

# Eigenvalue programming beyond matrices

Masaru Ito\*      Bruno F. Lourenço†

November 8, 2023

## Abstract

In this paper we analyze and solve eigenvalue programs, which consist of the task of minimizing a function subject to constraints on the “eigenvalues” of the decision variable. Here, by making use of the FTvN systems framework introduced by Gowda, we interpret “eigenvalues” in a broad fashion going beyond the usual eigenvalues of matrices. This allows us to shed new light on classical problems such as inverse eigenvalues problems and also leads to new applications. In particular, after analyzing and developing a simple projected gradient algorithm for general eigenvalue programs, we show that eigenvalue programs can be used to express what we call *vanishing quadratic constraints*. A vanishing quadratic constraint requires that a given system of convex quadratic inequalities be satisfied and at least a certain number of those inequalities must be tight. As a particular case, this includes the problem of finding a point  $x$  in the intersection of  $m$  ellipsoids in such a way that  $x$  is also in the boundary of at least  $\ell$  of the ellipsoids, for some fixed  $\ell > 0$ . At the end, we also present some numerical experiments.

**Keywords:** Eigenvalue programming, eigenvalue optimization, FTvN systems, vanishing quadratic constraints

## 1 Introduction

In this paper, we consider problems of the following form.

$$\begin{aligned} \min_{x \in \mathcal{E}} \quad & f(x) && \text{(Eig-Prog)} \\ \text{subject to} \quad & \lambda(x) \in \mathcal{C}, \end{aligned}$$

where  $f : \mathcal{E} \rightarrow \mathbb{R}$  is a smooth function,  $\mathcal{E}$  is a finite dimensional Euclidean space,  $\mathcal{C} \subseteq \mathbb{R}^r$  and  $(\mathcal{E}, \mathbb{R}^r, \lambda)$  is a so-called *FTvN system*, which was originally introduced by Gowda [17]. Informally,  $\lambda : \mathcal{E} \rightarrow \mathbb{R}^r$  has the role of “eigenvalue map” and (Eig-Prog) corresponds to the task of minimizing  $f$  subject to the constraint that the “eigenvalues” of  $x$  must belong to a certain set  $\mathcal{C}$ . Because of this, we will call (Eig-Prog) an *eigenvalue program* and this will be the main subject matter of this work.

The motivation for this work is to provide a broad, yet simple framework, for solving optimization problems which requires constraints on the eigenvalues of  $x$ . In particular, we will cover the case where  $\mathcal{E}$  is  $\mathcal{S}^n$  (the space of  $n \times n$  real symmetric matrices) and  $\lambda$  is the usual matrix eigenvalue map, where the eigenvalues are ordered from largest to smallest. This problem, of course, has a long history. However, it may be fair to say that many of the previous works seem to be focused on controlling the the largest or the smallest eigenvalues (or singular values). This includes the particular case of rank constraints, since we may force a  $n \times n$  symmetric positive semidefinite matrix to have rank less or

---

\*Department of Mathematics, College of Science and Technology, Nihon University, 1-8-14 Kanda-Surugadai, Chiyoda-Ku, Tokyo 101-8308, Japan. This author was supported partly by the JSPS Grant-in-Aid for Early-Career Scientists 21K17711. Email: (ito.masaru@nihon-u.ac.jp).

†Department of Statistical Inference and Mathematics, Institute of Statistical Mathematics, Japan. This author was supported partly by the JSPS Grant-in-Aid for Early-Career Scientists 23K16844. Email: (bruno@ism.ac.jp).

equal than  $r$  by requiring that the  $n - r$  smallest eigenvalues vanish. And, slightly more generally, we may constraint the rank of a general matrix by forcing the smaller singular values to vanish. But, if we would like to control the range of the whole eigenvalue map, this leads to constraints that, relatively speaking, have been less studied in the literature.

In addition, as we will discuss in Section 5, there are also other interesting problems that can be modelled as an eigenvalue program beyond matrix problems. For example, suppose that we have  $m$  ellipsoids and we would like to find an intersection point that is at the boundary of at least  $\ell$  of those ellipsoids, for some  $\ell \leq m$ . Surprisingly, such a problem can also be cast as an eigenvalue program in an appropriate setting without making use of the usual matrix eigenvalues. More generally, given  $m$  quadratic inequalities, the problem of finding points that satisfy at least  $\ell$  of those inequalities exactly, can also be cast as an eigenvalue program, as we will discuss in Section 5.

The problem in (Eig-Prog) at first glance seems highly nonconvex, because, as exemplified previously, we may use it to express rank constraints. For the sake of concreteness, let  $P$  be a polyhedral set in  $\mathbb{R}^r$ , let  $\mathcal{E} = \mathcal{S}^r$  be the space of real  $r \times r$  symmetric matrices and  $\lambda$  the usual eigenvalue map. If we set  $\mathcal{C} = (\mathbb{R}_+^s \times \{0\}^{r-s}) \cap P$ , the feasible set of (Eig-Prog) consists of positive semidefinite matrices with rank less or equal than  $s$  such that their eigenvalues are in  $P$ . Clearly,  $\lambda^{-1}(\mathcal{C})$  (i.e., the feasible set of (Eig-Prog)) will be nonconvex in general and, from an optimization point of view, it seems to be quite hard to handle directly.

However, in [17], Gowda made several fundamental contributions and he showed that linear optimization over “ $\lambda^{-1}(\mathcal{C})$ ” can be essentially converted to a linear optimization problem over  $\mathcal{C}$  and, similarly, in order to project onto “ $\lambda^{-1}(\mathcal{C})$ ”, it is enough to know how to project a vector  $v \in \mathbb{R}^r$  onto  $\mathcal{C}$ . This is notable, because in the example discussed just now, linear optimization over  $\mathcal{C}$  would correspond to a *linear program*, in stark contrast to linear optimization over  $\lambda^{-1}(\mathcal{C})$ .

With these points in mind, our motivation for this work is to leverage the theoretical findings of [17] into a simple yet practical algorithm and showcase some novel modelling applications of eigenvalue programs. Our contributions are as follows.

- We use the FTvN system framework to analyze and implement a simple projected gradient algorithm for solving (Eig-Prog), see Section 4.
- We discuss some of the modelling capabilities of (Eig-Prog), see Section 5. We start by revisiting *inverse eigenvalue problem* but with general eigenvalue constraints. Next, we show that when  $\mathcal{E}$  corresponds to the Jordan algebra associated to a direct product of second-order cones, we can solve feasibility problems with the so-called *vanishing quadratic constraints*, which require that a certain number of convex quadratic inequalities be satisfied with equality. In particular, given  $m$  ellipsoids, we can model and solve the problem of finding an intersection point that is in the boundary of at least  $\ell$  of those ellipsoids using vanishing quadratic constraints, see Section 5.2.1.

This work is divided as follows. In Section 2, we set up the notation and review notions from variational analysis. In Section 3, we present a self-contained account of the theory of FTvN systems and discuss some fundamental examples. In Section 4 we discuss and analyze a projected gradient algorithm for solving (Eig-Prog). In Section 5 we present some modelling examples together with numerical experiments. Finally, in Section 6 we conclude this paper.

## 2 Preliminaries

Let  $\mathcal{E}$  be a finite dimensional real vector space equipped with an inner product  $\langle \cdot, \cdot \rangle$ . We denote by  $\|\cdot\| = \sqrt{\langle \cdot, \cdot \rangle}$  the induced norm on  $\mathcal{E}$ . We also write  $\|w\|_2 = \sqrt{w^T w}$  for  $w \in \mathbb{R}^r$ . For a set  $S \subset \mathcal{E}$ , we define

$$\text{dist}(x, S) := \inf_{z \in S} \|x - z\|, \quad \text{proj}_S(x) := \{z \in S \mid \|z - x\| = \text{dist}(x, S)\}.$$

We denote the indicator function of  $S$  by  $\delta_S$ . For a vector  $x \in \mathbb{R}^r$ , we denote by  $x^\downarrow \in \mathbb{R}^r$  the vector that corresponds to ordering the elements of  $x$  from largest to smallest, so that  $x_1^\downarrow \geq \dots \geq x_n^\downarrow$ . We also write  $S_\downarrow = \{x^\downarrow \mid x \in S\}$  for a set  $S \subseteq \mathbb{R}^r$ .

