accelerated scheme may converge at a geometric rate, that is, much faster than what our theoretical analysis guarantees. This suggests the open problem of improving the quality of our error bounds.

To compare our approaches to other algorithms for ℓ_1-minimization, we implemented them in MATLAB [14][1] and ran the `l1benchmark` suite by [47], which includes implementations of several other ℓ_1-minimization solvers.

In the benchmark by Yang et al., problem instances are generated as follows according to three parameters m (the number of variables), n (the number of observations), and k (the number of nonzero entries in a reference sparse signal):

1. the matrix $A \in \mathbb{R}^{n\times m}$ is generated with independent standard Gaussian entries; then, each column of A is normalized to have unit ℓ_2-norm;
2. a reference sparse signal $\hat{s} \in \mathbb{R}^m$ is generated with k of its m components, randomly selected, being nonzero entries uniformly distributed in $[-10, 10]$, and all remaining entries being zero;
3. the vector $b \in \mathbb{R}^n$ is generated as $b = A\hat{s}$.

With probability 1 over the randomness of A, such reference $\hat{s}$ is the sparsest solution to $As = b$, as long as $k < n/2$ [24, Lemma 2.1]. Moreover, there exists $\rho > 0$ such that when $k \le \rho n$, $\hat{s}$ is also guaranteed (with overwhelming probability as n grows) to be the unique optimal solution of (BP) [24, Theorem 2.4]. Thus, in the benchmark the iterates produced by the algorithms are compared against the signal $\hat{s}$.

A representative comparison is shown in Fig. 1.

The figure plots the relative error[2] of the algorithms as a function of computation time, averaged on 20 randomly generated instances (with $m = 1000$, $n = 800$, and $k = n/4$ nonzeros in the reference solution $\hat{s}$). The implementations based on our approaches are:

- the Primal Gradient Scheme of Sect. 5.1 (PGS, with $\beta = 4$, $\delta = 10^{-15}$),
- the Accelerated Gradient Scheme of Sect. 5.2 (AGS, with $\beta = 3.5$, $\delta = 10^{-15}$, $\tau_k = 2/(k+3)$),
- a revised Accelerated Gradient Scheme, which we formulate below (AGS2, with $\beta = 1.1$, $\delta = 10^{-15}$, $\tau_k = 10^{-15}$).

Other algorithms measured in the experiment are the Homotopy method, the primal and dual augmented Lagrangian methods (PALM, DALM), the primal-dual interior point method (PDIPA), and the approximate message passing method (AMP). We refer the reader to [47] for definition and discussion of these other methods.

[1] Experiments used MATLAB R2020b on a PC with Intel i5-8600K CPU at 3.6 GHz and 16 Gb RAM.

[2] The relative error of a point s is defined as $(\|s\|_1 - \|\hat{s}\|_1)/\|\hat{s}\|_1$.

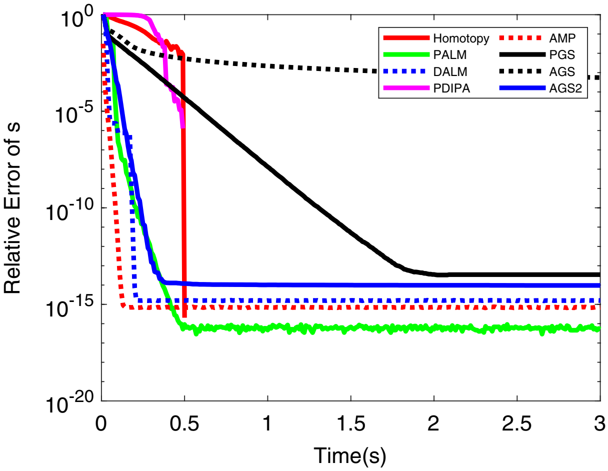

Fig. 1 Results of the `l1benchmark` benchmark. Parameters: $m = 1000$, $n = 800$, $k = 200$

Table 2 Results for $m = 1000$, $n = 400$, with a timeout of 3 seconds, averaged on 20 instances

