A Unified Framework for the Convergence Analysis of Optimization Algorithms via Sum-of-Squares

Sandra S. Y. Tan*, Antonios Varvitsiotis* and Vincent Y. F. Tan*†

* Department of Electrical and Computer Engineering, National University of Singapore
† Department of Mathematics, National University of Singapore
sandra_tsy@u.nus.edu, avarvits@gmail.com, vtan@nus.edu.sg

***Abstract*—Given the popularity of machine learning and large-scale data analysis, understanding the convergence properties of the optimization algorithms that constitute the workhorse in these fields is of fundamental theoretical importance. However, existing proofs of convergence consist mostly of ad-hoc arguments and case-by-case analysis. In this paper, we introduce a novel optimization framework for bounding the convergence rates of various important optimization algorithms by means of sum-of-squares certificates. Our approach allows us to recover known convergence bounds for three widely-used first-order algorithms in a unified manner and puts forward a promising framework for unifying convergence analyses of optimization algorithms.**

I. Introduction

With the rise of fields such as machine learning and computer vision, optimization algorithms—the workhorse of these fields—have become increasingly important. There is a need to evaluate the performance of various low-complexity optimization algorithms (e.g., first-order optimization algorithms that utilize (sub-)gradient information) and to provide convergence guarantees for them.

There exists extensive literature on the convergence analysis of many families of algorithms with respect to various performance metrics, e.g., see [1]–[4] and references therein. However, the proofs of these results typically consist of ad-hoc arguments and case-by-case analysis.

In this paper, we introduce a unified framework for deriving worst-case upper bounds on the convergence rates of optimization algorithms, through the use of sum-of-squares (SOS) certificates. SOS optimization is an active research area with important practical applications, e.g., see [5]–[8]. The key idea underlying SOS optimization is the use of semidefinite programming (SDP) to identify sufficient conditions for the nonnegativity of a polynomial over a semi-algebraic set.

To illustrate the main ingredients of our approach, consider the problem of minimizing a function $f:\mathbb{R}^n \to \mathbb{R}$ over $\mathbb{R}^n$, i.e.,

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}). \tag{1}$$

Solving problem (1) entails choosing a black-box algorithm $\mathcal{A}$ that generates a sequence of iterates $\{\mathbf{x}_i\}_{i\geq 1}$. Our goal is then to estimate the convergence rate of $\mathcal{A}$ with respect to a fixed family of functions $\mathcal{F}$ and an appropriate measure of performance (e.g., distance to optimality, residual gradient norm, and objective function accuracy).

For concreteness, let $f_i = f(\mathbf{x}_i)$ and $\mathbf{g}_i \in \partial f(\mathbf{x}_i)$ for all i. Fixing a monotone algorithm $\mathcal{A}$ (i.e., an algorithm that makes progress in terms of reducing the objective function value at each step) and using as performance metric the objective function accuracy (defined as the distance between the value of the objective function at the current iterate and the optimal value f_*), we seek to find the smallest scalar $t\in(0,1)$ satisfying

$$f_{k+1} - f_* \leq t(f_k - f_*) \tag{2}$$

This work is partially funded by an NRF Fellowship (R-263-000-D02-281).

for all $k\in\mathbb{N}$, all functions $f\in\mathcal{F}$, and all sequences of iterates $\{\mathbf{x}_k\}$ generated by $\mathcal{A}$. As this optimization problem is hard, we relax it into a tractable convex program in two steps. In the first step we derive necessary conditions that are expressed as polynomial (in)equalities $h_i(\mathbf{z}) \geq 0,\ g_j(\mathbf{z}) = 0$ in terms of the variables $\mathbf{z} = (f_*, f_k, f_{k+1}, \mathbf{x}_*, \mathbf{x}_k, \mathbf{x}_{k+1}, \mathbf{g}_*, \mathbf{g}_k, \mathbf{g}_{k+1})$, which are dictated by the choice of the algorithm and the corresponding class of functions. Having identified these necessary polynomial constraints, the first relaxation of (2) is to find the minimum $t\in(0,1)$ such that the polynomial $t(f_k - f_*) - (f_{k+1} - f_*)$ is nonnegative over the semi-algebraic set $K = \{\mathbf{z}\in\mathbb{R}^n : h_i(\mathbf{z}) \geq 0,\ g_j(\mathbf{z}) = 0\}$. Nevertheless, as this second problem is also hard in general, in the second step we further relax this constraint by demanding that the nonnegativity of the polynomial $t(f_k - f_*) - (f_{k+1} - f_*)$ over K is certified by an SOS decomposition. Lastly, the use of SOS certificates allows to estimate the optimal rate using SDPs.

To demonstrate the value of our approach we recover in a unified manner known bounds for three methods: Gradient Descent (GD) with exact line search, Proximal Gradient Method (PGM) with constant step size, as well as PGM with exact line search. Although in this paper we only focus on (monotone) first-order algorithms and we use as performance measure the objective function accuracy, our framework can also accommodate more general classes of algorithms and function classes. This will be the topic of future work.

A closely related approach to deriving worst-case bounds on the performance of first-order optimization algorithms was proposed recently by Drori and Teboulle [9]. Specifically, they developed a framework that casts the search for these bounds as an optimization problem in itself, known as the *Performance Estimation Problem* (PEP). In most instances where the PEP was applied, SOS certificates appeared in the derivation of their bounds, e.g., see [10] [11]. Our technique makes the search for the SOS certificates explicit, and hence, provides a more principled approach for deriving such bounds. In addition, our framework is able to handle more general constraints, and furthermore, the verification of the derived bounds using our technique is more straightforward than in the PEP.

The paper is organized as follows: Section II introduces the SOS technique, provides a step-by-step guide to applying it, and discusses its relation with the PEP. Examples shown in Sections III to V work out the derivation of bounds for specific algorithms. Finally, Section VI provides some implementation details, while Section VII contains some concluding remarks.

II. The Sum-of-Squares Approach

Before we provide the details of our approach we need to introduce some necessary notation and definitions. For any $\mathbf{a}\in\mathbb{N}^n$, where $\mathbb{N}$ is the set of nonnegative integers, we denote by $\mathbf{z}^{\mathbf{a}}$ the monomial $z_1^{a_1}\dots z_n^{a_n}$. The degree of the monomial $\mathbf{z}^{\mathbf{a}}$ is defined to be $|\,\mathbf{a}\,| = \sum_{i=1}^n a_i$. Let $\mathbb{R}[\mathbf{z}]_{n,d}$ denote the set of polynomials in n variables

$z_1, \ldots, z_n$, of degree at most d, i.e., each element of $\mathbb{R}[\mathbf{z}]_{n,d}$ can be written as a linear combination of monomials of degree-d monomials.