**Some variational analysis** Before we move forward we recall some notions from variational analysis, for more details see [30, 28]. Let  $f : \mathcal{E} \rightarrow \mathbb{R} \cup \{+\infty\}$  be a function. We say that  $d$  is a *regular subgradient of  $f$  at  $x$*  if

$$\liminf_{\substack{v \rightarrow 0 \\ v \neq 0}} \frac{f(x+v) - f(x) - \langle d, v \rangle}{\|v\|} \geq 0. \quad (2.1)$$

The set of regular subgradient of  $f$  at  $x$  is called the *regular subdifferential* of  $f$  at  $x$  and is denoted by  $\hat{\partial}f(x)$ .

We say that  $d$  is an *approximate subgradient* (also called *limiting subgradient*) of  $f$  at  $x$  if there are sequences  $\{x^k\}, \{d^k\} \subset \mathcal{E}$  such that  $d^k \in \hat{\partial}f(x^k)$  holds for every  $k$  and the following limits hold:

$$x^k \rightarrow x, \quad f(x^k) \rightarrow f(x), \quad d^k \rightarrow d.$$

The set of approximate subgradients of  $f$  at  $x$  is denoted by  $\partial f(x)$  and is called the *limiting subdifferential of  $f$  at  $x$* . We also define  $\text{dom } \partial f(x) = \{x \in \mathcal{E} \mid \partial f(x) \neq \emptyset\}$ . If  $x$  is such that  $0 \in \partial f(x)$ , then  $x$  is said to be a *critical point*.

Considering different subdifferentials of the indicator function of a given subset  $S \subseteq \mathcal{E}$  leads to different notions of normal cones. In particular, the regular and the limiting normal cones of a set  $S \subseteq \mathcal{E}$  at some  $x \in S$  are given by  $\hat{N}_S(x) := \hat{\partial}\delta_S(x)$  and  $N_S(x) := \partial\delta_S(x)$ , respectively. For more details, see [30, Chapters 6, 8 and Exercise 8.14].

### 3 FTvN systems

The notion of Fan-Theobald-von Neumann system introduced by Gowda is a broad generalization of several notions developed throughout the literature to handle objects that behave similarly to the usual matrix eigenvalue map [17, 18].

**Definition 3.1.** For a map  $\lambda : \mathcal{E} \rightarrow \mathbb{R}^r$ , we say that  $(\mathcal{E}, \mathbb{R}^r, \lambda)$  is a *Fan-Theobald-von Neumann (FTvN) system* if

- (i)  $\|\lambda(x)\|_2 = \|x\|, \forall x \in \mathcal{E}$
- (ii)  $\langle x, y \rangle \leq \langle \lambda(x), \lambda(y) \rangle, \forall x, y \in \mathcal{E}$
- (iii) For any  $c \in \mathcal{E}$  and  $\mu \in \lambda(\mathcal{E})$ , the following set is nonempty:

$$U(c, \mu) := \{z \in \mathcal{E} \mid \lambda(z) = \mu, \langle c, z \rangle = \langle \lambda(c), \lambda(z) \rangle\}. \quad (3.1)$$

*Remark 3.2.* The definition given in [17, 18] is more general than Definition 3.1, because it is considered over arbitrary real inner product spaces, while here we restrict ourselves to the finite dimensional case where one of the spaces is  $\mathbb{R}^r$ . Also, item (iii) is stated in a slightly different manner.

The first two items in Definition 3.1 are relatively natural: they require that the eigenvalue map  $\lambda$  be compatible with the norm used in  $\mathcal{E}$  and that the inner product satisfy an inequality analogous to a classical inequality over symmetric matrices (e.g., [22, Theorem 2.2]). The third item is somewhat more technical, but can be seen as complementing item (ii) as follows: if  $c, y \in \mathcal{E}$  are fixed, item (ii) implies that  $\langle c, y \rangle \leq \langle \lambda(c), \lambda(y) \rangle$ . However, applying item (iii) to  $c$  and  $\lambda(y)$ , we get that this inequality becomes tight for at least one  $z \in \mathcal{E}$  that has the same eigenvalues as  $y$ .

For FTvN systems, the projected gradient algorithm discussed in this paper relies on evaluating the map  $\lambda(\cdot)$  and being able to obtain an element of the set  $U(c, \mu)$ . For many cases, this is done by ‘‘spectral decompositions’’ as we will see later.

We define the *rank* of  $x \in \mathcal{E}$  by

$$\text{rank}(x) := \text{number of nonzero elements in } \lambda(x). \quad (3.2)$$

For a FTvN system  $(\mathcal{E}, \mathbb{R}^r, \lambda)$ , a *spectral set* in  $\mathcal{E}$  is a subset  $S$  of  $\mathcal{E}$  of the form

$$S = \lambda^{-1}(Q) \text{ for some } Q \subseteq \mathbb{R}^r.$$

Observe that any spectral set  $S = \lambda^{-1}(Q)$  satisfies

$$\lambda(S) = \lambda(\mathcal{E}) \cap Q, \quad S = \lambda^{-1}(\lambda(S)). \quad (3.3)$$

A very useful property of FTvN systems is that some optimization problems over  $\mathcal{E}$  can be equivalently reformulated over  $\mathbb{R}^r$  which may lead to significant simplifications.

**Proposition 3.3** (Gowda [17]). *Let  $(\mathcal{E}, \mathbb{R}^r, \lambda)$  be a FTvN system and  $S$  be a spectral set in  $\mathcal{E}$ . For any real-valued function  $\varphi : \mathbb{R}^r \rightarrow \mathbb{R}$  and  $c \in \mathcal{E}$ , we have*

$$\sup_{x \in S} \{\langle c, x \rangle + \varphi(\lambda(x))\} = \sup_{w \in \lambda(S)} \{\langle \lambda(c), w \rangle + \varphi(w)\}. \quad (3.4)$$

Moreover,  $x^* \in S$  attains the supremum of the left-hand side if and only if  $\langle c, x^* \rangle = \langle \lambda(c), \lambda(x^*) \rangle$  and  $\lambda(x^*)$  attains the supremum of the right-hand side. In other words, this is if and only if  $x^* \in U(c, w^*)$  for some  $w^* \in \lambda(S)$  that attains the supremum of the right hand side.

*Proof.* The assertion is essentially [17, Corollary 3.3] which follows from more general results therein. Here, we give a direct proof for the sake of self-containment. Let  $L$  and  $R$  denote the left and the right hand sides of (3.4), respectively.

Let  $x \in S$ . By item (ii) of Definition 3.1,  $w := \lambda(x)$  is a feasible solution to right-hand side of (3.4) satisfying  $\langle c, x \rangle \leq \langle \lambda(c), w \rangle$ . Therefore,  $L \leq R$ .

Conversely, for any  $w \in \lambda(S)$ , the set  $U(c, w)$  is nonempty, by item (iii) of Definition 3.1. With that we pick *any*  $x \in U(c, w)$  and, by definition, we have  $\lambda(x) = w$  and  $\langle c, x \rangle = \langle \lambda(c), \lambda(x) \rangle = \langle \lambda(c), w \rangle$ . These equalities imply that

$$\langle c, x \rangle + \varphi(\lambda(x)) = \langle \lambda(c), w \rangle + \varphi(w).$$

We also have  $x \in S$ , since  $x \in \lambda^{-1}(w) \subseteq \lambda^{-1}(\lambda(S)) = S$  (see (3.3)). In conclusion, for the two maximization problems associated to  $L$  and  $R$ , we can recover a feasible solution of the former from a feasible solution of the latter, without changing the objective value. This indicates  $L \geq R$  which shows that (3.4) holds.

