A Bridge Between Past and Present: Exchange and Conditional Gradient Methods are Equivalent

Armin Eftekhari and Andrew Thompson*

May 1, 2018

Abstract

We study an optimization program over nonnegative Borel measures that encourages sparsity in its solution. Efficient solvers for this program are increasingly needed, as it arises when learning from data generated by a “continuum-of-subspaces” model, a recent trend with applications in signal processing, machine learning, and statistics. We prove that the conditional gradient method (CGM) applied to this infinite-dimensional program, as proposed recently in the literature, is equivalent to the exchange method (EM) applied to its Lagrangian dual, which is a semi-infinite program. In doing so, we formally connect such infinite-dimensional programs to the well-established field of semi-infinite programming. On the one hand, the equivalence established in this paper allows us to provide a rate of convergence for EM which is more general than those existing in the literature. On the other hand, this connection and the resulting geometric insights might in the future lead to the design of improved variants of CGM for infinite-dimensional programs, which has been an active research topic. CGM is also known as the Frank-Wolfe algorithm.

1 Introduction

We consider the following affinely-constrained optimization over nonnegative Borel measures:

\[
\begin{align*}
\min_x & \quad L \left( \int_I \Phi(t)x(dt) - y \right) \\
\text{subject to} & \quad \|x\|_{TV} \leq 1 \\
& \quad x \in B_+(I).
\end{align*}
\]

(1)

Here, \(I\) is a compact subset of Euclidean space, \(B_+(I)\) denotes all nonnegative Borel measures supported on \(I\), and

\[\|x\|_{TV} = \int_I x(dt)\]

(2)

is the total variation of measure \(x\), see for example [1]. We are particularly interested in the case where \(L : \mathbb{C}^m \to \mathbb{R}\) is a differentiable loss function and \(\Phi : I \to \mathbb{C}^m\) is a continuous function. Note that Program (1) is an infinite-dimensional problem and that the constraints ensure that the problem is bounded. Program (1) therefore searches for a nonnegative measure on \(I\) that minimizes the loss above, while controlling its total variation. This problem and its closely related variants have received significant attention [2, 3, 4, 5, 6, 7, 8] in signal processing and machine learning, see Section 2 for more details.

It was recently proposed in [3] to solve Program (1) using the celebrated conditional gradient method (CGM) [9], also known as the Frank-Wolfe algorithm, adapted to optimization over nonnegative Borel measures. The CGM algorithm minimizes a differentiable, convex function over a compact convex set, and proceeds by iteratively minimizing linearizations of the objective function over the feasible set, generating a new descent direction in each iteration. The classical algorithm performs a descent step in each new direction generated, while in the fully-corrective CGM, the objective is minimized over the subspace spanned by all previous directions [10]. It is the fully-corrective version of the algorithm which we consider in this paper. It was shown in [3] that, when applied to Program (1), CGM generates a

*AE and AT have contributed equally to this work. AE is with the Alan Turing Institute in London. AT is with the Mathematical Institute, University of Oxford.
sequence of finitely supported measures, with a single parameter value \( t^l \in \mathbb{I} \) being added to the support in the \( l \)th iteration. Moreover, [3] established that the convergence rate of CGM here is \( \mathcal{O} \left( \frac{1}{l} \right) \), where \( l \) is the number of iterations, thereby extending the standard results for finite-dimensional CGM. A full description of CGM and its convergence guarantees can be found in Section 3. 

On the other hand, the (Lagrangian) dual of Program (1) is a finite-dimensional optimization problem with infinitely many constraints, often referred to as a semi-infinite program (SIP), namely

\[
\begin{aligned}
\max_{\lambda, \alpha} & \quad \Re \langle \lambda, y \rangle - L_\circ (-\lambda) - \alpha \\
\text{subject to} & \quad \Re \langle \lambda, \Phi(t) \rangle \leq \alpha, \quad t \in \mathbb{I} \\
& \quad \alpha \geq 0,
\end{aligned}
\]

(3)

where \( \langle \cdot, \cdot \rangle \) denotes the standard Euclidean inner product over \( \mathbb{C}^m \). Above,

\[ L_\circ(\lambda) = \sup_{z \in \mathbb{C}^m} \Re \langle \lambda, z \rangle - L(z) \]

(4)

denotes the Fenchel conjugate of \( L \). As an example, when \( L(\cdot) = \frac{1}{2} \| \cdot \|_2^2 \), it is easy to verify that \( L_\circ = L \). 

For the sake of completeness, we verify the duality of Programs (1) and (3) in Appendix B. Note that the Slater’s condition for the finite-dimensional Program (3) is met and there is consequently no duality gap between the two Programs (1) and (3).

There is a large body of research on SIPs such as Program (3), see for example [11, 12], and we are particularly interested in solving Program (3) with exchange methods. In one instantiation – which for ease we will refer to as the exchange method (EM) – one forms a sequence of nested subsets of the constraints in Program (3), adding in the \( l \)th iteration a single new constraint corresponding to the parameter value \( t^l \in \mathbb{I} \) that maximally violates the constraints. The finite-dimensional problem with these constraints is then solved and the process repeated. Convergence of EM has been established under somewhat general conditions, but results concerning rate of convergence are restricted to more specific SIPs, see Section 4 for a full description of the EM.

**Contribution.** The main contribution of this paper is to establish that, for Program (1) and provided the loss function \( L \) is both strongly smooth and strongly convex, CGM and EM are dual-equivalent. More precisely, the iterates of the two algorithms produce the same objective value and the same finite set of parameters in each iteration; for CGM, this set is the support of the current iterate of CGM and, for EM, this set is the choice of constraints in the dual program. The EM method can also be viewed as a bundle method for Program (3) as discussed in Section 6, and the duality of CGM and bundle methods is well known for finite-dimensional problems. This paper establishes dual-equivalence in the emerging context of optimization over measures on the one hand and the well-established semi-infinite programming on the other hand. On the one hand, the equivalence established in this paper allows us to provide a rate of convergence for EM which is more general than those existing in the literature; see Section 6 for a thorough discussion of the prior art. On the other hand, this connection and the resulting geometric insights might lead to the design of improved variants to CGM, another active research topic [3].

**Outline.** We begin in Section 2 with some motivation, describing the key role of Program (1) in data and computational sciences. Then in Sections 3 and 4, we give a more technical introduction to CGM and EM, respectively. We present the main contributions of the paper in Section 5, establishing the dual-equivalence of CGM and EM for Problems (1) and (3), and deriving the rate of convergence for EM. Related work is reviewed in Section 6 and we conclude in Section 7 with some geometric insights into the inner workings of CGM and EM.

