

# First-Order Algorithms for Polynomial Optimization

**Zoom Chow**

*Department of Electrical and Computer Engineering, UC San Diego*

**Theodore Fung**

*Department of Mathematics, UC San Diego*

ZCHOW@UCSD.EDU

TYFUNG@UCSD.EDU

## Abstract

Semidefinite programming (SDP) relaxations are a powerful approach for solving polynomial optimization problems (POPs), yet their scalability remains a challenge. This paper surveys advancements in first-order methods for SDP relaxations, focusing on inexact projected gradient methods (iPGM) as proposed by [Yang et al. \(2021\)](#). Their method integrates projected gradient updates with rounding and nonlinear lifting strategies to efficiently solve rank-one SDP relaxations, enabling faster convergence and improved scalability. We contextualize this approach by reviewing related methodologies, including regularization techniques by [Nie and Wang \(2012\)](#) and Alternating Direction Method of Multipliers (ADMM) based sum-of-squares methods by [Zheng et al. \(2019\)](#). To validate iPGM’s effectiveness, we reproduce numerical results from [Yang et al. \(2021\)](#) using the STRIDE algorithm and detail key proofs of the STRIDE algorithm. This study provides empirical insights into computational efficiency, convergence behavior, and solution accuracy, offering a comparative assessment of first-order methods for SDP relaxations in polynomial optimization.

**Keywords:** Polynomial Optimization · Convergence Analysis · Semidefinite Relaxation

## 1. Introduction

Polynomial optimization problems arise in various domains, including control theory, machine learning, and combinatorial optimization. These problems involve minimizing a polynomial objective function subject to polynomial equality and inequality constraints. Due to their inherent nonconvexity, a common approach is to relax the problem using semidefinite programming. However, solving large-scale SDP relaxations remains computationally challenging, particularly for high-dimensional problems.

### 1.1. Semidefinite Relaxations for Polynomial Optimization

A general polynomial optimization problem can be formulated as:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & p(x) \\ \text{subject to} \quad & g_i(x) \geq 0, \quad i = 1, \dots, m, \\ & f_j(x) = 0, \quad j = 1, \dots, k, \end{aligned} \tag{1}$$

where  $p, g_i, f_j$  are multivariate polynomials with real coefficients. The Lasserre hierarchy provides a systematic approach to approximate the solution using SDP relaxations. The first-order relaxation is given by:

$$\min_{X \succeq 0} \langle C, X \rangle \quad \text{subject to} \quad \mathcal{A}(X) = b, \tag{2}$$

where  $X$  is a moment matrix containing information about the original polynomial optimization problem.

## 2. Regularization for SDP Relaxations in Large-Scale POPs (Nie and Wang, 2012)

Solving large-scale SDP relaxations of polynomial optimization problems poses significant computational challenges due to memory constraints and numerical instability. Regularization techniques improve the efficiency and stability of SDP solvers by modifying the problem structure while preserving feasibility. This section details the regularization methods proposed by Nie and Wang (2012), particularly for SDP relaxations with block diagonal structures.

### 2.1. Block Structure in SDP Relaxations

Let  $\mathcal{K}$  denote the Cartesian product of several semidefinite cones:

$$\mathcal{K} = \mathbb{S}_+^{N_1} \times \cdots \times \mathbb{S}_+^{N_\ell}, \quad (3)$$

where  $\mathbb{S}_+^{N_i}$  represents the cone of  $N_i \times N_i$  symmetric positive semidefinite matrices. The feasible solution set belongs to the space:

$$\mathcal{M} = \mathbb{S}^{N_1} \times \cdots \times \mathbb{S}^{N_\ell}, \quad (4)$$

where each  $X \in \mathcal{M}$  is a block diagonal matrix  $X = (X_1, X_2, \dots, X_\ell)$ , satisfying  $X_i \succeq 0$ . The self-dual cone is given by:

$$\mathcal{K}^* = \{Y \in \mathcal{M} \mid \langle Y, X \rangle \geq 0, \quad \forall X \in \mathcal{K}\}. \quad (5)$$

For any symmetric matrix  $W$ , its projections onto the positive and negative semidefinite cones are defined via spectral decomposition:

$$(W)_+ = \sum_{\lambda_i > 0} \lambda_i u_i u_i^T, \quad (6)$$

$$(W)_- = \sum_{\lambda_i < 0} \lambda_i u_i u_i^T. \quad (7)$$

The block-diagonal structure of SDP relaxations for polynomial optimization motivates the development of specialized regularization techniques to handle large-scale instances efficiently.

