



Research article

The ADMM algorithm for audio signal recovery and performance modification with the dual Douglas-Rachford dynamical system

Andrew Calcan* and Scott B. Lindstrom

Centre for Optimisation and Decision Science, Curtin University, Perth, Western Australia, Australia

* **Correspondence:** Email: andrew.calcan@student.curtin.edu.au.

Abstract: Practitioners employ operator splitting methods—such as alternating direction method of multipliers (ADMM) and its “dual” Douglas-Rachford method (DR)—to solve many kinds of optimization problems. We provide a gentle introduction to these algorithms, and illustrations of their duality-like relationship in the context of solving basis pursuit problems for audio signal recovery. Recently, researchers have used the dynamical systems associated with the iterates of splitting methods to motivate the development of schemes to improve performance. These developments include a class of methods that act by iteratively minimizing surrogates for a Lyapunov function for the dynamical system. An exemplar of this class is currently state-of-the-art for the feasibility problem of finding wavelets with special structure. Early experimental evidence has also suggested that, when implemented in a primal-dual (ADMM and DR) framework, this exemplar may provide improved performance for basis pursuit problems. We provide a reasonable way to compute the updates for this exemplar, and we study the application of this method to the aforementioned basis pursuit audio problems. We provide experimental results and visualizations of the dynamical system for the dual DR sequence. We observe that for highly structured problems with real data, the algorithmic behavior is noticeably different than for randomly generated problems.

Keywords: ADMM; Douglas-Rachford; Lyapunov functions; basis pursuit; signal recovery; audio

Mathematics Subject Classification: 90-05, 90C25

1. Introduction

Throughout, \mathcal{H} denotes a Hilbert space. Additionally, $\Gamma_0(\mathcal{H})$ denotes the set of proper lower semicontinuous convex functions from \mathcal{H} to $[\infty_-, \infty_+]$. For readers unfamiliar with these notions, it suffices to know that the functions f, g, f^*, g^* from the problems described herein all belong to the class $\Gamma_0(\mathbb{R}^r)$.

The alternating direction method of multipliers (ADMM) solves problems of the form

$$\underset{(x,z) \in \mathbb{R}^n \times \mathbb{R}^m}{\text{minimize}} \quad f(x) + g(z) \quad \text{subject to} \quad Jx + Bz - v = 0, \quad (\mathbf{p})$$

where f and g are convex functions, $v \in \mathbb{R}^r$, $J \in \mathbb{R}^{r \times n}$, and $B \in \mathbb{R}^{r \times m}$. The ADMM algorithm deals with the two functions f and g separately, and so it is said to be an operator splitting method. Depending upon how we choose the functions f and g , we can model many different problems in this very flexible form. These include regression and classification problems, image and signal denoising, probabilistic optimization problems like progressive hedging, and basis pursuit [13]. We can use the latter to recover an approximate audio signal from a compressed set of sampling data.

This paper is outlined as follows. In Section 2 we introduce the basis pursuit audio problem. In Section 3 we introduce ADMM, motivating with basis pursuit. In Section 4, we describe the Douglas-Rachford method (DR) and use the basis pursuit problem to illustrate DR's duality-like relationship with ADMM. In Section 5, we describe the exemplar Lyapunov surrogate method with which we will work and provide (Theorem 2), a clean way to compute it. The remainder of the paper contains our experimental results, which we summarize in Section 7. This paper is intended to be widely accessible, including to researchers who are unfamiliar with ADMM. As such, we adopt a somewhat tutorial approach to the topic.

2. Basis pursuit for audio signal recovery

When digital audio is recorded, a microphone's diaphragm vibrates in response to sound waves, and N measurements are taken of the diaphragm's displacement. A typical rate at which these samples are collected is 44,100 per second (44.1 kHz). Recording over a time interval of length L , we denote by v_i the measurement value at sample time:

$$t_i := \left(\frac{2i-1}{2} \right) \left(\frac{L}{N} \right). \quad (2.1)$$

We can define a continuous function

$$V(t) := \frac{x_0}{\sqrt{N}} + \frac{\sqrt{2}}{\sqrt{N}} \sum_{k=1}^{N-1} x_k \cos\left(\frac{k\pi t}{L}\right) \quad \text{such that} \quad V(t_i) = v_i \quad \forall i \in \{1, \dots, N\}.$$

The unique constants x_k can be computed as follows. First, writing down the equality for the i th sample yields

$$\begin{aligned} v_i = V(t_i) &= \frac{x_0}{\sqrt{N}} + \frac{\sqrt{2}}{\sqrt{N}} \sum_{k=1}^{N-1} x_k \cos\left(\frac{k\pi t_i}{L}\right) \quad (\text{just the definition}) \\ &= \frac{x_0}{\sqrt{N}} + \frac{\sqrt{2}}{\sqrt{N}} \sum_{k=1}^{N-1} x_k \cos\left(\frac{k\pi (2i-1)L}{L \cdot 2N}\right) \quad (\text{substituting (2.1)}) \\ &= \frac{\sqrt{2}}{\sqrt{N}} \left(\frac{x_0}{\sqrt{2}} + \sum_{k=1}^{N-1} x_k \cos\left(\frac{k\pi(2i-1)}{2N}\right) \right). \end{aligned}$$

Since we have N such equalities (one for each sample (t_i, v_i)) in N unknowns (x_0, \dots, x_{N-1}) , we can write them as the following linear system:

$$\frac{\sqrt{2}}{\sqrt{N}} \underbrace{\begin{bmatrix} \frac{1}{\sqrt{2}} & \cos\left(\frac{\pi}{2N}\right) & \cos\left(\frac{2\pi}{2N}\right) & \cdots & \cos\left(\frac{(N-1)\pi}{2N}\right) \\ \frac{1}{\sqrt{2}} & \cos\left(\frac{3\pi}{2N}\right) & \cos\left(\frac{2 \cdot 3\pi}{2N}\right) & \cdots & \cos\left(\frac{(N-1)3\pi}{2N}\right) \\ \frac{1}{\sqrt{2}} & \cos\left(\frac{5\pi}{2N}\right) & \cos\left(\frac{2 \cdot 5\pi}{2N}\right) & \cdots & \cos\left(\frac{(N-1)5\pi}{2N}\right) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\sqrt{2}} & \cos\left(\frac{(2N-1)\pi}{2N}\right) & \cos\left(\frac{2(2N-1)\pi}{2N}\right) & \cdots & \cos\left(\frac{(N-1)(2N-1)\pi}{2N}\right) \end{bmatrix}}_{=:M} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{N-1} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_N \end{bmatrix}.$$

The matrix M is the inverse discrete cosine transform matrix, and it is orthogonal. Its inverse M^T —the discrete cosine transform matrix, allows us to compute the x values as $x = M^T v$. The nonzero values in x determine the shifts and dilates of the cosine function (i.e., the frequencies) that are present in the signal v . For this reason, x is said to be the spectrum of v .

2.1. Signal compression

To reduce the size of audio files, we can delete some of the samples. The deletion of a sample (t_i, v_i) is reflected in the system by the removal of row i from M and v . For example, removing samples (t_2, v_2) and (t_4, v_4) , the previous linear system becomes

$$\frac{\sqrt{2}}{\sqrt{N}} \underbrace{\begin{bmatrix} \frac{1}{\sqrt{2}} & \cos\left(\frac{\pi}{2N}\right) & \cos\left(\frac{2\pi}{2N}\right) & \cos\left(\frac{3\pi}{2N}\right) & \cos\left(\frac{4\pi}{2N}\right) \\ \frac{1}{\sqrt{2}} & \cos\left(\frac{5\pi}{2N}\right) & \cos\left(\frac{2 \cdot 5\pi}{2N}\right) & \cos\left(\frac{3 \cdot 5\pi}{2N}\right) & \cos\left(\frac{4 \cdot 5\pi}{2N}\right) \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}}_{=: \tilde{M}} \underbrace{\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ \vdots \end{bmatrix}}_{=: \tilde{x}} = \underbrace{\begin{bmatrix} v_1 \\ v_3 \\ v_5 \\ \vdots \end{bmatrix}}_{=: \tilde{v}}.$$

Of course, the solution \tilde{x} is no longer unique. The set of solutions is a subspace of dimension 2, since 2 is the number of samples we have deleted and therefore the number of free variables we have gained. We can still use the Moore-Penrose pseudo-inverse \tilde{M}^T to obtain a spectrum $\tilde{x} = \tilde{M}^T \tilde{v}$. The signal \tilde{V} that corresponds to this spectrum \tilde{x} still matches the samples we have not thrown away. Assuming we have not thrown away many samples, \tilde{V} may still provide a good approximation of V . In practice, of course, we want to reduce file size appreciably. When we remove 90% of samples (as is typical), the spectrum $\tilde{M}^T \tilde{v}$ no longer provides a good approximation signal. Note that while we removed entries 2 and 4 for illustrative purposes, there is nothing special about removing even-numbered entries; in our experiments, we remove 90% of entries randomly.