Method	Average rel. error	Average rel. distance	Average time (s)	Average iterations
Homotopy	8.00e−17	8.23e−16	0.127	242
PALM	*3.85e−17*	*1.92e−16*	3	9296
DALM	8.66e−16	5.09e−16	3	32121
PDIPA	1.59e−06	4.16e−06	0.268	25
AMP	1.76e−03	8.23e−03	3	61739
PGS	4.37e−15	1.41e−14	1.81	454
AGS	2.40e−02	6.31e−02	3	757
AGS2	3.96e−15	4.78e−15	3	793

The lowest relative error and relative distance are given in italics

A full set of experiments with different values of m and n is reported in Tables 2, 3, 4, 5, 6 and 7. Note that for a fixed number of variables m, instances get harder as n decreases, as this corresponds to a reconstruction problem with a smaller number of observations. As we varied both m and n, we held the sparsity parameter k fixed, to $n/4$. The tables report the relative error, the time and the number of iterations of each algorithm, averaged on 20 randomly generated instances. They also report the relative distance of the algorithm's final point s, which is defined as $\|s - \hat{s}\|_2/\|\hat{s}\|_2$ where $\hat{s}$ is the reference signal. We have emphasized in italics the lowest relative error and relative distance in each table.

We remark that the computation time can be shorter than the timeout for methods that have additional stopping criteria, which are often intrinsic to the method. In particular, for PDIPA, PGS, AGS and AGS2, the method is terminated early if the condition number of the linear system solved at each iteration rises above a certain

Table 3 Results for $m = 1000$, $n = 800$, with a timeout of 3 seconds, averaged on 20 instances

Method	Average rel. error	Average rel. distance	Average time (s)	Average iterations
Homotopy	1.01e−16	8.59e−16	0.369	315
PALM	*5.03e−17*	*2.31e−16*	3	4955
DALM	1.59e−15	1.10e−15	3	6362
PDIPA	1.68e−06	1.96e−06	0.44	18
AMP	7.32e−16	8.21e−16	3	32628
PGS	3.01e−14	3.60e−14	2.33	180
AGS	5.49e−04	6.63e−04	3	236
AGS2	9.53e−15	8.50e−15	3	235

The lowest relative error and relative distance are given in italics

Table 4 Results for $m = 1500$, $n = 600$, with a timeout of 5 seconds, averaged on 20 instances

Method	Average rel. error	Average rel. distance	Average time (s)	Average iterations
Homotopy	1.01e−16	9.81e−16	0.449	363
PALM	*2.36e−17*	*2.17e−16*	5	4720
DALM	8.91e−16	6.13e−16	5	11599
PDIPA	1.19e−06	3.16e−06	0.702	27
AMP	5.22e−16	9.56e−16	5	41036
PGS	4.24e−15	1.02e−14	3.92	433
AGS	2.47e−02	6.63e−02	5	551
AGS2	2.99e−15	3.89e−15	5	547

The lowest relative error and relative distance are given in italics

Table 5 Results for $m = 1500$, $n = 1200$, with a timeout of 5 seconds, averaged on 20 instances

Method	Average rel. error	Average rel. distance	Average time (s)	Average iterations
Homotopy	*5.40e−17*	9.59e−16	1.44	482
PALM	5.43e−17	*2.63e−16*	5	3020
DALM	1.80e−15	1.29e−15	5	3739
PDIPA	2.29e−06	2.70e−06	1.36	19
AMP	7.52e−16	9.30e−16	5	7832
PGS	4.32e−12	5.70e−12	5	127
AGS	1.21e−03	1.44e−03	5	128
AGS2	8.81e−−15	7.66e−15	5	127

The lowest relative error and relative distance are given in italics

Table 6 Results for $m = 2000$, $n = 800$, with a timeout of 7 seconds, averaged on 20 instances