### 2.2. General Conic Semidefinite Optimization Formulation

A general conic semidefinite optimization problem is given by:

$$\min_X \quad C \bullet X, \quad \text{s.t.} \quad A(X) = b, \quad X \in \mathcal{K}. \quad (8)$$

where  $C \in \mathcal{M}$ ,  $b \in \mathbb{R}^m$ , and  $A : \mathcal{M} \rightarrow \mathbb{R}^m$  is a linear operator. Its dual problem is:

$$\max_y \quad b^T y, \quad \text{s.t.} \quad A^*(y) + Z = C, \quad Z \in \mathcal{K}^*. \quad (9)$$

The SDP relaxations for polynomial optimization often exhibit a block diagonal structure, making traditional interior-point solvers computationally inefficient. To mitigate these challenges, two primary regularization techniques are introduced: Moreau-Yosida regularization and Augmented Lagrangian regularization.

### 2.3. Moreau-Yosida Regularization

The Moreau-Yosida regularization modifies the primal SDP problem by introducing a squared Frobenius norm penalty:

$$\min_{X, Y \in \mathcal{M}} C \bullet X + \frac{1}{2\sigma} \|X - Y\|^2, \quad \text{s.t.} \quad A(X) = b, \quad Y \in \mathcal{K}. \quad (10)$$

This formulation is equivalent to the original problem since the optimal  $Y$  satisfies  $Y = X$ , but it improves numerical stability by ensuring better-conditioned iterates.

### 2.4. Augmented Lagrangian Regularization

For the dual SDP problem, the augmented Lagrangian regularization takes the form:

$$\max_{y \in \mathbb{R}^m, Z \in \mathcal{M}} b^T y - (Z + A^*(y) - C) \bullet Y - \frac{\sigma}{2} \|Z + A^*(y) - C\|^2, \quad \text{s.t.} \quad Z \in \mathcal{K}. \quad (11)$$

When  $\mathcal{K} = \mathbb{S}_+^N$  consists of a single block, this is equivalent to solving:

$$\min_{X \in \mathcal{M}} C \bullet X + \frac{1}{2\sigma} \|X - Y\|^2 - y^T (A(X) - b) - Z \bullet X. \quad (12)$$

By fixing  $y$  and optimizing over  $Z$ , [Malick et al. \(2009\)](#) shows that this problem reduces to:

$$\max_{y \in \mathbb{R}^m} b^T y - \frac{\sigma}{2} \|A^*(y) - C + \frac{Y}{\sigma}\|_{\mathcal{K}}^2 + \frac{1}{2\sigma} \|Y\|^2. \quad (13)$$

### 2.5. Computational Benefits

Regularization techniques significantly improve the feasibility and numerical stability of large-scale SDP relaxations. The Moreau-Yosida and Augmented Lagrangian regularizations provide:

- Better-conditioned SDP constraints, reducing solver instability.
- Faster convergence in first-order methods, enabling efficient solutions for large  $N$ .
- Scalability to polynomial optimization problems with high-dimensional SDP relaxations.

These methods enhance the applicability of SDP relaxations in polynomial optimization, making them more suitable for real-world applications in control and combinatorial optimization.

## 3. Fast ADMM for Sum-of-Squares Optimization ([Zheng et al., 2019](#))

Sum-of-Squares (SOS) optimization reformulates polynomial optimization problems as semidefinite programs (SDPs), which can be computationally demanding for large-scale instances. Traditional solvers like SeDuMi and SDPT3 rely on interior-point methods, which do not scale efficiently. [Zheng et al. \(2019\)](#) propose a first order ADMM-based approach that exploits partial orthogonality in constraint matrices to enhance computational efficiency.