2.2. Basis pursuit for signal recovery

We would expect that the sparsest solution \tilde{x} consists of the frequencies that actually matter, while avoiding the superfluous ones. Finding the sparsest solution to a linear system $Ax = c$ is equivalent to the optimization problem:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \|x\|_1 \quad \text{subject to} \quad Ax - c = 0, \quad \text{with} \quad A \in \mathbb{R}^{r \times n}, \quad c \in \mathbb{R}^r. \quad (2.2)$$

This is called the basis pursuit problem. For our signal recovery application, we solve this optimization problem with $A = \tilde{M}$ and $c = \tilde{v}$. Using a Matlab code (**P-signal**) to generate a signal c , we compare

in Figure 1* the solution from (2.2) to a solution obtained using the Moore-Penrose pseudo-inverse. For this problem, c has 15 frequencies (i.e., $\text{Freq}=15$) denoted by $b_1, \dots, b_{15} \in \{1, 2, \dots, 500\}$ with corresponding amplitudes $a_1, \dots, a_{15} \in [0, 1]$. Here c represents 1/10th of a second of audio, and so is defined over domain $d = [0, \frac{1}{44100}, \frac{2}{44100}, \dots, \frac{4410}{44100}]$.

Matlab code 1. The code used for signal generation.

```
d = (0:(1/44100):(0.1-(1/44100)))';
a = rand(Freq,1);
b = randi(500,Freq,1);
c = 0;
for i = 1:Freq
    c = c + (a(i)*sin(pi*b(i)*d));
end
```

(P-signal)

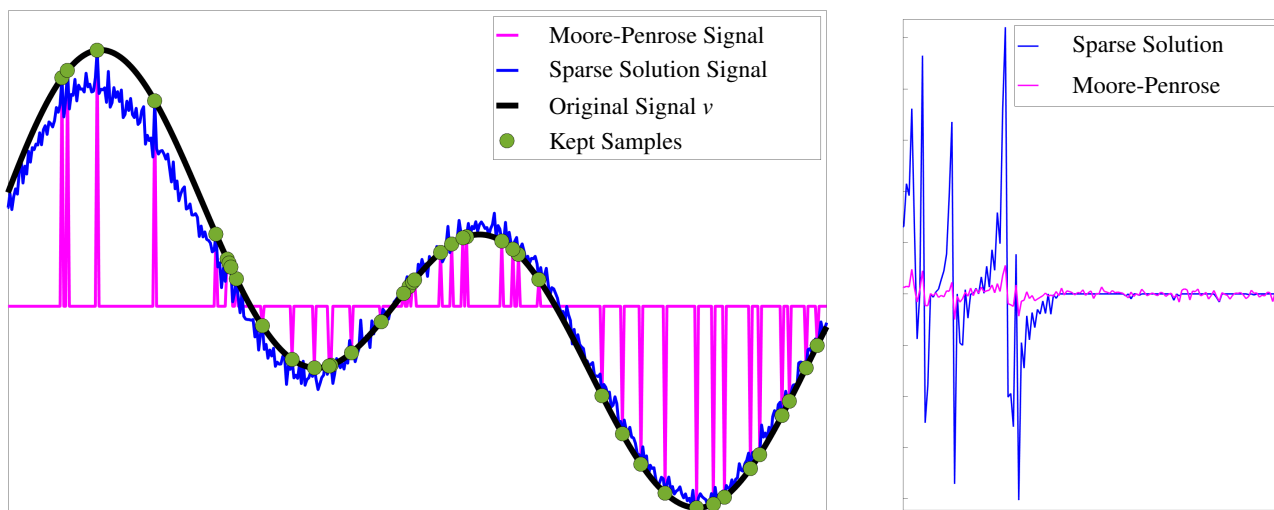


Figure 1. Portion of the reconstructed signals (left) and their most significant spectra (right).

3. The alternating direction method of multipliers

We can use ADMM to solve the basis pursuit problem. We first define the following set and associated function:

$$\text{Let } S := \{x \mid \tilde{M}x - c = 0\} \text{ and define } \delta_S := \begin{cases} 0, & \text{if } x \in S, \\ \infty, & \text{otherwise.} \end{cases}$$

We can use these to rewrite (2.2) in the following form:

$$\underset{(x,z)}{\text{minimise}} \delta_S(x) + \|z\|_1 \text{ subject to } x - z = 0. \quad \text{(BP-p)}$$

*The specific signal of Figure 1 can be generated by fixing the random generators with the additional code `rng(3)`.

The function δ_S is called the *indicator* of the set S . We introduce it so that we can take advantage of ADMM's ability to treat the functions $f(x) = \delta_S(x)$ and $g(z) = \|z\|_1$ separately. ADMM solves the problem

$$\min_{(x,z) \in \mathbb{R}^n} \max_{\lambda \in \mathbb{R}^m} L_\rho(x, z, \lambda) = \delta_S(x) + \|z\|_1 + (\lambda^k)^T(x - z) + \frac{\rho}{2}\|x - z\|^2, \quad (3.1)$$

by finding a triple (x^*, z^*, λ^*) . This triple is a saddle point for L_0 in the sense that (x^*, z^*) minimizes $(x, z) \mapsto L_0(x, z, \lambda^*)$ while λ^* maximizes $\lambda \mapsto L_0(x^*, z^*, \lambda)$. See, for example, [9, Section 5.5] and [10, Section 4.3]. Here ρ is a chosen constant. The *multiplier* variable λ generalizes the classical notion of a Lagrange multiplier for a general convex optimization problem, and is also called a *dual* variable. The *primal* variables x and z have the property that the optimal pair (x^*, z^*) is a solution to (BP-p). The ADMM update steps are

$$x^{k+1} := \operatorname{argmin}_x \{L_\rho(x, z^k, \lambda^k)\}, \quad (3.2a)$$

$$z^{k+1} := \operatorname{argmin}_z \{L_\rho(x^{k+1}, z, \lambda^k)\}, \quad (3.2b)$$

$$\lambda^{k+1} := \lambda^k + \rho(x^{k+1} - z^{k+1}). \quad (3.2c)$$

The multiplier update (3.2c) maximizes a concave quadratic approximation to $\lambda \mapsto L_\rho(x^{k+1}, z^{k+1}, \lambda)$. ADMM draws its name from this iterative process of minimizing with respect to the primal variables x and z , and then maximizing over the dual variable λ .