## 2 Motivation

Program (1) has diverse applications in data and computational sciences. In signal processing for example, each \( \Phi(t) \in \mathbb{C}^m \) is an atom and the set of all atoms \( \{ \Phi(t) \}_{t \in \mathbb{I}} \) is sometimes referred to as the dictionary. In radar, for instance, \( \Phi(t) \) is a copy of a known template, arriving at time \( t \). In this context, we are interested in signals that have a sparse representation in this dictionary, namely signals that can be written as the superposition of a small number of atoms. Any such signal \( \hat{y} \in \mathbb{C}^m \) can be written as

\[
\hat{y} = \int_i \Phi(t) \hat{x}(dt),
\]

(5)
where \( \hat{x} \) is a sparse measure, selecting the atoms that form \( \hat{y} \). More specifically,

\[
\hat{x} = \sum_{i=1}^{k} \hat{a}_i \cdot \delta_{\hat{t}_i},
\]

for an integer \( k \), positive amplitudes \( \{\hat{a}_i\}_{i=1}^{k} \), and parameters \( \{\hat{t}_i\}_{i=1}^{k} \subset I \). Here, \( \delta_{\hat{t}_i} \) is the Dirac measure located at \( \hat{t}_i \in I \). We can therefore rewrite (5) as

\[
\hat{y} = \int \Phi(t) \hat{x}(dt) = \sum_{i=1}^{k} \Phi(\hat{t}_i) \cdot \hat{a}_i.
\]

In words, \( \{\hat{t}_i\}_{i=1}^{k} \) are the parameters that construct the signal \( \hat{y} \) and \( I \) is the parameter space. We often receive \( y \in \mathbb{C}^m \), a noisy copy of \( \hat{y} \), and our objective in signal processing is to estimate the hidden parameters \( \{t_i\}_{i=1}^{k} \), given the noisy copy \( y \). See Figure 1 for an example.

![Figure 1](image)

Figure 1: In this numerical example, (a) depicts the measure \( \hat{x} \), see (6). Let \( \phi(t) = e^{-100t^2} \) be a Gaussian window. With the choice of sampling locations \( \{s_j\}_{j=1}^{m} \subset [0, 1] \) and \( \Phi(t) = [\phi(t-s_j)]_{j=1}^{m} \in \mathbb{R}^m \), (b) depicts \( \hat{y} \in \mathbb{R}^m \), see (2). Note that the entries of \( \hat{y} \) are in fact samples of \( \phi(x)(s) = \int_0^1 \phi(t-s) x(dt) \) at locations \( s \in \{s_j\}_{j=1}^{m} \), which forms the red curve in (b). Our objective is to estimate the locations \( \{\hat{t}_i\}_{i=1}^{k} \) from \( \hat{y} \). (Given an estimate of the locations, the amplitudes \( \{\hat{a}_i\}_{i=1}^{k} \) can also be estimated with a simple least-squares program.) This is indeed a difficult task: Even given the red curve \( \phi \cdot \hat{x} \) (from which \( \hat{y} \) is sampled), it is hard to see that there is an impulse located at \( \hat{t}_3 \). Solving Program (1) with \( \|x\|_{TV} \leq b \) for large enough \( b \) uniquely recovers \( x \), as proved in [1]. In this paper, we describe Algorithms 1 and 2 to solve Program (1), and establish their equivalence.

To that end, Program (1) searches for a nonnegative measure \( \hat{x} \) supported on \( I \) that minimizes the loss \( L(\int_0^1 \Phi(t)x(dt) - y) \), while penalizing its sparsity through the total variation constraint \( \|x\|_{TV} \leq 1 \). Under certain conditions on \( \Phi \) and when \( L = \frac{1}{2} \|\cdot\|_2^2 \), a minimizer \( \hat{x} \) of Program (1) is a robust estimate of the true measure \( x \) in the sense that \( d(\hat{x}, \hat{x}) \leq c \cdot L(y - \hat{y}) \) for a known factor \( c \) and in a certain metric \( d \) [1, 13, 6, 14].

The super-resolution problem outlined above is an example of learning under a “continuum-of-subspaces” model, in which data belongs to the union of infinitely-many subspaces. For super-resolution in particular, each subspace corresponds to fixed locations \( \{t_i\}_{i=1}^{k} \). This model is a natural generalization of the “union-of-subspaces” model, which is a central object in compressive sensing [15], wavelets [16], and feature selection in statistics [17]. The use of continuum-of-subspaces models is on the rise as it potentially addresses the drawbacks of the union-of-subspaces models, see for example [18]. As another application of Program (1), \( y \) represents the training labels in machine learning or, in the classic moments problem, \( y \) collects the moments of an unknown distribution. Various other examples are given in [3].
Note that Program (1) is an infinite-dimensional problem as the search is over all measures supported on $\mathbb{I}$. It is common in practice to restrict the support of $x$ to a uniform grid on $\mathbb{I}$, say $\{\tau_i\}_{i=1}^n \subset \mathbb{I}$, so that $x = \sum_{i=1}^n a_i \delta_{\tau_i}$ for nonnegative amplitudes $\{a_i\}_{i=1}^n$. Let $a \in \mathbb{R}_n^+$ be the vector formed by the amplitudes and concatenate the vectors $\{\Phi(t_i)\}_{i=1}^n \subset \mathbb{C}^{m}$ to form a (often very flat) matrix $\Phi \in \mathbb{C}^{m \times n}$. Then we may rewrite Program (1) as

\[
\begin{align*}
\min_a & \quad L(\Phi \cdot a - y) \\
\text{subject to} & \quad \langle 1_n, a \rangle \leq 1 \\
& \quad a \geq 0,
\end{align*}
\]

where $1_n \in \mathbb{R}^n$ is the vector of all ones. When $L(\cdot) = \frac{1}{2}\|\cdot\|^2$ in particular, Program (7) reduces to the well-known nonnegative Lasso [19]. The problem with this “gridding” approach is that there is often a mismatch between the atoms $\{\Phi(t_i)\}_{i=1}^n$ that are present in $\hat{y}$ and the atoms listed in $\Phi$, namely $\{\Phi(t_i)\}_{i=1}^n$. As a result, $\hat{y}$ often does not have a sufficiently sparse representation in $\Phi$. In the context of signal processing, this problem is known as the “frequency leakage”, see Figure 2. Countering the frequency leakage by excessively increasing the grid size $n$ leads to numerical instability of the recovery procedure because the (nearby) columns of $\Phi$ become increasingly similar or coherent. This discussion encourages directly solving the infinite-dimensional Program (1) without discretization; it is this direction that is pursued in this work and in [3, 20, 21, 22, 23, 24].

![Figure 2: (a) depicts a translated Gaussian window, namely $\phi(t-t_1) = e^{-100(t-t_1)^2}$ for translation $t_1 \in [0, 1]$. Equivalently, $\phi(t - t_1) = (\phi * \delta_{t_1})(t)$, as represented in (b). On the other hand, (c) shows the coefficients of the least-squares approximation of the translated window $\phi(t - t_1)$ in the dictionary $\{\phi(t - i/N)\}_{i=1}^N$ for $N = 66$. By comparing (b) and (c), we observe that $\phi(t - t_1)$ loses its sparse representation after gridding. This problem is alleviated by choosing finer and finer uniform grids for the interval $[0, 1]$ but at the cost of numerical instability in practice. This observation encourages directly solving the infinite-dimensional Program (1), as opposed to discretizing it.](image-url)
3  Conditional Gradient Method

In this section and the next one, we review two algorithms for solving Program (1). The first one is the conditional gradient method [9], a popular first-order algorithm for constrained optimization. The popularity of CGM partly stems from the fact that it is projection free, unlike projected gradient descent for example which requires projection onto the feasible set in every iteration.

