On semidefinite programming under perturbations with unknown boundaries

Jonathan D. Hauenstein * Tingting Tang †

March 26, 2018

Abstract

This paper shows that solving a family of semidefinite programming problems (SDPs) under affine perturbations can be converted to solving a system of quasilinear partial differential equations (PDEs) utilizing the Davidenko differential equations within the *a priori* unknown maximal perturbation set. We develop a second-order sweeping Euler scheme to simultaneously approximate the boundary of the maximal set of perturbations and solve the SDPs within this set. We prove local and global error bounds for this second-order sweeping Euler scheme and demonstrate results on several examples.

1 Introduction

A semidefinite program (SDP) is the problem of optimizing a linear objective function over a linear slice of the cone of semidefinite matrices, called a spectrahedron [5]. Semidefinite programs can be efficiently solved both theoretically and practically [24] with applications of SDPs including control theory and combinational optimization [2, 6]. There are many active research areas related to semidefinite programming such as the development and advancement in convex programming including interior point and simplex methods [2, 13, 16, 19]. Another one is the theoretical and practical use of semidefinite programming relaxations of hard combinational optimization problems [7]. The intersection of optimization, algebraic geometry, and convex geometry, is an emerging area known as convex algebraic geometry, e.g., see [5].

In addition to the increasing number of practical applications of semidefinite programming, problems involving semidefinite programming under perturbations is of increasing interest and is the focus of this article. There are a number of reasons to introduce uncertainties in the coefficients of a semidefinite program. Several of them are listed in [4] including uncertainty about the future, errors in the data, model uncertainty, and implementation errors. Different approaches have been developed over the past twenty years to handle the problem of semidefinite programs under perturbations from different angles. The papers [4, 14] developed notions for so-called robust semidefinite programming and showed how to compute robust solutions given a known set of perturbations under certain conditions. Stochastic semidefinite programming has been introduced to work with random perturbations and probabilistic satisfaction of the linear matrix inequality constraints [3]. Sensitivity analysis of the central solutions to the changes on the right hand side of the constraints are investigated in [20, 21] under the assumption that the perturbations are infinitesimal.

To the best of our knowledge, no studies have been done to compute the boundary of the maximal set of perturbations within which the semidefinite problem is feasible nor sensitivity studies with perturbations in the coefficients of the objective and constraints. These two topics (computing maximal set of perturbations and sensitivity of solutions) motivate our study as they arise in many applications. The key aspect of our approach is replacing solving SDPs with solving quasilinear partial differential equations which can be solved using efficient numerical schemes.

To formulate SDPs with affine perturbations under consideration, let $C, D_j, A_i, E_{ij} \in \mathbb{R}^{n\times n}$ be symmetric matrices and $b \in \mathbb{R}^k$. For two symmetric matrices $M, N \in \mathbb{R}^{n\times n}$, define the operator $\bullet$ via

$$M \bullet N = \operatorname{trace}(M \cdot N)$$

*Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, Notre Dame, IN 46556 (hauenstein@nd.edu, www.nd.edu/~jhauenst). This author was supported in part by ONR N00014-16-1-2722 and Sloan Research Fellowship BR2014-110 TR14.

†Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, Notre Dame, IN 46556 (ttang@nd.edu, sites.nd.edu/tingting-tang). This author was supported in part by ONR N00014-16-1-2722.

and say that $M \succeq 0$ if M is a positive semidefinite matrix, i.e., all eigenvalues of M are nonnegative. For the perturbation vector $\delta \in \mathbb{R}^\ell$, we consider the following primal SDP with affine perturbations:

$$\begin{aligned} \text{minimize} \quad & \left(C + \sum_{j=1}^{\ell} \delta_j D_j\right) \bullet X \\ \text{s.t.} \quad & \left(A_i + \sum_{j=1}^{\ell} \delta_j E_{ji}\right) \bullet X = b_i, \quad \text{for } i = 1, \ldots, k \\ & X \succeq 0. \end{aligned} \tag{1}$$

The dual SDP can be written as the following:

$$\begin{aligned} \text{maximize} \quad & m \\ \text{s.t.} \quad & m = b^T y, \\ & S = C - \sum_{i=1}^{k} y_i A_i + \sum_{j=1}^{\ell} \delta_j \left(D_j - \sum_{i=1}^{k} y_i E_{ji}\right), \\ & S \succeq 0. \end{aligned} \tag{2}$$

Assuming, for example, that the feasible regions for both (1) and (2) when $\delta = 0$ have nonempty interior, we seek to find the maximal set of perturbations where the feasible regions still have nonempty interior and approximate the solution at any point within such set. Since solving the primal (1) and dual (2) problems are equivalent when the feasible regions have nonempty interior, they can be referred to interchangeably.

The remainder of this paper is organized as follows. In Section 2, we show how to convert (1) and (2) into a system of well-posed quasilinear partial differential equations (PDEs). In Section 3, we develop a second-order sweeping Euler scheme to efficiently approximate the solution to the PDEs along characteristics and compute the boundary of the maximal perturbation set simultaneously. Local and global bounds are provided in Theorem 7. Section 4 considers examples which numerically demonstrate the second-order convergence and compares it to a first-order finite difference scheme as well as solving using a popular SDP solver. Finally, we summarize our results and give some concluding remarks in Section 5.

2 Reformulating SDPs with Perturbations

We first reformulate solving (1) and (2) into the standard first-order Karush-Kuhn-Tucker (KKT) necessary conditions for optimality, namely, $X, S \succeq 0$ and

$$F(m, y, S, X; \delta) = \begin{pmatrix} m - b^T y \\ \left(A_i + \sum_{j=1}^{\ell} \delta_j E_{ji}\right) \bullet X - b_i, \quad \text{for } i = 1, \ldots, k \\ C - \sum_{i=1}^{k} y_i A_i + \sum_{j=1}^{\ell} \left(D_j - \sum_{i=1}^{k} y_i E_{ji}\right) - S \\ S \cdot X \end{pmatrix} = 0. \tag{3}$$

In particular, if δ such that both (1) and (2) have feasible regions with nonempty interior, then the KKT conditions are both necessary and sufficient conditions for optimality.

As formulated, the system F in (3), which is parameterized by $\delta \in \mathbb{R}^\ell$, consists of $1 + k + 2n^2$ equations in $1 + k + n + n^2$ variables. The overdeterminedness of F arises from the last two families of equations, which are matrix equations. Since all matrices are symmetric, the first matrix equation only yields $(n^2 + n)/2$ distinct equations. The second matrix equation, corresponding to the complementary slackness condition, namely $S \cdot X = 0$, can be handled in various ways. For example, one approach is to simply take the $(n^2 + n)/2$ equations corresponding to the upper triangular part. Another approach is to take the upper triangular part of a symmetrized version of this equation, namely

$$\frac{1}{2}(S \cdot X + X \cdot S) = 0.$$

For simplicity, our examples utilize the first approach thus yielding a well-constrained system of $1 + k + n + n^2$ equations in $1 + k + n + n^2$ variables which we also write as $F(m, y, S, X; \delta) = 0$.

To simplify the presentation, we will let $c = 1 + k + n + n^2$ and let V be the vector of length c corresponding to the variables of F, namely

$$V = \begin{pmatrix} m \\ y \\ S(:) \\ X(:) \end{pmatrix},$$

where

$$S(:) = (s_{11}, s_{12}, \ldots, s_{1n}, s_{22}, \ldots, s_{2n}, \ldots, s_{nn})^T \quad \text{and} \quad X(:) = (x_{11}, x_{12}, \ldots, x_{1n}, x_{22}, \ldots, x_{2n}, \ldots, x_{nn})^T.$$

Suppose that $\delta = 0$ is such that both (1) and (2) have feasible regions with nonempty interior. For the corresponding solution $V(0)$ to the KKT system, suppose that $J_V F(V(0); 0)$ is invertible where $J_Z G$ is the Jacobian matrix of a system G with respect to Z. The implicit function theorem shows that there is a neighborhood $U \subset \mathbb{R}^\ell$ of 0 and an analytic function $V(\delta)$ on U satisfying

$$F(V(\delta); \delta) = 0. \tag{4}$$

In particular, for $j = 1, \ldots, \ell$, the Davidenko differential equation associated with (4) shows that

$$\frac{\partial V}{\partial \delta_j} = -(J_V F)^{-1} J_{\delta_j} F. \tag{5}$$

Remark 1. *For $j = 1, \ldots, \ell$, the sensitivity of the solution to the SDPs* (1) *and* (2) *to each perturbation parameter δ_j is described by* (5).

We seek to understand the maximal perturbation neighborhood $U \subset \mathbb{R}^\ell$ of 0 for which $V(\delta)$ exists, is analytic, and solves the KKT system on U as schematically shown in Figure 1.