3.1. ADMM updates for basis pursuit

Let us look at how these updates are computed for the basis pursuit problem. We begin with the x update:

$$\begin{aligned} x^{k+1} &= \operatorname{argmin}_x \{ \delta_S(x) + \|z^k\|_1 + (\lambda^k)^T(x - z^k) + \frac{\rho}{2}\|x - z^k\|^2 \} \\ &= \operatorname{argmin}_x \left\{ \delta_S(x) + \left\| x - z^k + \frac{1}{\rho}(\lambda^k)^T \right\|^2 \right\} \quad \left(\begin{array}{l} \text{completing the square \&} \\ \text{dropping terms without } x \end{array} \right) \\ &= \operatorname{argmin}_{x \in S} \left\| x - \left(z^k + \frac{1}{\rho} \lambda^k \right) \right\|^2 =: P_S \left(z^k - \frac{\lambda^k}{\rho} \right). \end{aligned} \quad (3.3)$$

Here P_S is the closest-point projection operator for the set S . As S is closed and convex in a Hilbert space, P_S has nonempty, single-valued images [3]. Moreover, since S is an affine subspace defined by $\tilde{M}x = c$, P_S has a closed form [11]. We can use that closed form to write (3.3) explicitly:

$$x^{k+1} = \left(z^k - \frac{\lambda^k}{\rho} \right) + \tilde{M}^T (\tilde{M} \tilde{M}^T)^{-1} (c - \tilde{M} (z^k - \frac{\lambda^k}{\rho})) = (I - \tilde{M}^T (\tilde{M} \tilde{M}^T)^{-1} \tilde{M}) (z^k - \frac{\lambda^k}{\rho}) + \tilde{M}^T (\tilde{M} \tilde{M}^T)^{-1} c.$$

We compute the z update as follows:

$$\begin{aligned} z^{k+1} &= \operatorname{argmin}_z \{ \delta_S(x^{k+1}) + \|z\|_1 + (\lambda^k)^T(x^{k+1} - z) + \frac{\rho}{2}\|x^{k+1} - z\|^2 \} \\ &= \operatorname{argmin}_z \left\{ \|z\|_1 + \left\| x^{k+1} + \frac{1}{\rho}(\lambda^k)^T - z \right\|^2 \right\} \quad \left(\begin{array}{l} \text{completing the square \&} \\ \text{omitting terms without } z \end{array} \right) \\ &=: \mathcal{S}_{\frac{1}{\rho}} \left(x^{k+1} + \frac{1}{\rho} \lambda^k \right). \end{aligned}$$

The operator $\mathcal{S}_{\frac{1}{\rho}}(u)$ is known as the soft thresholding function. It is computable as follows:

$$(\forall i \in \{1, 2, \dots, n\}) \quad \mathcal{S}_{\frac{1}{\rho}}(u)_i := \begin{cases} u_i - \frac{1}{\rho}, & \text{if } u_i > \frac{1}{\rho}, \\ 0, & \text{if } |u_i| \leq \frac{1}{\rho}, \\ u_i + \frac{1}{\rho}, & \text{if } u_i < -\frac{1}{\rho}. \end{cases}$$

Both the projection operator P_S and the shrinkage operator $\mathcal{S}_{\frac{1}{\rho}}$ are examples of proximity operators. The proximity operator for a function $h \in \Gamma_0(\mathcal{H})$ is defined as follows:

$$\text{prox}_h : \mathcal{H} \rightarrow \mathcal{H} : w \mapsto \operatorname{argmin}_{u \in \mathcal{H}} (f(u) + \frac{1}{2} \|w - u\|^2).$$

Straight from the definition, one sees that the projection operator P_S is prox_{δ_S} , while the soft thresholding operator $\mathcal{S}_{\frac{1}{\rho}}$ is $\text{prox}_{\|\cdot\|_1}$. Such operators are used to compute the steps for operator splitting methods.

4. ADMM and DR duality

We will next look at another operator splitting method, one that has a special *duality-like* relationship with ADMM. Let I be the identity map and R_h denote the reflected proximal mapping:

$$R_h := 2\text{prox}_h - I. \quad (4.1)$$

The Douglas-Rachford algorithm (DR) minimizes $f + g$ where $f, g \in \Gamma_0(\mathcal{H})$ by generating a sequence $(y^n)_{n=1}^\infty$ as

$$y^{n+1} \in T_{g,f}(y^n) \text{ where } T_{g,f} := \frac{1}{2}(I + R_f R_g). \quad (4.2)$$

In [2], Attouch showed[†] that if f and g are convex and 0 belongs to the interior of $\{x : f(x) < \infty\} - \{x : g(x) < \infty\}$, then $f + g$ satisfies the necessary conditions for the classical DR convergence result of Lions and Mercier.

Theorem 1. ([2, 26]) *If $f, g \in \Gamma_0(\mathcal{H})$ and Attouch's criterion holds, then y^k from (4.2) converges weakly to some $y \in \mathcal{H}$ such that $\text{prox}_g y$ minimizes $f + g$.*

For our context, we are interested in the setting where the Douglas-Rachford method is applied to the problem

$$\underset{y}{\text{minimize}} \quad f^*(-A^T y) + g^*(y), \quad (\mathbf{d})$$

where A is a matrix of appropriate dimension. Here h^* denotes the Fenchel-Moreau-Rockafellar conjugate of h [3, 24], defined as

$$h^*(y) = \sup\{y^T x - h(x)\}.$$

[†]Attouch's actual requirements are less restrictive than what we have presented.

We are particularly interested in the case when $A = I$, $f^* = \delta_S^*$, and $g^* = \|\cdot\|_1^*$. The conjugates are

$$\delta_S^*(y) = \sup_{x \in S} \{y^T x\} \text{ and } \|\cdot\|_1^*(y) = \begin{cases} 0, & \text{if } \|x\|_\infty \leq 1, \\ \infty, & \text{otherwise.} \end{cases}$$

Note that Attouch’s condition is satisfied for δ_S^* and $\|\cdot\|_1^*$, as long as the system $Ax = b$ is consistent. The general problem **(d)** is said to be dual to the general problem **(p)**, while the specific problem

$$\text{minimize}_y \delta_S^*(-y) + \|y\|_1^* \tag{BP-d}$$

is dual to the primal basis pursuit problem **(BP-p)**. For fully convex problems, DR and ADMM are closely related, with ADMM for **(p)** being equivalent to DR for **(d)** [19, 20, 24, 25]. The linear relationship between DR and ADMM variables is outlined in Table 1.

Table 1. The relationship between ADMM and DR variables.

Primal	Dual
$\rho Ax^{k+1} = \text{prox}_{\rho f^* \circ (-AT)}(R_{\rho g^*} y^k) - R_{\rho g^*}(y^k)$	$T_{\rho g^*, \rho f^* \circ (-AT)} y^{k-1} = y^k = \lambda^k + \rho z^k$
$\rho z^k = y^k - \text{prox}_{\rho g^*} y^k$	$R_{\rho g^*} y^k = \lambda^k - \rho z^k$
$\lambda^k = \text{prox}_{\rho g^*} y^k$	$R_{\rho f^* \circ (-AT)} R_{\rho g^*} y^k = \lambda^k - \rho z^k + 2\rho Ax^{k+1}$

From Theorem 1 and the relationships in the table, it is clear that the multiplier sequence λ^k converges to the solution of **(d)**. For the basis pursuit problem **(BP-p)** and its dual problem **(BP-d)**, the relationship in Table 1 is illustrated in Figure 2. Because the DR operator T_{g^*, f^*} has particularly desirable properties, DR convergence results are often cited as particularly elegant proofs of the convergence of ADMM, and the two algorithms are frequently studied together [22, 23].

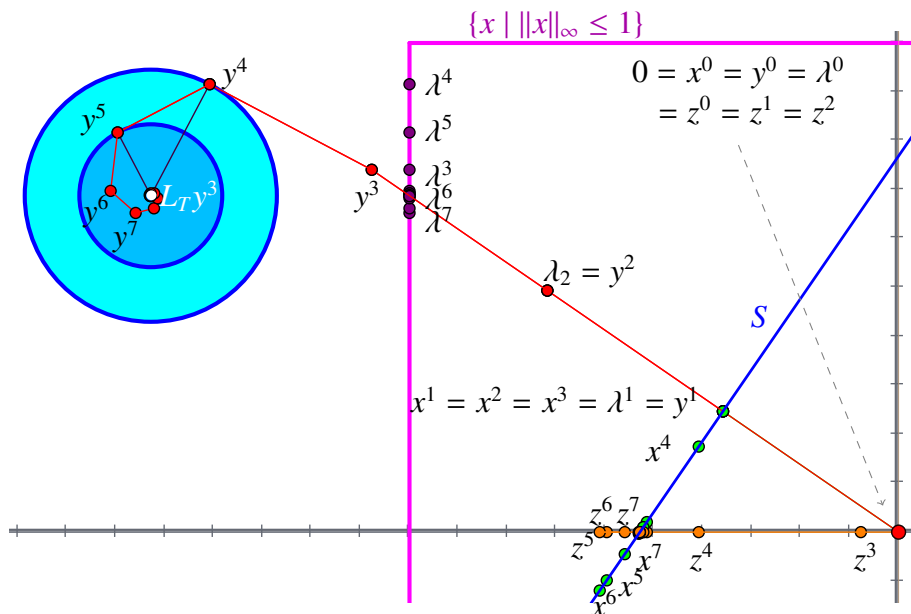


Figure 2. The ADMM sequence $(x^n, z^n, \lambda^n)_{n \in \mathbb{N}}$ for solving **(BP-p)** and the associated dual DR sequence $(y^n)_{n \in \mathbb{N}}$ for solving **(BP-d)**. They have the relationship in Table 1.