More specifically, CGM solves the general constrained optimization problem

$$\min_{x \in \mathcal{F}} f(x)$$

where $f(x)$ is a differentiable function and $\mathcal{F}$ is a compact convex set. Given the current iterate $x^{l-1}$, CGM finds a search direction $s^l$ which minimizes the linearized objective function, namely $s^l$ is a solution to

$$\min_{s \in \mathcal{F}} f(x^{l-1}) + \langle s - x^{l-1}, \nabla f(x^{l-1}) \rangle$$

Note that we may remove the terms independent of $s$ without changing the minimizers of Program (8). The classical CGM algorithm then takes a step along the direction $s^l - x^{l-1}$, namely

$$x^l = x^{l-1} + \gamma^l \cdot (s^l - x^{l-1}),$$

for some step size $\gamma^l \in (0, 1]$. In a similar spirit, fully-corrective CGM chooses $x^l$ within the convex hull of all previous update directions [10]. To be specific, fully-corrective CGM (which we simply refer to as CGM henceforth) sets $x^l$ to be a minimizer of

$$\begin{cases} 
\min f(x) \\
\text{subject to } x \in \text{conv}(s^1, \ldots, s^l).
\end{cases}$$

In the context of sparse regression and classification, CGM is particularly appealing because it produces sparse iterates. Indeed, because the objective function in Program (8) is linear in $s$, there always exist a minimizer of Program (8) that is an extreme point of the feasible set $\mathcal{F}$. In our case,

$$\mathcal{F} = \{x \in B_1(I) : \|x\|_{TV} \leq 1\},$$

and any extreme point of $\mathcal{F}$ is therefore of the form $\delta_t$ with $t \in \mathbb{I}$. It follows that each iterate $x^l$ of CGM is at most $l$-sparse, namely supported on a subset of $I$ of size at most $l$.

In light of the discussion above, CGM applied to (1) is summarized in Algorithm 1. Note that we might interpret Algorithm 1 as follows. Let $x_p$ be a minimizer of Program (1), supported on the index set $T_p \subset I$. If an oracle gave us the correct support $T_p$, we could have recovered $x_p$ by solving Program (1) restricted to the support $T_p$, rather than $I$. Since we do not have access to such an oracle, at iteration $l$, Algorithm 1

1. finds an atom $\Phi(t^l)$ that reduces the objective of Program (1) the most, namely an atom that is least correlated with the gradient at the current residual $\int I \Phi(t)(\tau)x^{l-1} d\tau - y$, and then

2. adds $t^l$ to the support.

When $L(\cdot) = \frac{1}{2}\|\cdot\|_2^2$, in particular, Algorithm 1 reduces to the well-known orthogonal matching pursuit (OMP) for sparse regression [25], adapted to measures.

The convergence rate of CGM has been established in [26, 3], and is reviewed next for the sake of completeness. For the rest of this paper, let us assume that $L$ is both strongly smooth and strongly convex, namely there exists $\gamma \geq 1$ such that

$$\frac{\|x - x'\|^2}{2\gamma} \leq L(x) - L(x') - \langle x - x', \nabla L(x') \rangle \leq \frac{\gamma}{2} \|x - x'\|^2,$$  \hspace{1cm} (11)

for every $x, x' \in \mathbb{C}^m$. In words, $L$ can be approximated by quadratic functions at any point of its domain. For example, $L(\cdot) = \frac{1}{2}\|\cdot\|_2^2$ satisfies (11) with $\gamma = 1$. Let us also define

$$r := \max_{t \in I} \|\Phi(t)\|_2.$$  \hspace{1cm} (12)

The convergence rate of Algorithm 1 is given by the following result, which is similar to the result originally given in [3], except that we replace the curvature condition in [3] with the strongly smooth and convex assumption in (11), see Appendix A for the proof.
Algorithm 1: CGM for solving Program (1)

Input: Compact set \( I \), continuous function \( \Phi : I \to \mathbb{C}^m \), differentiable function \( L : \mathbb{C}^m \to \mathbb{R} \), vector \( y \in \mathbb{C}^m \), and tolerance \( \epsilon \geq 0 \).

Output: Nonnegative measure \( \hat{x} \) supported on \( I \).

Initialize: Set \( l = 1 \), \( T^0 = \emptyset \), and \( x^0 \equiv 0 \).

While \( \| \nabla L(\int_I \Phi(\tau)x^{l-1}(d\tau) - y) \|_2 > \epsilon \), do

1. Let \( t^l \) be a minimizer of
   \[
   \min_{t \in I} \left\langle \Phi(t), \nabla L \left( \int_I \Phi(\tau)x^{l-1}(d\tau) - y \right) \right\rangle. \tag{9}
   \]

2. Set \( T^l = T^{l-1} \cup \{ t^l \} \).

3. Let \( x^l \) be a minimizer of
   \[
   \min_{x} \begin{cases} 
   L \left( \int_I \Phi(t)x(dt) - y \right) \\
   \text{subject to} \\
   \|x\|_{TV} \leq 1 \\
   \text{supp}(x) \subseteq T^l \\
   x \in B_+(I).
   \end{cases} \tag{10}
   \]

Return: \( \hat{x} = x^l \).

Proposition 1. (Convergence rate of Algorithm 1) For \( \gamma \geq 1 \), suppose that \( L \) satisfies (11).\footnote{Strictly speaking, strong convexity is not required for Proposition 1. That is, the far left term in (11) can be replaced with zero.}

Suppose that Program (9) is solved to within an accuracy of \( 2\gamma r^2 \epsilon \) in every iteration of Algorithm 1. Let \( v_p \) be the optimal value of Program (1). Let also \( v^l_{\text{CGM}} \) be the optimal value of Program (10). Then, at iteration \( l \geq 1 \), it holds that

\[
 v^l_{\text{CGM}} - v_p \leq \frac{4\gamma r^2(1 + \epsilon)}{l + 2}. \tag{13}
\]

Assuming that \( L \) satisfies (11), it is not difficult to verify that Program (1) is a convex and strongly smooth problem. Therefore CGM achieves the same convergence rate of \( 1/l \) that the projected gradient descent achieves for such problems [27].

4 Exchange Method

EM is a well-known algorithm to solve SIPs and, in particular, Program (3). In every iteration, EM adds a new constraint out of the infinitely many in Program (3), thereby forming an increasingly finer discretization of \( I \) as the algorithm proceeds. The new constraints are added where needed most, namely at \( t \in I \) that maximally violates the constraints in Program (3). In other words, a new constraint is added at \( t \in I \) that maximizes the “polynomial” Re\( \langle \lambda^l, \Phi(t) \rangle \), where \((\lambda^l, \alpha^l)\) is the current iterate. EM is summarized in Algorithm 2.

Let \((\lambda_d, \alpha_d)\) be a maximizer of Program (3). Also assume that \( T_d \subset I \) is the set of active constraints in Program (3), namely Re\( \langle \lambda_d, \Phi(t) \rangle = \alpha_d \) for every \( t \in T_d \). If an oracle tells us what the active constraints \( T_d \) are in advance, we can simply find the optimal pair \((\lambda_d, \alpha_d)\) by solving Program (3) with \( T_d \) instead of \( I \). Alas, such an oracle is not at hand. Instead, at iteration \( l \), Algorithm 2

1. solves Program (3) restricted to the current constraints \( T^{l-1} \) to find \((\lambda^l, \alpha^l)\), and then
2. if \((\lambda^l, \alpha^l)\) does not violate the constraints of Program (3) on \( I \setminus T^{l-1} \), the algorithm terminates because it has found a maximizer of Program (3), namely \((\lambda^l, \alpha^l)\). Otherwise, EM adds to its support a point \( t^l \in I \) that maximally violates the constraints of Program (3).