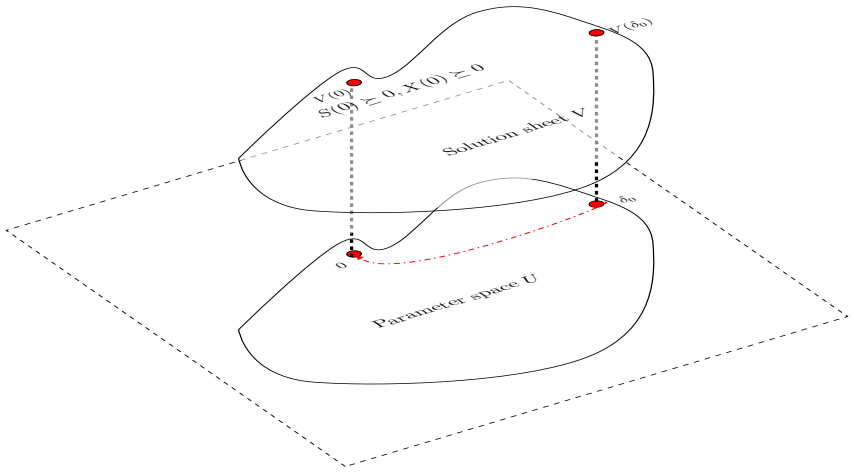


Figure 1: Illustration of solving $F(V(\delta); \delta) = 0$ with solution sheet $V(\delta)$ on maximal perturbation set U

Theorem 2. *Suppose that $\delta = 0$ is such that both* (1) *and* (2) *have feasible regions with nonempty interior with corresponding solution $V(0)$ to the KKT system such that $J_V F(V(0); 0)$ is invertible. Then, there is a set $U \subset \mathbb{R}^\ell$, the maximal perturbation neighborhood of* 0*, and a unique analytic function $V(\delta)$ satisfying the KKT conditions on U where U is the maximal connected subset of*

$$\left\{\delta \in \mathbb{R}^\ell \;\middle|\; \text{there exists } V = (m, y, S(:), X(:))^T \in \mathbb{R}^c \text{ with } S, X \succeq 0, F(V; \delta) = 0, \det J_V F(V; \delta) \neq 0\right\}$$

containing the origin, which is an open subset of $\mathbb{R}^\ell$.

Proof. By convexity, if $\delta \in \mathbb{R}^\ell$ such that there exists $V(\delta) = (m, y, S(:), X(:))^T \in \mathbb{R}^c$ with $S, X \succeq 0$, $F(V(\delta); \delta) = 0$, and $\det J_V F(V(\delta); \delta) \neq 0$, then $V(\delta)$ is the unique solution to the KKT system at δ yielding the solution to (1) and (2). Hence, the implicit function theorem yields a local neighborhood of analyticity for V satisfying (5). On a connected subset of such δ, we know that this must be the same analytic function $V(\delta)$ satisfying (5). The openness of U follows immediately from $\det J_V F \neq 0$. □

With this reduction, we aim to simultaneously compute $V(\delta)$ on U as well as the boundary, ∂U, of U. Our approach for this computation is to utilize derivative information of the solution and recover the solution along different perturbation directions. In particular, we use the Davidenko differential equations (5) to further convert the problem of solving a system of polynomials with changing coefficients to solve a system of first-order quasilinear partial differential equations with the perturbation parameters as variables. Moreover, we view the solution along different perturbation directions as solving a system of PDEs along its characteristics.

It is important to keep in mind the power of the Davidenko differential equations (5) during this process: we are able to obtain the derivative along any direction at any point in U using these equations. Consequently, we can establish the desired system of PDEs giving the solution to the SDPs (1) and (2) in U with its characteristics along any directions from the Davidenko differential equations.

For $\delta \in \mathbb{R}^\ell$, define

$$\operatorname{dist}(\delta, \partial U) = \min_{y \in \partial U} \|y - \delta\|$$

where $\|\cdot\|$ is the Euclidean norm on $\mathbb{R}^\ell$. For $\epsilon > 0$, consider a connected open set $W_\epsilon \subset U$ with smooth boundary such that

$$W_\epsilon \subset \{\delta \in U \mid \operatorname{dist}(\delta, \partial U) > \epsilon\} \subset U$$

as illustrated in Figure 2. The value of $\epsilon > 0$ will be selected according to numerical requirements on how accurate one aims to approximate the boundary ∂U. In particular, since W_ϵ is an open subset of U whose boundary, ∂W_ϵ, is smooth and strictly contained in U, we can compute the analytic function $V(\delta)$ on the closure of W_ϵ, namely $\overline{W}_\epsilon = W_\epsilon \cup \partial W_\epsilon$. The boundary condition, say $G(\delta)$, is simply restricting $V(\delta)$ to ∂W_ϵ. The following theorem computes $V(\delta)$ on W_ϵ via a system of first-order quasilinear PDEs using the method of characteristics.

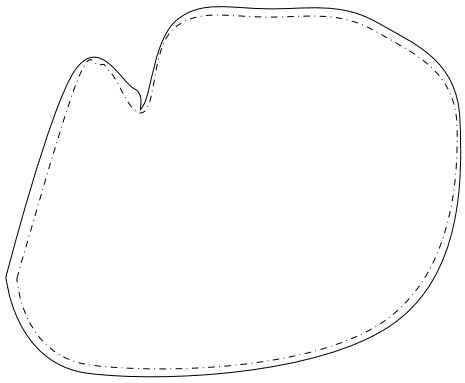


Figure 2: Illustration of W_ϵ with a smooth boundary (dashed) inside of U which could possibly have a nonsmooth boundary (solid).

Theorem 3. *For almost every nonzero $\alpha \in \mathbb{R}^\ell$, with the setup above, there exists a unique solution to the system*

$$\begin{cases} \dfrac{\partial \mathcal{V}}{\partial \delta} \cdot \alpha &= H(V(\delta), \delta) & \text{in } W_\epsilon \\ \mathcal{V}\Big|_{\Gamma_\epsilon} &= G & \text{in } \Gamma_\epsilon = \partial W_\epsilon, \end{cases} \tag{6}$$

where

$$H(V(\delta), \delta) = \begin{pmatrix} \dfrac{\partial V_1}{\partial \delta} \cdot \alpha \\ \vdots \\ \dfrac{\partial V_c}{\partial \delta} \cdot \alpha \end{pmatrix}.$$

The unique solution $\mathcal{V}(\delta)$ is equal to $V(\delta)$ and solves the SDPs (1) *and* (2) *for* $\delta \in \overline{W}_\epsilon = W_\epsilon \cup \partial W_\epsilon$.

Proof. Since V is analytic from Theorem 2, we know $H \in C^1(W_\epsilon)$ and $G \in C^1(\Gamma_\epsilon)$. Since, for $\ell = 2$, this result was established in [11], we only need to extend this result for $\ell > 2$.

First, we notice from (6) that the PDEs have identical projected characteristic manifold which makes it readily possible to use method of characteristics. Since Γ_ϵ is smooth, for any fixed $\delta^0 \in \Gamma_\epsilon$, there exists a neighborhood Q_{δ^0} of δ^0 such that we can parameterize Γ_ϵ in Q_{δ^0} locally by

$$\Gamma_{\epsilon,\delta^0}(r) = (\gamma_1(r), \ldots, \gamma_\ell(r))$$

where $r \in \mathbb{R}^{\ell-1}$ with $\Gamma_{\epsilon,\delta^0}(0) = \delta^0$. The corresponding system of ODEs to (6) can be rewritten as

$$\begin{cases} \dfrac{d\delta_1}{ds} = \alpha_1 \\ \quad\vdots \\ \dfrac{d\delta_\ell}{ds} = \alpha_\ell \\ \dfrac{dV}{ds} = H, \end{cases} \qquad \begin{cases} \delta_1(0,r) = \gamma_1(r) \\ \quad\vdots \\ \delta_\ell(0,r) = \gamma_\ell(r) \\ V(0,r) = G(r). \end{cases} \tag{7}$$

From Picard-Lindelöf Theorem [10], we know that there exists a unique solution $V(\delta(s,r))$ to the ODE system in Q_{δ^0}. When $r = 0$, the solution to (7) gives us a solution of system (6) along the characteristic curve that crosses Γ_ϵ at δ^0 [15]. If the characteristic is parallel to Γ_ϵ at δ^0, we only obtain the solution at δ^0. So, let's assume that Γ_ϵ is non-characteristic at δ^0. Notice that using the first ℓ equations from (7), we can solve for δ in terms of s and r:

$$\begin{aligned} \delta_1 &= \alpha_1 s + \gamma_1(r) \\ &\vdots \\ \delta_\ell &= \alpha_\ell s + \gamma_\ell(r) \end{aligned} \tag{8}$$

Let $X = (s, r)$. Algebraically, using the inverse function theorem, we need $\det J_X \delta(0) \neq 0$. Describing this condition geometrically, this is equivalent to the characteristic curve not being parallel to Γ_ϵ at δ^0. Since Γ_ϵ is a $\ell - 1$ dimensional compact hypersurface with a null boundary and α determines a family of parallel lines, say Ω_α, in ℓ dimensional space, it is clear that for almost every α, there is at most a set with Lebesgue measure zero on Γ_ϵ in which an element is tangent to some line in Ω_α (see Figure 3 for illustration). Hence, we can obtain a solution $V(s(\delta), r(\delta))$ of (6) for δ in a small neighborhood, P_{δ^0}, of δ^0. By letting δ_0 vary, we can find the solution to (6) in some neighborhood of Γ_ϵ. Then, we can generate a new boundary condition Γ'_ϵ which is contained in W_ϵ. Iterating this process above, we obtain a solution to (6) which covers the whole space $\overline{W}_\epsilon$.