A polynomial $p(\mathbf{z}) \in \mathbb{R}[\mathbf{z}]_{n,2d}$ is called an SOS (of degree d) if there exist polynomials $q_1, \ldots, q_m \in \mathbb{R}[\mathbf{z}]_{n,d}$ satisfying

$$p(\mathbf{z}) = \sum_{k=1}^{m} q_k^2(\mathbf{z}). \tag{3}$$

It is instructive to think of the existence of an SOS decomposition as in (3) as a tractable certificate for global nonnegativity. Indeed, it is evident from (3) that any SOS polynomial $p(\mathbf{z})$ is also globally nonnegative, i.e., $p(\mathbf{z}) \geq 0$ for all $\mathbf{z} \in \mathbb{R}^n$. Furthermore, although less obvious, it is well-known that checking the existence of an SOS decomposition can be done efficiently using SDPs [8]. Concretely, a polynomial $p(\mathbf{z}) = \sum_{\mathbf{a}} p_{\mathbf{a}} \mathbf{z}^{\mathbf{a}} \in \mathbb{R}[\mathbf{z}]_{n,2d}$ admits an SOS decomposition where the q_k's have degree at most d if and only if there exists a positive semidefinite matrix Q indexed by all vectors $\{\mathbf{a} \in \mathbb{N}^n : |\,\mathbf{a}\,| \leq d\}$ that satisfies

$$p_{\mathbf{a}} = \sum_{\mathbf{b},\, \mathbf{c}\,:\, \mathbf{b}+\mathbf{c}=\mathbf{a}} Q_{\mathbf{b},\mathbf{c}}, \ \forall\, \mathbf{a}\,. \tag{4}$$

This statement admits an easy proof: express $p(\mathbf{z})$ in its quadratic form $m_d^T Q m_d$, where m_d denotes the vector of monomials in $\mathbf{z}$ of degree at most d and Q is positive semidefinite. Matching coefficients of the monomials in $\mathbf{z}$ produces the set of affine constraints given in (4). Thus, deciding whether $p(\mathbf{z})$ is an SOS is an SDP with matrix variables of size $\binom{n+d}{d}$, which can be solved efficiently [8].

Moving beyond global nonnegativity, a more general problem of significant practical importance is to certify the nonnegativity of a polynomial $p(\mathbf{z})$ over a (basic) closed semi-algebraic set

$$K = \{\mathbf{z} \in \mathbb{R}^n : \ h_i(\mathbf{z}) \geq 0 \ (1 \leq i \leq m), \ g_j(\mathbf{z}) = 0 \ (1 \leq j \leq \ell)\}\,,$$

i.e., to certify that $p(\mathbf{z}) \geq 0$ for all $\mathbf{z} \in K$. Analogously to the SOS case above, we look for certificates that can be found efficiently using SDPs. One such choice is given by:

$$p(\mathbf{z}) = s_0(\mathbf{z}) + \sum_{i=1}^{m} s_i(\mathbf{z}) h_i(\mathbf{z}) + \sum_{j=1}^{\ell} v_j(\mathbf{z}) g_j(\mathbf{z}), \tag{5}$$

where the s_i's are themselves SOS polynomials and the v_j's are arbitrary (i.e., not necessarily SOS) polynomials. Clearly, the expression (5) serves as a certificate that $p(\mathbf{z}) \geq 0$ for all $\mathbf{z} \in K$ and moreover, the existence of such a representation (for a fixed degree d) can be done by means of SDPs. Indeed, it is well-known (see [8]) that $p(\mathbf{z})$ admits an SOS decomposition as in (5) if and only if there exist $m+1$ positive semidefinite matrices $Q^0, \ldots, Q^m$ such that

$$p_{\mathbf{a}} = \sum_{\substack{\mathbf{b},\, \mathbf{c}\,: \\ \mathbf{b}+\mathbf{c}=\mathbf{a}}} Q^0_{\mathbf{b},\mathbf{c}} + \sum_{\substack{\mathbf{b},\, \mathbf{c},\, \mathbf{d}\,: \\ \mathbf{b}+\mathbf{c}+\mathbf{d}=\mathbf{a}}} \sum_{i=1}^{m} Q^i_{\mathbf{b},\mathbf{c}} h_{i,\mathbf{d}} + \sum_{\substack{\mathbf{a},\, \mathbf{b}\,: \\ \mathbf{b}+\mathbf{c}=\mathbf{a}}} \sum_{j=1}^{\ell} v_{j,\mathbf{b}} g_{j,\mathbf{c}}, \tag{6}$$

where $h_{i,\mathbf{d}}$ refers to the coefficient of the monomial $\mathbf{z}^{\mathbf{d}}$ in the polynomial h_i. Again, these affine constraints (6) are obtained by expressing each SOS term in its quadratic form and matching coefficients of the monomials in $\mathbf{z}$. The problem of finding positive semidefinite matrices $Q^0, Q^1, \ldots, Q^m$ satisfying the affine constraints (6) is an instance of a (block diagonal) SDP feasibility problem.

A. SOS certificates for analyzing optimization algorithms

As mentioned in the introduction, in this paper, we focus on (monotone) first-order algorithms and use as performance metric the objective function accuracy. Fixing a family of functions $\mathcal{F}$ and a (monotone) first-order algorithm $\mathcal{A}$, our goal is to estimate the best contraction factor $t \in (0,1)$ that is valid over all functions in $\mathcal{F}$ and all sequences of iterates that can be generated using the algorithm $\mathcal{A}$. Concretely, we aim to estimate the minimum $t \in (0,1)$ satisfying

$$f_{k+1} - f_* \leq t(f_k - f_*) \tag{7}$$

for all $k \in \mathbb{N}$, all functions $f \in \mathcal{F}$, and all sequences of iterates $\mathbf{x}_{k+1} = \mathcal{A}\left(\mathbf{x}_0, \ldots, \mathbf{x}_k; f_0, \ldots, f_k; \mathbf{g}_0, \ldots, \mathbf{g}_k\right)$. The novelty of our approach is that we address this question using SOS certificates. To employ the SOS approach, we identify polynomial inequalities $h_i(\mathbf{z}) \geq 0$ and polynomial equalities $g_j(\mathbf{z}) = 0$ in the variables