5. A Lyapunov surrogate method for basis pursuit

Benoist, Dao, Tam, Giladi, and Rüffer used Lyapunov functions to describe the dynamics of DR in cases when it exhibited spiraling patterns similar to the one in Figure 2 [8, 14, 21]. All of their Lyapunov functions U shared the property that

$$\langle \nabla U(y^n), y^{n-1} - y^n \rangle = 0. \quad (5.1)$$

Borrowing inspiration from their constructions, the method introduced in [24] provides an update candidate that minimizes a spherical surrogate for such a Lyapunov function. Figure 2 shows an example of such a surrogate, and the update obtained from it. When y^k, y^{k+1}, y^{k+2} are collinear, the construction is not possible and so a normal algorithmic update would be accepted. Otherwise, this update, denoted $L_T(y^k)$, is constructed to belong to the 2-dimensional affine subspace containing y^k, y^{k+1} , and y^{k+2} , and to have the property that $\langle L_T y^k - y^{k+1}, y^{k+1} - y^k \rangle = \langle L_T y^k - y^{k+2}, y^{k+2} - y^{k+1} \rangle = 0$. A candidate update for the multiplier is

$$\lambda^{L_T} := \text{prox}_{\rho g^*} L_T(y^k) = P_{\{x \mid \|x\|_\infty \leq 1\}}(L_T(y^k)).$$

We can propagate the update to the primal variables by $z^{L_T} \leftarrow L_T y^k - \lambda^{L_T}$ and $x^{L_T} \leftarrow P_S(z^{L_T} - \lambda^{L_T})$. Then x^{L_T} is a candidate for updating x . There are various criteria we can use to decide whether to accept it or reject it in favor of a regular ADMM update. In this paper, the main criterion we use in our experiments is whether x_{L_T} has a smaller 1-norm value than a regular update would. Based on the symbol for the operator $L_T : \mathbb{R}^r \rightarrow \mathbb{R}^r$, we refer to the full algorithm in this case as LT. For comparison, in some experiments we instead accept the x_{L_T} update always. We denote the algorithm in such cases by LTA. The operator L_T may be computed efficiently, as we now show.

Theorem 2. *Let $y^0, y^1, y^2 \in \mathcal{H}$. Let $w^1 = y^1 - y^0$ and $w^2 = y^2 - y^0$. If $\|w^1\|^2 \|w^2\|^2 - (\langle w^1, w^2 \rangle)^2 = 0$, then y^0, y^1, y^2 are collinear. Otherwise, let $w = y^0 + \mu_1 w^1 + \mu_2 w^2$ where*

$$\begin{aligned} \delta &:= \|w^1\|^2 \|w^2\|^2 - (\langle w^1, w^2 \rangle)^2, \\ \mu &:= \frac{1}{\delta} \begin{bmatrix} \|w^2\|^2 & -\langle w^1, w^2 \rangle \\ -\langle w^1, w^2 \rangle & \|w^1\|^2 \end{bmatrix} \begin{bmatrix} \|w^1\|^2 \\ \|w^2\|^2 - \langle w^1, w^2 \rangle + \|w^1\|^2 \end{bmatrix}, \end{aligned}$$

and w is the unique point in the 2-dimensional affine subspace of \mathcal{H} containing $\{y^0, y^1, y^2\}$ that satisfies

$$\langle w - y^1, y^1 - y^0 \rangle = \langle w - y^2, y^2 - y^1 \rangle = 0.$$

Proof. Letting $u = w - y^0$, the linear system rewrites to

$$\langle u - w^2, w^2 - w^1 \rangle = 0 \quad \text{and} \quad \langle u - w^1, w^1 \rangle = 0.$$

The second equality yields $\langle u, w^1 \rangle = \|w^1\|^2$, whereupon the first equality yields

$$\begin{aligned} \langle u, w^2 - w^1 \rangle &= \langle w^2, w^2 - w^1 \rangle = \|w^2\|^2 - \langle w^2, w^1 \rangle \\ \implies \langle u, w^2 \rangle &= \|w^2\|^2 - \langle w^2, w^1 \rangle + \langle u, w^1 \rangle = \|w^2\|^2 - \langle w^2, w^1 \rangle + \|w^1\|^2. \end{aligned}$$

As w belongs to the 2-dimensional affine subspace containing $\{y^0, y^1, y^2\}$, there must exist $\mu_1, \mu_2 \in \mathbb{R}$ such that $u = \mu_1 w^1 + \mu_2 w^2$. Substituting this identity into the two equalities above yields the system:

$$\begin{aligned} & \langle w^1, \mu_1 w^1 + \mu_2 w^2 \rangle = \|w^1\|^2, \\ \text{and } & \langle w^2, \mu_1 w^1 + \mu_2 w^2 \rangle = \|w^2\|^2 - \langle w^2, w^1 \rangle + \|w^1\|^2, \\ \Leftrightarrow & \begin{bmatrix} \|w^1\|^2 & \langle w^1, w^2 \rangle \\ \langle w^1, w^2 \rangle & \|w^2\|^2 \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} = \begin{bmatrix} \|w^1\|^2 \\ \|w^2\|^2 - \langle w^1, w^2 \rangle + \|w^1\|^2 \end{bmatrix}. \end{aligned}$$

The left matrix is invertible so long as its determinant δ is nonzero. Bearing in mind that $\|w^1\| \|w^2\| \cos(\theta) = \langle w^1, w^2 \rangle$ where θ is the angle between w^1 and w^2 , we have that this determinant is zero exactly when the angle between w^1 and w^2 is 0 or π (i.e., y^0, y^1, y^2 are collinear). When the determinant is nonzero, we take the inverse of the matrix on the left and obtain the claimed identity for μ . \square

Remark 1 (Checking collinearity). *In computational practice, to decide whether the determinant δ in Theorem 2 is nonzero, one must choose some $\epsilon > 0$ and replace the condition $\delta \neq 0$ with $|\delta| > \epsilon$. In our experiments, we used the threshold $\epsilon = 10^{-10}$, and never encountered the case when $|\delta| < \epsilon$. If ϵ is chosen very small and the condition number is very large, the computed μ may suffer numerical inaccuracy. This is another important reason for the inclusion of the objective function check.*

Interestingly, for structured feasibility problems of the kind studied in [8, 14, 21], it was shown in [24] that the circumcentered reflections method (see, for example, [1, 4–7, 15, 17]) is another example of an algorithm that returns minimizers for surrogates of a Lyapunov function for the DR dynamical system. For reasons described in [24], the natural generalization of circumcentered reflections method does not work for basis pursuit problems, and so we do not include it here. Indeed, a key motivation for the algorithm LT is that whenever a Lyapunov function has the property (5.1), LT and LTA return minimizers of spherical surrogates for that Lyapunov function [24]. This means that they may be useful for a wider variety of applications than those listed here. Algorithm 1 shows their implementation for basis pursuit specifically. In Table 2, we summarize the criteria that we use in our experiments to decide whether to accept an update candidate derived from $L_T(y^k)$ or reject it in favor of a regular update.

Algorithm 1 LT / LTA.

Data: $A^{m \times n}$, absolute and relative tolerances such that $\epsilon_{\text{abs}}, \epsilon_{\text{rel}} > 0$, a tolerance $\epsilon_{\text{det}} > 0$, and a data vector b .