Method	Average	Average	Average	Average
	rel. error	rel. distance	time (s)	iterations
Homotopy	6.92e−17	8.50e−16	1.02	474
PALM	*4.56e−17*	*2.32e−16*	7	3199
DALM	8.63e−16	6.88e−16	7	6959
PDIPA	2.04e−06	5.28e−06	1.53	27
AMP	3.48e−03	1.16e−02	7	12824
PGS	5.83e−15	1.45e−14	6.41	335
AGS	2.47e−02	6.65e−02	7	371
AGS2	3.39e−15	4.56e−15	7	368

The lowest relative error and relative distance are given in italics

Table 7 Results for $m = 2000$, $n = 1600$, with a timeout of 12 seconds, averaged on 20 instances

Method	Average	Average	Average	Average
	rel. error	rel. distance	time (s)	iterations
Homotopy	7.91e−17	1.44e−15	3.47	649
PALM	*2.85e−17*	*3.01e−16*	12	3979
DALM	1.97e−15	1.47e−15	12	4837
PDIPA	3.22e−06	3.78e−06	2.87	19
AMP	7.75e−16	1.05e−15	12	7840
PGS	3.82e−14	4.69e−14	11.7	142
AGS	9.97e−04	1.18e−03	12	145
AGS2	9.34e−15	8.30e−15	12	145

The lowest relative error and relative distance are given in italics

threshold (10^{14} for PDIPA, 10^{24} for PGS/AGS/AGS2). For the Homotopy method, early termination will occurr if the regularization coefficient drops below 10^{-6} (see [47] for an empirical justification).

In the experiments, PGS exhibited a geometric convergence rate, which is much better than what Theorem 5.2 guarantees, strongly suggesting that an improved theoretical analysis may be possible. Over time, PGS essentially reaches the machine precision barrier ($\approx 10^{-15}$), in contrast with other methods, notably the interior point method (PDIPA).

AGS, on the other hand, appears to be rather inaccurate in practice and does not exhibit a substantially better behavior than what is guaranteed by Theorem 5.4. This suggests that the entropic form of the updates – used in PGS but not in AGS – might have a high impact in practice. Therefore, we also benchmark a revised algorithm (AGS2) obtained by adopting an entropic form of the AGS updates (5.10)–(5.11), as follows (colored terms are new):

$$y_j^k = \max\left\{\delta, x_j^k - x_j^k \cdot \beta^{-1}\left[\nabla f(x^k)\right]_j\right\} \tag{6.1}$$

$$z_j^k = \max\left\{\delta, x_j^0 - x_j^0 \cdot \beta^{-1}\left[\sum_i \alpha_i \nabla f(x^i)\right]_j\right\} \tag{6.2}$$

$$x^{k+1} = \tau_k z^k + (1 - \tau_k) y^k. \tag{6.3}$$

The resulting scheme AGS2 is seen in Fig. 1 to exhibit a geometric convergence rate and to be competitive against some of the best results in the benchmark, such as those of the primal augmented Lagrangian method (PALM).

7 Conclusions

We proposed a novel exact reformulation of the basis pursuit problem, which leads to a new family of gradient-based, IRLS-type methods for its solution. We then analyzed the iteration complexity of a natural optimization approach to the reformulation, based on the mirror descent scheme, as well as the iteration complexity of an accelerated gradient method. The first scheme can be seen as the discretization of a Hessian gradient flow and also as a variant on the Physarum dynamics, derived purely from optimization principles. The accelerated method, on the other hand, improves the error dependency for IRLS-type methods for basis pursuit, from $\epsilon^{-8/3}$ to ϵ^{-2}. The experimental behavior of the first scheme, as well as that of a simple variant the second scheme, is consistent with a geometric convergence rate. We interpret this as evidence that the dissipation minimization perspective may stimulate even more approaches to the design and analysis of efficient and practical IRLS-type methods.

An interesting open problem is whether the proposed approach can solve generalizations of the ℓ_1-norm minimization problem, such as the graphical lasso (GLASSO) problem [11, 34, 37].

Funding Open access funding provided by Università degli Studi Roma Tre within the CRUI-CARE Agreement.