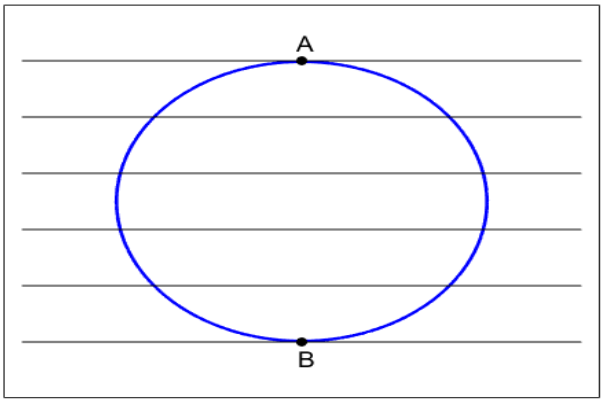


Figure 3: Illustration of Γ_ϵ and a family Ω_α of parallel lines where there are two points, A and B, on Γ_ϵ at which an element of Ω_α is parallel to Γ_ϵ.

Next, we show the solution is unique. Assuming that V and $\mathcal{V}$ are two different solutions to (6), let $Z = V - \mathcal{V}$. Then, Z satisfies the following system:

$$\begin{aligned} \frac{\partial Z}{\partial \delta} \cdot \alpha &= 0, \qquad \text{in } W_\epsilon, \\ Z\Big|_\Gamma &= 0, \qquad \text{in } \Gamma = \partial W_\epsilon. \end{aligned} \tag{9}$$

A similar argument as above provides that $Z \equiv 0$ which contradicts the assumption that $\mathcal{V}$ is different from V thereby showing uniqueness. Hence, the unique solution is the solution $V(\delta)$ to the KKT system associated with the SDPs (1) and (2). □

Remark 4. *One key aspect of Theorem 3 is that* (6) *has the same solution, namely the solution to the SDPs* (1) *and* (2)*, for almost all values of* α*. In particular, we can approximate the solution to SDPs* (1) *and* (2) *along various directions emanating from the origin, as illustrated in Figure 4, using the system of ODEs in* (7)*. This makes it possible for us to approximate the first point on the boundary of* U *in the direction of* α *starting at the origin arbitrarily close by changing* ϵ*. Even though the boundary condition function* G *is typically not explicitly available in practice, this reduction to a system of ODEs allows for the efficient approximation of the solution to SDPs* (1) *and* (2) *in the direction* α *starting at the origin.*

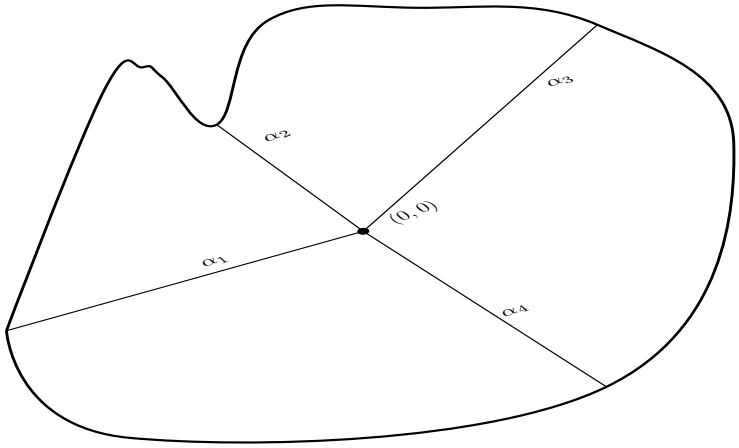


Figure 4: Illustration of approximating the boundary of W_ϵ along different choices of characteristics.

In order to make Theorem 3 effective for computing the boundary of W_ϵ or U, we need a concrete criteria to determine whether a point along the characteristic curve is within the set or not. By observing the solution closely, we find the following property of the solution to (1) and (2) makes numerically tracking the boundary of W_ϵ practical.

Proposition 5. *The minimum eigenvalues,* $\lambda_{ms}(\delta)$ *and* $\lambda_{mx}(\delta)$*, of matrices* $S(\delta)$ *and* $X(\delta)$*, respectively, are identically zero in* U*. In addition, along any direction* α*, the derivative of* $\lambda_{ms}(\delta)$ *and* $\lambda_{mx}(\delta)$ *along any characteristic curve with respect to its parameter is zero in* $\overline{W}_\epsilon$*.*

Proof. This trivially follows from the complimentary slackness condition $S \cdot X = 0$. □

Before describing our numerical scheme, we demonstrate the main topics in this section with an illustrative example where perturbations only occurs in the constraint function of (1) for simplicity.

Example 6. *For* $k = \ell = 1$*, we consider the following:*

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad D_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad A_1 = E_{11} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \textit{and} \quad b_1 = 2.$$

Since $D_1 = 0$*, only the constraint in* (1) *is changed by the perturbation* $\delta = \delta_1$*. Hence,* (1) *and* (2) *are*

$$\begin{array}{ll} \textit{minimize} & x_{11} + x_{22} \\ \textit{s.t.} & 2(1+\delta)x_{12} = 2 \\ & X = \begin{bmatrix} x_{11} & x_{12} \\ x_{12} & x_{22} \end{bmatrix} \succeq 0 \end{array} \qquad \begin{array}{ll} \textit{maximize} & m \\ \textit{s.t.} & m = 2y \\ & S = \begin{bmatrix} 1 & -(1+\delta)y \\ -(1+\delta)y & 1 \end{bmatrix} \succeq 0. \end{array} \tag{10}$$

When $\delta = 0$, it is easy to see that the solution to (10) is

$$m = 2, \quad y = 1, \quad S = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \quad X = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad \text{so that} \quad V(0) = (2, 1, 1, -1, 1, 1, 1, 1)^T.$$

Since $\ell = 1$, i.e., only one perturbation parameter, the maximal perturbation set U is an interval on $\mathbb{R}$. Thus, we seek its two endpoints. The corresponding KKT system is $S, X \succeq 0$ with

$$F = \begin{pmatrix} m - 2y \\ (1+\delta)x_{12} - 1 \\ s_{11} - 1 \\ s_{12} + (1+\delta)y \\ s_{22} - 1 \\ s_{11}x_{11} + s_{12}x_{12} \\ s_{11}x_{12} + s_{12}x_{22} \\ s_{12}x_{12} + s_{22}x_{22} \end{pmatrix} = 0 \quad \text{and} \quad J_V F = \begin{pmatrix} 1 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1+\delta & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1+\delta & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & x_{11} & x_{12} & 0 & s_{11} & s_{12} & 0 \\ 0 & 0 & x_{12} & x_{22} & 0 & 0 & s_{11} & s_{12} \\ 0 & 0 & 0 & x_{12} & x_{22} & 0 & s_{12} & s_{22} \end{pmatrix}. \quad (11)$$

Using (5) and (11), we obtain the derivative

$$\frac{dV}{d\delta} = H(V, \delta) = \begin{pmatrix} -\dfrac{2\left((\delta+1)s_{12}x_{12}y - s_{22}\left(s_{11}x_{12} + (\delta+1)x_{22}y\right) + s_{12}^2 x_{12}\right)}{(\delta+1)^2\left(s_{12}x_{12} - s_{22}x_{22}\right)} \\ -\dfrac{(\delta+1)s_{12}x_{12}y - s_{22}\left(s_{11}x_{12} + (\delta+1)x_{22}y\right) + s_{12}^2 x_{12}}{(\delta+1)^2\left(s_{12}x_{12} - s_{22}x_{22}\right)} \\ 0 \\ \dfrac{\left(s_{12}^2 - s_{11}s_{22}\right)x_{12}}{(\delta+1)\left(s_{12}x_{12} - s_{22}x_{22}\right)} \\ 0 \\ \dfrac{s_{22}x_{12}\left(s_{11}x_{12} - s_{12}x_{22}\right)}{(\delta+1)s_{11}\left(s_{12}x_{12} - s_{22}x_{22}\right)} \\ -\dfrac{x_{12}}{\delta+1} \\ \dfrac{x_{12}\left(s_{11}x_{12} - s_{12}x_{22}\right)}{(\delta+1)\left(s_{12}x_{12} - s_{22}x_{22}\right)} \end{pmatrix}. \quad (12)$$

With the initial condition given by $V(0)$, the boundary of U can be found by searching along the positive and negative direction starting from 0. This corresponds, respectively, with solving the two systems:

$$\begin{cases} \dfrac{d\delta}{ds} = 1 \\ \dfrac{dV}{ds} = H \\ \delta(0) = 0 \\ V_0 = V(0), \end{cases} \quad \text{and} \quad \begin{cases} \dfrac{d\delta}{ds} = -1 \\ \dfrac{dV}{ds} = H \\ \delta(0) = 0 \\ V_0 = V(0). \end{cases} \quad (13)$$

One is always able to solve in the positive direction, while the endpoint in the negative direction is $\delta = -1$. Hence, $U = (-1, \infty)$ which is easy to verify for this simple system. For more complicated problems, the following section develops a numerical scheme to solve such systems of differential equations.

3 Sweeping Euler Scheme

To solve SDPs (1) and (2) efficiently over an unknown perturbation set, we develop a second-order numerical scheme called a sweeping Euler scheme (SES) in this section. This scheme consists of three components: directional sweeping, numerical ODE method, and a boundary threshold criterion.