Initialisation:

$$i, x, z, \hat{z}, \lambda, y^i \leftarrow 0; r = x - z; s = \hat{z} - z;$$

$$\epsilon_{\text{pri}} = \sqrt{n}\epsilon_{\text{abs}} + \epsilon_{\text{rel}} \max\{\|x\|, \|\hat{z}\|\}; \epsilon_{\text{dual}} = \sqrt{n}\epsilon_{\text{abs}} + \epsilon_{\text{rel}}\|\lambda\|;$$

while $\|r\| \geq \epsilon_{\text{pri}}, \|s\| \geq \epsilon_{\text{dual}}$ **do**

if $i \geq 2$ **then**

$$w^1 \leftarrow y^{i-1} - y^{i-2};$$

$$w^2 \leftarrow y^i - y^{i-2};$$

$$\delta \leftarrow \|w^1\|^2 \|w^2\|^2 - (w^{1T} w^2)^2;$$

if $|\delta| > \epsilon_{\text{det}}$ **then**

$$\mu \leftarrow \frac{1}{\delta} \begin{bmatrix} \|w^2\|^2 & -w^{1T} w^2 \\ -w^{1T} w^2 & \|w^1\|^2 \end{bmatrix} \begin{bmatrix} \|w^1\|^2 \\ \|w^2\|^2 - w^{2T} w^1 + \|w^1\|^2 \end{bmatrix};$$

$$y^{LT} \leftarrow y^{i-2} + \mu_1 w^1 + \mu_2 w^2;$$

$$\lambda^{LT} \leftarrow \mathcal{S}_{\frac{1}{\rho}}(y^{LT});$$

$$z^{LT} \leftarrow y^{LT} - \lambda^{LT};$$

$$x^{LT} \leftarrow (I - A^T(AA^T)^{-1}A)(z_{LT} - \frac{\lambda_{LT}}{\rho}) + A^T(AA^T)^{-1}b;$$

$$\text{criterion} = \|x_{LT}\|_1 < \|x\|_1;$$

if *criterion* **then**

$$x \leftarrow x^{LT};$$

$$z \leftarrow z^{LT};$$

$$\lambda \leftarrow \lambda^{LT};$$

$$i \leftarrow 0;$$

end

end

end

$$i \leftarrow i + 1;$$

$$x \leftarrow (I - A^T(AA^T)^{-1}A)(z - \frac{\lambda}{\rho}) + A^T(AA^T)^{-1}b;$$

$$\hat{z} \leftarrow z; z \leftarrow \mathcal{S}_{\frac{1}{\rho}}(x + \frac{\lambda}{\rho}); \lambda \leftarrow \lambda + \rho(x - z); y^i \leftarrow \lambda + \rho z;$$

$$r = x - z; s = \hat{z} - z;$$

end

Table 2. The criteria for LT and LTA in Algorithm 1.

Algorithm variation	Criterion used
LT	$\ x^{LT}\ _1 < \ x\ _1$
LTA	true

6. Experimental results

Performance profiles are a popular way of comparing algorithm performance; see [18], whose notation we closely follow. We use them to compare the performances of LT, LTA, and ADMM. For each problem p and solver s , we define

$t_{p,s}$ = the number of iterations required to solve problem p by solver s .

The performance of each algorithm is evaluated relative to the superior algorithm in each instance. This is done according to the following ratio, where S denotes the set of solvers

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} : s \in S\}}.$$

Our performance profile graphs depict the cumulative distribution functions Φ_s , defined as

$$\Phi_s(\tau) = \frac{1}{n_p} \text{size}\{p \in P \mid r_{p,s} \leq \tau\},$$

where n_p is the total number of problems solved by s . For example, if $\Phi_s(1.5) = 0.8$, then algorithm s solved 80% of problems using no more than 150% of the passes through (3.2) required by the algorithm that used the fewest passes.

6.1. Random problems

In [24], LT was found to reliably outperform ADMM for problem (2.2), where the problems were randomly generated according to the specifications in Boyd, Parikh, Chu, Peleato and Eckstein's Matlab scripts [12]. The code for these problems is **(P-random)**. Here, $A \in \mathbb{R}^{500 \times 5000}$, with each entry drawn from the standard normal distribution, while $x \in \mathbb{R}^{5000}$ is sparse with approximately 500 nonzero $x_i \sim \mathcal{N}(0, 1)$. The data vector c is given by $c = Ax$. The performance profile for LT, LTA, and ADMM in solving 1000 problems of this kind is Figure 3b. For these problems, the algorithms stopped once locating solutions within $\pm 10^{-4}$ of optimality (i.e., $\epsilon_{\text{abs}} = \epsilon_{\text{rel}} = 10^{-4}$).

Matlab code 2. The code used for randomly-generating basis pursuit problems.

```
n = 5000;
m = 500;
A = randn(m, n);
x = sprandn(n, 1, 0.1*n);
c = A*x.
```

(P-random)

Following [24], we say that an algorithm shows signs of spiraling when $\|\text{prox}_{g^*} y^{k+1} - \text{prox}_{g^*} y^k\|$ (i.e., $\|\lambda^{k+1} - \lambda^k\|$) exhibits a distinct “tombstone” pattern (observable in Figure 3a). This phenomenon is often associated with the existence of a Lyapunov function satisfying condition (5.1). Problems generated using **(P-random)** typically exhibit signs of spiraling.

In Figure 4, we illustrate the behavior of the DR dual sequence $(y^k)_{k \in \mathbb{N}}$ for a problem generated using **(P-random)**[‡]. We visualize (y^k) 's behavior by projecting it onto 10 different 2-dimensional subspaces

[‡]For the example shown, the random seed state was `rng(8)`.

(i.e., plotting coordinate pairs). Namely, we plot (y_1^n, y_2^n) (red), (y_1^n, y_3^n) (neon green), (y_1^n, y_4^n) (blue), (y_1^n, y_5^n) (cyan), (y_2^n, y_3^n) (magenta), (y_2^n, y_4^n) (yellow), (y_2^n, y_5^n) (purple), (y_3^n, y_4^n) (teal), (y_3^n, y_5^n) (orange), and (y_4^n, y_5^n) (green). In Figure 4b and 4c, we zoom in on the tail of the sequence (y_2^n, y_3^n) and plot the sequence $((y_{LT}^n)_2, (y_{LT}^n)_3)$ of coordinate pairs (blue shades) as well as the sequence $((y_{LTA}^n)_2, (y_{LTA}^n)_3)$ (green shades). The accepted L_T updates are shown as squares.

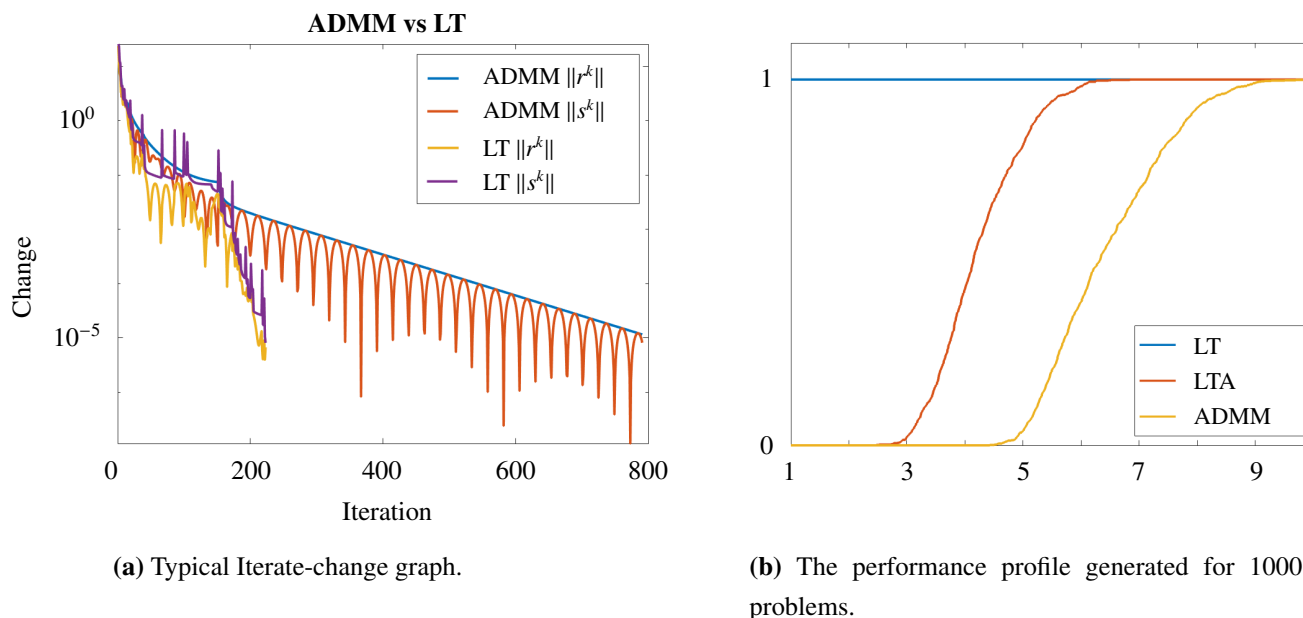


Figure 3. Algorithm performance for basis pursuit problems generated using (**P-random**).