Having reviewed both CGM and EM for solving Program (1) in the past two sections, we next establish their equivalence.
Algorithm 2 EM for solving Program (3).

\textbf{Input:} Compact set $\mathbb{I}$, continuous functions $\Phi : \mathbb{I} \rightarrow \mathbb{C}^m$ and $w : \mathbb{I} \rightarrow \mathbb{R}^+$, differentiable function $L : \mathbb{C}^m \rightarrow \mathbb{R}$, $y \in \mathbb{C}^m$ and tolerance $\epsilon \geq 0$.

\textbf{Output:} Vector $\hat{\lambda} \in \mathbb{C}^m$ and $\hat{\alpha} \geq 0$.

\textbf{Initialize:} $l = 1$ and $T^0 = \emptyset$.

\textbf{While} $\max_{t \in \mathbb{I}} \Re \langle \lambda^l, \Phi(t) \rangle > \alpha^l + \epsilon$, \textbf{do}

1. Let $(\lambda^l, \alpha^l)$ be a maximizer of

$$\begin{aligned}
\max_{\lambda, \alpha} & \quad \Re \langle \lambda, y \rangle - L_o (-\lambda) - \alpha \\
\text{subject to} & \quad \Re \langle \lambda, \Phi(t) \rangle \leq \alpha \quad t \in T^{l-1} \\
& \quad \alpha \geq 0,
\end{aligned} \quad (14)$$

where $L_o$ is the Fenchel conjugate of $L$, see (4).

2. Let $t^l$ be the solution to

$$\max_{t \in \mathbb{I}} \Re \langle \lambda^l, \Phi(t) \rangle. \quad (15)$$

3. Set $T^l = T^{l-1} \cup t^l$.

\textbf{Return:} $(\hat{\lambda}, \hat{\alpha}) = (\lambda^l, \alpha^l)$.

5 \hspace{1em} \textbf{Equivalence of CGM and EM}

Cgm solves Program (1) and adds a new atom in every iteration whereas EM solves the dual problem (namely Program (3)) and adds a new active constraint in every iteration, and both algorithms do so “greedily”. Their connection goes deeper: Consider Program (1) restricted to a finite support $T \subset \mathbb{I}$, namely the program

$$\begin{aligned}
\min_x & \quad L \left( \int_{\mathbb{I}} \Phi(t)x(dt) - y \right) \\
\text{subject to} & \quad \|x\|_{TV} \leq 1 \\
& \quad \text{supp}(x) \subseteq T \\
& \quad x \in B_+(\mathbb{I}).
\end{aligned} \quad (16)$$

The dual of Program (16) is

$$\begin{aligned}
\max_{\lambda, \alpha} & \quad \Re \langle \lambda, y \rangle - L_o (-\lambda) - \alpha \\
\text{subject to} & \quad \Re \langle \lambda, \Phi(t) \rangle \leq \alpha \quad t \in T \\
& \quad \alpha \geq 0.
\end{aligned} \quad (17)$$

Indeed, Program (17) is the restriction of Program (3) to $T$. Note that the complementary slackness forces any minimizer of Program (16) to be supported on the set of active constraints of Program (17). Note also that Programs (16) and (17) appear respectively in CGM and EM but with different support sets. The following result states that CGM and EM are in fact equivalent algorithms to solve Program (1), see Appendix C for the proof.

\textbf{Proposition 2. (Equivalence of Algorithms 1 and 2)} For $\gamma \geq 1$, suppose that $L$ satisfies (11). Assume also that CGM and EM update their supports according to the same rule, e.g., selecting the smallest solutions if $\mathbb{I} \subset \mathbb{R}$. Then CGM and EM are equivalent in the sense that $T^l_{\text{CGM}} = T^l_{\text{EM}}$ for every iteration $l \geq 0$. Here, $T^l_{\text{CGM}}$ and $T^l_{\text{EM}}$ (both subsets of $\mathbb{I}$) are the support sets of CGM and EM at iteration $l$, respectively.

Furthermore, $v^l_{\text{CGM}} = v^l_{\text{EM}}$, where $v^l_{\text{CGM}}$ and $v^l_{\text{EM}}$ denote the optimal values of Programs (10) and (14) in CGM and EM, respectively.
The above equivalence allows us to carry convergence results from one algorithm to another. In particular, the convergence rate of CGM in Proposition 1 determines the convergence rate of EM, as the following result indicates, see Appendix D for the proof.

**Proposition 3. (Convergence of Algorithm 2)** For $\gamma \geq 1$, suppose that $L$ satisfies (11). Recall the definition of $r$ in (12) and, for $\epsilon \geq 0$, suppose that Program (15) is solved to within an accuracy of $2\gamma r^2 \epsilon$ in every iteration. Let $v_0$ be the optimal value of Program (3) and $(\lambda_d, \alpha_d)$ be its unique maximizer. Likewise, let $v_{EM}$ be the optimal value of Program (14). At iteration $l \geq 1$, it then holds that

$$v_{EM} - v_0 \leq \frac{4\gamma r^2 (1 + \epsilon)}{l + 2},$$

$$||\lambda^l - \lambda_d||_2 \leq \sqrt{\frac{8\gamma^2 r^4 (1 + \epsilon)}{l + 2}},$$

$$|\alpha^l - \alpha_d| \leq \sqrt{\frac{8\gamma^2 r^4 (1 + \epsilon)}{l + 2}}.$$  \hfill (18) \hfill (19)

Furthermore, it holds that

$$\max_{t \in [1]} \langle \lambda^t, \Phi(t) \rangle \leq \alpha_d + \sqrt{\frac{8\gamma^2 r^4 (1 + \epsilon)}{l + 2}}.$$  \hfill (20)

namely the iterates $\{\lambda^1\}$ of Algorithm 2 gradually become feasible for Program (3).

Proposition 3 states that the Program (3), which has infinitely many constraints, can be solved as fast as a smooth convex program with finitely many constraints. More specifically, it is not difficult to verify that the objective function of Program (3) is convex and strongly smooth, see Section 7. Then, (18) states that EM solves Program (3) at the rate of $1/l$, the same rate at which the projected gradient descent solves a finite-dimensional problem under the assumptions of convexity and strong smoothness [27]. This is perhaps remarkable given that Program (3) has infinitely many constraints. Note however that the convergence of the iterates $\{(\lambda^t, \alpha^t)\}_l$ of EM to the unique maximizer $(\lambda_d, \alpha_d)$ of Program (3) is much slower as given in (20), namely at the rate of $1/\sqrt{l}$.

We remark that Proposition 3 is novel in providing a rate of convergence for EM for a general class of nonlinear SIPs, whereas the literature on SIP only gives rates of convergence for specific problems. See Section 6 for a more detailed literature review.

## 6 Related Work

The conditional gradient method (CGM), also known as the Frank-Wolfe algorithm, is one of the earliest algorithms for constrained optimization [9]. The version of the algorithm considered in this paper is the fully-corrective Frank-Wolfe algorithm, also known as the simplicial decomposition algorithm, in which the objective is optimized over the convex hull of all previous atoms [10, 28]. The algorithm was proposed for optimization over measures, the context considered in this paper, in [3].

Semi-infinite programs (SIPs) have been much studied, both theoretically in terms of optimality conditions and duality theory, and algorithmically in terms of design and analysis of numerical methods for their solution. We refer the reader to the review articles [11] and [12] for further background.