Based on Theorem 3, we aim to solve along various directions α. One way to parameterize all unit

vectors in $\mathbb{R}^\ell$ is using the following sphere coordinate system:

$$\alpha(\theta_1,\theta_2,\ldots,\theta_{\ell-1}) = \begin{pmatrix} \cos(\theta_1) \\ \sin(\theta_1)\cos(\theta_2) \\ \sin(\theta_1)\sin(\theta_2)\cos(\theta_3) \\ \vdots \\ \sin(\theta_1)\cdots\sin(\theta_{\ell-2})\cos(\theta_{\ell-1}) \\ \sin(\theta_1)\cdots\sin(\theta_{\ell-2})\sin(\theta_{\ell-1}) \end{pmatrix}.$$

For $\theta_1,\ldots,\theta_{\ell-2}\in[0,\pi)$ and $\theta_{\ell-1}\in[0,2\pi)$, the direction vector α sweeps the entire $(\ell-1)$-dimensional unit sphere. This provides a consistent strategy to approximate ∂U by discretizing the angles $\theta_1,\ldots,\theta_{\ell-1}$.

The second component is solving (7) iteratively along each direction selected by the directional sweeping component. In our computations, we apply Heun's method to (7) along the characteristic determined by α, parameterized by s, starting at the origin. We introduce the following notation for simplicity. We use $h>0$ to denote the mesh size along one projected characteristic that is parameterized by s. The i^{th} mesh point is represented by $s_i = ih$ so that $s_0 = 0$. In addition, we use s_{α,δ_0} to denote the parameter value at which the characteristic starting at δ_0 and determined by α first intersects the boundary. The corresponding point on the characteristic curve is given by $\delta_i = s_i\alpha$. We also introduce V_i and H_i to represent the approximation of $V(\delta(s_i))$ and $H(\delta(s_i))$ at s_i, respectively. The numerical scheme solving the system of ODE (7) starting at 0 is

$$\begin{aligned} V_0 &= V(0) \\ V_{i+1} &= V_i + \frac{h}{2}\left(H_i + H(s_{i+1},K_i)\right) \quad \text{where } K_i = V_i + hH_i \quad \text{for } i\geq 0. \end{aligned} \tag{14}$$

Although this numerical scheme is known to have global order two, e.g., [1], to the best of our knowledge, an explicit upper bound on the global error of this method for a system of ODEs has not been provided. We provide such a bound in the following with $E_i = \|V(\delta(s_i)) - V_i\|$ denoting the global error at $s = s_i$ while err_i denotes the local error at s_i.

Theorem 7. *For* (14)*, there exist constants* M *and* D*, defined in* (16) *and* (19)*, respectively, which are independent of step size* h *such that the local error satisfies*

$$err_i \leq Dh^3,$$

and the global error satisfies

$$E_i \leq \frac{D}{M}(e^{Ms_i}-1)h^2.$$

Proof. Consider the function $\Phi : \mathbb{R}\times\mathbb{R}^c\times\mathbb{R}\to\mathbb{R}^c$ defined by

$$\Phi(s,V(s),h) = H(V(s),s) + H(s+h, hH(V(s),s)+V(s)).$$

It easy to obtain that

$$\|\Phi(s,V_1,h) - \Phi(s,V_2,h)\| \leq M\|V_1 - V_2\|, \tag{15}$$

where

$$M = \max_{t\in[0,1]}\left\|\frac{\partial H}{\partial V}(s, tV_1+(1-t)V_2)\right\|. \tag{16}$$

Next, we consider the function

$$d(s,V(s),h) = V(s+h) - V(s) - \frac{h}{2}\Phi(s,V(s),h).$$

For $j=1,\ldots,c$, let V^j and H^j denote the j^{th} element in the corresponding vector. By Taylor expansion,

$$\begin{aligned} V^j(s+h) &= V^j(s) + h\tfrac{dV^j}{ds}(s) + \tfrac{h^2}{2}\tfrac{d^2V^j}{ds^2}(s) + \tfrac{h^3}{3!}\tfrac{d^3V^j}{ds^3}(s) + \tfrac{h^4}{4!}\tfrac{d^4V^j}{ds^4}(\eta_1) \\ H^j\big(s+h, V(s)+hH(V(s),s)\big) &= H^j(V(s),s) + h\tfrac{\partial H^j}{\partial s}(V(s),s) + \tfrac{\partial H^j}{\partial V}(V(s),s)hH(V(s),s) \\ &\quad + \tfrac{h^2}{2}\tfrac{\partial^2 H^j}{\partial s^2}(V(s),s) + 2\tfrac{h}{2}\tfrac{\partial^2 H^j}{\partial s\partial V}hH(V(s),s) \\ &\quad + \tfrac{1}{2}\big(hH(V(s),s)\big)^T\tfrac{\partial^2 H^j}{\partial V^2}(V(s),s)hH(V(s),s) \\ &\quad + \tfrac{h^3}{3!}\tfrac{\partial^3 H^j}{\partial s^3}(\eta_2,V(s)) + 3\tfrac{h^2}{3!}\tfrac{\partial^3 H^j}{\partial s^2\partial V}(\eta_3,\xi_1)hH(V(s),s) \\ &\quad + 3\tfrac{h}{3!}\big(hH(V(s),s)\big)^T\tfrac{\partial^3 H^j}{\partial s\partial V^2}(\eta_4,\xi_2)hH(V(s),s) \\ &\quad + \tfrac{1}{3!}\tfrac{\partial^3 H^j}{\partial V^3}(s,\xi_3)\otimes\big(hH(V(s),s)\big), \end{aligned} \tag{17}$$

where $\eta_k \in [s, s+h]$ for $k = 1, \ldots, 4$ and ξ_q for $q \in 1, 2, 3$ is a point on the line segment connecting $V(s)$ and $V(s) + hH(V(s), s)$. The matrix $\frac{\partial^2 H^j}{\partial V^2}$ and tensor $\frac{\partial^3 H^j}{\partial V^3}$ are the second and third derivative of H^j with respect to V, respectively. In particular, if A is a $p \times p \times p$ tensor and $b \in \mathbb{R}^p$, then

$$A \otimes b = \sum_{i,j,k}^{p} a_{ijk} b_i b_j b_k$$

From (7), we have that

$$\begin{aligned}
\tfrac{dV^j}{ds}(s) &= H(V(s), s) \\
\tfrac{d^2V^j}{ds^2}(s) &= \tfrac{dH^j}{ds}(V(s), s) = \tfrac{\partial H^j}{\partial s}(V(s), s) + \tfrac{\partial H^j}{\partial V}(V(s), s) H(V(s), s) \\
\tfrac{d^3V^j}{ds^3}(s) &= \tfrac{\partial^2 H^j}{\partial s^2}(V(s), s) + 2\tfrac{\partial^2 H^j}{\partial s \partial V}(V(s), s) H(V(s), s) + \tfrac{\partial H^j}{\partial V}(V(s), s) \tfrac{\partial H}{\partial V}(V(s), s) H(V(s), s) \\
&\quad + \tfrac{\partial H^j}{\partial V}(V(s), s) \tfrac{\partial H}{\partial s}(V(s), s) + \big(H(V(s), s)\big)^T \tfrac{\partial^2 H^j}{\partial V^2}(V(s), s) H(V(s), s),
\end{aligned} \tag{18}$$

By substituting (18) and (17) into the j^{th} entry of $d(s, V(s), h)$ yields

$$\begin{aligned}
d^j(s, V(s), h) &= \tfrac{h^3}{6} \tfrac{\partial H^j}{\partial V}(V(s), s) \tfrac{\partial H}{\partial V}(V(s), s) H(V(s), s) + \tfrac{h^3}{6} \tfrac{\partial H^j}{\partial V}(V(s), s) \tfrac{\partial H}{\partial s}(V(s), s) \\
&\quad - \tfrac{h^3}{12} \tfrac{\partial^2 H^j}{\partial s^2}(V(s), s) - \tfrac{h^3}{6} \tfrac{\partial^2 H^j}{\partial s \partial V}(V(s), s) H(V(s), s) \\
&\quad - \tfrac{h^3}{12} \big(H(V(s), s)\big)^T \tfrac{\partial^2 H^j}{\partial V^2} H(V(s), s) + \tfrac{h^4}{4!} \tfrac{d^4 V}{ds^4}(s) - R
\end{aligned}$$

where R represents the last four terms in (17). Define

$$\begin{aligned}
L &= \max_{1 \le j \le c} \max_{s \in [0, s_{\alpha,\delta_0}]} |V^j(s)|, \\
L_1 &= \max_{1 \le i,j \le c} \max_{s \in [0, s_{\alpha,\delta_0}]} \left\{ \left| \tfrac{\partial H^i}{\partial V^j}(V(s), s) \right|, \left| \tfrac{\partial H^i}{\partial s}(V(s), s) \right| \right\}, \\
L_2 &= \max_{1 \le i,j,k \le c} \max_{s \in [0, s_{\alpha,\delta_0}]} \left\{ \left| \tfrac{\partial^2 H^i}{\partial V^k \partial V^j}(V(s), s) \right|, \left| \tfrac{\partial^2 H^i}{\partial s \partial V^j}(V(s), s) \right|, \left| \tfrac{\partial^2 H^i}{\partial s}(V(s), s) \right| \right\}.
\end{aligned}$$

Then, for sufficient small h and $j = 1, \ldots, c$, we have

$$|d^j(s, V(s), h)| \le \frac{h^3}{6} L L_1^2 c^2 + \frac{h^3}{6} L_1^2 c + \frac{h^3}{12} L_2 + \frac{h^3}{6} L L_2 c + \frac{h^3}{12} L^2 L_2 c^2 + h^3$$

showing that

$$\|d(s, V(s), h)\| \le D h^3 \quad \text{where} \quad D = \frac{1}{6} L L_1^2 c^2 + \frac{1}{6} L_1^2 c + \frac{1}{12} L_2 + \frac{1}{6} L L_2 c + \frac{1}{12} L^2 L_2 c^2 + 1. \tag{19}$$

From the numerical scheme (14), we have

$$V_{i+1} = V_i + \frac{h}{2} \Phi(s_i, V_i, h). \tag{20}$$

By definition of $d(s, V(s), h)$, we obtain

$$V(s_{i+1}) = V(s_i) + \frac{h}{2} \Phi(s_i, V(s_i), h) + d(s_i, V(s_i), j). \tag{21}$$

Subtracting (20) from (21) and applying the triangle inequality yields

$$E_{i+1} \le E_i + \frac{h}{2} \|\Phi(s_i, V_i, h) - \Phi(s_i, V(s_i), h)\| + \|d(s_i, V(s_i), h)\|.$$

Replacing the last two terms using (15) and (19), we have

$$E_{i+1} \le E_i \left(1 + \frac{h}{2} M\right) + D h^3. \tag{22}$$

which implies that

$$E_i \le \frac{D}{M} (e^{M s_i} - 1) h^2$$

yielding the global error bound. The local error bound is immediately obtained by assuming there is no error in previous step, i.e., if $E_i = 0$, then $E_{i+1} \le D h^3$.