The development done so far implies that if  $x^* \in S$  is an optimal solution to the left-hand side of (3.4), then  $w := \lambda(x^*)$  must be an optimal solution to the right-hand side. In view of  $L = R$ , we have  $\langle c, x^* \rangle + \varphi(\lambda(x^*)) = \langle \lambda(c), \lambda(x^*) \rangle + \varphi(\lambda(x^*))$ , which leads to  $\langle c, x^* \rangle = \langle \lambda(c), \lambda(x^*) \rangle$ . Conversely, suppose that  $x^* \in S$  is such that  $\langle c, x^* \rangle = \langle \lambda(c), \lambda(x^*) \rangle$  and  $\lambda(x^*)$  attains the supremum of the right-hand side of (3.4). Again, in view of  $L = R$ ,  $x^*$  must be optimal to the left-hand side of (3.4).  $\square$

For instance, when  $\varphi(\cdot) \equiv 0$ , this assertion says that linear optimization  $\sup_{x \in S} \langle c, x \rangle$  for a spectral set  $S$  can be reduced to  $\sup_{w \in \lambda(S)} \langle \lambda(c), w \rangle$  whose solution  $w^*$  can be used to recover a solution to  $\sup_{x \in S} \langle c, x \rangle$  through the computation of an element  $x^* \in U(c, w^*)$ .

Taking  $\varphi(x) = -\|x\|^2/2$ , we obtain the following useful fact regarding projections, see also [17, Corollary 3.8].

**Corollary 3.4.** *Let  $(\mathcal{E}, \mathbb{R}^r, \lambda)$  be a FTvN system and  $S$  be a spectral set in  $\mathcal{E}$ . For any  $z \in \mathcal{E}$ , we have*

$$\text{proj}_S(z) = \bigcup_{w^* \in \text{proj}_{\lambda(S)}(\lambda(z))} U(z, w^*).$$

*Proof.* Since  $\|x - z\|^2/2 = \|x\|^2/2 - \langle z, x \rangle + \|z\|^2/2$  holds,  $\text{proj}_S(z)$  is the set of solutions of the problem of maximizing

$$\langle z, x \rangle - \frac{1}{2}\|x\|^2 = \langle z, x \rangle - \frac{1}{2}\|\lambda(x)\|_2^2,$$

over  $x \in S$ , where the equality follows from item (i) of Definition 3.1. Therefore, the result follows by taking  $\varphi(x) := -\|x\|^2/2$  in Proposition 3.3.  $\square$

We mention in passing that in the context of spectral sets over symmetric matrices there is a related earlier result by Lewis and Malick [23, Theorem A.1].

### 3.1 Examples of FTvN systems

We give some examples of FTvN systems. In these examples, we will always check that the eigenvalue mapping  $\lambda$  is *semialgebraic*, which will significantly simplify the analysis of our projected gradient method. We recall that a semialgebraic set in  $\mathbb{R}^n$  a finite union of sets of the form

$$\{x \in \mathbb{R}^m \mid p_i(x) = 0, i = 1, \dots, k \text{ and } q_j(x) < 0, j = 1, \dots, \ell\},$$

where  $p_i$  and  $q_j$  are polynomials of real coefficients. A *semialgebraic function* is a map  $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$  whose graph  $\{(x, y) \mid y = f(x)\}$  is a semialgebraic set in  $\mathbb{R}^{n+m}$ . Basic facts on semialgebraic sets and functions can be seen in [19], although we only make use of the most elementary properties.

We now start with what is perhaps the most basic example of FTvN system.

**Example 3.5** (The symmetric matrices). The vector space  $\mathcal{S}^n$  of  $n$  by  $n$  real symmetric matrices can be seen as a FTvN system  $(\mathcal{S}^n, \mathbb{R}^n, \lambda)$  with the eigenvalue map  $\lambda : \mathcal{S}^n \rightarrow \mathbb{R}_\downarrow^n$  defined as follows:

$$\lambda(X) := (\lambda_1(X), \dots, \lambda_n(X)),$$

where  $\lambda_i(X)$  is the  $i$ -th largest eigenvalue of  $X$ . Here, we assume that  $\mathcal{S}^n$  is equipped with the trace inner product, so that  $\|X\| = \|\lambda(X)\|_2$  holds for all  $X \in \mathcal{S}^n$ . Item (ii) of Definition 3.1 follows from a classical inequality (e.g., [22, Theorem 2.2]). Finally, we take a look at item (iii) of Definition 3.1. For  $C \in \mathcal{S}^n$  and  $\mu \in \mathbb{R}_\downarrow^n$ , an element of  $U(C, \mu)$  for  $C \in \mathcal{S}^n$  and  $\mu \in \lambda(\mathbb{R}^n) = \mathbb{R}_\downarrow^n$  can be computed as follows. Letting  $C = U^T \text{Diag}(\lambda(C))U$  an eigenvalue decomposition of  $C$  with an orthogonal matrix  $U \in \mathbb{R}^{n \times n}$ , we obtain

$$U^T \text{Diag}(\mu)U \in U(C, \mu).$$

An entirely analogous analysis can also be carried out for complex Hermitian matrices. That  $\lambda$  is a semialgebraic map is well-known, but it also follows from a more general result on Jordan algebras, see Proposition 3.6.

#### 3.1.1 Jordan algebras

We discuss an important subclass of FTvN systems, the Euclidean Jordan algebras. For more details see [11, 12]. A finite-dimensional real inner product space  $(\mathcal{E}, \langle \cdot, \cdot \rangle)$  is called a *Euclidean Jordan algebra* if it admits an operation  $\circ : \mathcal{E} \times \mathcal{E} \rightarrow \mathcal{E}$  (called a *Jordan product*) which has an identity element  $e \in \mathcal{E}$  (i.e.,  $x \circ e = x$  for all  $x \in \mathcal{E}$ ) and satisfies

$$x \circ y = y \circ x, \quad x \circ (x^2 \circ y) = x^2 \circ (x \circ y), \quad \langle z \circ x, y \rangle = \langle z, x \circ y \rangle,$$

for all  $x, y, z \in \mathcal{E}$ , where  $x^2 = x \circ x$ .

The *cone of squares* associated to  $\mathcal{E}$  is given by  $\mathcal{K} = \{x \circ x \mid x \in \mathcal{E}\}$  and it is known to be a *symmetric cone*, i.e., a self-dual homogeneous cone. Conversely, every symmetric cone in finite dimensions is known to arise as the cone of squares of some Euclidean Jordan algebra.

An element  $v \in \mathcal{E}$  is called an *idempotent* if  $v \circ v = v$ . Two idempotents  $u, v \in \mathcal{E}$  are *orthogonal* if  $u \circ v = 0$ . Note that  $\langle u, v \rangle = \langle u^2, v \rangle = \langle u, u \circ v \rangle = 0$  for orthogonal idempotents  $u$  and  $v$ .

The maximal number of nonzero idempotents which are pairwise orthogonal is well-defined and called the *rank* of  $\mathcal{E}$ . A remarkable fact about Euclidean Jordan algebras  $\mathcal{E}$  is that they admit a form of spectral decomposition very similar to the matrix case. Namely if  $\mathcal{E}$  has rank  $r$ , then any  $x \in \mathcal{E}$  can be expressed as

$$x = \lambda_1 v_1 + \cdots + \lambda_r v_r,$$

for some  $(\lambda_1, \dots, \lambda_r) \in \mathbb{R}_{\downarrow}^r$  and the  $v_1, \dots, v_r \in \mathcal{E}$  are nonzero such that  $v_i^2 = v_i$ ,  $v_i \circ v_j = 0$  ( $i \neq j$ ),  $\sum_{i=1}^r v_i = e$ . Here,  $\lambda_1, \dots, \lambda_r$  are called the *eigenvalues* of  $x$  and  $\{v_1, \dots, v_r\}$  is called a *Jordan frame* of  $x$ . The eigenvalues of  $x$  are unique up to permutation.

The trace map  $\text{tr}(x) := \lambda_1 + \cdots + \lambda_r$  yields an inner product  $(x, y) \mapsto \text{tr}(x \circ y)$  on  $\mathcal{E}$ . With that, we can define the map  $\lambda : \mathcal{E} \rightarrow \mathbb{R}_{\downarrow}^r$  that maps an element to its eigenvalues ordered from largest to smallest.