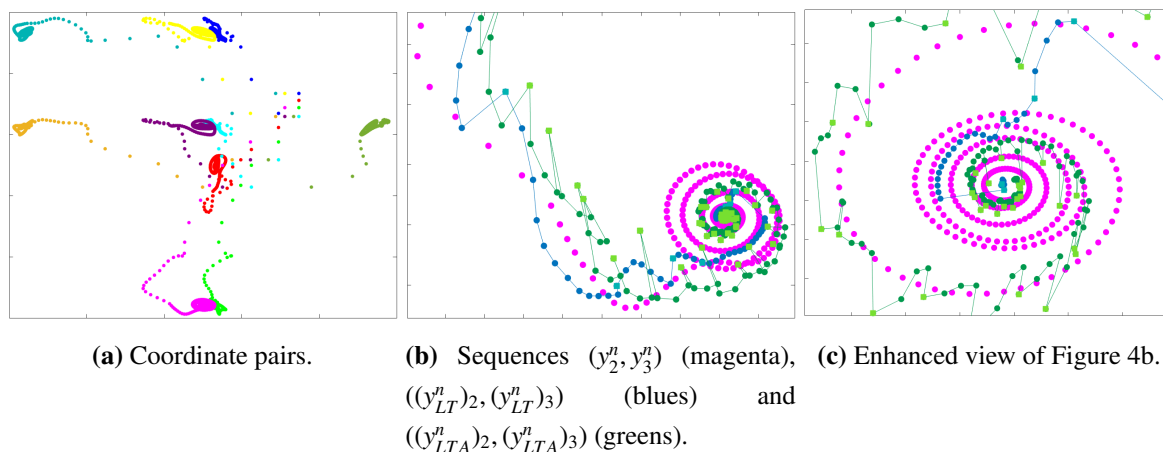


Figure 4. A basis pursuit problem showing signs of spiralling. Squares (shaded lighter) denote accepted L_T updates.

Because LT and LTA afforded measurable improvement for solving basis pursuit problems with random structure, we next benchmarked their performance on the audio compression and recovery problem.

6.2. Audio signal recovery

To evaluate the performance of LT and LTA for audio recovery problems, we generated signals using **(P-signal)**. $d = [0, \frac{1}{44100}, \frac{2}{44100}, \dots, \frac{4410}{44100}]$ was used as the domain of each c , while c is the sum of shifted and dilated sine functions with amplitudes and frequencies specified by $a_1, \dots, a_{\text{Freq}} \in [0, 1]$ and $b_1, \dots, b_{\text{Freq}} \in [1, 2, \dots, 500]$. The number of frequencies (Freq) corresponds to the number of such functions composing c , with a higher number of frequencies resulting in more complex signals. To assess the algorithms on a variety of signals, we ran experiments with Freq = 2, 5 and 10. For each Freq value, 1000 problems were solved where $A = \tilde{M}^{441 \times 4410}$. We include the performance profiles for these problems in Figure 5.

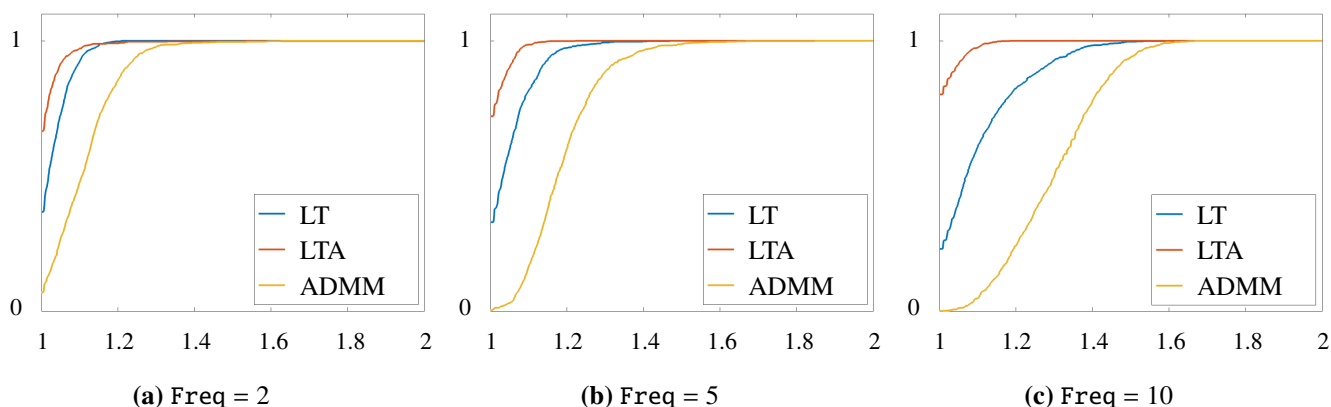


Figure 5. The performance profiles generated for LT, LTA, and ADMM in solving 1000 problems with $A = \tilde{M}^{441 \times 4410}$ and c generated using **(P-signal)**.

We provide a sense of how the algorithms pursue a sparse solution in Figure 6, where we define the superfluous x_i as having absolute values less than 0.7. This threshold was determined based on plotting the spectra for various solved problems (as in Figure 1). Figures 6a–c depict Mx^5 , Mx^{30} and Mx^* ($x^* = x^{158}$), as produced by ADMM solving (2.2) with $A = \tilde{M}$ and c specified by **(P-signal)** with Freq = 15[§]. We graph the percentage of x_i^k that satisfy $|x_i^k| > 0.7$ for $k = 5, 30, 158$, in Figure 6d. These signals were reconstructed with $\epsilon_{\text{abs}} = \epsilon_{\text{rel}} = 10^{-3}$ accuracy. Although not perfect, we found that lower ϵ_{abs} and ϵ_{rel} values did not result in better approximations for V , so this accuracy is used throughout our experiments.

[§]For the example shown, the random seed state was `rng(3)`.