□

The upshot of Theorem 7 is that our numerical scheme is convergent and consistent, e.g., see [9]. By discretizing over the angles, one can then assemble the solution inside of W_ϵ obtained from the mesh points together using bicubic interpolation method (for two-dimensional perturbations) or nearest-neighbor interpolation (for perturbations in any dimension) to produce numerical solutions to (1) and (2).

The third component of our sweeping Euler scheme is to determine where the boundary is located, i.e., determine if ∂U lies between the current mesh point and the next mesh point. We will discuss two options resulting from Prop. 5.

Eigenvalue threshold Inside of U, the minimum eigenvalue of both $X(\delta(s))$ and $S(\delta(s))$ is zero. One option after crossing the boundary is for one of these minimum eigenvalues to become negative. We can use a small negative constant C_1 to decide that a minimum eigenvalue has become negative.

Derivative threshold From Prop. 5, the derivatives of the minimum eigenvalues, namely $\frac{d\lambda_{ms}}{ds}$ and $\frac{d\lambda_{mx}}{ds}$, are zero inside of U. Hence, we can threshold, say with a positive constant C_2, based on the absolute value of an approximation of these derivatives.

Thresholding on the minimum eigenvalue may be more intuitive since this is the condition to ensure that the matrices remain positive semidefinite. A small negative constant C_1 is chosen instead of 0 due to numerical error. One major disadvantage is the potential difficulty in selecting this constant C_1 since it is challenging to predict the magnitude of deviation of the minimum eigenvalue. This makes the performance of the eigenvalue threshold approach unstable as demonstrated on an example in Table 4.2.

Inside of U, the derivatives of the minimum eigenvalues $\frac{d\lambda_{ms}}{ds}$ and $\frac{d\lambda_{mx}}{ds}$ are both identically zero. At the boundary ∂U, at least one of them becomes undefined due to either divergence to infinity as one approaches the boundary or one of the matrices becomes more rank deficient. This leads us to determine ∂U via one of these derivatives having a sufficiently large magnitude. In our numerical experiments, taking $C_2 = 1$ is sufficient to decide the derivatives of the eigenvalues, which should be zero, are becoming undefined.

From Prop. 5, we have

$$\lambda_{mx}(X(s)) = \lambda_{ms}(S(s)) = 0 \quad \text{for} \quad s \in \widetilde{W}_\epsilon$$

where $\widetilde{W}_\epsilon = S^{-1}(W_\epsilon)$ and $S : \mathbb{R}^c \to \mathbb{R}$ is the mapping from δ to s. Taking the derivative with respect to s along a characteristic curve starting at δ_0 determined by α yields

$$\begin{array}{rcccl} 0 &=& \frac{d\lambda_{mx}(X(s))}{ds} &=& \sum_{i=1}^{n} \frac{\partial \lambda_{mx}(X(s))}{\partial x_{ii}(s)} \frac{dx_{ii}(s)}{ds} + 2\sum_{j=i+1}^{n} \sum_{i=1}^{n-1} \frac{\partial \lambda_{mx}(X(s))}{\partial x_{ij}(s)} \frac{dx_{ij}(s)}{ds}, \quad 0 \le s \le s_{\alpha,\delta_0} \\ 0 &=& \frac{d\lambda_{ms}(S(s))}{ds} &=& \sum_{i=1}^{n} \frac{\partial \lambda_{ms}(S(s))}{\partial s_{ii}(s)} \frac{ds_{ii}(s)}{ds} + 2\sum_{j=i+1}^{n} \sum_{i=1}^{n-1} \frac{\partial \lambda_{ms}(S(s))}{\partial s_{ij}(s)} \frac{ds_{ij}(s)}{ds}, \quad 0 \le s \le s_{\alpha,\delta_0}. \end{array} \tag{23}$$

Following [8], we know

$$\frac{\partial \lambda_{mx}(X(s))}{\partial x_{ij}(s)} = \frac{vx_i(s) \cdot wx_j(s)}{vx(s) \cdot wx(s)} \quad \text{and} \quad \frac{\partial \lambda_{ms}(S(s))}{\partial s_{ij}(s)} = \frac{vs_i(s) \cdot ws_j(s)}{vs(s) \cdot ws(s)}$$

where $vx(s)$,$vs(s)$ and $wx(s), ws(s)$ are the left and right eigenvectors of $X(s)$ and $S(s)$, respectively. We note that the numerators are a scalar product of entries and the denominators are a dot product of vectors. In addition, from (8), we obtain

$$\frac{dx_{ij}(s)}{ds} = H^{1+k+\frac{n+n^2}{2}+\frac{(2n-i)(i-1)}{2}+j}(V(s), s) \quad \text{and} \quad \frac{ds_{ij}(s)}{ds} = H^{1+k+\frac{(2n-i)(i-1)}{2}+j}(V(s), s).$$

Thus, (23) yields

$$\begin{array}{rcl} \frac{d\lambda_{mx}(X(s))}{ds} &=& \sum_{i=1}^{n} \frac{vx_i(s) \cdot wx_j(s)}{vx(s) \cdot wx(s)} H^{1+k+\frac{n+n^2}{2}+\frac{(2n-i)(i-1)}{2}+j}(V(s), s) \\ && + 2\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{vx_i(s) \cdot wx_j(s)}{vx(s) \cdot wx(s)} H^{1+k+\frac{(2n-i)(i-1)}{2}+j}(V(s), s), \\ \frac{d\lambda_{ms}(S(s))}{ds} &=& \sum_{i=1}^{n} \frac{vs_i(s) \cdot ws_j(s)}{vs(s) \cdot ws(s)} H^{1+k+\frac{(2n-i)(i-1)}{2}+j}(V(s), s) \\ && + 2\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{vs_i(s) \cdot ws_j(s)}{vs(s) \cdot ws(s)} H^{1+k+\frac{(2n-i)(i-1)}{2}+j}(V(s), s). \end{array} \tag{24}$$

Example 8. *In Ex. 6, the solution as a function of* δ *on* $(-1,\infty)$ *is*

$$y=\frac{1}{1+\delta},\quad m=2y,\quad X=y\begin{pmatrix}1&1\\1&1\end{pmatrix},\quad S=\begin{pmatrix}1&-y(1+\delta)\\-y(1+\delta)&1\end{pmatrix}=\begin{pmatrix}1&-1\\-1&1\end{pmatrix}. \tag{25}$$

One clearly sees that $X\succeq 0$ *for* $\delta>-1$*,* X *diverges to infinity as* $\delta\to -1^+$*, and* X *has a negative eigenvalue when* $\delta<-1$*. This shows a potential difficulty of observing a negative eigenvalue by having to pass over a divergent point occurring at* $\delta=-1$*.*

On the other hand, from (12)*, we have*

$$\lim_{\delta\to -1^+} H=\begin{pmatrix}-\infty\\-\infty\\0\\0\\0\\-\infty\\-\infty\\-\infty\end{pmatrix}$$

so that (24) *can be used to identify the ill-conditioning of computing the undefined derivative of the minimum eigenvalue of* X *at* $\delta=-1$*, which is on* ∂U*. We return to this example in Section 4.1.*

4 Numerical Examples

In this section, we demonstrate the sweeping Euler scheme (SES) developed in Section 3 to find the maximum connected set U in which the optimum depends continuously on the parameter values for several examples. Section 4.1 numerically demonstrates second-order convergence and compares the global error bound from Theorem 7 with actual error. Section 4.2 compares numerical error for computing a point on ∂U with the exact value computed symbolically.

To the best of our knowledge, there are no other specially designed numerical methods that compute ∂U for SDPs under perturbation. However, there are numerous established state of art SDP solvers, e.g., see [17], which can be used to find ∂U when coupled with the sweeping technique described in Section 3. To show the performance of the sweeping Euler method, we compare it with using the SDP solver SDPT3 [22] coupled with the sweeping technique on a SDP problem with two perturbation parameters. As described in Section 2, we reformulate the problem of computing ∂U as a PDE problem and thus we also compare the sweeping Euler method with a standard finite difference scheme for PDEs. All computations are executed using MATLAB running on a 2.5 GHz Intel Core i5-6500 processor.