Exchange methods are one of the three main families of popular methods for the numerical solution of SIPs, with the other two being discretization methods and localization methods. In discretization methods, the infinite constraints are replaced by a finite subset thereof and the resulting finite dimensional problem is solved as an approximation of the SIP. In localization methods, a sequence of local (usually quadratic) approximations to the problem are solved. Global convergence of discretization methods has been proved for linear SIPs [29], but no general convergence result exists for nonlinear SIPs [11]. Global convergence of exchange methods has been proved for general SIPs [11], but to the authors’ best knowledge there is no general proof of rate of convergence, except for more specific problems. For localization methods, local superlinear convergence has been proved assuming strong sufficient second-order optimality conditions, which do not hold for all SIPs [30]. The guarantees extend to global convergence of more sophisticated algorithms which combine localization methods with global search, see [12, Section 7.3] and references therein. We refer the reader to [11, 12] for more details on existing convergence analysis of SIPs. Against this background, the convergence rate of the EM, established here in Proposition 3 for a wide class of nonlinear SIPs, represents a new contribution.
A connection can also be made between EM and the bundle method for unconstrained optimization [28]. In the bundle method, for convex and smooth function \( u \) and convex (but not necessarily smooth) function \( v \), the function \( u + v \) is minimized by generating the sequence of iterates \( \{\lambda^i\} \) specified as

\[
\lambda^i \in \arg \min_{\lambda} \left( u(\lambda) + \max_{1 \leq i \leq l} \Re(\langle \lambda, \partial v(\lambda^i) \rangle) \right),
\]

where \( \partial v(\lambda^i) \) is a subgradient of \( v \) at \( \lambda^i \). To establish the connection with EM, note that Program (3) can be rewritten as the unconstrained problem

\[
\max_{\lambda \in \mathbb{C}^m} \Re(\lambda, y) - L_0(-\lambda) - \max_{t \in T} \langle \lambda, \Phi(t) \rangle.
\]

Setting \( u(\lambda) = -\Re(\lambda, y) + L_0(-\lambda) \) and \( v(\lambda) = \max_{t \in T} \langle \lambda, \Phi(t) \rangle \), and then applying the bundle method produces the iterates

\[
\lambda^i \in \arg \max_{\lambda \in \mathbb{R}} \Re(\lambda, y) - L_0(-\lambda) - \max_{t \in T^l} \langle \lambda, \Phi(t) \rangle \equiv \text{Program (14)}.
\]

That is, EM and the bundle method, applied to Program (3), produce the same iterates. The dual equivalence of CGM and the bundle method has previously been noted for various finite dimensional problems, see for example [28, 31, 32, 33]. However, we are not aware of any extension of these finite-dimensional results to SIPs and their dual problem of optimization over Borel measures. In this sense, the equivalence established in Proposition 2 is novel.

7 Geometric Insights

This section collects a number of useful insights about CGM/EM. Recalling the setup in Section 1, here we study Program (1) in the case where \( \mathbb{I} \subset \mathbb{R} \) is a compact subset of the real line and the function \( \Phi : \mathbb{I} \to \mathbb{C}^m \) is a Chebyshev system [34].

**Definition 4. (Chebyshev system)** Consider a compact interval \( \mathbb{I} \subset \mathbb{R} \) and a continuous function \( \Phi : \mathbb{I} \to \mathbb{C}^m \). Then \( \Phi \) is a Chebyshev system if \( \{\Phi(t_i)\}_{i=1}^m \subset \mathbb{C}^m \) are linearly independent vectors for any choice of distinct \( \{t_i\}_{i=1}^m \subset \mathbb{I} \).

Chebyshev systems are widely used in classical approximation theory and generalize the notion of ordinary polynomials. Many functions form Chebyshev systems, for example sinusoids or translated copies of the Gaussian window, and we refer the interested reader to [34, 35, 1] for more on their properties and applications. Let \( C_1 \subset \mathbb{C}^m \) be the convex hull of \( \{\Phi(t_i)\}_{i\in \mathbb{I}} \cup \{0\} \), namely

\[
C_1 := \left\{ \int_\mathbb{I} \Phi(t)x(dt) : x \in B_+(\mathbb{I}), \|x\|_{TV} \leq 1 \right\}.
\]

Note that \( \{x \in B_+(\mathbb{I}) : \|x\|_{TV} \leq 1\} \) is a compact set. Then, by the continuity of \( \Phi \) and with an application of the dominated convergence theorem, it follows that \( C_1 \) is a compact set too. Since \( \Phi \) is by assumption a Chebyshev system, \( C_1 \subset \mathbb{C}^m \) is in fact a convex body, namely a compact convex set with non-empty interior. Introducing \( z = \int_\mathbb{I} \Phi(t)x(dt) \), we note that Program (1) is equivalent to the program

\[
\min_{z \in C_1} L(z - y).
\]

The compactness of \( C_1 \) and the strong convexity of \( L \) in (11) together imply that Program (25) has a unique minimizer \( y_p \in C_1 \), which can be written as \( y_p = \int_\mathbb{I} \Phi(t)x_p(dt) \), where \( x_p \) itself is a minimizer of Program (1). For example, when \( L(t) = \frac{1}{2} \| \cdot \|_2^2 \) Program (25) projects \( y \) onto \( C_1 \). That is, \( y_p \) is the orthogonal projection of \( y \) onto the convex set \( C_1 \).

Given the equivalence of Programs (1) and (25), we might say that solving Program (1) “denoises” the signal \( y \) from a signal processing viewpoint, in the sense that it finds a nearby signal \( y_p = \int_\mathbb{I} \Phi(t)x_p(dt) \) that has a sparse representation in the dictionary \( \{\Phi(t)\}_{t \in \mathbb{I}} \). To be more specific, by Carathéodory’s theorem [36], every \( y_p \in C_1 \) can be written as a convex combination of at most \( m \) atoms of the dictionary \( \{\Phi(t)\}_{t \in \mathbb{I}} \). On the other hand, the Chebyshev assumption on \( \Phi \) implies that \( \{\Phi(t)\}_{t \in \mathbb{I}} \) are the extreme points of \( C_1 \) [34, Chapter II]. Hence, an extreme point of \( C_1 \) is a point in \( C_1 \) that cannot be written as a convex combination of other points in \( C_1 \). It then follows that this atomic

\[\text{Note that Definition 4 is slightly different from the standard one in [34] which requires } \Phi \text{ to be real-valued.}\]
decomposition of \( y_p \) is unique, and \( x_p \) is necessarily \( m \)-sparse. We may note the analogous result in the finite-dimensional case. Indeed, the Lasso problem is known to have a unique solution whose sparsity is no greater than the rank of the measurement matrix, provided the columns of the measurement matrix are in general position [37].