References

1. Alvarez, F., Bolte, J., Brahic, O.: Hessian Riemannian gradient flows in convex programming. SIAM J. Control Optim. **43**(2), 477–501 (2004)
2. Amari, S.: Information Geometry and Its Applications. Springer, Berlin (2016)
3. Arora, S., Hazan, E., Kale, S.: The multiplicative weights update method: a meta-algorithm and applications. Theory Comput. **8**(1), 121–164 (2012)
4. Bach, F., Jenatton, R., Mairal, J., Obozinski, G.: Optimization with sparsity-inducing penalties. Found. Trends Mach. Learn. **4**(1), 1–106 (2012)
5. Bansal, N., Gupta, A.: Potential-function proofs for gradient methods. Theory Comput. **15**(4), 1–32 (2019)
6. Bauschke, H., Bolte, J., Teboulle, M.: A descent lemma beyond Lipschitz gradient continuity: First-order methods revisited and applications. Math. Oper. Res. **42**(2), 330–348 (2017)
7. Beck, A.: On the convergence of alternating minimization for convex programming with applications to iteratively reweighted least squares and decomposition schemes. SIAM J Optim. **25**(1), 185–209 (2015)
8. Beck, A.: First-Order Methods in Optimization. SIAM, Philadelphia, PA (2017)
9. Beck, A., Teboulle, M.: Mirror descent and nonlinear projected subgradient methods for convex optimization. Oper. Res. Lett. **31**(3), 167–175 (2003)
10. Becker, R., Bonifaci, V., Karrenbauer, A., Kolev, P., Mehlhorn, K.: Two results on slime mold computations. Theor. Comput. Sci. **773**, 79–106 (2019)
11. Benfenati, A., Chouzenoux, É., Pesquet, J.: Proximal approaches for matrix optimization problems: Application to robust precision matrix estimation. Signal Process. **169**, 107417 (2020)
12. Bollobás, B.: Modern Graph Theory. Springer, New York, NY (1998)
13. Bonifaci, V.: On the convergence time of a natural dynamics for linear programming. In Proc. of the 28th Int. Symposium on Algorithms and Computation, pages 17:1–17:12. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl (2017)
14. Bonifaci, V.: MATLAB implementation of Laplacian-based gradient methods for L1-norm minimization. http://ricerca.mat.uniroma3.it/users/vbonifaci/soft/l1opt.zip, (2020)
15. Bonifaci, V., Mehlhorn, K., Varma, G.: Physarum can compute shortest paths. In Proc. of the 23rd ACM-SIAM Symposium on Discrete Algorithms, pages 233–240. SIAM, (2012)
16. Boyd, S., Vanderberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
17. Cai, J., Osher, S.J., Shen, Z.: Convergence of the linearized Bregman iteration for ℓ_1-norm minimization. Math. Comput. **78**(268), 2127–2136 (2009)
18. Candès, E., Romberg, J.: ℓ_1-magic: Recovery of sparse signals via linear programming. https://statweb.stanford.edu/ candes/l1magic/downloads/l1magic.pdf, (2005)
19. Chartrand, R., Yin, W.: Iteratively reweighted algorithms for compressive sensing. In Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pages 3869–3872. IEEE, (2008)
20. Chen, S.S., Donoho, D.L., Saunders, M.A.: Atomic decomposition by basis pursuit. SIAM Rev. **43**(1), 129–159 (2001)
21. Chin, H.H., Madry, A., Miller, G.L., Peng, R.: Runtime guarantees for regression problems. In Proc. of Innovations in Theoretical Computer Science, pages 269–282. ACM, (2013)
22. Christiano, P., Kelner, J.A., Madry, A., Spielman, D.A., Teng, S.-H.: Electrical flows, Laplacian systems, and faster approximation of maximum flow in undirected graphs. In Proc. of the 43rd ACM Symp. on Theory of Computing, pages 273–282. ACM, (2011)
23. Daubechies, I., DeVore, R., Fornasier, M., Güntürk, C.: Iteratively reweighted least squares minimization for sparse recovery. Comm. on Pure Appl. Math. **63**(1), 1–38 (2010)
24. Donoho, D.L.: For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution. Commun. Pure Appl. Math. **59**(6), 797–829 (2006)
25. Ene, A., Vladu, A.: Improved convergence for ℓ_1 and ℓ_∞ regression via iteratively reweighted least squares. In Proceedings of the 36th International Conference on Machine Learning, pages 1794–1801, (2019)
26. Facca, E., Cardin, F., Putti, M.: Physarum dynamics and optimal transport for basis pursuit. arXiv:1812.11782v1 [math.NA], (2019)
27. Foucart, S., Rauhut, H.: A Mathematical Introduction to Compressive Sensing. Birkhäuser, New York, NY (2013)