### 3.1. ADMM for Homogeneous Self-Dual Embedding

The proposed approach is based on solving the homogeneous self-dual embedding of the conic program:

$$\max_{y,z} \quad b^T y \quad \text{s.t.} \quad A^T y + z = c, \quad z \in K^*, \quad (14)$$

where  $K^*$  is the dual of the cone  $K$ . When strong duality holds, an optimal solution or a certificate of infeasibility can be extracted from the solution of the linear system:

$$\begin{bmatrix} 0 & -A^T & c \\ A & 0 & -b \\ -c^T & b^T & 0 \end{bmatrix} \begin{bmatrix} \xi \\ y \\ \tau \end{bmatrix} = \begin{bmatrix} z \\ s \\ \kappa \end{bmatrix}, \quad (15)$$

where  $(\xi, y, \tau) \in K \times \mathbb{R}^m \times \mathbb{R}_+$  and  $(z, s, \kappa) \in K^* \times \{0\}^m \times \mathbb{R}_+$ .

### 3.2. Iterative ADMM Updates

By defining:

$$u = \begin{bmatrix} \xi \\ y \\ \tau \end{bmatrix}, \quad v = \begin{bmatrix} z \\ s \\ \kappa \end{bmatrix}, \quad Q = \begin{bmatrix} 0 & -A^T & c \\ A & 0 & -b \\ -c^T & b^T & 0 \end{bmatrix}, \quad (16)$$

the homogeneous self-dual feasibility problem can be solved using a simplified ADMM update scheme:

$$\hat{u}^{(k)} = (I + Q)^{-1}(u^{(k-1)} + v^{(k-1)}), \quad (17)$$

$$u^{(k)} = \mathcal{P}_C \left( \hat{u}^{(k)} - v^{(k-1)} \right), \quad (18)$$

$$v^{(k)} = v^{(k-1)} - \hat{u}^{(k)} + u^{(k)}. \quad (19)$$

### 3.3. Exploiting Partial Orthogonality

For SOS optimization, the matrix  $Q$  is typically large, making direct factorization computationally prohibitive. Instead, [Zheng et al. \(2019\)](#) leverage the structure of the constraint matrix  $A$ , which exhibits partial orthogonality, to efficiently solve the linear system:

$$\begin{bmatrix} I & -A^T \\ A & I \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \end{bmatrix} = \begin{bmatrix} \hat{\omega}_1 \\ \hat{\omega}_2 \end{bmatrix}. \quad (20)$$

By eliminating  $\sigma_1$  from the second block equation, we obtain:

$$\sigma_1 = \hat{\omega}_1 + A^T \sigma_2, \quad (21)$$

$$(I + AA^T) \sigma_2 = -A \hat{\omega}_1 + \hat{\omega}_2. \quad (22)$$

Using the matrix inversion lemma :

$$(A + BC)^{-1} = A^{-1} - A^{-1}B(I + CA^{-1}B)^{-1}CA^{-1}, \quad (23)$$

the inverse of  $(I + AA^T)$  is computed as:

$$(I + AA^T)^{-1} = P^{-1} - P^{-1}A_1(I + A_1^T P^{-1}A_1)^{-1}A_1^T P^{-1}. \quad (24)$$

where  $P$  is a diagonal matrix and  $A_1$  contains the leading rows of  $A$  associated with the largest eigenvalues. Since  $P$  is diagonal, its inverse is computed efficiently, reducing the cost of solving  $(I + AA^T)\sigma_2$  to a small system of size  $t \times t$  (where  $t \ll m$ ), significantly improving efficiency.

#### 4. Projected Gradient Methods for Rank-One SDP Relaxations (Yang et al., 2021)

Solving SDP relaxations of POPs often results in solutions that do not satisfy the rank-one constraint needed for extracting feasible polynomial solutions. Yang et al. address this issue by introducing an Inexact Projected Gradient Method (iPGM) with rounding and nonlinear lifting strategies. Given a rank-one SDP relaxation:

$$\min_{X \succeq 0} \langle C, X \rangle \quad \text{subject to} \quad A(X) = b, \quad \text{rank}(X) = 1, \quad (25)$$

standard SDP solvers relax the rank constraint, leading to high-rank solutions. The iPGM method combines gradient-based updates with rounding and nonlinear lifting techniques to solve rank-one semidefinite relaxations efficiently. The method iteratively updates a positive semidefinite matrix  $X^k$  via:

$$X^k \approx \Pi_{\mathcal{F}_P} \left( X^{k-1} - \sigma_k C \right), \quad (26)$$

where  $\Pi_{\mathcal{F}_P}$  denotes projection onto the feasible set, and  $\sigma_k$  is a step size parameter. The key innovation is the rank-one lifting and rounding procedure, which ensures that the solution remains close to a rank-one matrix. The rounding step is performed by eigenvalue decomposition:

$$X = \sum_i \lambda_i v_i v_i^T. \quad (27)$$

The top eigenvector is selected to construct a rank-one approximation:

$$\hat{x} = \text{rounding}(v_1), \quad (28)$$

where the rounding function selects the best candidate  $\hat{x}$  that satisfies the original polynomial constraints. Yang et al. (2021) implement this method in the STRIDE algorithm, demonstrating that iPGM outperforms traditional SDP solvers in terms of computational efficiency and scalability, particularly for large-scale rank-one constrained SDP relaxations.

##### 4.1. Inexact Projected Gradient Method

Recall the primal semidefinite relaxation given by (2). Then, the Lagrangian dual of the problem is given by:

$$\max_{y \in \mathbb{R}^m, S \succeq 0} \langle b, y \rangle \quad \text{subject to} \quad \sum_{i=1}^m y_i A_i + S = C, \quad (29)$$

with  $(X^*, y^*, S^*)$  used to denote an optimal solution to (2) and (29). The natural questions that follow are how fast does the iPGM method seen in (26) converges to the optimal solution in (2).

**Theorem 1** *Let  $\{\epsilon_k\}$  be a nonnegative sequence such that  $\{k\epsilon_k\}$  summable and  $\{\sigma_k\}$  be a nondecreasing positive sequence. Further let any sequence  $\{(X^k, y^k, S^k)\} \in \mathbb{S}_+^n \times \mathbb{R}^m \times \mathbb{S}_+^n$  generated by (26) and for all  $k \geq 0$ ,  $\|y^k\| \leq M$  for some  $M > 0$  and satisfies:*

$$\|A(X^k) - b\| \leq \epsilon_k, \quad (30)$$

$$\sum_{i=1}^m y_i^k A_i + S^k - C - \frac{1}{\sigma_k} (X^k - X^{k-1}) = 0, \quad (31)$$

$$\langle X^k, S^k \rangle = 0, \quad (32)$$

$$\max\{\eta_p, \eta_d, \eta_g\} \leq \gamma,$$

where  $\gamma > 0$  is given and  $\eta_p, \eta_d, \eta_g$  are defined as follows:

$$\eta_p = \frac{\|A(X^k) - b\|}{1 + \|b\|}, \eta_d = \frac{\|\sum_{i=1}^m y_i^k A_i + S^k - C\|}{1 + \|C\|}, \eta_g = \frac{|\langle C, X^k \rangle - \langle b, y^k \rangle|}{1 + |\langle C, X^k \rangle| + |\langle b, y^k \rangle|}. \quad (33)$$

Then, for  $k \geq 1$ , the following holds:

$$\begin{aligned} -\|y^*\| \epsilon_k &\leq \langle C, X^k - X^* \rangle \leq \frac{1}{k} \left( \frac{1}{2\sigma_0} \|X^0 - X^*\|^2 + 2M \sum_{i=1}^k i \epsilon_i \right), \\ \|A(X^k) - b\| &\leq \epsilon_k \leq O\left(\frac{1}{k}\right), \\ \left\| \sum_{i=1}^m y_i^k A_i + S^k - C \right\| &\leq O\left(\frac{1}{\sqrt{k}\sigma_k}\right). \end{aligned}$$

**Lemma 2** If  $\sigma_k \geq O(\sqrt{k})$  for all  $k \geq 1$ , that is, if a sufficiently large lower bound is further imposed on  $\sigma_k$ . Then the implication from Theorem 1 can be written as:

$$\max \left\{ \langle C, X^k - X^* \rangle, \|A(X^k) - b\|, \left\| \sum_{i=1}^m y_i^k A_i + S^k - C \right\| \right\} \leq O\left(\frac{1}{k}\right), \quad \forall k \geq 1. \quad (34)$$

However, even with a sufficiently large lower bound for  $\sigma_k$ , the convergence of iPGM is still quite slow. To combat this issue, Yang et al. (2021) proposes a problem-dependent acceleration strategy to improve the computational time of large-scale SDPs. Algorithm 1 presents the pseudocode for the aforementioned acceleration technique on a standard POP with a binary constraint.