At iteration \( I \) of CGM, let \( C^I \subseteq \mathbb{C}^m \) be the convex hull of \( \{ \Phi(t) \}_{t \in T_I} \cup \{ 0 \} \), namely

\[
C^I := \left\{ \sum_{t \in T_I} \Phi(t) \cdot a_t : \sum_{t \in T_I} a_t \leq 1, a_t \geq 0, \forall t \in T^I \right\}.
\]

(26)

Similar to the argument above, we observe that Program (10) is equivalent to

\[
\min_{z \in C^I} L(z - y)
\]

(27)

As with Program (25), Program (27) has a unique minimizer \( y^I \in C^I \) that satisfies \( y^I = \int_{T_I} \Phi(t)x^I(dt) \) and \( x^I \) is a minimizer of Program (10). By Carathéodory’s theorem again, \( x^I \) is at most \( m \)-sparse. In other words, there always exists an \( m \)-sparse minimizer \( x^I \) to Program (10); iterates of CGM are always sparse and so are the iterates of EM by their equivalence in Proposition 2. In addition, note that the chain \( C^1 \subseteq C^2 \subseteq \cdots \subseteq C^I \) provides a sequence of increasingly better approximations to \( C \). CGM eventually terminates when \( y_p = y^I \in C^I \subseteq C \), which happens as soon as \( C^I \) contains the face of \( C \) to which \( y^I \) belongs. It is however common to use different stopping criteria to terminate CGM when \( y^I \) is sufficiently close to \( y_p \).

Let us now rewrite Program (3) in a similar way. First let \( C_{\lambda,0} \subseteq \mathbb{C}^m \) be the polar of \( C \), namely

\[
C_{\lambda,0} = \{ \lambda : \Re \langle \lambda, z \rangle \leq 1, \forall z \in C \} = \{ \lambda : \Re \langle \lambda, \Phi(t) \rangle \leq 1, \forall t \in I \},
\]

where the second identity follows from the definition of \( C \). Let also \( g_{C_{\lambda,0}} = \gamma_C \) denote the gauge function associated with \( C_{\lambda,0} \) and the support function associated with \( C \), respectively [38]. That is, for \( \lambda \in \mathbb{C}^m \), we define

\[
g_{C_{\lambda,0}}(\lambda) := \begin{cases}
\min_{\alpha} \alpha \\
\text{subject to} \quad \lambda \in \alpha \cdot C_{\lambda,0} = \max_{t \in I} \Re \langle \lambda, \Phi(t) \rangle = \max_{z \in C} \Re \langle \lambda, z \rangle := \gamma_C(\lambda),
\end{cases}
\]

(28)

where \( \alpha \cdot C_{\lambda,0} = \{ \alpha \cdot \lambda : \lambda \in C_{\lambda,0} \} \). In words, \( g_{C_{\lambda,0}}(\lambda) = \gamma_C(\lambda) \) is the smallest \( \alpha \) at which the inflated “ball” \( \alpha \cdot C_{\lambda,0} \) first reaches \( \lambda \). By usual convention, the optimal value above is set to infinity when the problem is infeasible, namely when the ray that passes through \( \lambda \) does not intersect \( C_{\lambda,0} \). It is also not difficult to verify that \( g_{C_{\lambda,0}} = \gamma_C \) is a positively-homogeneous convex function. Using (28), we may rewrite Program (3) as

\[
\begin{cases}
\max_{\lambda, \alpha} L_\circ (-\lambda) + \Re \langle \lambda, y \rangle - \alpha \\
\text{subject to} \quad \lambda \in \alpha \cdot C_{\lambda,0} \quad \equiv \max_{\lambda \in \mathbb{C}^m} \Re \langle \lambda, y \rangle - L_\circ (-\lambda) - g_{C_{\lambda,0}}(\lambda).
\end{cases}
\]

(29)

By the assumption in (11), \( L \) is strongly smooth and consequently \( L_\circ \) is strongly convex [27]. Therefore Program (3) has a unique maximizer, which we denote by \( (\lambda_d, \alpha_d) \). The optimality of \( (\lambda_d, \alpha_d) \) also immediately implies that

\[
\alpha_d = g_{C_{\lambda,0}}(\lambda_d).
\]

(30)

Thanks to Proposition 2, we likewise define the polar of \( C^{\lambda-1} \) and the corresponding gauge function to rewrite the main step of EM in Algorithm 2, namely

\[
\begin{cases}
\max_{\lambda, \alpha} L_\circ (-\lambda) + \Re \langle \lambda, y \rangle - \alpha \\
\text{subject to} \quad \lambda \in \alpha \cdot C^{\lambda-1}_{\circ} \quad \equiv \max_{\lambda \in \mathbb{C}^m} \Re \langle \lambda, y \rangle - L_\circ (-\lambda) - g_{C^{\lambda-1}}(\lambda).
\end{cases}
\]

(31)

and the unique minimizer of the above three programs is \( (\lambda^*, \alpha^*) \), where the uniqueness again comes from the strong convexity of \( L_\circ \). Similar to (30), the optimality of \( (\lambda^*, \alpha^*) \) immediately implies that

\[
\alpha^* = g_{C_{\lambda^*}}(\lambda^*).
\]

(32)
It is not difficult to verify that
\[ C_1 \supseteq C_2 \supseteq \cdots \supseteq C_r, \quad (33) \]
That is, as EM progresses, \( C^t \) gradually “zooms into” \( C_r \). As with CGM, EM eventually terminates as soon as \( C^t \) includes the face of \( C_r \) to which \( \lambda_d/\alpha_d \) belongs, at which point \( (\lambda^t, \alpha^t) = (\lambda_d, \alpha_d) \). In light of the argument in Appendix C, in every iteration, we also have that
\[ \langle y^t, \lambda^t/\alpha^t \rangle = 1, \quad (34) \]
namely the pair \( (y^t, \lambda^t/\alpha^t) \in C^t \times C^t \) satisfies the generalized Holder inequality \( g_{C^t} \cdot g_{C^t} \leq 1 \) with equality \([38]\). Here, \( g_{C^t} \) and \( g_{C^t} \) are the gauge functions of \( C^t \) and \( C^t \), respectively. It is worth pointing out that, with the choice of \( L(\cdot) = L_\alpha(\cdot) = \frac{1}{2} \| \cdot \|^2_2 \), the maximizer of (31) is the same as the (unique) minimizer of
\[ \min_{\lambda \in C^m} \frac{\| \lambda - y \|^2_2}{2} + g_{C^t}^{-1}(\lambda), \]
which might be interpreted as a generalization of Lasso and other standard tools for sparse denoising \([17]\). That is, each iteration of CGM/EM can be interpreted as a simple denoising procedure.

Acknowledgements

The authors are grateful to Mike Wakin, Jared Tanner, Mark Davenport, Greg Ongie, Stéphane Chrétien, and Martin Jaggi for their helpful feedback and comments. AE is supported by the Alan Turing Institute under the EPSRC grant EP/N510129/1 and also by the Turing Seed Funding grant SF019.

References

[1] A. Thompson B. Toader H. Tyagi A. Eftekhari, J. Tanner. Sparse non-negative super-resolution — simplified and stabilised. arXiv preprint arXiv:1804.01490, 2018.

[2] G. Schiebinger, E. Robeva, and B. Recht. Superresolution without separation. Information and Inference, page iax006, 2017.

[3] N. Boyd, G. Schiebinger, and B. Recht. The alternating descent conditional gradient method for sparse inverse problems. SIAM Journal on Optimization, 27(2):616–639, 2017.

[4] E. Candès and C. Fernandez-Granda. Super-resolution from noisy data. Journal of Fourier Analysis and Applications, 19(6):1229–1254, 2013.

[5] E. Candès and C. Fernandez-Granda. Towards a mathematical theory of super-resolution. Communications on Pure and Applied Mathematics, 67(6):906–956, 2014.

[6] Y. De Castro and F. Gamboa. Exact reconstruction using Beurling minimal extrapolation. Journal of Mathematical Analysis and applications, 395(1):336–354, 2012.