4.1 Covergence rate and numerical error

For an illustrative demonstration of the convergence rate and numerical error associated with SES, we return to the setup from Examples 6 and 8.

First, to numerically demonstrate the second order convergence of SES, we compare the exact value

$$m(0.05)=\frac{2}{1+0.05}=\frac{40}{21}$$

with numerical approximations using various step sizes. The results are summarized in Table 1 which clearly shows second order convergence for this problem.

Next, using a step size of 3/1000, we compare the global error bound from Theorem 7 applied to $m=\frac{2}{1+\delta}$ for $\delta>-1$ with the actual numerical error. Figure 5 shows that the global error bound from Theorem 7 does indeed provide an upper bound on the actual numerical error. As is typical, this error bound can be quite pessimistic, even for this illustrative example, as δ moves towards the boundary of U.

Figure 6 provides a clearer picture as to the behavior of the actual numerical error for $\delta<0$ on the left and $\delta>0$ on the right with the same step size of 3/1000. As δ approaches the boundary of U at -1, the error increases rapidly. In the graph on the right, the error is bounded by $2\cdot 10^{-7}$. Note that these approximations arise from only knowing the solution at $\delta=0$. To reduce numerical error, especially as one more towards a point on the boundary of U, one can always use the numerical approximation to solve the SDP, e.g., via the KKT system, and "recenter" the computations at this new value.

Table 1: Numerically testing order of convergence.

Step size	Error	Order
1/20	$4.3735 \cdot 10^{-4}$	
1/40	$1.1260 \cdot 10^{-4}$	1.9576
1/80	$2.8546 \cdot 10^{-5}$	1.9798
1/160	$7.1853 \cdot 10^{-6}$	1.9901
1/320	$1.8024 \cdot 10^{-6}$	1.9951
1/640	$4.5136 \cdot 10^{-7}$	1.9976
1/1280	$1.1293 \cdot 10^{-7}$	1.9988
1/2560	$2.8245 \cdot 10^{-8}$	1.9994

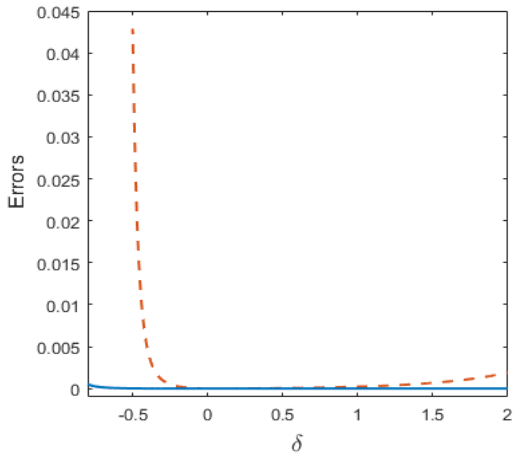


Figure 5: Comparison of the global error bound (dashed) with the actual numerical error (solid).

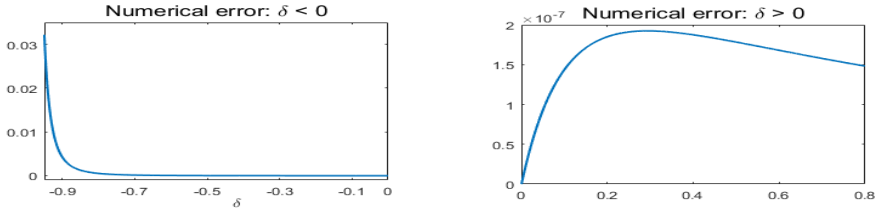


Figure 6: Plot of actual numerical error.

4.2 Comparison of computing boundary point

For the illustrative system presented in Ex. 6, one could explicitly compute both the solution and the terms needed for the global error bound presented in Theorem 7 as shown in Fig. 5. However, it may be difficult to compute the terms in the global error exactly for more complicated examples. In such scenarios, one can use numerical approximations computed via a central difference scheme. For example, consider the following SDPs:

$$\begin{aligned} \text{minimize} \quad & (1+2\delta)x_{11} - 2(1+\delta)x_{12} + (2+3\delta)x_{22} \\ \text{s.t.} \quad & (2+\delta)x_{11} + 2(3+2\delta)x_{12} + (1+3\delta)x_{22} = -1 \\ & X = \begin{bmatrix} x_{11} & x_{12} \\ x_{12} & x_{22} \end{bmatrix} \succeq 0, \end{aligned} \tag{26}$$

$$\begin{aligned} \text{maximize} \quad & m \\ \text{s.t.} \quad & m = y \\ & S = \begin{bmatrix} 1+2y+\delta(2+y) & -1+3y+\delta(2y-1) \\ -1+3y+\delta(2y-1) & 2+y+\delta(3+3y) \end{bmatrix} \succeq 0. \end{aligned}$$

In order to compare an approximation of the global error bound with the actual numerical error for $m(\delta)$, we first need to derive an exact symbolic expression for m used to compute the actual numerical error.

From the complimentary slackness condition $S \cdot X = 0$ exploited by Prop. 5, we know that

$$0 = \det S = -(\delta^2 + 7\delta + 11)y^2 + (13\delta^2 + 19\delta + 9)y + 5\delta^2 + 5\delta + 1.$$

Since we want to maximize $m = y$, this yields

$$m = \frac{13\delta^2 + 19\delta + 9 + \sqrt{189\delta^4 + 654\delta^3 + 959\delta^2 + 590\delta + 125}}{2(\delta^2 + 7\delta + 11)}. \tag{27}$$

Using this expression for m, Figure 7 compares an approximation of the global error bound from Theorem 7 with the actual numerical error with step size $1/100$ as one head towards the boundary of U in the negative direction starting from 0. As with Fig. 5, the global error bound becomes more pessimistic as one moves towards the boundary.

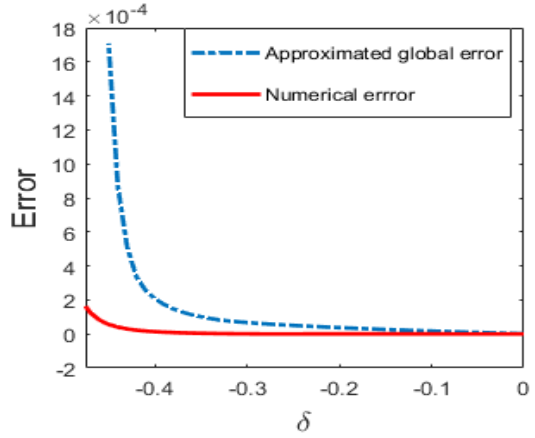


Figure 7: Comparing an approximation of the global error bound (dashed) with numerical error (solid).

We now turn to computing the boundary point of the interval U in the negative direction. In order to compare the numerical results, we first compute this boundary point explicitly. In this case, this occurs at the largest negative root of quartic polynomial under square root in (27), namely

$$\delta_c = \frac{\gamma - 109}{126} + \frac{1}{2}\sqrt{\frac{239104}{3969\gamma} - \frac{1030}{1323} - \frac{\alpha + \beta}{567}} \approx -0.49347124103$$

where

$$\alpha = \sqrt[3]{11113649 + 34560\sqrt{24019}}, \quad \beta = \sqrt[3]{11113649 - 34560\sqrt{24019}}, \quad \text{and} \quad \gamma = \sqrt{7(\alpha + \beta) - 1545}.$$

We now compare approximating the corresponding boundary point using the eigenvalue and derivative threshold criteria described in Section 3 starting from $\delta = 0$. To this end, Table 4.2 compares using different step sizes with different threshold values C_1 and C_2, respectively. In both cases, the best one can hope for is to compute δ_c with an error bounded by the step size. This table shows that this is precisely the case for the derivative threshold criterion. In particular, the derivative threshold criterion is more stable than the eigenvalue threshold criterion for approximating the boundary starting at $\delta = 0$.

Table 2: Comparison of eigenvalue and derivative threshold for approximating the boundary.

	Eigenvalue threshold C_1			Derivative threshold C_2		
Step size	-10^{-5}	-10^{-4}	-10^{-3}	0.1	1	10
1/10	−0.5	−0.5	−0.5	−0.5	−0.5	−0.5
1/100	−0.33	−0.44	−0.49	−0.49	−0.50	−0.50
1/1000	−0.475	−0.490	−0.494	−0.493	−0.493	−0.494

Since these criterion can only approximate the boundary up to the step size, it can be computationally expensive to produce highly accurate approximations. However, one can use these approximations as initial values to procedures to more accurately compute the boundary point. For example, one such approach is to build a Puiseux series centered at the unknown boundary point. For singularities of multiplicity 2, which is the generic case [12], we can build a Puiseux series with half powers [18]. In this case, for δ near δ_c, there exists $a_j \in \mathbb{R}$ such that

$$m(\delta) = \sum_{j=0}^{\infty} a_j (\delta - \delta_c)^{\frac{j}{2}}.$$

Using n data points, say $(\delta_j, m(\delta_j))$ for $j = 0, \ldots, n-1$, one can simultaneously approximate the boundary point δ_c and the first $n-1$ coefficients $a_0 = m(\delta_c), a_1, \ldots, a_{n-2}$ using interpolation. Table 3 compares using two different step sizes $\Delta\delta$, namely $1/100$ and $1/1000$, with three choices of n, namely $3, 5, 7$, with the actual values. In this table, we list δ_0 with $\delta_j = \delta_0 - j\Delta\delta$ for $j = 1, \ldots, n-1$. The values are presented up to the first incorrect digit with the correct digits highlighted in red. In particular, we see that just with the knowledge of the boundary to 3 digits of accuracy, the Puiseux series interpolation can quickly be used to approximate the singularity with 7 digits of accuracy. One could continue to play this modification of the "endgame" of [18] to compute the boundary point as accurately as required.