$$\mathbf{z} = (f_*, f_k, f_{k+1}, \mathbf{x}_*, \mathbf{x}_k, \mathbf{x}_{k+1}, \mathbf{g}_*, \mathbf{g}_k, \mathbf{g}_{k+1}) \in \mathbb{R}^{6n+3} \tag{8}$$

that should be necessarily satisfied, following the choice of the class of functions $\mathcal{F}$ and the first-order algorithm $\mathcal{A}$. Letting K to be the semi-algebraic set defined by the h_i's and the g_j's, if a fixed rate $t \in (0,1)$ is admissible (in the sense of (7)), the polynomial

$$p(\mathbf{z}) := t(f_k - f_*) - (f_{k+1} - f_*) \tag{9}$$

is nonnegative on the set K. In view of this, we set out to find the least $t \in (0,1)$ for which $p(\mathbf{z})$ is nonnegative on K, i.e.,

$$\begin{aligned} \min \quad & t \\ \text{s.t.} \quad & p(\mathbf{z}) \geq 0, \ \forall\, \mathbf{z} \in K, \\ & t \in (0,1). \end{aligned} \tag{10}$$

As (10) is not tractable, we relax the constraint that $p(\mathbf{z})$ is nonnegative over K with the SOS certificate introduced in (5). Concretely, for any $d \in \mathbb{N}$, this leads to the SDP:

$$\begin{aligned} \min \quad & t \\ \text{s.t.} \quad & p(\mathbf{z}) = s_0(\mathbf{z}) + \sum_{i=1}^{m} s_i(\mathbf{z}) h_i(\mathbf{z}) + \sum_{j=1}^{\ell} v_j(\mathbf{z}) g_j(\mathbf{z}), \\ & t \in (0,1), \\ & s_i(\mathbf{z}) : \text{SOS of degree } d, \ v_j(\mathbf{z}) : \text{degree } d. \end{aligned} \tag{11}$$

As discussed previously, for any fixed d, the optimization problem (11) is an instance of an SDP, and consequently, it can be solved efficiently. Summarizing, our strategy for estimating the best rate t satisfying (7) can be broken down in the following steps:

1) Identify polynomial inequality and equality constraints $h_i(\mathbf{z}) \geq 0$, $g_j(\mathbf{z}) = 0$ in the variable $\mathbf{z}$ (recall (8)) that are implied by choosing a function class and an algorithm.
2) Fix a degree $d \in \mathbb{N}$ for the SOS certificate, i.e., for the degree of the s_i's and v_j's. Higher degree certificates allow for tighter bounds but are more difficult to find due to the size of the SDP.
3) Numerically solve the SDP (11) using degree-d SOS certificates. The optimal solution, denoted by t_d, is the best contraction factor that can certified using degree-d SOS certificates.

An optional step which we use to reduce the complexity of the SDP is to substitute equality constraints that linear in $\mathbf{z}$ into the rest of the constraints and eliminate the corresponding variables.

Lastly, to find a symbolic expression for the optimal rate t_d, we vary the parameters in the coefficients of the h_i's and g_j's and 'guess' the symbolic expression of the SOS certificate and the rate t_d.

B. Related Work

The PEP framework was introduced in [9] to bound the worst-case performance of first-order black-box optimization methods by studying the following optimization problem:

$$\begin{aligned}\max_{\mathbf{z}} \ & f_k - f_* \\ \text{s.t. } & g_j(\mathbf{z}) = 0, \quad \forall\, j, \\ & h_i(\mathbf{z}) \geq 0, \quad \forall\, i, \\ & f_k - f_* \leq R.\end{aligned}$$

Under some conditions, the PEP can be relaxed/formulated into an SDP. In the cases analyzed thus far, when the authors multiplied each g_j and h_i (excluding the boundedness constraint) with its (symbolic) dual optimal variable and summed these up, they can be rearranged to form expressions of the form

$$f_{k+1} - f_* \leq t(f_k - f_*) - \text{an SOS term}, \tag{12}$$

which implies (7), since the SOS term is nonnegative [10] [11]. However, it is not easy to complete the squares to produce the SOS term nor is it immediately obvious why SOS certificates should appear.

Our approach instead makes the search for this SOS term explicit, with the advantage that instead of having rearrange the terms to find an SOS decomposition, we only need check the positive semidefiniteness of Q; this can be easily done through symbolic software.

In addition, the PEP requires all constraints to be in terms of inner products of iterates and their sub-gradients, $\mathbf{x}_i$'s and $\mathbf{g}_i$'s, in order to be formulated as an SDP. The SOS approach does not suffer from such a restriction. It is able to handle general constraints, albeit requiring higher degree certificates.

III. GD with Exact Line Search

A. Function Class

For all three examples, we consider the class of *L-smooth, μ-strongly convex* functions, denoted $\mathcal{F}_{\mu,L}(\mathbb{R}^n)$. Given a proper, closed convex function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ and constants μ, L (where $0 \leq \mu < L \leq +\infty$), f is *L-smooth, μ-strongly convex* if:

1) $\frac{1}{L}\|\mathbf{g}_1 - \mathbf{g}_2\| \leq \|\mathbf{x}_1 - \mathbf{x}_2\|$ for all $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$ and their corresponding sub-gradients $\mathbf{g}_1 \in \partial f(\mathbf{x}_1), \mathbf{g}_2 \in \partial f(\mathbf{x}_2)$.
2) $f(\mathbf{x}) - \frac{\mu}{2}\|\mathbf{x}\|^2$ is convex.

Taylor et al. [12] developed a set of necessary and sufficient conditions on data triples $\{(\mathbf{x}_i, f_i, \mathbf{g}_i)\}_{i\in I}$ that guarantees the existence of a function in $\mathcal{F}_{\mu,L}(\mathbb{R}^n)$ that could possibly have generated these data triples. Such a set of data triples is termed *(μ, L)-interpolable*.