Equipped with the trace inner product on  $\mathcal{E}$ , the tuple  $(\mathcal{E}, \mathbb{R}^r, \lambda)$  forms a FTvN system, see [17, Section 4]. For the sake of completeness, we check the details. First, item (i) of Definition 3.1 follows by definition of the inner product. Then, item (ii) follows from [2, Theorem 23] which states that  $\langle x, y \rangle \leq \langle \lambda(x), \lambda(y) \rangle$ . Moreover, given  $c \in \mathcal{E}$  and  $\mu \in \lambda(\mathcal{E}) = \mathbb{R}_{\downarrow}^r$ , considering the spectral decomposition  $c = \lambda_1(c)v_1 + \cdots + \lambda_r(c)v_r$ , we construct the following element of  $U(c, \mu)$ :

$$\mu_1 v_1 + \cdots + \mu_r v_r \in U(c, \mu). \quad (3.5)$$

We also observe that the rank of  $x \in \mathcal{E}$  as defined in (3.2) corresponds to the number of nonzero eigenvalues that  $x$  has.

Finally, let us check that the map  $\lambda$  is semialgebraic, which may be obvious for those familiar with Jordan algebras. However, since this fact is not clearly stated in the classical references, we provide a proof.

**Proposition 3.6.** *In a finite-dimensional Euclidean Jordan Algebra  $\mathcal{E}$  of rank  $r$ , the map  $\lambda : \mathcal{E} \rightarrow \mathbb{R}_{\downarrow}^r$  is semialgebraic.*

*Proof.* Consider the function  $\det : \mathcal{E} \rightarrow \mathbb{R}$  such that  $\det(x) = \prod_{i=1}^r \lambda_i(x)$  holds. It is known that the composition of an elementary symmetric polynomial with  $\lambda$  leads to a polynomial in  $x$ , see [11, Theorem III.1.2]. In particular, since  $\det$  is the composition of  $\lambda$  with the elementary symmetric polynomial  $\prod_{i=1}^r t_i$ , we conclude that  $\det$  is a polynomial function. Furthermore, since  $\lambda_i(x - te) = \lambda_i(x) - t$  holds for all  $x$  and  $i = 1, \dots, r$ , the eigenvalues of a given  $x \in \mathcal{E}$  are exactly the roots of the polynomial  $t \mapsto \det(x - te)$ . With this, the graph of  $\lambda$  can be written as the semialgebraic set

$$\{(x, u) \in \mathcal{E} \times \mathbb{R}_{\downarrow}^r \mid \forall t \in \mathbb{R}, \det(x - te) = (t - u_1) \cdots (t - u_r)\},$$

so  $\lambda$  is indeed a semialgebraic function<sup>1</sup>. □

It is well-known that the  $n \times n$  real symmetric matrices as in Example 3.5 form a Jordan algebra with the Jordan product being  $X \circ Y = (XY + YX)/2$  and corresponding symmetric cone being the positive semidefinite matrices. Next, we recall another well-known example.

**Example 3.7** (The algebra of second-order cones). Consider the vector space  $\mathcal{E} := \mathbb{R}^n \times \mathbb{R}$  with the Jordan product  $\circ$  such that for  $(x, t), (y, u) \in \mathbb{R}^n \times \mathbb{R}$  we have

$$(x, t) \circ (y, u) := \frac{\sqrt{2}}{2}(ux + ty, \langle x, y \rangle + tu), \quad (3.6)$$

where  $\langle \cdot, \cdot \rangle$  is the usual dot product. By convention we write  $\mathbb{R}^{n+1}$  instead of  $\mathbb{R}^n \times \mathbb{R}$  and for  $(x, t) \in \mathbb{R}^{n+1}$ , we assume that  $x \in \mathbb{R}^n$  and  $t \in \mathbb{R}$ . With that,  $\mathbb{R}^{n+1}$  can be seen as a Euclidean Jordan algebra, e.g., see [11, Chapter II, Section 1], [15, Section 2] or [31, page 141]. Depending on the reference, the

<sup>1</sup>Here we used the fact that the solution set of a first-order formula quantified over semialgebraic sets and functions must be semialgebraic as well, e.g., [19, Proposition 2.2.4].

$\frac{\sqrt{2}}{2}$  factor may be omitted in (3.6). In any case, the corresponding cone of squares is the second-order cone  $\mathcal{L}_2^{n+1} := \{(x, t) \in \mathbb{R}^n \times \mathbb{R} \mid \|x\|_2 \leq t\}$ , where  $\|\cdot\|_2$  is the usual Euclidean norm.

Given  $(x, t) \in \mathbb{R}^{n+1}$ , we have the spectral decomposition

$$(x, t) = \lambda_+ e_+ + \lambda_- e_-, \quad \lambda_{\pm} = \frac{\sqrt{2}}{2}(t \pm \|x\|_2), \quad e_{\pm} = \frac{\sqrt{2}}{2}(\pm w_x, 1), \quad (3.7)$$

where  $w_x = x/\|x\|_2$  if  $x \neq 0$ , otherwise  $w_x$  can be taken to be any vector of unit norm.

The inner product in  $\mathbb{R}^{n+1}$  satisfies  $\text{tr}((x, t) \circ (y, u)) = \langle x, y \rangle + tu$ , which coincides with the usual Euclidean inner product on  $\mathbb{R}^{n+1}$  and explains the  $\frac{\sqrt{2}}{2}$  factor in (3.6). With the eigenvalue map  $\lambda(x, t) := (\lambda_+, \lambda_-)$ , the tuple  $(\mathbb{R}^{n+1}, \mathbb{R}^2, \lambda)$  forms a FTvN system, where the inner products in  $\mathbb{R}^{n+1}, \mathbb{R}^2$  are the usual Euclidean one.

Following (3.5), given  $c = (u, t)$  and  $\mu = (\mu_+, \mu_-) \in \mathbb{R}_+^2$ , the spectral decomposition  $c = \lambda_+ e_+ + \lambda_- e_-$  of  $c$  allows us to compute an element of  $U(c, \mu)$  as follows.

$$\mu_+ e_+ + \mu_- e_- \in U(c, \mu). \quad (3.8)$$

### 3.1.2 Singular values

For positive integers  $m, n$  and  $r = \min(m, n)$ , we denote by  $\sigma(X) = (\sigma_1(X), \dots, \sigma_r(X))$  the ordered vector of singular values of  $X \in \mathbb{R}^{m \times n}$  so that  $\sigma_1(X) \geq \dots \geq \sigma_r(X)$ . We use the inner product

$$\langle X, Y \rangle = \text{tr}(X^T Y),$$

for matrices  $X, Y \in \mathbb{R}^{m \times n}$ . With that, we check that the tuple  $(\mathbb{R}^{m \times n}, \mathbb{R}^r, \sigma)$  is a FTvN system. Item (i) of Definition 3.1 holds by definition, as for item (ii), it follows from [24, Theorem 4.6]. Finally, to compute an element of  $U(C, \mu)$  for  $C \in \mathbb{R}^{m \times n}$  and  $\mu \in \sigma(\mathbb{R}^{m \times n}) = (\mathbb{R}_+^r)_\downarrow$ , let  $C = U^T \text{Diag}(\sigma(C))V$  be a singular value decomposition of  $C$  with square orthogonal matrices  $U \in \mathbb{R}^{n \times n}$  and  $V \in \mathbb{R}^{m \times m}$ . Then, we have

$$U^T \text{Diag}(\mu)V \in U(C, \mu).$$

Finally, the map  $\sigma$  is semialgebraic because  $\sigma(X)$  correspond to the square-root of the eigenvalues of  $XX^T$ .

Complex analogues can also be constructed with the inner product  $\text{Re}(\text{tr}(X^* Y))$  where  $X^*$  denotes the adjoint of  $X \in \mathbb{C}^{m \times n}$ .