[7] C. Fernandez-Granda. Support detection in super-resolution. In Proceedings of the 10th International Conference on Sampling Theory and Applications, 2013.

[8] Q. Denoyelle, V. Duval, and G. Peyré. Support recovery for sparse deconvolution of positive measures. Journal of Fourier Analysis and its Applications, 23(5):1153–1194, 2017.

[9] M. Frank and P. Wolfe. An algorithm for quadratic programming. Naval Research Logistics Quarterly, 3:95–110, 1956.

[10] C. Holloway. An extension of the Frank and Wolfe method of feasible directions. Mathematical Programming, 6(1):14–27, 1974.

[11] R. Hettich and K. Kortanek. Semi-infinite programming: theory, methods and applications. SIAM Review, 35(3):380–429, 1993.

[12] M. Lopez and G. Still. Semi-infinite programming. European Journal of Operational Research, 180(2):491–518, 2005.

[13] V. Duval. A characterization of the non-degenerate source condition in super-resolution. arXiv preprint arXiv:1712.06373, 2017.

[14] E.J. Candès and C. Fernandez-Granda. Towards a mathematical theory of super-resolution. Communications on Pure and Applied Mathematics, 67(6):906–956, 2014.
[15] E. Candès and M. Wakin. An introduction to compressive sampling. *IEEE signal processing magazine*, 25(2):21–30, 2008.

[16] S. Mallat. *A Wavelet Tour of Signal Processing: The Sparse Way*. Elsevier Science, 2008.

[17] T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical Learning with Sparsity: The LASSO and Generalizations*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. CRC Press, 2015.

[18] Z. Zhu and M. Wakin. Approximating sampled sinusoids and multiband signals using multiband modulated DPSS dictionaries. *Journal of Fourier Analysis and Applications*, 23(6):1263–1310, 2017.

[19] M. Slawski and M. Hein. Non-negative least squares for high-dimensional linear models: Consistency and sparse recovery without regularization. *Electronic Journal of Statistics*, 7:3004–3056, 2013.

[20] V. Duval and G. Peyré. Sparse spikes super-resolution on thin grids II: the continuous basis pursuit. *Inverse Problems*, 33(9):095008, 2017.

[21] V. Duval and G. Peyré. Sparse spikes super-resolution on thin grids I: the lasso. *Inverse Problems*, 33(5):055008, 2017.

[22] A. Eftekhari, J. Romberg, and M.B. Wakin. Matched filtering from limited frequency samples. *IEEE Transactions on Information Theory*, 59(6):3475–3496, 2013.

[23] A. Eftekhari and M.B. Wakin. Supplementary material for “Greed is super: A fast algorithm for super-resolution”. Technical report, Colorado School of Mines, 2015.

[24] A. Eftekhari and M.B. Wakin. Greed is super: A new iterative method for super-resolution. In *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2013.

[25] J. Tropp and A. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on information theory*, 53(12):4655–4666, 2007.

[26] M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *ICML (1)*, pages 427–435, 2013.

[27] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Applied Optimization. Springer US, 2013.

[28] D. Bertsekas and H. Yu. A unifying polyhedral approximation framework for convex optimization. *SIAM Journal on Optimization*, 21(1):333–360, 2011.

[29] M. Goberna and M. Lopez. *Linear semi-infinite optimization*. Wiley, Chichester, 1998.

[30] R. Fontecilla, T. Steihaug, and R. Tapia. A convergence theory for a class of quasi-Newton methods for constrained optimization. *SIAM Journal on Numerical Analysis*, 24(5):1133–1151, 1987.

[31] F. Bach. Learning with submodular functions: A convex optimization perspective. *Foundations and Trends in Machine Learning*, 6(2-3):145–373, 2013.

[32] F. Bach. Duality between subgradient and conditional gradient methods. *SIAM Journal on Optimization*, 25(1):115–129, 2015.

[33] S. Lacoste-Julien, M. Jaggi, M. Schmidt, and P. Pletscher. Block-coordinate Frank-Wolfe optimization for structural SVMs. In *international Conference on Machine Learning*, 2013.

[34] S. Karlin and W. Studden. *Tchebycheff systems: with applications in analysis and statistics*. Pure and applied mathematics. Interscience Publishers, 1966.

[35] M. Krein and A. Nudelman. *The Markov moment problem and extremal problems*. American Mathematical Society, 1977.

[36] A. Barvinok. *A Course in Convexity*. Graduate studies in mathematics. American Mathematical Society, 2002.

[37] R. Tibshirani. The lasso problem and uniqueness. *Electronic Journal of Statistics*, 7, 2012.

[38] R. Rockafellar. *Convex Analysis*. Princeton Landmarks in Mathematics and Physics. Princeton University Press, 2015.
A Proof of Proposition 1

Recall the compact convex set $C_1 \subset \mathbb{C}^m$, defined in (24). By Carathéodory’s theorem [36], every point in $C_1$ can be written as a combination of at most $m + 1$ points within $\{\Phi(t)\}_{t \in I} \cup \{0\}$, allowing us to write that

$$C_1 = \left\{ \sum_{t \in I} \Phi(t) \cdot a_t : \sum_{t \in I} a_t \leq 1, \ a_t \geq 0, \ \forall t \in I \right\} \subset \mathbb{C}^m,$$

for any finite subset $\Gamma \subset I$. Recall also the equivalent form of Program (1) given by Program (25), and let $v_p$ be the optimal value of both these programs. Similarly, recall $C^l \subset \mathbb{C}^m$, defined in (26), and the equivalent form of Program (16) in Algorithm 2 given by Program (27), letting $v^I_{CGM}$ denote the optimal value of both of these programs. Suppose that Program (16) is solved to to an accuracy of $\theta \cdot \epsilon$ in every iteration, where

$$v^I_{CGM} - v_p \leq \frac{2\theta(1 + \epsilon)}{I + 2}.$$

Let us next bound $\theta$ in terms of the known quantities. Due to the assumption (11), for any feasible pair $(z, z')$ in (36), we have that

$$L(z' - y) - L(z - y) - \langle z' - z, \nabla L(z - y) \rangle \leq \frac{\gamma}{2} \|z' - z\|^2$$

$$\leq \frac{\gamma}{2} \|s - z\|^2 \quad (z' = z + \rho(s - z))$$

$$\leq \frac{\gamma}{2} (\|s\|^2 + \|z\|^2)$$

$$\leq \gamma \rho^2 r^2, \quad \text{(see (36))}$$

which immediately implies that $\theta \leq 2\gamma r^2$, and (13) now follows.