Theorem 1 ($\mathcal{F}_{\mu,L}$-interpolability)**.** *Given a set $\{(\boldsymbol{x}_i, f_i, \boldsymbol{g}_i)\}_{i\in I}$, there exists $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^n)$ where $f_i = f(\boldsymbol{x}_i)$ and $\boldsymbol{g}_i \in \partial f(\boldsymbol{x}_i)$ for all i if and only if*

$$\begin{aligned}f_i - f_j - \boldsymbol{g}_j^T(\boldsymbol{x}_i - \boldsymbol{x}_j) \geq \frac{1}{2(1-\mu/L)}\Big(\frac{1}{L}\|\boldsymbol{g}_i - \boldsymbol{g}_j\|^2 \\ + \mu\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 - 2\frac{\mu}{L}(\boldsymbol{g}_j - \boldsymbol{g}_i)^T(\boldsymbol{x}_j - \boldsymbol{x}_i)\Big) \ \forall\, i \neq j.\end{aligned}$$

This result applied to the index set $I = \{k, k+1, *\}$ produces 6 (μ, L)-interpolability conditions.

B. Algorithmic Constraint

de Klerk et al. [10] identified the following properties of GD with exact line search: For all k,

$$\mathbf{x}_{k+1} - \mathbf{x}_k + \gamma\mathbf{g}_k = 0 \text{ for some } \gamma \geq 0, \tag{13}$$
$$\mathbf{g}_{k+1}^T(\mathbf{x}_{k+1} - \mathbf{x}_k) = 0. \tag{14}$$

In general, $\mathbf{x}_{k+1} = \mathbf{x}_k + \gamma\mathbf{d}_k$. Equation (13) is the update rule for gradient descent (where $\mathbf{d}_k = -\mathbf{g}_k$), while (14) captures the idea for exact line search, i.e., for $\hat{\gamma} = \arg\min_\gamma f(\mathbf{x}_k + \gamma\mathbf{d}_k)$, we find that $\nabla f(\mathbf{x}_k + \hat{\gamma}\mathbf{d}_k)^T\mathbf{d}_k = 0$ which implies (14).

A consequence of (13) and (14) is

$$\mathbf{g}_{k+1}^T\mathbf{g}_k = 0. \tag{15}$$

Following de Klerk et al.'s [10] methodology, we use the relaxed conditions of (14) and (15) instead of (13) and (14) to obtain γ-independent constraints.

C. Searching for SOS Certificates

Thus far, we have identified:

- 6 (μ, L)-interpolability inequality constraints, denoted by $h_i(\mathbf{z}) \geq 0$
- 2 algorithmic equality constraints: (14) and (15), denoted by $g_j(\mathbf{z}) = 0$

Since $\mathbf{g}_* = 0$, we can remove this variable from $\mathbf{z} = (\mathbf{x}_i, f_i, \mathbf{g}_i)_{i=k,k+1,*}$ and set any term containing $\mathbf{g}_*$ in the other constraints to 0. Hence, the updated $\mathbf{z}$ is given by $(f_*, f_k, f_{k+1}, \mathbf{x}_*, \mathbf{x}_k, \mathbf{x}_{k+1}, \mathbf{g}_k, \mathbf{g}_{k+1}) \in \mathbb{R}^{5n+3}$.

We choose to search for a mixed 0/1-degree certificate, where the s_0 of (5) is an SOS of degree 1, and the s_i's (for $1 \leq i \leq m$) and the v_j's are degree 0. Thus, we are searching for a certificate of the following form:

$$t(f_k - f_*) - (f_{k+1} - f_*) = m_1^T Q m_1 + \sum_i \lambda_i h_i(\mathbf{z}) + \sum_j \lambda'_j g_j(\mathbf{z}), \tag{16}$$

where $m_1 = (1, \mathbf{z}) = (1, f_*, \ldots, \mathbf{g}_{k+1,n})$, λ_i's are nonnegative scalars, and λ'_j's are unconstrained scalars.

Matching coefficients of the monomials in (16), we obtain a set of affine constraints given in (27) in Appendix A. By minimizing t subject to $Q \succeq 0$, $\lambda_i \geq 0$ and the affine constraints in (27), we arrive at following result.

Theorem 2. *Given a function $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^n)$ and a sequence of iterates $\{\boldsymbol{x}_i\}_{i\geq 1}$ generated by gradient descent with exact line search, we can certify that*

$$f_{k+1} - f_* \leq \left(\frac{L-\mu}{L+\mu}\right)^2 (f_k - f_*)$$

holds for all $k \in \mathbb{N}$ using SOS certificates of degree-1.

The minimum contraction factor $t = \left(\frac{L-\mu}{L+\mu}\right)^2$ obtained matches de Klerk et al.'s results [10]. Refer to Appendix A for full details, and to Appendix B for the link to the code verifying this result.

D. Discussion

Higher Degree Certificates: To find the minimum t, we have chosen to search for a mixed 0/1-degree certificate. We have found that the optimal contraction factor and multipliers for a fully degree-1 certificate reduces to the mixed 0/1-degree case. While higher degree certificates (beyond degree 1) could potentially produce tighter bounds, it would also be much harder to obtain the symbolic expressions for the optimal contraction factor and multipliers.

Consider equation (6). Let $\bar{v}_j$ denote the vector of coefficients $v_{j,\mathbf{b}}$. If the s_i's and v_j's of (5) have degrees greater than 0, then the Q^i's and $\bar{v}_j$ are no longer scalars but full matrices and vectors respectively. We would have to guess symbolic expressions for every entry of all the Q^i's and $\bar{v}_j$'s.

Multivariate Case: When matching coefficients of the monomials, we assumed that the iterates $\mathbf{x}_i$ are univariate i.e., $n = 1$. Let us consider the multivariate case. Instead of matching coefficients for say the $x_* x_k$ term, we have to consider $x_{*,1}x_{k,1},\ x_{*,1}x_{k,2},\ \ldots,\ x_{*,n}x_{k,n}$.

Presently, any term containing $\mathbf{x}_i$ or $\mathbf{g}_i$ within the constraints appear only as inner products. Hence the coefficient for $x_* x_k$ in the univariate case is the same coefficient for $x_{*,i}x_{k,i},\ i = 1, \ldots, n$ in the multivariate case. The coefficient for $x_{*,i}x_{k,j}$ for $i \neq j$ is 0.