### 3.1.3 Direct products

Other FTvN systems can be constructed by taking the direct product of the above examples. Given  $m$  FTvN systems  $(\mathcal{E}^{(i)}, \mathbb{R}^{r_i}, \lambda^{(i)})$ ,  $i = 1, 2, \dots, m$ , defining

$$\mathcal{E} = \mathcal{E}^{(1)} \times \dots \times \mathcal{E}^{(m)}, \quad r = r_1 + \dots + r_m, \quad \lambda = (\lambda^{(1)}, \dots, \lambda^{(m)}), \quad (3.9)$$

the tuple  $(\mathcal{E}, \mathbb{R}^r, \lambda)$  forms a FTvN system. Here, the inner product is given by

$$\langle (x_1, \dots, x_m), (y_1, \dots, y_m) \rangle = \sum_{i=1}^m \langle x_i, y_i \rangle_i,$$

where  $\langle \cdot, \cdot \rangle_i$  indicates the inner product over each  $\mathcal{E}^{(i)}$ . Note that, given  $c = (c_1, \dots, c_m) \in \mathcal{E}$  and  $\mu = (\mu_1, \dots, \mu_m) \in \lambda(\mathcal{E})$ , we have

$$U(c, \mu) = U(c_1, \mu_1) \times \dots \times U(c_m, \mu_m). \quad (3.10)$$

In (3.10) the “ $\supseteq$ ” inclusion follows from the definition of the  $U(c_i, \mu_i)$ . As for the inclusion “ $\subseteq$ ”, it follows from the fact that if  $z \in U(c, \mu)$ , then  $\lambda^{(i)}(z_i) = \mu_i$  holds for every  $i$  and  $\sum_{i=1}^m (\langle \lambda^{(i)}(c_i), \lambda^{(i)}(z_i) \rangle -$

$\langle c_i, z_i \rangle = 0$  holds. However, each term of the summation is nonnegative by item (ii) of Definition 3.1, so  $\langle \lambda^{(i)}(c_i), \lambda^{(i)}(z_i) \rangle = \langle c_i, z_i \rangle$  holds for all  $i$  and we have  $z_i \in U(c_i, \mu_i)$  for every  $i$ .

In the particular case where each  $\mathcal{E}_i$  is a Euclidean Jordan algebra, there is another way to see  $\mathcal{E}$  as a FTvN system employing the eigenvalue map

$$x \mapsto \lambda^\downarrow(x) := (\lambda^{(1)}(x), \dots, \lambda^{(m)}(x))^\downarrow. \quad (3.11)$$

In this case, the tuple  $(\mathcal{E}, \mathbb{R}^r, \lambda^\downarrow)$  also forms a FTvN system, since  $\mathcal{E}$  is also a Jordan algebra and, in fact,  $\lambda^\downarrow$  correspond to the same eigenvalue map discussed in Section 3.1.1.

Still, for  $c = (c_1, \dots, c_m) \in \mathcal{E}$  and  $\mu = (\mu_1, \dots, \mu_m) \in \lambda^\downarrow(\mathcal{E})$ , it makes sense to discuss how to compute an element in  $U(c, \mu)$  using the  $U(c_i, \mu_i)$ , since in applications we would typically consider spectral decompositions according to the block structure of  $\mathcal{E}$ . In order to obtain an element in  $U(c, \mu)$ , we sort the  $r = r_1 + \dots + r_m$  real values in  $\{\lambda_j^{(i)}(c_i) \mid i = 1, \dots, m; j = 1, \dots, r_i\}$  (which are the eigenvalues of  $c$ ) in descending order so that  $\lambda_{j_k}^{(i_k)}(c_{i_k})$  denotes  $k$ -th largest eigenvalue of  $c$ . Then, we have

$$\sum_{k=1}^r \mu_{i_k} \tilde{v}_{j_k}^{(i_k)} \in U(c, \mu), \quad (3.12)$$

where  $\tilde{v}_j^{(i)}$  is defined as

$$\tilde{v}_j^{(i)} := (0, \dots, 0, \underbrace{v_j^{(i)}}_{i\text{-th position}}, 0, \dots, 0),$$

and  $\{v_1^{(i)}, \dots, v_{r_i}^{(i)}\}$  is a Jordan frame for  $c^{(i)}$ .

The difference between  $(\mathcal{E}, \mathbb{R}^r, \lambda^\downarrow)$  and  $(\mathcal{E}, \mathbb{R}^r, \lambda)$  boils down to whether it is desirable to order all eigenvalues or to only order eigenvalues inside each block  $\mathcal{E}_i$ . We will use both  $(\mathcal{E}, \mathbb{R}^r, \lambda^\downarrow)$  and  $(\mathcal{E}, \mathbb{R}^r, \lambda)$  in our examples in Section 5.

## 4 Projected gradient and related topics

For a spectral set  $\lambda^{-1}(\mathcal{C})$  with  $\mathcal{C} \subseteq \mathbb{R}^r$ , it follows from Corollary 3.4 that in order to project  $x$  onto  $\lambda^{-1}(\mathcal{C})$  it is enough to do as follows. First, we let  $w$  be such that

$$w \in \arg \min \{\|\lambda(x) - v\|_2 \mid v \in \lambda(\mathcal{E}), v \in \mathcal{C}\}.$$

Then, we compute an element of  $U(x, w)$  in order to obtain the desired projection. In particular, we have

$$\text{proj}_{\lambda^{-1}(\mathcal{C})}(x) \supseteq U(x, w).$$

Being able to compute the projection onto  $\lambda^{-1}(\mathcal{C})$ , allows us (in theory) to use a simple projected gradient algorithm for solving (Eig-Prog) by considering iterations that satisfy

$$x_{k+1} \in \text{proj}_{\lambda^{-1}(\mathcal{C})}(x_k - \alpha_k \nabla f(x_k)), \quad (4.1)$$

where  $\alpha_k > 0$  is a step-size, see Algorithm 1.

---

### Algorithm 1: Projected gradient method for (Eig-Prog)

---

**Input:**  $f : \mathcal{E} \rightarrow \mathbb{R}, \mathcal{C} \subset \mathbb{R}^r, x_0 \in \mathcal{E}$

- 1  $k \leftarrow 0$
- 2 **while** *stopping criteria not satisfied* **do**
- 3     Select a step-size  $\alpha_k > 0$ .
- 4     Set  $y_k = x_k - \alpha_k \nabla f(x_k)$ .
- 5     Compute  $w_k \in \arg \min_v \{\|\lambda(y_k) - v\|_2 \mid v \in \lambda(\mathcal{E}), v \in \mathcal{C}\}$ .
- 6     Compute  $x_{k+1} \in U(y_k, w_k)$ .
- 7      $k \leftarrow k + 1$
- 8 **end**

---

First we analyze the convergence properties of Algorithm 1 under the assumption that the problem data is semialgebraic, which is enough to cover a wide range of applications.

**Theorem 4.1** ([1]). *Let  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  be a differentiable semialgebraic function with  $L$ -Lipschitz continuous gradient and  $S \subset \mathbb{R}^n$  be a closed semialgebraic set. Let  $\{x_k\}$  be generated by*

$$x_{k+1} \in \text{proj}_S(x_k - \alpha_k \nabla f(x_k))$$

*with step-size  $\alpha_k \in (\varepsilon, 1/L - \varepsilon)$  for some  $\varepsilon \in (0, 1/(2L))$ . If  $\{x_k\}$  is bounded, then it converges to some  $x^*$  such that  $0 \in \nabla f(x^*) + N_S(x^*)$ . Moreover,  $\sum_k \|x_{k+1} - x_k\|^2 < +\infty$ .*

*Proof.* This is a special case of Theorem 5.3 in [1] for semialgebraic functions. See also the observation following Theorem 5.3.  $\square$

**Theorem 4.2.** *Let  $\mathcal{C}$  be a closed subset of  $\mathbb{R}^r$  and  $f : \mathcal{E} \rightarrow \mathbb{R}$  be a  $C^1$  function with  $L$ -Lipschitz continuous gradient. Suppose that  $f$ ,  $\lambda$  and  $\mathcal{C}$  are semialgebraic. Let  $\{x_k\}$  be generated by Algorithm 1 with step-size  $\alpha_k \in (\varepsilon, 1/L - \varepsilon)$  for some  $\varepsilon \in (0, 1/(2L))$ . If  $\{x_k\}$  is bounded then it converges to a critical point  $x^*$  of  $f + \delta_{\lambda^{-1}(\mathcal{C})}$ , i.e.,  $0 \in \nabla f(x^*) + N_{\lambda^{-1}(\mathcal{C})}(x^*)$ . Moreover,  $\sum_k \|x_{k+1} - x_k\|^2 < +\infty$  holds.*