B Duality of Programs (1) and (3)

We show here that the dual of Program (1) is Program (3). We first observe that Program (1) is equivalent to

$$\begin{align*}
\min_{z,s} & \quad L(z - y) \\
\text{subject to} & \quad z = \int_I \Phi(t)x(dt) \\
& \quad \|x\|_{TV} \leq 1 \\
& \quad x \in B_1(\mathbb{I}).
\end{align*}$$

(38)

Introducing Lagrange multipliers $\lambda \in \mathbb{C}^m$ and $\alpha \geq 0$ for the two respective constraints, the Lagrangian $\mathcal{L}(x, \lambda, \alpha)$ for Program (38) is

$$\mathcal{L}(x, \lambda, \alpha) = L(z - y) - \text{Re} \left( \lambda, \left( z - \int_I \Phi(t)x(dt) \right) \right) + \alpha (\|x\|_{TV} - 1)$$

$$= L(z - y) + \text{Re}(\lambda, z) + \int_I (\alpha - \text{Re}(\lambda, \Phi(t))) x(dt),$$

and so the dual of Program (38) is

$$\max_{\lambda \in \mathbb{C}^m, \alpha \geq 0} \left\{ \inf_{x \in \mathbb{C}^n} [L(z - y) + \text{Re}(\lambda, z - y)] + \inf_{\mu \in \Theta_1(\mathbb{I})} \left[ \int_I (\alpha - \text{Re}(\lambda, \Phi(t))) x(dt) \right] + \text{Re}(\lambda, y) - \alpha \right\}$$
where $B_+(I)$ is the set of all nonnegative Borel measures supported on $I$. Using the definition of the Fenchel conjugate in (4), the above problem is equivalent to

\[
\begin{aligned}
\max_{\lambda, \alpha} & \quad -L_0(-\lambda) + \text{Re}(\lambda, y) - \alpha \\
\text{subject to} & \quad \text{Re}(\lambda, \Phi(t)) \leq \alpha \quad t \in I \\
& \quad \alpha \geq 0,
\end{aligned}
\]

which is Program (3).

## C Proof of Proposition 2

By construction, $T_{CGM}^0 = T_{EM}^0 = \emptyset$. Fix iteration $l \geq 1$ and assume that $T_{CGM}^{l-1} = T_{EM}^{l-1} = T^{l-1}$. We next show that $T_{CGM}^l = T_{EM}^l = T^l = T^{l-1} \cup \{t^l\}$, namely the two algorithms add the same point $t^l$ to their support sets in iteration $l$. We opt for a geometric argument here that relies heavily on Section 7.

Recall that Program (10) is equivalent to Program (27). Recall also that $y^l = \int_{t^l} \Phi(t)x^l(dt)$ is the unique minimizer of Program (27), where $x^l$ is a minimizer of Program (10). On the other hand, recall that Program (14) is equivalent to Program (31), and both programs have the unique minimizer $(\lambda^l, \alpha^l)$. Since Program (14) only has linear constraints, Slater’s condition is met and the tuple $(\lambda^l, \alpha^l)$ satisfies the KKT conditions, namely

\[
y^l \in C^{l-1}, \quad \lambda^l \in \alpha^l \cdot C_{\alpha}^{l-1}, \quad \alpha^l \geq 0,
\]

where

\[
\lambda^l = -\nabla L(y^l - y), \quad \langle y^l, \lambda^l \rangle = \alpha^l.
\]

From the above expression for $\lambda^l$, it follows immediately that the same point is added to the support in both Programs (10) and (14), which implies that $T_{CGM}^l = T_{EM}^l = T^{l-1} \cup \{t^l\}$. Finally, the above argument reveals that $v_{CGM}^l = v_{EM}^l$, which completes the proof of Proposition 2.

## D Proof of Proposition 3

Note that

\[
v_{EM}^l - v_d = v_{CGM}^{l-1} - v_d \quad \text{(see Proposition 2)}
\]

\[
= v_{CGM}^{l-1} - v_p \quad \text{(strong duality between Programs (1) and (3))}
\]

\[
\leq 4\gamma^2(1 + \epsilon)/\ell \quad \text{(see Proposition 1)}
\]

which proves the first claim in Proposition 3. To prove the second claim there, first recall the setup in Section 7. Let us first show that the minimizer of Program (14), namely $(\lambda^l, \alpha^l)$, converge to the minimizer of Program (3), namely $(\lambda_d, \alpha_d)$. To that end, recall the equivalent formulation of Programs (3, 14) given in (29, 31), and let

\[
h_{C_{l,s}}(\lambda) := \text{Re}(\lambda, y) - L_0(-\lambda) - g_{C_{l,s}}(\lambda),
\]

\[
h_{C_l}(\lambda) := \text{Re}(\lambda, y) - L_0(-\lambda) - g_{C_{l-1}}(\lambda),
\]

\[
(40)
\]

\[
\text{denote their objective functions, respectively. In particular, note that}
\]

\[
h_{C_{l,s}}(\lambda_d) = v_d, \quad h_{C_l}(\lambda^l) = v_{EM}^l.
\]

\[
(41)
\]

By assumption in (11), $L$ is $\gamma$-strongly smooth and therefore $L_0$ is $(\gamma^{-1})$-strongly convex [27]. Consequently, $-h_{C_{l,s}}$ is also $(\gamma^{-1})$-strongly convex, which in turn implies that

\[
\frac{1}{2\gamma}\|\lambda^l - \lambda_d\|^2 \leq -h_{C_{l,s}}(\lambda_d) + h_{C_{l}}(\lambda^l) + (\lambda_d - \lambda^l, \nabla h_{C_{l}}(\lambda^l))
\]

\[
= -h_{C_{l}}(\lambda_d) + v_{EM}^l. \quad \text{(see (41))}
\]

\[
(42)
\]
Let us next control \( h_{C_l} (\lambda_d) \) in the last line above by noting that
\[
\begin{align*}
  h_{C_l} (\lambda^d) &= \Re \langle \lambda_d, y \rangle - L_{c}(\lambda_d) - g_{C_l} (\lambda_d) \quad \text{(see (40))} \\
  &\geq \Re \langle \lambda_d, y \rangle - L_{c}(\lambda_d) - g_{C_{c, o}} (\lambda_d) \quad \left( C_{c} \supseteq C_{c, o} \text{ in (33)} \right) \\
  &= h_{C_{c, o}} (\lambda_d) \\
  &= v_d. \quad \text{(see (41)) (44)}
\end{align*}
\]

By substituting the bound above back into (43), we find that
\[
\| \lambda^l - \lambda_d \|^2 \leq 2 \gamma (v_{EM} - v_d) \leq \frac{8 \gamma^2 r^2 (1 + \epsilon)}{l + 2}. \quad \text{(see (39)) (45)}
\]
The above bound also allows us to find the convergence rate of \( \alpha^l \) to \( \alpha_d \). Indeed, note that
\[
\begin{align*}
  |\alpha^l - \alpha_d| &= \left| g_{C_l} (\lambda^l) - g_{C_{c, o}} (\lambda_d) \right| \quad \text{(see (32,30))} \\
  &= \left| \max_{t \in T_1} \langle \lambda^l, \Phi(t) \rangle - \max_{t \in T_1} \langle \lambda_d, \Phi(t) \rangle \right| \quad \text{(see (28))} \\
  &\leq \max_{t \in T_1} \left| \langle \lambda^l - \lambda_d, \Phi(t) \rangle \right| \\
  &\leq \| \lambda^l - \lambda_d \|_2 \max_{t \in T_1} \| \Phi(t) \|_2 \\
  &\leq \sqrt{\frac{8 \gamma^2 r^2 (1 + \epsilon)}{l + 2}} \cdot r. \quad \text{(see (45,12)) (46)}
\end{align*}
\]

With an argument similar to (46), we also find that
\[
\left| \max_{t \in T_1} \langle \lambda^l, \Phi(t) \rangle - \alpha_d \right| \leq \frac{8 \gamma^2 r^2 (1 + \epsilon)}{l + 2} \cdot r, \quad \text{(47)}
\]
which completes the proof of Proposition 3.