---

**Algorithm 1** Acceleration Technique for Binary POP

---

**Input** :  $r \geq 1, \bar{X}^k = \sum_i \lambda_i v_i v_i^T \in \mathbb{S}_+^n$  be the  $k$ -th iterate primal variable produced by iPGM.

**Output**:  $\hat{X}^k \in \mathbb{S}_+^n$ .

**begin**

**for**  $i \leftarrow 1$  **to**  $r$  **do**

$$v_i^k \leftarrow \frac{v_i}{v_i[1]}$$

        ▷ Projection Step

$$\bar{x}_i^k \leftarrow \text{sgn}(v_i[x])$$

$$\hat{x}_i^k \leftarrow \text{nlp}(\bar{x}_i^k)$$

        ▷ Local Search

**end**

$$\hat{x}^k \leftarrow \text{argmin}_{\hat{x}_i^k} p(\hat{x}_i^k)$$

$$\hat{X}^k \leftarrow [\hat{x}^k]_\kappa [\hat{x}^k]_\kappa^T$$

**end**

---

Although both the projection and local search are problem-dependent, the projection step can be discarded if the projection is proven to be too challenging. However, the same cannot be said about local search.

A crucial part of Algorithm 1 is the local search function denoted by nlp. Additionally, the usage of local search is heuristic and can only be supported by the numerical experiments seen in Section 5.2. Lastly, there will be instances where Algorithm 1 can fail to produce any feasible option with respect the output of iPGM; however, this will not have any negative effects to Algorithm 2 in regards to its convergence.

## 4.2. Spectrahedron Inexact Projected Gradient Descent Along Vertices

Recall from 4.1, the convergence of iPGM can be rather slow. As a result, a problem-dependent acceleration algorithm is employed to improve computational time of large-scale SDPs. In this section, we will introduce pseudocode for Spectrahedron Inexact Projected Gradient Descent Along Vertices (STRIDE) in Algorithm 2, along with its convergence analysis in Theorem 4.

---

### Algorithm 2 Spectrahedron Inexact Projected Gradient Descent Along Vertices (STRIDE)

---

**Input** :  $(X^0, y^0, S^0) \in \mathbb{S}_+^n \times \mathbb{R}^m \times \mathbb{S}_+^n$ ,  $\eta = \max\{\eta_p, \eta_d, \eta_g\}$ ,  $\delta > 0$ ,  $m > 0$ ,  $r \in [1, n]$ ,  $\{\epsilon_k\}$  be a nonnegative sequence such that  $\{k\epsilon_k\}$  summable and  $\{\sigma_k\}$  be a nondecreasing positive sequence.

**Output**:  $(X^k, y^k, S^k) \in \mathbb{S}_+^n \times \mathbb{R}^m \times \mathbb{S}_+^n$ .

**begin**

```

 $X^k \leftarrow 0, \mathcal{V} \leftarrow \emptyset.$ 
if  $X^0$  in feasible set of (2) then
    |  $\mathcal{V} \leftarrow \mathcal{V} \cup \{X^0\}$ 
end
for  $k \leftarrow 1$  to  $m$  do
    |  $\bar{X}^k \leftarrow$  (26) satisfying (30) - (32).
    | if  $\eta < \delta$  then
    | | Output  $(\bar{X}^k, y^k, S^k)$ 
    | | Break
    | end
    |  $\hat{X}^k \leftarrow$  Algorithm 1 with  $r$  and  $\bar{X}^k$  as its parameters.
    | if  $\langle C, \hat{X}^k \rangle < \min\{\min_X \{\langle C, X \rangle \mid X \in \mathcal{V}\}, \langle C, \bar{X}^k \rangle\}$  then
    | |  $X^k \leftarrow \hat{X}^k$ 
    | else
    | |  $X^k \leftarrow \bar{X}^k$ 
    | end
    | if  $X^k = \hat{X}^k$  then
    | |  $\mathcal{V} \leftarrow \mathcal{V} \cup \{\hat{X}^k\}$ 
    | end
end

```

**end**

---

**Remark 3** In STRIDE, a safeguarding policy is implemented and can be seen after Algorithm 1 is called in Algorithm 2. This policy ensures a strict decrease of our objective value of the original SDP seen in (2). We now proceed to Theorem 4 where the convergence analysis of STRIDE is covered.