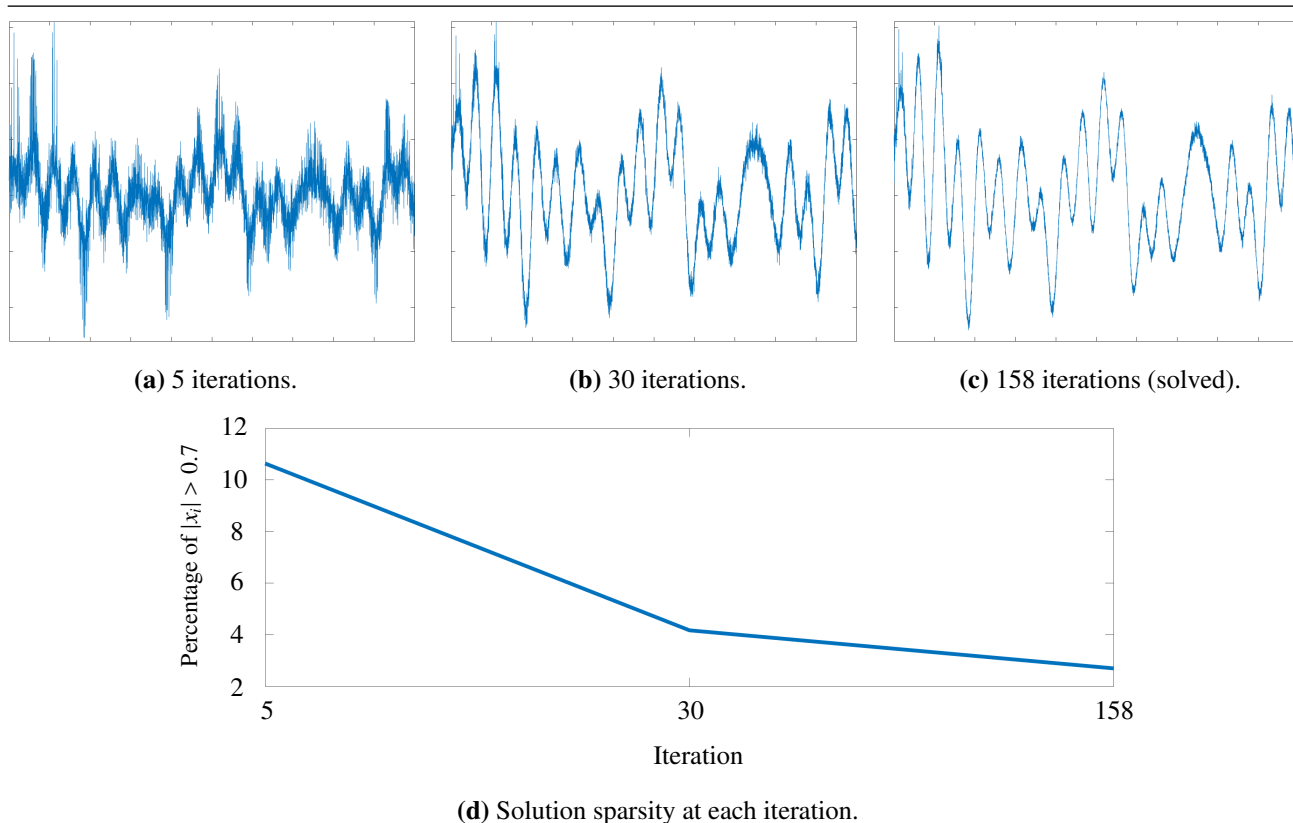


Figure 6. The signals reproduced at different stages of ADMM solving basis pursuit with $A = \tilde{M}$ and c generated using (**P-signal**).

6.2.1. Experiments on real audio

We likewise applied LT and LTA to recovering the waveform signal of a performance of Frédéric Chopin's Scherzo No. 2. We decomposed the 2 minute and 14 second performance into 1340 problems, each recovering 0.1 seconds of audio. Using the file's sampling rate of 44800 Hz, each problem was formulated with $A = \tilde{M}^{448 \times 4480}$. The performance profile for this experiment is Figure 7.

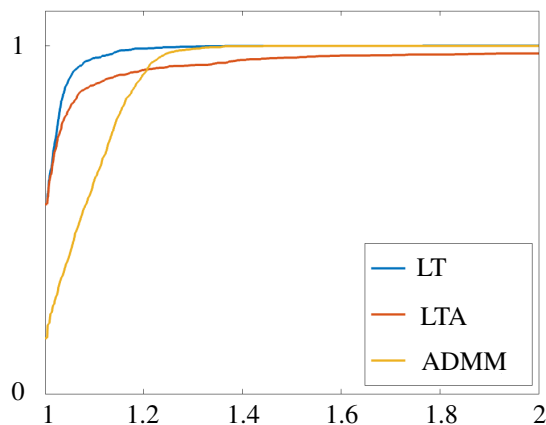


Figure 7. The performances of LT, LTA, and ADMM in recovering Chopin's Op. 31. The profile depicts 1340 problems with $A = \tilde{M}^{448 \times 4480}$.

6.3. Experiments: Discussion

The meta algorithms exhibited noticeably less improvement relative to ADMM when the random problem structure of **(P-random)** was replaced by the structure of the simulated audio problem **(P-signal)**, and even less improvement for a real audio problem. For problems we solved using \tilde{M} , we typically did not observe signs of spiraling for the dual sequence y^k ; when we did observe signs of spiraling, we only observed it after the signal was recovered (i.e. when computing more than necessary). Figure 8 shows y^k coordinate pairs for an example problem generated using **(P-signal)**[¶] with $\text{Freq} = 10$. Enhanced views of sequences (y_1^n, y_3^n) and (y_1^n, y_4^n) for this problem are included in Figure 9a. Figure 9b enhances the sequence (y_2^n, y_3^n) , and also shows the sequence of coordinate pairs $((y_{LT}^n)_2, (y_{LT}^n)_3)$ generated by LT, as well as the sequence of coordinate pairs $((y_{LTA}^n)_2, (y_{LTA}^n)_3)$ generated by the variant LTA. For the latter two sequences, regular DR updates are circles while updates accepted from the surrogate are squares.

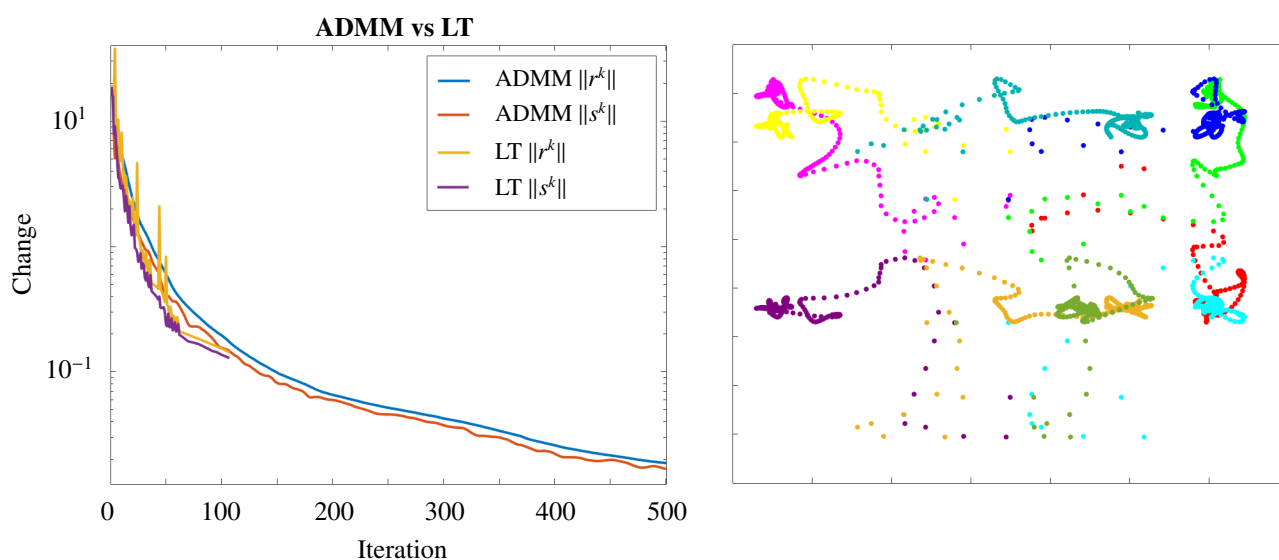
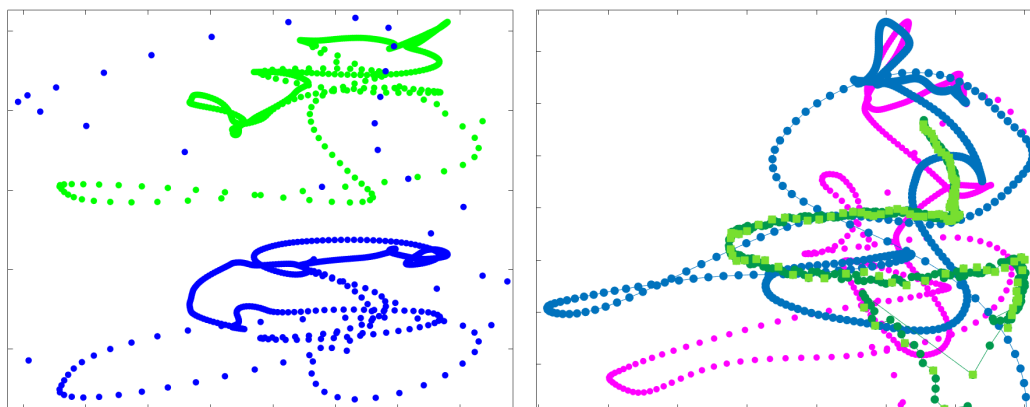


Figure 8. Convergence plot (left) and plot of coordinate pairs (right) for a problem not displaying signs of spiraling.