*Proof.* Algorithm 1 is the projected gradient method (4.1) for the constraint set  $\lambda^{-1}(\mathcal{C})$ . Since  $\mathcal{C}$  is semialgebraic, the spectral set  $\lambda^{-1}(\mathcal{C})$  is also semialgebraic, see [19, Proposition 2.2.7]. Hence, the assertion follows by Theorem 4.1. We also remark that since  $f$  is  $C^1$ , we have  $\partial(f + \delta_{\lambda^{-1}(\mathcal{C})})(x^*) = \nabla f(x^*) + N_{\lambda^{-1}(\mathcal{C})}(x^*)$ , by [30, Exercise 8.8].  $\square$

Next, we briefly comment on local convergence rates for Algorithm 1, which, in general, requires further assumptions. In what follows, let  $F : \mathcal{E} \rightarrow \mathbb{R} \cup \{+\infty\}$  be a proper closed function. Then,  $F$  is said to be a *Kurdyka-Lojasiewicz (KL) function* with exponent  $\alpha$  (e.g., see [25, Definition 2.2]) at  $\bar{x} \in \text{dom } \partial F$  if there exists  $c, \epsilon > 0$  and  $\nu \in (0, \infty]$  so that

$$\text{dist}(0, \partial F(x)) \geq c(F(x) - F(\bar{x}))^\alpha \quad (4.2)$$

holds whenever  $\|x - \bar{x}\| \leq \epsilon$  and  $F(\bar{x}) < F(x) < F(\bar{x}) + \nu$ . We also call (4.2) the *KL property of  $F$  at  $\bar{x}$* .

Next, let  $F$  be defined by  $F(x) := f(x) + \delta_{\lambda^{-1}(\mathcal{C})}(x)$ , where we recall that  $\delta_{\lambda^{-1}(\mathcal{C})}$  denotes the indicator function of  $\lambda^{-1}(\mathcal{C})$ . If  $F$  is a KL function with exponent  $\alpha$ , then a local convergence rate for Algorithm 1 can be obtained through [13, Theorem 3.4]. In particular, if  $\alpha = 1/2$ , the method has local linear convergence, see also the discussion in [25, Section 5.3]. We note this as a proposition.

**Proposition 4.3.** *Let  $F := f(\cdot) + \delta_{\lambda^{-1}(\mathcal{C})}(\cdot)$  and let  $\{x_k\}$  be generated by Algorithm 1 with step-size  $\alpha_k \in (\varepsilon, 1/L - \varepsilon)$  for some  $\varepsilon \in (0, 1/(2L))$ . If  $\{x_k\}$  converges to  $\bar{x}$  and  $F$  has the KL property with exponent  $1/2$  at  $\bar{x}$ , then  $\{x_k\}$  converges linearly to  $x^*$ .*

*Proof.* As remarked previously, this result follows from [13, Theorem 3.4], but here we discuss some of the details. In reality, in order to invoke [13, Theorem 3.4], it is necessary to check that  $x_k$  satisfy certain conditions which were labeled  $H_1, H_2$  and  $H_3$  therein. This is indeed the case, because  $x_k$  is a particular case of the iteration discussed at the beginning of [13, Section 4], see Equation (12) therein and the subsequent comments. The other assumption that needs to be checked is that  $\{x_k\}$   $F$ -converges to  $\bar{x}$ , which means that  $x_k \rightarrow \bar{x}$  and  $F(x_k) \rightarrow F(\bar{x})$ . The former holds by assumption and the latter holds because of the continuity of  $f$  and the fact that  $x_k \in \lambda^{-1}(\mathcal{C})$  for all  $k$ . This takes care of all the assumptions required to invoke case (ii) of [13, Theorem 3.4] which leads to linear convergence, or in the parlance of [13], *exponential convergence*.  $\square$

Let us briefly examine the KL condition for  $F$ . Since  $f$  is smooth, we have  $\partial F(x) = \nabla f(x) + \partial \delta_{\lambda^{-1}(\mathcal{C})}(x)$ , see [30, Exercise 8.8]. We recall, however, that  $\partial \delta_{\lambda^{-1}(\mathcal{C})}(x)$  is the normal cone of  $\lambda^{-1}(\mathcal{C})$  at  $x$ . So, for  $x \in \lambda^{-1}(\mathcal{C})$ , (4.2) can be written as

$$\text{dist}(\nabla f(x), -N_{\lambda^{-1}(\mathcal{C})}(x)) \geq c(f(x) - f(\bar{x}))^\alpha \quad (4.3)$$

In view of (4.3), it is important to analyze the normal cone  $N_{\lambda^{-1}(\mathcal{C})}(x)$ . In the particular case where  $\mathcal{E}$  is a Jordan algebra and  $\mathcal{C} \subseteq \mathbb{R}^r$  is a permutation invariant set<sup>2</sup>, the computation of the normal cone  $N_{\lambda^{-1}(\mathcal{C})}(x)$  can be related to the normal cone of  $\mathcal{C}$  at  $\lambda(x)$  as follows.

**Proposition 4.4.** *Let  $\mathcal{E}$  be a Euclidean Jordan algebra of rank  $r$  and let  $\mathcal{C} \subseteq \mathbb{R}^r$  be a permutation invariant closed set. Then*

$$N_{\lambda^{-1}(\mathcal{C})}(x) = \{s \in \mathcal{E} \mid \exists \mathcal{J} \in \mathcal{J}(x, s) \text{ with } \text{diag}(s, \mathcal{J}) \in N_{\mathcal{C}}(\lambda(x))\},$$

where  $\mathcal{J}(x, s)$  is the set of Jordan frames  $\{e_1, \dots, e_r\}$  for which  $x = \lambda_1(x)e_1 + \dots + \lambda_r(x)e_r$  and  $s = a_1e_1 + \dots + a_re_r$  hold<sup>3</sup> and  $\text{diag}(s, \mathcal{J})$  is the vector  $(a_1, \dots, a_r)$ .

*Proof.* It follows directly from [27, Theorem 27] applied to  $\delta_{\mathcal{C}}$  since  $\delta_{\mathcal{C}} \circ \lambda$  is the indicator function of  $\lambda^{-1}(\mathcal{C})$  and  $N_{\mathcal{C}}(u) = \partial\delta_{\mathcal{C}}$ .  $\square$

## 4.1 Application to feasibility problems and convergence rates

In this section, we consider a feasibility problem of the following form:

$$\text{Find } x \in \bar{\mathcal{C}} \cap \lambda^{-1}(\mathcal{C}), \quad (4.4)$$

where  $(\mathcal{E}, \mathbb{R}^r, \lambda)$  is a FTvN system,  $\mathcal{C} \subset \mathbb{R}^r$  and  $\bar{\mathcal{C}} \subset \mathcal{E}$  is a convex set for which  $\text{proj}_{\bar{\mathcal{C}}}(\cdot)$  is assumed to be available. This can be reformulated as the following optimization problem.

$$\begin{aligned} \min_{x \in \mathcal{E}} \quad & f(x) := \frac{1}{2} \text{dist}(x, \bar{\mathcal{C}})^2 \\ \text{subject to} \quad & \lambda(x) \in \mathcal{C}. \end{aligned} \quad (4.5)$$

The gradient of  $f$  is given by

$$\nabla f(x) = x - \text{proj}_{\bar{\mathcal{C}}}(x),$$

and if  $\bar{\mathcal{C}}$  is semialgebraic then  $f$  is semialgebraic as well [19, Proposition 2.2.8]. Note that  $\nabla f$  is 1-Lipschitz continuous. Therefore, we can use Algorithm 1, where the resulting iterative scheme can be described as

$$y_k := (1 - \alpha_k)x_k + \alpha_k \text{proj}_{\bar{\mathcal{C}}}(x_k), \quad x_{k+1} \in \text{proj}_{\lambda^{-1}(\mathcal{C})}(y_k), \quad k = 0, 1, 2, \dots \quad (4.6)$$

In particular, when  $\alpha_k \in (\varepsilon, 1 - \varepsilon)$  holds for some  $\varepsilon \in (0, 1)$  and  $f$  is semialgebraic, we obtain the convergence guarantee described in Theorem 4.2.