28. Ghosh, A., Boyd, S., Saberi, A.: Minimizing effective resistance of a graph. SIAM Rev. **50**(1), 37–66 (2008)
29. Godsil, C., Royle, G.: Algebraic Graph Theory. Springer, Berlin (2001)
30. Goldstein, T., Studer, C.: Phasemax: Convex phase retrieval via basis pursuit. IEEE Trans. Inf. Theory **64**(4), 2675–2689 (2018)
31. Green, P.J.: Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. J. R. Statist. Soc., Series B **46**(2), 149–192 (1984)
32. Hofbauer, J., Sigmund, K.: Evolutionary Games and Population Dynamics. Cambridge University Press, Cambridge (1998)
33. Horn, R.A., Johnson, C.R.: Matrix Analysis. Cambridge University Press, Cambridge (2013)
34. Kao, J., Tian, D., Mansour, H., Ortega, A., Vetro, A.: Disc-glasso: Discriminative graph learning with sparsity regularization. In Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing, pages 2956–2960. IEEE (2017)
35. Lu, H., Freund, R.M., Nesterov, Y.: Relatively smooth convex optimization by first-order methods, and applications. SIAM J. Optim. **28**(1), 333–354 (2018)
36. Magnus, J.R., Neudecker, H.: Matrix Differential Calculus with Applications in Statistics and Econometrics. Wiley, Oxford (2019)
37. Mazumder, R., Hastie, T.: The graphical lasso: New insights and alternatives. Electron. J. Statist. **6**, 2125–2149 (2012)
38. Nesterov, Y.: Smooth minimization of non-smooth functions. Math. Program. **103**(1), 127–152 (2005)
39. Osborne, M.R.: Finite Algorithms in Optimization and Data Analysis. Wiley, Oxford (1985)
40. Rao, C., Toutenburg, H., Heumann, S.: Linear Models and Generalizations. Springer, Berlin (2008)
41. Rockafellar, R.T.: Convex Analysis. Princeton University Press, Princeton, NJ (1970)
42. Saha, T., Srivastava, S., Khare, S., Stanimirovic, P.S., Petkovic, M.D.: An improved algorithm for basis pursuit problem and its applications. Appl. Math. Comput. **355**, 385–398 (2019)
43. Straszak, D., Vishnoi, N.K.: IRLS and slime mold: Equivalence and convergence. arXiv:1601.02712 [cs.DS], (2016)
44. Straszak, D., Vishnoi, N.K.: Natural algorithms for flow problems. In Proc. of the 27th ACM-SIAM Symposium on Discrete Algorithms, pages 1868–1883. SIAM, (2016)
45. Tero, A., Kobayashi, R., Nakagaki, T.: A mathematical model for adaptive transport network in path finding by true slime mold. J. Theor. Biol. **244**, 553–564 (2007)
46. Wilson, A.: Lyapunov arguments in optimization. Ph.D. dissertation, University of California at Berkeley, (2018)
47. Yang, A.Y., Zhou, Z., Balasubramanian, A.G., Sastry, S.S., Ma, Y.: Fast ℓ_1-minimization algorithms for robust face recognition. IEEE Trans. Image Process. **22**(8), 3234–3246 (2013)
48. Yin, W., Osher, S.J., Goldfarb, D., Darbon, J.: Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing. SIAM J. Imaging Sci. **1**(1), 143–168 (2008)