**Theorem 4** Let  $\{(\bar{X}^k, y^k, S^k)\} \in \mathbb{S}_+^n \times \mathbb{R}^m \times \mathbb{S}_+^n$  be any sequence generated by Algorithm 2 and there exist some  $M > 0$  such that  $\|y^k\| \leq M$  for all  $k \geq 0$ . If strong duality holds and  $\sigma_k \geq O(\sqrt{k})$  for all  $k \geq 1$ , then  $\{\langle C, \bar{X}^k \rangle\}$  converges to the optimal value of (2).

**Proof** Immediately, we see that the proof of convergence for Algorithm 2 has two initialization cases for the set  $\mathcal{V}$  (i.e.  $\mathcal{V} = \emptyset$  and  $\mathcal{V} = \{X^0\}$ ). We will consider the  $\mathcal{V} = \emptyset$  initialization case as the  $\mathcal{V} = \{X^0\}$  initialization case can be found in Theorem 2 of Yang et al. (2021).

Suppose  $|\mathcal{V}| = \infty$ , then we have an infinite sequence of feasible solutions to (2), and the objective value of (2) converges to  $-\infty$ . Equivalently, (2) is now unbounded and a contradiction is reached as strong duality holds by assumption.

Suppose  $|\mathcal{V}| < \infty$ , then there are 2 subcases to consider.

1. Let us consider  $|\mathcal{V}| = 0$ , or equivalently, all computed  $\hat{X}^k$  from Algorithm 1 has been denied by the safeguarding policy. Then STRIDE will simply proceed as the iPGM which converges with  $O(\frac{1}{k})$  to the optimal solution to (2) by Theorem 1 and Lemma 2.
2. Let us consider  $|\mathcal{V}| > 0$ , or equivalently, some computed  $\hat{X}^k$  have been accepted by the safeguarding policy. Then STRIDE will proceed as the iPGM with the latest accepted  $X^k$  along with its dual variables as its initial point. Again, this will converge to optimal solution to (2) with  $O(\frac{1}{k})$  by Theorem 1 and Lemma 2.

■

#### 4.3. The Projection Subproblem

Recall the Inexact Projected Gradient Method (iPGM), a natural question is to ask what methods can be employed to solve the subproblem seen in (26). This problem can be formulated as follows

$$\min_X \left\{ \frac{1}{2} \|X - Z\|^2 \mid X \text{ is a feasible solution to (2)} \right\}, \quad (35)$$

$$\max_{W, \xi} \left\{ \frac{1}{2} \left\| W + \sum_{i=1}^m \xi_i A_i + Z \right\|^2 - \langle b, \xi \rangle \mid \xi \in \mathbb{R}^m, W \in \mathbb{S}_+^n \right\}, \quad (36)$$

where (35) is the primal subproblem and (36) is the dual subproblem. Informally speaking, (35) is equivalent to finding the matrix  $X$  such that it is both feasible and the closest approximate to a given matrix  $Z$  from a Frobenius norm sense.

While Dykstra's projection algorithm can solve (35), its computational speed is rather undesirable. As a result, we will consider the dual problem (36), where the problem can be solved much more efficiently. Yang et al. (2021) proposes a two-phase algorithm to solve (36), which consist of a symmetric Gauss-Seidel based accelerated proximal gradient method and a modified limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm, which can be located in Algorithm 4 and Algorithm 5, respectively. Lastly, we will study the convergence result the symmetric Gauss-Seidel based proximal gradient method in Theorem 5.