Let the matrix Q for the univariate and multivariate case be donated Q_u and Q_m respectively. Let the element in Q_u corresponding to $x_* x_k$ contain the value c. In the multivariate case, this single element is replaced with a diagonal matrix $c \cdot I_n$ in Q_m (where I_n is the identity matrix of size n). In general, all the elements corresponding to inner products of $\mathbf{x}_i$'s and $\mathbf{g}_i$'s (forming a submatrix within Q_u) are similarly replaced. Thus, submatrix of Q_m = submatrix of $Q_\mathrm{u} \otimes I_n$. If Q_u is positive semidefinite, Q_m is too and thus it suffices to solve the univariate case.

IV. PGM with Constant Step Size

We first describe the proximal gradient method (PGM) with constant step size [13]. Consider a composite convex minimization problem:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \left\{ F(\mathbf{x}) := f(\mathbf{x}) + h(\mathbf{x}) \right\}, \tag{17}$$

where $f \in \mathcal{F}_{\mu,L}$ and $h \in \mathcal{F}_{0,\infty}$. We assume that f is differentiable and the proximal operator of h,

$$\mathbf{p}_{\gamma h}(\mathbf{x}) := \arg\min_{\mathbf{y} \in \mathbb{R}^n} \left\{ \gamma h(\mathbf{y}) + \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|^2 \right\},$$

exists at every $\mathbf{x}$. The proximal gradient method (PGM) with constant step size γ generates iterates as follows:

$$\mathbf{x}_{k+1} = \mathbf{p}_{\gamma h}\left(\mathbf{x}_k - \gamma \mathbf{g}_k\right), s \tag{18}$$

where $0 \leq \gamma \leq \frac{2}{L}$ and $\mathbf{g}_k \in \partial f(\mathbf{x}_k)$.

Taylor et al. [11] reformulated (18) so that it is not in terms of the proximal operator. First, we introduce some additional notation. Let $\mathbf{x}_k'$ be the "intermediate" step,

$$\mathbf{x}_k' = \mathbf{x}_k - \gamma \mathbf{g}_k,$$

such that

$$\mathbf{x}_{k+1} = \mathbf{p}_{\gamma h}(\mathbf{x}_k').$$

Taking the derivative of $\mathbf{p}_{\gamma h}(\mathbf{x}_k')$ wrt $\mathbf{y}$ and setting it to 0, we obtain

$$\gamma \mathbf{s}_y + \mathbf{y} - \mathbf{x}_k' = 0, \tag{19}$$

where $\mathbf{s}_y \in \partial h(\mathbf{y})$. Substituting $\mathbf{y} = \mathbf{x}_{k+1}$ and the definition of $\mathbf{x}_k'$ into (19), we obtain

$$\begin{aligned} &\gamma \mathbf{s}_{k+1} + \mathbf{x}_{k+1} - \mathbf{x}_k + \gamma \mathbf{g}_k = 0 \\ \Longrightarrow\ &\mathbf{x}_{k+1} = \mathbf{x}_k - \gamma\left(\mathbf{g}_k + \mathbf{s}_{k+1}\right). \end{aligned} \tag{20}$$

A. Searching for SOS Certificates

The constraints characterizing this function class and algorithm are

- 6 (μ, L)-interpolability inequality constraints on f
- 6 (μ, L)-interpolability inequality constraints on h with $\mu = 0, L = \infty$
- 1 algorithmic equality constraint (20)
- 1 optimality condition: $\mathbf{g}_* + \mathbf{s}_* = 0$

We can eliminate variables $\mathbf{x}_{k+1}$ and $\mathbf{s}_*$ by substituting the equality conditions into the rest of the constraints. We thus have $\mathbf{z} = (f_*, f_k, f_{k+1}, h_*, h_k, h_{k+1}, \mathbf{x}_*, \mathbf{x}_k, \mathbf{g}_*, \mathbf{g}_k, \mathbf{g}_{k+1}, \mathbf{s}_k, \mathbf{s}_{k+1}) \in \mathbb{R}^{7n+6}$. Again, we search for a mixed 0/1-degree SOS certificate:

$$\begin{aligned} p(\mathbf{z}) &:= t(f_k + h_k) + (1-t)(f_* + h_*) - (f_{k+1} + h_{k+1}) \\ &= m_1^T Q m_1 + \sum_i \lambda_i h_i(\mathbf{z}). \end{aligned}$$

Similarly, we matched the coefficients for the monomials to obtain a set of linear equalities given in (1) in the supplementary document; see Appendix B. Taylor et al. [11] found that the step size $\gamma = \frac{2}{L+\mu}$ minimizes the contraction factor t. Thus, we solve the SDP for $\gamma = \frac{2}{L+\mu}$ and obtain the following result.

Theorem 3. *Given a function $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^n)$ and a sequence of iterates $\{\boldsymbol{x}_i\}_{i \geq 1}$ generated by the proximal gradient method with constant step size $\gamma = \frac{2}{L+\mu}$, we can certify that*

$$f_{k+1} - f_* \leq \left(\frac{L-\mu}{L+\mu}\right)^2 (f_k - f_*)$$

holds for all $k \in \mathbb{N}$ using SOS certificates of degree-1.

Details of the derivation as well as the code verifying this result can be found in the supplementary document; see Appendix B. The optimal t obtained matches the results found by Taylor et al. [11, Theorem 3.3].

We additionally note that if the equality constraints had not been removed, the SDP would still be feasible if the multipliers for the equality constraints were chosen to be degree 1 polynomials. The optimal contraction factor, t, achieved would be the same. However, to obtain an SDP formulation under the PEP framework, these equality constraints must be removed.

V. PGM with Exact Line Search

The optimal contraction factor for this method has been derived by Taylor et al. [11] as well, although only as a corollary of their bound for PGM with constant step size. This contraction factor was shown to be tight/optimal. We aim to recover this bound by analyzing the algorithm directly.