Table 3: Comparison of using Puisuex series to approximate boundary point.

Step size	δ_0	n	Approximation of δ_c	Approximation of $a_0 = m(\delta_c)$
$1/100$	-0.49	3	-0.498	0.14
		5	-0.4933	0.18
		7	-0.49349	0.178
$1/1000$	-0.493	3	-0.4935	0.177
		5	-0.49347121	0.1790763
		7	-0.49347125	0.179075
Actual values			-0.493471241	0.17907623

4.3 A problem with two perturbation parameters

All of the previous examples had a single perturbation parameter in which case the sweeping Euler scheme (SES) only "sweeps" in the positive and negative directions. Thus, in this section, we demonstrate the scheme on an example where S and X are 3×3 symmetric matrices with two perturbation parameters. For simplicity, we will only write the dual SDP (2) with $S(:,j)$ denoting the j^{th} column of S:

$$\begin{aligned}
\text{maximize} \quad & m \\
\text{s.t.} \quad & m = y_1 - 4y_2 \\
& S(:,1) = \begin{bmatrix} \delta_1(-8y_1 - 4y_2 + 4) + \delta_2(10y_1 - 10y_2 + 6) - 2y_1 - 4y_2 - 8 \\ \delta_1(-6y_1 + 2y_2 + 5) + \delta_2(2y_1 + 9y_2 - 5) - 4y_1 + 6y_2 + 2 \\ \delta_1(-4y_1 + 5y_2 + 2) + \delta_2(-y_1 - 2y_2 + 6) + 9y_1 - 4y_2 + 8 \end{bmatrix} \\
& S(:,2) = \begin{bmatrix} \delta_1(-6y_1 + 2y_2 + 5) + \delta_2(2y_1 + 9y_2 - 5) - 4y_1 + 6y_2 + 2 \\ \delta_1(4y_1 + 6y_2 - 8) + \delta_2(10y_1 - 10y_2 - 6) - 10y_1 + 2 \\ \delta_1(-2y_1 - 9y_2) + \delta_2(4y_1 + 2y_2 - 3) + 6y_1 + 7y_2 + 6 \end{bmatrix} \\
& S(:,3) = \begin{bmatrix} \delta_1(-4y_1 + 5y_2 + 2) + \delta_2(-y_1 - 2y_2 + 6) - 9y_1 - 4y_2 + 8 \\ \delta_1(-2y_1 - 9y_2) + \delta_2(4y_1 + 2y_2 - 3) + 6y_1 + 7y_2 + 6 \\ \delta_1(-2y_1 - 6) + \delta_2(10y_1 - 8y_2 - 4) - 8y_1 - 4y_2 \end{bmatrix} \\
& S \succeq 0.
\end{aligned} \tag{28}$$

We first use SES to approximate the solution to problem (28) inside of the *a priori* unknown maximal perturbation set U. Following Section 3, we first sweep using direction vectors

$$\alpha(\theta) = \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix} \quad \text{for} \quad \theta \in [0, 2\pi).$$

Using an angular mesh step size of $\pi/180$, for each direction vector $\alpha(\theta)$ selected, we start at the origin and solve the corresponding ODEs (7) along $\alpha(\theta)$ with step size $1/1000$ until the derivative threshold criterion provides an approximation of the first boundary of U in the direction of $\alpha(\theta)$. In Figure 8, the left-hand side picture shows the approximation of m in all directions, while the right-hand side picture is restricted to $\delta_1, \delta_2 \in [0, 0.05]$.

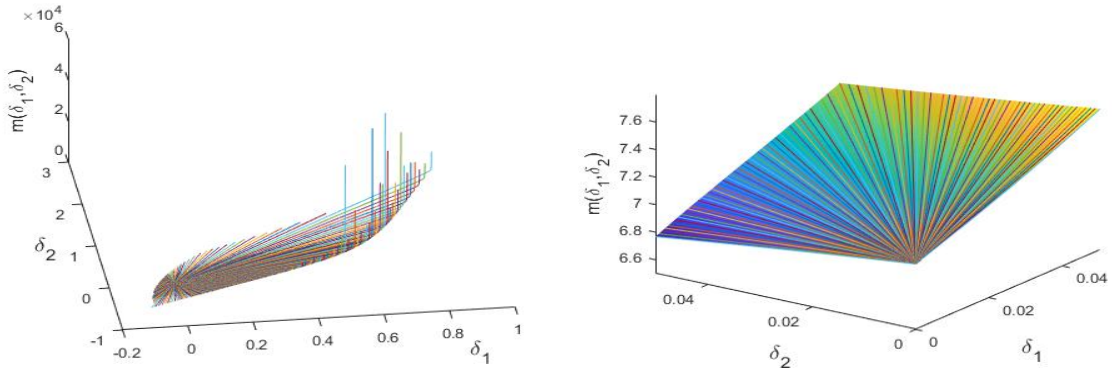


Figure 8: Plot of solution computing using SES in all directions (left) and on $[0, 0.05]^2$ (right).

There are several possibilities that create the boundary of U. First, the optimum of the SDP could become unbounded. This is observed in Figure 8 by the rays becoming nearly vertical at the boundary. Second, the optimum could become singular with respect to the KKT conditions. This could happen at the boundary between feasible and infeasible SDPs as well as having multiple solution sheets of the KKT system intersecting. The complete boundary of U is shown in Figure 9.

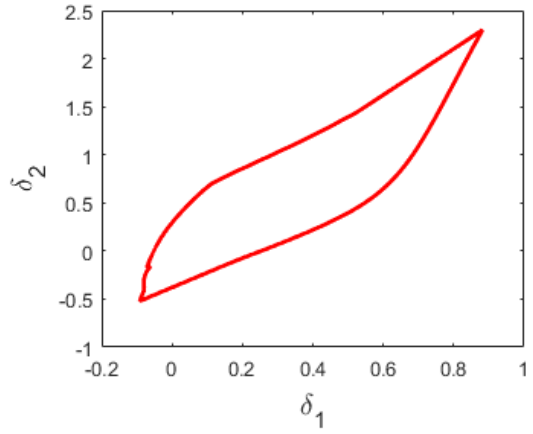


Figure 9: Plot of the approximation of the boundary of U computed by SES.

We next compare the performance of SES with an alternative approach to approximate ∂U: use SDPT3 [22] to solve at each mesh point derived by the sweeping technique. We chose to compare with SDPT3 is due to its popularity in optimization modeling language, its advantages in solving small to medium scale SDP problems [17, 23], and an easy-to-use interface via the MATLAB package CVX. In this comparison case, a boundary point is recorded whenever SDPT3 reports that the status is "not solved." Hence, equating "not solved" with infeasibility yields an outer approximation of ∂U. One difficulty in using such a naive approach is that one may have jumped over ∂U where the SDP is feasible an the solution lies on a different solution sheet while our SES approach identifies where such a switch occurs.

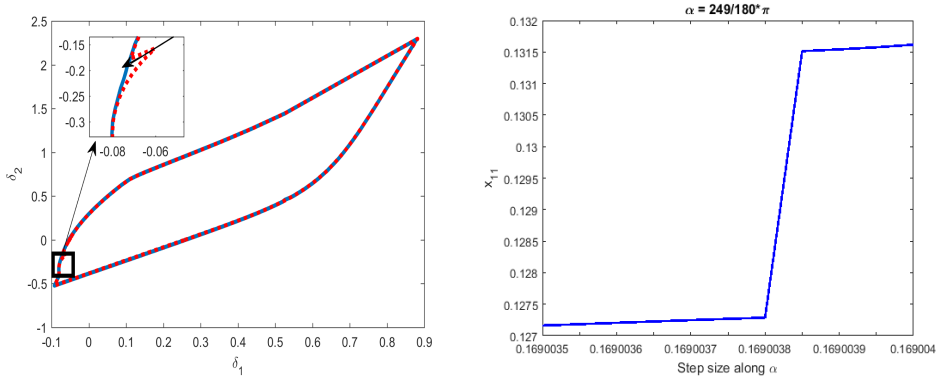
Our comparison utilized an angular mesh size of $\pi/180$ and step size of $1/1000$. Table 4 lists the computational time, which shows that our SES method is more than 100 times faster than using SDPT3 via CVX. To compare the accuracy of the methods, we compare the L_1 error of the approaches for $(\delta_1, \delta_2) \in [0, 0.05]^2$ which Table 4 shows are comparable.

Figure 10(a) compares the boundaries computed using SES with SDPT3 via CVX. We observe that both provide an almost identical approximation of ∂U which reaffirms of the ability of SES in approxi-

Table 4: Comparison of CPU time (in seconds) and L_1 error.