Regarding convergence rates, it seems natural that this must depend on the geometry of the intersection between  $\bar{\mathcal{C}}$  and  $\lambda^{-1}(\mathcal{C})$ . In what follows, we will verify that if the intersection is *transversal* at  $x$ , then  $f + \delta_{\lambda^{-1}(\mathcal{C})}$  satisfy the KL property with exponent  $1/2$  at  $x$ . We recall that  $S_1, S_2 \subseteq \mathcal{E}$  are said to be *transversal* at  $\bar{x}$  if

$$N_{S_1}(\bar{x}) \cap (-N_{S_2}(\bar{x})) = \{0\}. \quad (4.7)$$

Transversality is related to well-known constraint qualifications used in several contexts. With that we have the following result. Note that this will not require that  $\lambda$  or  $\bar{\mathcal{C}}$  be semialgebraic.

**Proposition 4.5.** *If the iteration (4.6) converges to a critical point  $\bar{x}$  of  $f = \frac{1}{2} \text{dist}(\cdot, \bar{\mathcal{C}})^2 + \delta_{\lambda^{-1}(\mathcal{C})}(\cdot)$  and  $f$  satisfies the KL property with exponent  $1/2$  at  $\bar{x}$ , then the convergence rate is linear. In particular, this is true if  $\bar{x} \in \bar{\mathcal{C}} \cap \lambda^{-1}(\mathcal{C})$  and  $\bar{\mathcal{C}}$  and  $\lambda^{-1}(\mathcal{C})$  are transversal at  $\bar{x}$ .*

<sup>2</sup>That is,  $P(\mathcal{C}) = \mathcal{C}$  holds for any  $r \times r$  permutation matrix  $P$ .

<sup>3</sup>Because the eigenvalues are unique, the  $a_i$ 's must be the eigenvalues of  $s$ . Overall,  $\{e_1, \dots, e_r\}$  is a Jordan frame that “diagonalizes” both  $x$  and  $s$ , but only the eigenvalues of  $x$  are required to be in nonincreasing order.

*Proof.* If  $\bar{\mathcal{C}}$  and  $\lambda^{-1}(\mathcal{C})$  intersect transversally at  $\bar{x}$ , then  $f$  is a KL function with exponent  $1/2$  by [5, Theorem 5]<sup>4</sup>. In any case, the result follows from Proposition 4.3.  $\square$

It is interesting to note that the limiting case where  $\alpha_k$  is taken to be 1 in (4.6) corresponds to the *alternating projections algorithm*, where each iteration consists of successively projecting onto  $\bar{\mathcal{C}}$  and  $\lambda^{-1}(\mathcal{C})$ . Although the alternating projections algorithm falls outside of the scope of Theorem 4.2, it is known that it locally converges with a linear rate under a weaker assumption called *intrinsic transversality*, see [10, Theorem 6.1] and [29] for a related work. On the other hand, it is not clear if the global guarantee in Theorem 4.2 also holds for alternating projections under semialgebraic assumptions.

**The convex case** In the special case where  $\bar{\mathcal{C}}$  and  $\lambda^{-1}(\mathcal{C})$  are convex the situation is far more favourable. First of all, the iteration (4.6) always converges to a point in the intersection  $\bar{\mathcal{C}} \cap \lambda^{-1}(\mathcal{C})$ , provided that the problem is feasible. This is a consequence of classical convergence results of the proximal gradient method for convex functions, but it can also be obtained by observing that the iteration in (4.6) falls under the scope of several frameworks for analyzing convex feasibility problems and beyond. These frameworks are also able to take care of the limiting case  $\alpha_k = 1$ , e.g., [3, pg. 2], [4].

In the convex case, the interesting question is not *whether* (4.6) converges but *how fast does it converge*. Under convexity, it is known that the convergence rate of many algorithms is related to the underlying error bound that holds between sets. In particular, when a so-called *Lipschitz error bound* (also called *bounded linear regularity* in some papers, e.g., [3]) holds, several algorithms are known to converge at linear rate. What is somewhat less known is that under even weaker error bound assumptions *concrete*<sup>5</sup> convergence rates are still obtainable. For example, see [4, Section 3] for the case of Hölderian error bounds in the setting of fixed point problems or [26, Sections 4 and 5] for the case of general error bounds for convex feasibility problems.

Given that the convex case is easier to handle, a related question is to understand when does  $\lambda^{-1}(\mathcal{C})$  become a convex set. In the case of Jordan Algebras this is very well-understood. In particular, if  $\mathcal{C}$  is permutation invariant, then  $\lambda^{-1}(\mathcal{C})$  is convex if and only if  $\mathcal{C}$  is convex [20, Theorem 3], see also [2, Theorem 27]. In the general case of FTvN systems, the situation is more subtle and additional conditions seem necessary in order to obtain similar results, see [21].

## 5 Eigenvalue programming examples

In this section, we discuss two examples of the eigenvalue programming problem (**Eig-Prog**) and show numerical experiments for the projected gradient method described in Algorithm 1. In all examples, the problem data is semialgebraic so we are under the scope of Theorem 4.2. All experiments are implemented in Julia and conducted on a 3.0GHz Intel Xeon E5-1680v2 processor with 64GB of RAM.<sup>6</sup>

### 5.1 Inverse eigenvalue problems

Given a FTvN system  $(\mathcal{E}, \mathbb{R}^r, \lambda)$ ,  $\lambda^* \in \lambda(\mathcal{E})$  and  $a_i \in \mathcal{E}$  ( $i = 0, 1, \dots, d$ ), we consider the *inverse eigenvalue problem*:

$$\text{Find } c \in \mathbb{R}^d \text{ such that } \lambda(a_0 + c_1 a_1 + \dots + c_d a_d) = \lambda^*. \quad (5.1)$$

For the special case  $\mathcal{E} = \mathcal{S}^n$ , this problem has been extensively researched [14, 6, 8, 9, 7]. However, beyond  $\mathcal{S}^n$ , we are not aware of a systematic study of inverse eigenvalue problems.

<sup>4</sup>The correspondence between the notation in [5, Theorem 5] and our case is as follows.  $A$  is the identity map,  $D$  is  $\bar{\mathcal{C}}$  and  $\mathcal{C}$  is  $\delta_{\lambda^{-1}(\mathcal{C})}$ .

<sup>5</sup>By *concrete* we mean that the convergence rate is upper bounded in terms of a function of the iteration number  $k$  (e.g., see [4, Theorem 3.1]).

<sup>6</sup>The implementation is available in <https://github.com/ito-masaru/pgm-eig-prog>

The problem (5.1) is equivalent to the feasibility problem (4.4) with

$$\bar{C} := a_0 + \text{span}\{a_1, \dots, a_d\}, \quad C := \{\lambda^*\}.$$

In this case, the iterative scheme (4.6) reduces to

$$y_k = (1 - \alpha_k)x_k + \alpha_k \text{proj}_{\bar{C}}(x_k), \quad x_{k+1} \in U(y_k, \lambda^*), \quad (5.2)$$

since  $\text{proj}_{\lambda^{-1}(C)}(y) = U(y, \lambda^*)$  holds by Corollary 3.4.

### 5.1.1 Numerical experiment

Here we show some numerical results of the method (5.2) applied to the inverse eigenvalue problem (5.1) in the case

$$\mathcal{E} = \underbrace{\mathbb{R}^{n+1} \times \dots \times \mathbb{R}^{n+1}}_{m \text{ times}} \times \mathcal{S}^n, \quad (5.3)$$

where we regard  $\mathbb{R}^{n+1}$  and  $\mathcal{S}^n$  as the Jordan algebras in Examples 3.7 and 3.5, respectively. We examine both the block-wise eigenvalue maps  $\lambda$  and the ordered one  $\lambda^\downarrow$  described in (3.9) and (3.11), respectively. The goal of this experiment is to check the behavior of Algorithm 1 and see how far an initial point must be in order to ensure convergence to a solution point.