Given the optimization problem in (17), the PGM with exact line search generates iterates as follows:

$$\gamma_k = \arg\min_{\gamma} F\left[\mathbf{p}_{\gamma h}\left(\mathbf{x}_k - \gamma \mathbf{g}_k\right)\right], \tag{21}$$

$$\mathbf{x}_{k+1} = \mathbf{p}_{\gamma h}\left(\mathbf{x}_k - \gamma_k \mathbf{g}_k\right). \tag{22}$$

In the same way (18) was written as (20), we can rewrite equations (21) and (22) as

$$\gamma_k = \arg\min_{\gamma} F\left[\mathbf{x}_k - \gamma(\mathbf{g}_k + \mathbf{s}_{k+1})\right], \tag{23}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \gamma_k(\mathbf{g}_k + \mathbf{s}_{k+1}). \tag{24}$$

Taking the derivative of (23) with respect to γ and setting it to 0, we obtain

$$\tilde{\nabla} F\left[\mathbf{x}_k - \gamma(\mathbf{g}_k + \mathbf{s}_{k+1})\right]^T\left[-(\mathbf{g}_k + \mathbf{s}_{k+1})\right] = 0,$$

where $\tilde{\nabla} F(\mathbf{x}) \in \partial F(\mathbf{x})$. Substituting $\gamma = \gamma_k$, we obtain

$$\begin{aligned} \tilde{\nabla} F(\mathbf{x}_{k+1})^T(\mathbf{g}_k + \mathbf{s}_{k+1}) &= 0, \\ (\mathbf{g}_{k+1} + \mathbf{s}_{k+1})^T(\mathbf{g}_k + \mathbf{s}_{k+1}) &= 0. \end{aligned} \tag{25}$$

Substituing (24) into (25), we obtain

$$(\mathbf{g}_{k+1} + \mathbf{s}_{k+1})^T(\mathbf{x}_{k+1} - \mathbf{x}_k) = 0. \tag{26}$$

Our goal now is to use (25) and (26) as our algorithmic constraints in the SOS decomposition problem.

A. Searching for SOS Certificates

The constraints characterising this function class and algorithm are

- 6 (μ, L)-interpolability inequality constraints on f
- 6 (μ, L)-interpolability inequality constraints on h with $\mu = 0, L = \infty$
- 2 algorithmic equality constraints: (25) and (26)
- 1 optimality condition: $\mathbf{g}_* + \mathbf{s}_* = 0$

Here, we have $\mathbf{z} = (f_*, f_k, f_{k+1}, h_*, h_k, h_{k+1}, \mathbf{x}_*, \mathbf{x}_k, \mathbf{x}_{k+1}, \mathbf{g}_*, \mathbf{g}_k, \mathbf{g}_{k+1}, \mathbf{s}_*, \mathbf{s}_k, \mathbf{s}_{k+1}) \in \mathbb{R}^{8n+6}$. We then search for a mixed 0/1-degree SOS certificate:

$$\begin{aligned} p(\mathbf{z}) &:= t(f_k + h_k) + (1-t)(f_* + h_*) - (f_{k+1} + h_{k+1}) \\ &= m_1^T Q m_1 + \sum_i \lambda_i h_i(\mathbf{z}) + \sum_j \lambda'_j g_j(\mathbf{z}). \end{aligned}$$

After matching coefficients and solving the resultant SDP, given in (2) of the supplementary document, our result can be stated as follows:

Theorem 4. *Given a function $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^n)$ and a sequence of iterates $\{\boldsymbol{x}_i\}_{i \geq 1}$ generated by the proximal gradient method with exact line search, we can certify that*

$$f_{k+1} - f_* \leq \left(\frac{L-\mu}{L+\mu}\right)^2 (f_k - f_*)$$

holds for all $k \in \mathbb{N}$ using SOS certificates of degree-1.

This matches Taylor et al.'s bound [11]. (The derivation details and code can be found in the supplementary document; see Appendix B.) It is interesting to note that this method, PGM with exact line search, does not guarantee a better convergence rate than PGM with constant step size $\gamma = \frac{2}{L+\mu}$ in Theorem 3.

It turns out that the bound for PGM with exact line search (as given in Theorem 4) is tight for quadratic functions [1, Example on p. 68], which has $\gamma = \frac{2}{L+\mu}$ as the optimal step size at every iteration. Hence, the contraction factor $t = \left(\frac{L-\mu}{L+\mu}\right)^2$ cannot be improved.

VI. Implementation Details

In the examples described above, we searched for mixed 0/1-degree certificates. Matching coefficients and listing the set of affine constraints was done by hand to demonstrate the SOS technique. The resultant SDP was solved using CVX, an SDP solver. Doing these by hand is tedious but possible (as seen by the details in the appendix). However, this quickly becomes impractical for higher-degree certificates.

Fortunately, there are many SOS optimization solvers such as YALMIP that automate the process of matching coefficients and constructing the SDP. The user only needs to input the constraints and set the degree of the certificate he/she wishes to search for. However, these solvers can only check for whether there is an SOS decomposition (i.e., check feasibility of the SDP for a fixed t, and not minimize t directly).

Thus, the user can first fix t to be 0.5, bisecting the interval $(0, 1)$, and check for feasibility for the fixed t. In the next run, fix t to bisect its updated feasible region (i.e., 0.25 or 0.75) and check feasibility again. Repeat this, bisecting the interval each time until the user gets the desired degree of accuracy for t. The link to the MATLAB code for both CVX and YALMIP can be found in Appendix B.

VII. Conclusion

This paper proposes a new technique for bounding the convergence rate for various algorithms, by searching for SOS certificates. Our approach allows the problem to be formulated as an SDP. It also has several advantages over the PEP framework; it is able to handle more general constraints, and its method of verifying the bound (checking the positive semidefiniteness of a matrix symbolically) is more straightforward than the method suggested by the PEP framework [10] [11]. With this technique, we recover previously-known bounds for three first-order optimization algorithms: GD with exact line search, PGM with constant step size and PGM with exact line search.

However, our technique does not necessarily produce tight bounds, since it entails two relaxation steps. For one, the constraints characterizing the function class or algorithm may be relaxed. Secondly, we relax the constraint that $p(\mathbf{z})$ be nonnegative to the constraint that $p(\mathbf{z})$ is an SOS. Recall that while SOS implies nonnegativity, the converse is not necessarily true. Proving the tightness of the derived bounds will have to be done via other means.

A. Future Work

This technique can be applied to other algorithms, for which convergence rates may not be known, e.g., GD with Armijo rule. Additionally, while we have focused on finding bounds on objective function accuracy of the form stated in (7), this technique may be extended to bounding other performance metrics, e.g., $\|\mathbf{x}_{k+1} - \mathbf{x}_*\| \leq t\|\mathbf{x}_k - \mathbf{x}_*\|$ or $f_{k+1} - f_* \leq t\|\mathbf{x}_k - \mathbf{x}_*\|$.