Method	CPU time	L_1 error
SES	144.3	$1.07 \cdot 10^{-9}$
CVX-SDPT3	14,599.5	$2.00 \cdot 10^{-9}$

mating ∂U. A discrepancy occurs for the angle $\theta = 249\pi/180$ (illustrated by the arrow in Figure 10(a)) where SES indicates a boundary near 0.169 units along this direction while the SDPT3 approach yields 0.204. The reason for this is illustrated in Figure 10(b) which shows a jump in the x_{11} coordinate between solution sheets of the optimum between 0.1690038 and 0.1690039. Thus, the optimum is not differentiable which is identified by our SES approach. By simply solving at mesh points, SDPT3 identified that the SDP had solutions on both sides of this so it continued to look for infeasibility which occurred much later. Therefore, the sensitivity of our SES approach combined with being two orders of magnitude faster with similar error show its dominance for computing ∂U over simply solving the SDP at mesh points looking for infeasibility.



(a) (b)

Figure 10: (a): Comparion of ∂U for (28) using SES and a mesh solving approach using SDPT3 via CVX. The dashed line represents using SES while the solid line are the results from CVX-SDPT3 combination. (b): Demonstrating a jump in the value of x_{11}-coordinate of the optimum along the direction $249\pi/180$.

We next compare our SES method with a first-order finite difference scheme (FDS) for solving (6). To setup notation, we note that the vector of variables V has 15 entries. For simplicity, we write V^k and H^k to be the k^{th} coordinate of V and H, respectively. Let Δ_i denote the mesh size for perturbation δ_i with $\delta_i^j = \delta_i^0 + j\Delta_i$. We let $V_{i,j}^k$ and $H_{i,j}^k$ denote the approximation of $V^k(\delta_1^i, \delta_2^j)$ and $H^k(\delta_1^i, \delta_2^j)$, respectively. Thus, the numerical scheme is

$$\frac{V_{i+1,j}^k - V_{i,j}^k}{\Delta_1} + \frac{V_{i,j}^k - V_{i,j-1}^k}{\Delta_2} = H_{i,j}^k \quad \text{for} \quad k = 1, \ldots, 15 \tag{29}$$

with given boundary conditions

$$V^k(0,j) = V^k(\delta_1^0, \delta_2^j) \quad \text{and} \quad V^k(i,0) = V^k(\delta_1^i, \delta_2^0). \tag{30}$$

For example, we apply this finite difference scheme given the boundary conditions along

$$(\delta_1, \delta_2) \in (\{0\} \times [0, 0.05]) \cup ([0, 0.05] \times \{0\}).$$

Since the solution from SES is not on a uniform grid in terms of δ_1 and δ_2, we utilize a two-dimensional cubic interpolation to approximate the solution on the uniform grid for comparison with the FDS. Figure 11 compares the exact solution with the approximation using the finite difference scheme (FDS) and sweeping Euler scheme (SES) after cubic interpolation for $\delta_1 = 0.05$ and $\delta_2 \in [0, 0.05]$. This clearly shows that the improved performance of the SES over the FDS when using the same mesh size of $\Delta_1 = \Delta_2 = 1/1000$.

Table 5 compares the L_1 error for two different meshes. In addition to the improved performance, we note that the sweeping Euler scheme only needs the solution at one point, e.g., the origin where there is no perturbations, while the finite difference scheme requires additional boundary conditions.

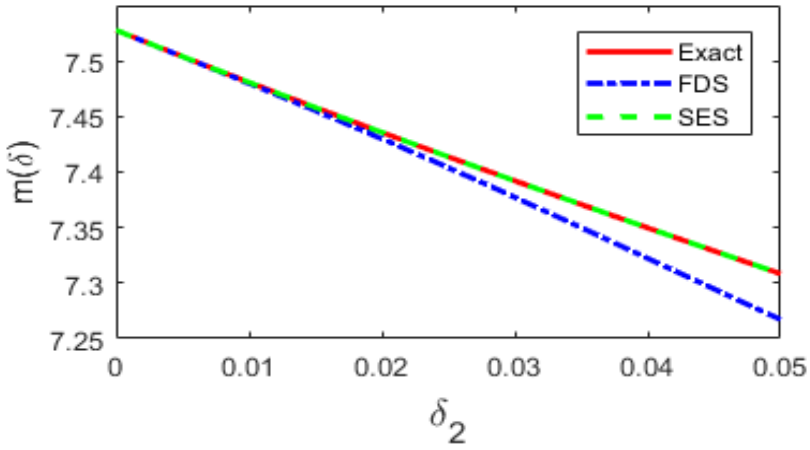


Figure 11: For $\delta_1 = 0.05$, plot of of the exact solution and numerical approximations computed using the finite difference scheme (FDS) and sweeping Euler scheme (SES) with cubic interpolation.

Table 5: Comparison of L_1 error.

mesh size $\Delta_1 = \Delta_2$	1/200	1/1000
sweeping Euler scheme (SES)	$2.69 \cdot 10^{-8}$	$1.19 \cdot 10^{-18}$
finite difference scheme (FDS)	$3.38 \cdot 10^{-7}$	$2.46 \cdot 10^{-7}$

5 Conclusion and remarks

In this article, we solve semidefinite programs (SDPs) under affine perturbations via solving a system of quasilinear partial differential equations (PDEs). We developed a second-order sweeping Euler scheme (SES) to solve the system of quasilinear PDEs and approximate the unknown boundary of the maximal perturbation set U. Several examples are used to numerically demonstrate the convergence rate and approximation of the boundary of U.

We also compared our SES method with using SDPT3 [22] via CVX in MATLAB and a finite difference scheme (FDS) on a family of SDPs parameterized by two affine perturbations. The data shows that the SES is clearly advantageous over the other two methods. In addition, SES only requires one point while the FDS requires additional boundary conditions.

Finally, we note that this method is not limited to just affine perturbations. In fact, both the theory and numerical computations needs for the SES method naturally extend to SDPs with sufficiently smooth nonlinear perturbations.

References

[1] A. S. Ackleh, E. J. Allen, R. B. Kearfott, and P. Seshaiyer. *Classical and Modern Numerical Analysis: Theory, Methods and Practice*. Taylor and Francis Publishing, 2009.

[2] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM journal on Optimization*, 5(1):13–51, 1995.

[3] K. A. Ariyawansa and Y. Zhu. Stochastic semidefinite programming: a new paradigm for stochastic optimization. *4OR*, 4(3):239–253, Sep 2006.

[4] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. Robust semidefinite programming. *Handbook on Semidefinite Programming*, 27, 1998.

[5] G. Blekherman, P. A. Parrilo, and R. R. Thomas. *Semidefinite optimization and convex algebraic geometry*. SIAM, 2012.

[6] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear matrix inequalities in system and control theory*. SIAM, Philadelphia, PA, 1994.

[7] S. Boyd and L. Vandenberghe. Semidefinite programming relaxations of non-convex problems in control and combinatorial optimization. In *Communications, Computation, Control, and Signal Processing*, pages 279–287. Springer, 1997.

[8] H. Caswell. *Matrix Population Models: Construction, Analysis, and Interpretation*. Sinauer Associates, Sunderland,MA, second edition, 2001.

[9] D. M. Causon and C. G. Mingham. *Introductory finite difference methods for PDEs*. Bookboon, 2010.

[10] E. A. Coddington and N. Levinson. *Theory of Ordinary Differential Equations*. McGraw-Hill Education, New York, 1955.

[11] R. Courant. Cauchy's problem for hyperbolic quasi-linear systems of first order partial differential equations in two independent variables. *Communications on Pure and Applied Mathematics*, 14(3):257–265, 1961.

[12] J. P. Dedieu and M. Shub. On simple double zeros and badly conditioned zeros of analytic functions of n variables. *Mathematics of computation*, pages 319–327, 2001.

[13] A. Dehghani, J. L. Goffin, and D. Orban. A primal–dual regularized interior-point method for semidefinite programming. *Optimization Methods and Software*, 32(1):193–219, 2017.

[14] L. El Ghaoui, F. Oustry, and H. Lebret. Robust solutions to uncertain semidefinite programs. *SIAM Journal on Optimization*, 9(1):33–52, 1998.

[15] L. C. Evans. *Partial Differential Equations–second edition, volume 19 of Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2010.

[16] S. Hayashi, T. Okuno, and Y. Ito. Simplex-type algorithm for second-order cone programmes via semi-infinite programming reformulation. *Optimization Methods and Software*, 31(6):1272–1297, 2016.

[17] H. D. Mittelmann. The state-of-the-art in conic optimization software. In *Handbook on Semidefinite, Conic and Polynomial Optimization*, pages 671–686. Springer, 2012.

[18] A. P. Morgan, A. J. Sommese, and C. W. Wampler. A power series method for computing singular solutions to nonlinear analytic systems. *Numerische Mathematik*, 63(1):391–409, 1992.

[19] G. Pataki. Cone-lp's and semidefinite programs: Geometry and a simplex-type method. *Integer programming and combinatorial optimization*, pages 162–174, 1996.

[20] A. Shapiro. First and second order analysis of nonlinear semidefinite programs. *Mathematical Programming*, 77(1):301–320, 1997.

[21] J. F. Sturm and S. Zhang. On sensitivity of central solutions in semidefinite programming. *Mathematical Programming*, 90(2):205–227, 2001.

[22] K.C. Toh, R.H. Tütüncü, and M.J. Todd. SDPT3 – a matlab software package for semidefinite-quadratic-linear programming. Available at `www.math.nus.edu.sg/~mattohkc/sdpt3.html`.

[23] R. H. Tütüncü, K. C. Toh, and M. J. Todd. Solving semidefinite-quadratic-linear programs using sdpt3. *Mathematical programming*, 95(2):189–217, 2003.

[24] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM review*, 38(1):49–95, 1996.