[¶]For the example shown, the random seed state was `rng(4)`



(a) Sequences (y_1^n, y_3^n) (green) and (y_1^n, y_4^n) (blue). (b) Sequence (y_2^n, y_3^n) (magenta), $((y_{LT}^n)_2, (y_{LT}^n)_3)$ (blues) and $((y_{LTA}^n)_2, (y_{LTA}^n)_3)$ (greens).

Figure 9. Enhanced views of sequences for the problem in Figure 8. Squares (shaded lighter) denote accepted L_T updates. All three coordinate pair sequences entered from bottom right and terminated at top right.

7. Conclusions

Our results highlight the importance of testing algorithms on problems with real data. While the Lyapunov surrogate variant LTA is state-of-the-art (much faster than DR and equally reliable) for the real world problem of finding wavelets [16], its performance for the real world audio compression and recovery problem is less impressive than experiments on random problems might suggest.

Additionally, our results are consistent with the hypothesis in [24] that Lyapunov surrogate methods are more likely to deliver improvement in situations when signs of spiraling are present, as we suspect that LT's inconsistent performance is due to differences in the dynamics of the underlying DR dual algorithm.

Finally, the visualizations in Figures 4 and 9 are interesting in two ways. First, they support the hypothesis that signs of spiraling do, in fact, reflect the dynamic they purport to. The dynamics we observe for the simulated audio problem in Figure 9, when signs of spiraling were not present, appear much less regular than those we observed for the fully random problem in Figure 4b, when signs of spiraling were present. Second, the latter dynamics still exhibit patterns that may lend themselves to extrapolation methods or other dynamics-based schemes. We suggest this as future research.

Use of AI tools declaration

The authors declare that they have not used artificial intelligence tools in the creation of this article.

Conflict of interest

The authors declare no conflict of interest.

Data Availability

The code used to generate all examples in this paper is available at <https://github.com/AndrewCalcan/Basis-Pursuit-via-LT-LTA-ADMM.git>.

References

1. R. Arefidamghani, R. Behling, Y. Bello-Cruz, A. Iusem, L. Santos, The circumcentered-reflection method achieves better rates than alternating projections, *Comput. Optim. Appl.*, **79** (2021), 507–530. <http://dx.doi.org/10.1007/s10589-021-00275-6>
2. H. Attouch, On the maximality of the sum of two maximal monotone operators, *Nonlinear Anal.-Theor.*, **5** (1981), 143–147. [http://dx.doi.org/10.1016/0362-546X\(81\)90039-0](http://dx.doi.org/10.1016/0362-546X(81)90039-0)
3. H. Bauschke, P. Combettes, *Convex analysis and monotone operator theory in Hilbert spaces*, Cham: Springer, 2 Eds., 2017. <http://dx.doi.org/10.1007/978-3-319-48311-5>
4. H. Bauschke, H. Ouyang, X. Wang, On circumcenters of finite sets in Hilbert spaces, arXiv: 1807.02093.
5. R. Behling, Y. Bello-Cruz, L. Santos, Circumcentering the Douglas-Rachford method, *Numer. Algor.*, **78** (2018), 759–776. <http://dx.doi.org/10.1007/s11075-017-0399-5>
6. R. Behling, Y. Bello-Cruz, L. Santos, On the linear convergence of the circumcentered-reflection method, *Oper. Res. Lett.*, **46** (2018), 159–162. <http://dx.doi.org/10.1016/j.orl.2017.11.018>
7. R. Behling, Y. Bello-Cruz, L. Santos, On the circumcentered-reflection method for the convex feasibility problem, *Numer. Algor.*, **86** (2021), 1475–1494. <http://dx.doi.org/10.1007/s11075-020-00941-6>
8. J. Benoist, The Douglas-Rachford algorithm for the case of the sphere and the line, *J. Glob. Optim.*, **63** (2015), 363–380. <http://dx.doi.org/10.1007/s10898-015-0296-1>
9. D. Bertsekas, *Convex optimization theory*, Melrose: Athena Scientific, 2009.
10. J. Borwein, A. Lewis, *Convex analysis and nonlinear optimization: theory and examples*, New York: Springer, 2 Eds., 2006. <http://dx.doi.org/10.1007/978-0-387-31256-9>
11. S. Boyd, L. Vandenberghe, *Convex optimisation*, Cambridge: Cambridge University Press, 2004. <http://dx.doi.org/10.1017/CBO9780511804441>
12. S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *Matlab scripts for alternating direction method of multipliers*, 2011. Available from: <https://web.stanford.edu/~boyd/papers/admm/>.
13. S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *Distributed optimisation and statistical learning via the alternating direction method of multipliers*, New York: Now Foundations and Trends, 2010. <http://dx.doi.org/10.1561/22000000016>
14. M. Dao, M. Tam, A Luapunov-type approach to convergence of the Douglas-Rachford algorithm, *J. Glob. Optim.*, **73** (2019), 83–112. <http://dx.doi.org/10.1007/s10898-018-0677-3>
15. N. Dizon, J. Hogan, S. Lindstrom, Circumcentered reflections method for wavelet feasibility problems, *ANZIAM J.*, **62** (2020), C98–C111. <http://dx.doi.org/10.21914/anziamj.v62.16118>

16. N. Dizon, J. Hogan, S. Lindstrom, Centering projection methods for wavelet feasibility problems, In *Current trends in analysis, its applications and computation*, Cham: Birkhäuser, 2022. http://dx.doi.org/10.1007/978-3-030-87502-2_66
17. Neil Dizon, J. A. Hogan, and Scott B. Lindstrom, Circumcentering reflection methods for nonconvex feasibility problems. *Set-Valued Var. Anal.*, **30** (2022), 943–973. <http://dx.doi.org/10.1007/s11228-021-00626-9>
18. E. Dolan, J. Moré, Benchmarking optimisation software with performance profiles, *Math. Program.*, **91** (2022), 201–213. <http://dx.doi.org/10.1007/s101070100263>
19. J. Eckstein, W. Yao, Understanding the convergence of the alternating direction method of multipliers: theoretical and computational perspectives, *Pac. J. Optim.*, **11** (2015), 619–644.
20. D. Gabay, Applications of the method of multipliers to variational inequalities, *Studies in Mathematics and its Applications*, **15** (1983), 299–331. [http://dx.doi.org/10.1016/S0168-2024\(08\)70034-1](http://dx.doi.org/10.1016/S0168-2024(08)70034-1)
21. O. Giladi, B. Rüffer, A Lyapunov function construction for a non-convex Douglas-Rachford iteration, *J. Optim. Theory Appl.*, **180** (2019), 729–750. <http://dx.doi.org/10.1007/s10957-018-1405-3>
22. B. He, X. Yuan, On the $O(1/n)$ convergence rate of the Douglas-Rachford alternating direction method, *SIAM J. Numer. Anal.*, **50** (2012), 700–709. <http://dx.doi.org/10.1137/110836936>
23. B. He, X. Yuan, On the convergence rate of Douglas-Rachford operator splitting method, *Math. Program.*, **153** (2015), 715–722. <http://dx.doi.org/10.1007/s10107-014-0805-x>
24. S. B. Lindstrom, Computable centering methods for spiralling algorithms and their duals with motivations from the theory of Lyapunov functions, *Comput. Optim. Appl.*, **83** (2022), 999–1026. <http://dx.doi.org/10.1007/s10589-022-00413-8>
25. S. B. Lindstrom, B. Sims, Survey: sixty years of Douglas-Rachford, *J. Aust. Math. Soc.*, **110** (2021), 333–370. <http://dx.doi.org/10.1017/S1446788719000570>
26. P. Lions, B. Mercier, Splitting algorithms for the sum of two nonlinear operators, *SIAM J. Numer. Anal.*, **16** (1979), 964–979. <http://dx.doi.org/10.1137/0716071>



©2024 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)