**Theorem 5** *Let  $(W^*, \xi^*) \in \mathbb{S}_+^n \times \mathbb{R}^m$  be an optimal solution to (36) with  $\det(AA^*) \neq 0$ . Consider the sequence  $\{W^k, \xi^k\}$  generated by Algorithm 4 in Yang et al. (2021), then*



$$0 \leq f(W^k, \xi^k) - f(W^*, \xi^*) \leq O\left(\frac{1}{k^2}\right), \quad (37)$$

where  $f$  is defined identically as in Section 4.1 of [Yang et al. \(2021\)](#).

**Proof** Let  $g, q_k$  also be defined identically as in Section 4.1 of [Yang et al. \(2021\)](#), then the conditions (11) and (12) in [Jiang et al. \(2012\)](#) holds for all  $k$ . Furthermore, define a linear operator  $H_k$  as follows:

$$H_k := (D + U)D^{-1}(D + U^*), \quad (38)$$

where  $U, D$  are defined as follows:

$$U := \begin{bmatrix} 0 & A^* \\ 0 & 0 \end{bmatrix} \quad D := \begin{bmatrix} I & 0 \\ 0 & AA^* \end{bmatrix} \quad (39)$$

This choice of  $H_k \succ 0$  for all  $k$  immediately follows from  $AA^*$  is nonsingular by assumption. Then, from Theorem 2.1 in [Jiang et al. \(2012\)](#), there exists a constant  $c \geq 0$  such that

$$0 \leq f(W^k, \xi^k) - f(W^*, \xi^*) \leq \frac{c}{(k+1)^2}. \quad (40)$$

■

## 5. Outlier Robust Wahba Problem ([Yang et al., 2021](#)), ([Yang and Carlone, 2019](#))

The main motivation behind the implementation of STRIDE is the numerical scalability of the algorithm on large number of constraints when dealing with SDPs. In this section, we will cover the quaternion-based outlier robust Wahba problem, which will give us better insight at the scalability of STRIDE when benchmarked against traditional solvers such as MOSEK and SeDuMi where an interior point method is employed.

To begin, the Wahba problem is a rotational matrix estimation problem which is defined as follows:

$$\min_{R \in \text{SO}3} \sum_i w_i \|v_i - Ru_i\|^2, \quad (41)$$

where the set  $\text{SO}3 = \{R \in \mathbb{R}^{3 \times 3} \mid RR^T = R^T R = I, \det(R) = 1\}$ ,  $w_i^2$  are weights known a priori with respect to  $(v_i, u_i) \in \mathbb{R}^3 \times \mathbb{R}^3$ . Informally, this problem requires one to determine a rotation matrix  $R$  such that it minimizes the distance between all given pairs of vectors  $v_i$  and rotated  $u_i$  in a squared Euclidean norm sense.

### 5.1. Quaternion-based Outlier Robust Wahba Problem

Let us consider an equivalent problem using quaternions along with the addition of outliers from some given data instead. Then, (41) can be formulated as follows:

$$\min_{q \in S^3} \sum_i \min \left\{ \frac{\|\hat{v}_i - q \odot \hat{u}_i \odot q^{-1}\|^2}{c_i^2}, 1 \right\}, \quad (42)$$

where the set  $\mathcal{S}^3 = \{q \in \mathbb{R}^4 \mid \|q\| = 1\}$ ,  $\hat{v}_i = [v_i^T, 0]^T$  and  $\hat{u}_i$  is defined similarly. Lastly, the operation  $\odot$  denotes the standard quaternion product and  $q^{-1}$  denotes the standard inverse quaternion.

We can further reformulate (42) to be binary constrained by introducing  $\theta_i \in \{-1, +1\}$  to label inliers from outliers. For outliers, we can pick  $\theta_i = -1$  and  $\theta_i = 1$  for inliers, then (42) becomes:

$$\min_{q \in \mathcal{S}^3, \theta_i \in \{\pm 1\}} \sum_i \frac{1 + \theta_i}{2} \frac{\|\hat{v}_i - q \odot \hat{u}_i \odot q^{-1}\|^2}{c_i^2} + \frac{1 - \theta_i}{2}. \quad (43)$$