References

[1] D. P. Bertsekas, *Nonlinear programming*, 2nd ed. Athena Scientific, 1999.

[2] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2009.

[3] A. I. Cohen, "Rate of convergence of several conjugate gradient algorithms," *SIAM Journal on Numerical Analysis*, vol. 9, no. 2, p. 248–259, 1972.

[4] Y.-H. Dai, "Nonlinear conjugate gradient methods," *Wiley Encyclopedia of Operations Research and Management Science*, 2011.

[5] P. A. Parrilo, "Semidefinite programming relaxations for semialgebraic problems," *Mathematical Programming*, vol. 96, no. 2, pp. 293–320, May 2003. [Online]. Available: https://doi.org/10.1007/s10107-003-0387-5

[6] J. Lasserre, "A sum of squares approximation of nonnegative polynomials," *SIAM Review*, vol. 49, no. 4, pp. 651–669, 2007. [Online]. Available: https://doi.org/10.1137/070693709

[7] M. Laurent, *Sums of Squares, Moment Matrices and Optimization Over Polynomials*. New York, NY: Springer New York, 2009, pp. 157–270. [Online]. Available: https://doi.org/10.1007/978-0-387-09686-5_7

[8] P. A. Parrilo, *Polynomial Optimization, Sums of Squares, and Applications*. Society for Industrial and Applied Mathematics, 2013.

[9] Y. Drori and M. Teboulle, "Performance of first-order methods for smooth convex minimization: a novel approach," *Mathematical Programming*, vol. 145, no. 1, pp. 451–482, Jun 2014. [Online]. Available: https://doi.org/10.1007/s10107-013-0653-0

[10] E. de Klerk, F. Glineur, and A. B. Taylor, "On the worst-case complexity of the gradient method with exact line search for smooth strongly convex functions," *Optimization Letters*, vol. 11, no. 7, pp. 1185–1199, Oct 2017. [Online]. Available: https://doi.org/10.1007/s11590-016-1087-4

[11] A. B. Taylor, J. M. Hendrickx, and F. Glineur, "Exact worst-case convergence rates of the proximal gradient method for composite convex minimization," *Journal of Optimization Theory and Applications*, vol. 178, no. 2, pp. 455–476, Aug 2018. [Online]. Available: https://doi.org/10.1007/s10957-018-1298-1

[12] ——, "Smooth strongly convex interpolation and exact worst-case performance of first-order methods," *Mathematical Programming*, vol. 161, no. 1, pp. 307–345, Jan 2017. [Online]. Available: https://doi.org/10.1007/s10107-016-1009-3

[13] A. Beck, *The Proximal Gradient Method*. Society for Industrial and Applied Mathematics, 2017, ch. 10, pp. 269–329. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9781611974997.ch10

APPENDIX A
FULL DETAILS FOR GD WITH EXACT LINE SEARCH

Equation (16) can be fully expressed as

$$\begin{aligned}
p(\mathbf{z}) = \sum_{a,b} \mathbf{z}_1^a \mathbf{z}_1^b Q(a,b) &+ \lambda_1 \Big\{ f_k - f_{k+1} - \mathbf{g}_{k+1}^T(\mathbf{x}_k - \mathbf{x}_{k+1}) \\
&- \alpha \Big[\frac{1}{L}\left\|\mathbf{g}_k - \mathbf{g}_{k+1}\right\|^2 + \mu\|\mathbf{x}_k - \mathbf{x}_{k+1}\|^2 \\
&- 2\frac{\mu}{L}(\mathbf{g}_{k+1} - \mathbf{g}_k)^T(\mathbf{x}_{k+1} - \mathbf{x}_k)\Big]\Big\} \\
&+ \lambda_2 \Big\{ f_k - f_* - \alpha \Big[\frac{1}{L}\|\mathbf{g}_k\|^2 + \mu\|\mathbf{x}_k - \mathbf{x}_*\|^2 \\
&+ 2\frac{\mu}{L}\mathbf{g}_k^T(\mathbf{x}_* - \mathbf{x}_k)\Big]\Big\} \\
&+ \lambda_3 \Big\{ f_{k+1} - f_k - \mathbf{g}_k^T(\mathbf{x}_{k+1} - \mathbf{x}_k) - \alpha \Big[\frac{1}{L}\left\|\mathbf{g}_{k+1} - \mathbf{g}_k\right\|^2 \\
&+ \mu\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 - 2\frac{\mu}{L}(\mathbf{g}_k - \mathbf{g}_{k+1})^T(\mathbf{x}_k - \mathbf{x}_{k+1})\Big]\Big\} \\
&+ \lambda_4 \Big\{ f_{k+1} - f_* - \alpha \Big[\frac{1}{L}\left\|\mathbf{g}_{k+1}\right\|^2 \\
&+ \mu\|\mathbf{x}_{k+1} - \mathbf{x}_*\|^2 + 2\frac{\mu}{L}\mathbf{g}_{k+1}^T(\mathbf{x}_* - \mathbf{x}_{k+1})\Big]\Big\} \\
&+ \lambda_5 \Big\{ f_* - f_k - \mathbf{g}_k^T(\mathbf{x}_* - \mathbf{x}_k) - \alpha \Big[\frac{1}{L}\|\mathbf{g}_k\|^2 \\
&+ \mu\|\mathbf{x}_* - \mathbf{x}_k\|^2 - 2\frac{\mu}{L}\mathbf{g}_k^T(\mathbf{x}_k - \mathbf{x}_*)\Big]\Big\} \\
&+ \lambda_6 \Big\{ f_* - f_{k+1} - \mathbf{g}_{k+1}^T(\mathbf{x}_* - \mathbf{x}_{k+1}) - \alpha \Big[\frac{1}{L}\left\|\mathbf{g}_{k+1}\right\|^2 \\
&+ \mu\|\mathbf{x}_* - \mathbf{x}_{k+1}\|^2 - 2\frac{\mu}{L}\mathbf{g}_{k+1}^T(\mathbf{x}_{k+1} - \mathbf{x}_*)\Big]\Big\} \\
&+ \lambda_7 \Big\{ \mathbf{g}_{k+1}^T(\mathbf{x}_{k+1} - \mathbf{x}_k)\Big\} + \lambda_8 \Big\{ \mathbf{g}_{k+1}^T\mathbf{g}_k \Big\}
\end{aligned}$$

where $Q \succeq 0$, $\lambda_0, \ldots, \lambda_6 \geq 0$ and $\lambda_7, \lambda_8 \in \mathbb{R}$.