Then, utilizing the binary cloning technique and an SDP reformulation discussed in Proposition 3 to Proposition 5 of Yang and Carbone (2019). The problem seen in (43) can be reformulated into an semidefinite program as follows:

$$\begin{aligned} \min_{Z \succeq 0} \quad & \text{tr}(QZ) \\ \text{subject to} \quad & ([Z]_{qq}) = 1 \\ & [Z]_{q_i q_i} = [Z]_{qq}, \quad \forall i \\ & \text{rank}(Z) = 1, \end{aligned} \quad (44)$$

where  $Q, Z, q_i, q$  are defined identically as in Proposition 3 to Proposition 5 of Yang and Carbone (2019). Finally, we will relax the rank 1 constraint seen in (44), and the resulting problem becomes:

$$\begin{aligned} \min_{Z \succeq 0} \quad & \text{tr}(QZ) \\ \text{subject to} \quad & ([Z]_{qq}) = 1 \\ & [Z]_{q_i q_i} = [Z]_{qq}, \quad \forall i. \end{aligned}$$

## 5.2. Numerical Experiments

In this section, we benchmarked STRIDE against proprietary and open source solvers such as MOSEK and SeDuMi in two numerical experiments using 50 and 100 data points, which we will denote as Wahba 50 and Wahba 100, respectively. Additionally, we utilized a heuristic method proposed by Yang et al. (2021) called Graduated Non-Convexity (GNC) along with STRIDE which significantly decrease computational time as the solution is not certified for optimality. Despite this lack of certification, it is able to compute the optimal solution with an extremely high success rate.

| Solver     | Wahba 50 Time (Seconds) | Wahba 100 Time (Seconds) |
|------------|-------------------------|--------------------------|
| STRIDE+GNC | 1.684                   | 22.581                   |
| STRIDE     | 7.562                   | 46.270                   |
| MOSEK      | 13.086                  | 488.003                  |
| SeDuMi     | 1011.400                | OOM                      |

Table 1: Time Consumption for Wahba 50 and Wahba 100

From Table 1, we observed that the computational time needed for SeDuMi to solve a relatively small problem is almost 2 order-of-magnitude higher than MOSEK; for a relatively medium sized problem with a  $404 \times 404$  moment matrix with 31301 equality constraints (Wahba 100), MOSEK fails to compute a solution within the same order-of-magnitude as STRIDE, and SeDuMi faces severe numerical instability issues and out-of-memory(OOM) errors for both problems.

## Acknowledgments

We thank Professor Yang Zheng and Teaching Assistant Feng-Yi Liao

## References

- Kaifeng Jiang, Defeng Sun, and Kim-Chuan Toh. An inexact accelerated proximal gradient method for large scale linearly constrained convex sdp. *SIAM Journal on Optimization*, 22(3):1042–1064, 2012. doi: 10.1137/110847081.
- Jérôme Malick, Janez Povh, Franz Rendl, and Angelika Wiegele. Regularization methods for semidefinite programming. *SIAM Journal on Optimization*, 20(1):336–356, 2009. doi: 10.1137/070704575. URL <https://doi.org/10.1137/070704575>.
- Jiawang Nie and Li Wang. Regularization methods for sdp relaxations in large-scale polynomial optimization. *SIAM Journal on Optimization*, 22(2):408–428, 2012. doi: 10.1137/110825844. URL <https://doi.org/10.1137/110825844>.
- Heng Yang and Luca Carlone. A quaternion-based certifiably optimal solution to the wahba problem with outliers. *arXiv preprint*, arXiv:1905.12536, 2019.
- Heng Yang, Ling Liang, Luca Carlone, and Kim-Chuan Toh. An inexact projected gradient method with rounding and lifting by nonlinear programming for solving rank-one semidefinite relaxation of polynomial optimization. 2021. URL <https://arxiv.org/abs/2105.14033>.
- Yang Zheng, Giovanni Fantuzzi, and Antonis Papachristodoulou. Fast admm for sum-of-squares programs using partial orthogonality. *IEEE Transactions on Automatic Control*, 64(9):3869–3876, September 2019. ISSN 2334-3303. doi: 10.1109/tac.2018.2886170. URL <http://dx.doi.org/10.1109/TAC.2018.2886170>.