Without loss of generality, as discussed in Section III-D, we may assume that $n = 1$. Then $Q \in \mathbb{S}^9$. The set of linear equalities obtained from matching coefficients are as follows:

$$\begin{aligned}
& & 0 &= Q(1,1) \\
f_*: & & 1 - t &= 2Q(1,2) - \lambda_2 - \lambda_4 + \lambda_5 + \lambda_6 \\
f_k: & & t &= 2Q(1,3) + \lambda_1 + \lambda_2 - \lambda_3 - \lambda_5 \\
f_{k+1}: & & -1 &= 2Q(1,4) - \lambda_1 + \lambda_3 + \lambda_4 - \lambda_6 \\
& & 0 &= Q(1,i) \quad i = 5, \ldots, 9 \\
& & 0 &= Q(i,j) \quad i = 2, \ldots, 4,\ j = i, \ldots, 9 \\
x_*^2: & & 0 &= Q(5,5) - \alpha\mu\,(\lambda_2 + \lambda_4 + \lambda_5 + \lambda_6) \\
x_* x_k: & & 0 &= 2Q(5,6) + 2\alpha\mu\,(\lambda_2 + \lambda_5) \\
x_* x_{k+1}: & & 0 &= 2Q(5,7) + 2\alpha\mu\,(\lambda_4 + \lambda_6) \\
x_* g_k: & & 0 &= 2Q(5,8) - 2\alpha\frac{\mu}{L}\,(\lambda_2 + \lambda_5) - \lambda_5 \\
x_* g_{k+1}: & & 0 &= 2Q(5,9) - 2\alpha\frac{\mu}{L}\,(\lambda_4 + \lambda_6) - \lambda_6 \\
x_k^2: & & 0 &= Q(6,6) - \alpha\mu\,(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_5) \\
x_k x_{k+1}: & & 0 &= 2Q(6,7) + 2\alpha\mu\,(\lambda_1 + \lambda_3) \\
x_k g_k: & & 0 &= 2Q(6,8) + \lambda_3 + \lambda_5 \\
& & & \quad + 2\alpha\frac{\mu}{L}\,(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_5) \\
x_k g_{k+1}: & & 0 &= 2Q(6,9) - \lambda_1 - \lambda_7 - 2\alpha\frac{\mu}{L}\,(\lambda_1 + \lambda_3) \\
x_{k+1}^2: & & 0 &= Q(7,7) - \alpha\mu\,(\lambda_1 + \lambda_3 + \lambda_4 + \lambda_6) \\
x_{k+1} g_k: & & 0 &= 2Q(7,8) - 2\alpha\frac{\mu}{L}\,(\lambda_1 + \lambda_3) - \lambda_3 \\
x_{k+1} g_{k+1} & & 0 &= 2Q(7,9) + \lambda_1 + \lambda_6 + \lambda_7 \\
& & & \quad + 2\alpha\frac{\mu}{L}\,(\lambda_1 + \lambda_3 + \lambda_4 + \lambda_6) \\
g_k^2: & & 0 &= Q(8,8) - \frac{\alpha}{L}\,(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_5) \\
g_k g_{k+1}: & & 0 &= 2Q(8,9) + 2\frac{\alpha}{L}\,(\lambda_1 + \lambda_3) + \lambda_8 \\
g_{k+1}^2: & & 0 &= Q(9,9) - \frac{\alpha}{L}\,(\lambda_1 + \lambda_3 + \lambda_4 + \lambda_6).
\end{aligned} \tag{27}$$

The SDP minimizes t subject to these affine constraints as well as

$$\begin{aligned}
& Q \succeq 0 \\
& \lambda_i \geq 0 \quad i = 0, \ldots, 6 \\
& 0 \leq t \leq 1.
\end{aligned}$$

The optimal values of t and λ's correspond to

$$\begin{aligned}
t &= \left(\frac{L-\mu}{L+\mu}\right)^2 & \lambda_1 &= \frac{L-\mu}{L+\mu} & \lambda_5 &= 2\mu\frac{L-\mu}{(L+\mu)^2} \\
\lambda_6 &= \frac{2\mu}{L+\mu} & \lambda_7 &= -1 & \lambda_8 &= -\frac{2}{L+\mu}
\end{aligned}$$

and $\lambda_i = 0$ for $i = 2, \ldots, 4$. Note that these are all the variables we need to guess; Q is fully determined by t and the λ's through the set of linear equalities. The optimal Q is a sparse symmetric matrix, with the non-zero entries in the upper triangular portion being:

$$\begin{aligned}
Q(5,5) &= \frac{2L^2\mu^2}{(L+\mu)^2(L-\mu)} & Q(5,6) &= -\frac{L\mu^2}{(L+\mu)^2} \\
Q(5,7) &= -\frac{L\mu^2}{(L+\mu)(L-\mu)} & Q(5,8) &= \frac{L\mu}{(L+\mu)^2} \\
Q(5,9) &= \frac{L\mu}{(L+\mu)(L-\mu)} & Q(6,6) &= \frac{L\mu(L+3\mu)}{2(L+\mu)^2} \\
Q(6,7) &= -\frac{L\mu}{2(L+\mu)} & Q(6,8) &= -\frac{\mu(3L+\mu)}{2(L+\mu)^2} \\
Q(6,9) &= -\frac{\mu}{2(L+\mu)} & Q(7,7) &= \frac{L\mu}{2(L-\mu)} \\
Q(7,8) &= \frac{\mu}{2(L+\mu)} & Q(7,9) &= -\frac{\mu}{2(L-\mu)} \\
Q(8,8) &= \frac{L+3\mu}{2(L+\mu)^2} & Q(8,9) &= \frac{1}{2(L+\mu)} \\
Q(9,9) &= \frac{1}{2(L-\mu)}.
\end{aligned}$$

It may be verified by symbolic computation that Q is positive semidefinite, completing the proof that

$$f_{k+1} - f_* \leq \left(\frac{L-\mu}{L+\mu}\right)^2 (f_k - f_*).$$

APPENDIX B
SUPPLEMENTARY MATERIAL

Analysis details for the other two algorithms, as well as code for numerically and symbolically verifying the results, can be found at https://github.com/sandratsy/SumsOfSquares.