Given  $(n, m, d)$ , we examine 10 random instances of the problem (5.3) under the FTvN system (5.3) by setting  $\lambda^* := \lambda(a_0 + c_1 a_1 + \dots + c_d a_d) \in \mathcal{E}$ , where  $c_i \in \mathbb{R}$  and components of  $a_i \in \mathcal{E}$  are generated from the uniform distribution over  $[0, 1]$ . We run the method (5.2) with step-size  $\alpha_k = 0.99$  until it finds an iterate  $x_k$  satisfying the termination criterion

$$\text{dist}(x_k, \bar{C}) \leq 10^{-3}$$

(note that we have  $\lambda(x_k) = \lambda^*$  for  $k \geq 1$  by construction). If this condition fails to be satisfied after 10000 iterations, we restart the method and change the initial point. The initial point  $x_0$  is chosen as follows:

$$x_0 := x^* + 100\|x^*\|u/2^\ell,$$

where  $x^* := a_0 + c_1 a_1 + \dots + c_m a_m$  is an optimal solution,  $u \in \mathcal{E}$  is generated from the uniform distribution on the unit sphere of  $\mathcal{E}$  and  $\ell \geq 0$  is the number of restarts of the method. Note that  $\ell$  controls the relative distance from  $x_0$  to the optimal solution since  $\|x_0 - x^*\|/\|x^*\| = 100/2^\ell$  holds.

Tables 1 and 2 show numerical results for some problem instances under the eigenvalue maps  $\lambda$  and  $\lambda^\downarrow$ , respectively. For each  $(m, n)$ , the dimension  $d$  is chosen as  $d = \lfloor \dim(\mathcal{E})/\rho \rfloor$  for  $\rho \in \{0.2, 0.4, 0.6, 0.8\}$ . We see that the proposed method successfully converges to an optimal solution (i.e., a feasible solution to (5.1)) in many cases, while some instances require several restarts. Also employing the block-wise eigenvalue map  $\lambda$  is relatively better than the ordered one  $\lambda^\downarrow$  in average. Regarding the performance, we can see that the number of iterations is decreasing with respect to the parameter  $d$ . This is somewhat puzzling and at this point we can only speculate on the reasons for that. A possibility is that as  $d$  increases, the dimension of  $\bar{C}$  increases as well. Given the random nature of the instances,  $\bar{C} \cap \lambda^{-1}(\{\lambda^*\})$  might get be “better-conditioned” as  $d$  increases. For example, it could be case that the region corresponding to linear convergence as in Propositions 4.3 and 4.5 gets larger.

## 5.2 Vanishing Quadratic Constraints

In this section we take a look at the following problem. Given  $m$  convex quadratic inequalities and an integer  $\ell \in \{0, 1, \dots, m\}$ , we would like to find a common solution  $x \in \mathbb{R}^n$  that satisfies at least  $\ell$  with equality as follows.

$$\begin{cases} \|A_i x + b_i\|_2 \leq \langle c_i, x \rangle + d_i, & i = 1, \dots, m, \end{cases} \quad (5.4a)$$

$$\left\{ \begin{array}{l} \text{At least } \ell \text{ of these inequalities are tight.} \end{array} \right. \quad (5.4b)$$

$m$	$n$	$d$	Iterations				Restarts			
			mean	max	min	std	mean	max	min	std
0	10	11	1836.4	4136	574	1205.8	0.8	5	0	1.8
0	10	22	107.0	312	34	78.6	0.0	0	0	0.0
0	10	33	31.6	58	20	11.5	0.0	0	0	0.0
0	10	44	19.2	45	10	10.7	0.0	0	0	0.0
1	10	13	2255.9	4110	336	1135.0	3.1	7	0	3.4
1	10	26	82.4	152	47	37.6	0.0	0	0	0.0
1	10	39	47.1	63	29	10.8	0.0	0	0	0.0
1	10	52	32.0	56	15	12.2	0.0	0	0	0.0
5	10	22	3049.5	9666	564	2983.9	2.3	7	0	3.1
5	10	44	102.6	189	52	37.6	0.0	0	0	0.0
5	10	66	59.2	95	28	18.6	0.0	0	0	0.0
5	10	88	26.4	42	16	7.2	0.0	0	0	0.0

Table 1: Numerical results for problem (5.1) under the block-wise eigenvalue map  $\lambda$ . The column “Iterations” collects the summary of the number of iterations at the final run of the proposed method, showing the mean, the maximum, the minimum and the sample standard deviation. The summary of the number of restarts is given in the column “Restarts”.

$m$	$n$	$d$	Iterations				Restarts			
			mean	max	min	std	mean	max	min	std
0	10	11	1836.4	4136	574	1205.8	0.8	5	0	1.8
0	10	22	107.0	312	34	78.6	0.0	0	0	0.0
0	10	33	31.6	58	20	11.5	0.0	0	0	0.0
0	10	44	19.2	45	10	10.7	0.0	0	0	0.0
1	10	13	2255.9	4110	336	1135.0	3.1	7	0	3.4
1	10	26	82.4	152	47	37.6	0.0	0	0	0.0
1	10	39	56.8	160	29	37.4	0.0	0	0	0.0
1	10	52	39.8	64	24	12.6	0.0	0	0	0.0
5	10	22	2049.6	4340	494	1477.4	3.0	10	0	3.9
5	10	44	1937.7	8993	74	2856.9	1.5	6	0	2.5
5	10	66	78.8	127	46	30.5	0.0	0	0	0.0
5	10	88	24.6	36	15	6.3	0.0	0	0	0.0

Table 2: Numerical results for problem (5.1) under the ordered eigenvalue map  $\lambda^\downarrow$

Here,  $A_i \in \mathbb{R}^{n_i \times n}$ ,  $b_i \in \mathbb{R}^{n_i}$ ,  $c_i \in \mathbb{R}^n$ ,  $d_i \in \mathbb{R}$  and  $\ell \in \{0, 1, \dots, m\}$  are given. We refer to a constraint in format of the (5.4a) and (5.4b) as *quadratic inequality with a vanishing constraint* or a *vanishing quadratic constraint* for short. In this subsection, we will discuss how to reformulate a vanishing quadratic constraint in the form of a feasibility problem as in (4.4).

Let  $\mathcal{E} := \mathbb{R}^{n_1+1} \times \dots \times \mathbb{R}^{n_m+1}$  and  $\mathcal{K} := \mathcal{L}_2^{n_1+1} \times \dots \times \mathcal{L}_2^{n_m+1}$  be a direct product of second-order cones. We consider that each  $\mathbb{R}^{n_i+1}$  is a Euclidean Jordan algebra as in Example 3.7 and the corresponding cone is  $\mathcal{L}_2^{n_i+1}$ . With that,  $(\mathcal{E}, \mathbb{R}^{2m}, \lambda^\downarrow)$  is a FTvN system where  $\lambda^\downarrow$  is as in (3.11).

Next, we define

$$b := (b_1, d_1, \dots, b_m, d_m) \in \mathcal{E}, \quad A : \mathbb{R}^n \rightarrow \mathcal{E}, \quad Ax := (A_1x, \langle c_1, x \rangle, \dots, A_mx, \langle c_m, x \rangle).$$

We have the following result.

**Theorem 5.1** (Vanishing quadratic constraints as a rank constraint). *Suppose that  $x \in \mathbb{R}^n$  satisfies (5.4a) and (5.4b), then*

$$Ax + b \in \mathcal{K} \quad \text{and} \quad \text{rank}(Ax + b) \leq 2m - \ell. \quad (5.5)$$

*Conversely, if  $x \in \mathbb{R}^n$  satisfies (5.5) and  $(A_i x + b_i, \langle c_i, x \rangle + d_i) \neq 0$  holds for all  $i$ , then  $x$  satisfies (5.4a) and (5.4b).*

*Proof.* First, suppose that  $x \in \mathbb{R}^n$  satisfies (5.4a) and (5.4b). From, (5.4a) we obtain  $Ax + b \in \mathcal{K}$ . Next, let  $z := Ax + b$ , so that  $z = (z_1, t_1, \dots, z_m, t_m)$  can be divided in  $m$  blocks so that

$$(z_i, t_i) = (A_i x + b_i, \langle c_i, x \rangle + d_i)$$

holds for  $i \in \{1, \dots, m\}$ . We have that  $\|A_i x + b_i\|_2 = \langle c_i, x \rangle + d_i$  if and only if  $(z_i, t_i)$  belongs to the boundary of  $\mathcal{L}_2^{n_i+1}$ .

Equivalently, the block  $(z_i, t_i) \in \mathcal{L}_2^{n_i+1}$  belongs to the boundary of  $\mathcal{L}_2^{n_i+1}$  if and only if the smallest eigenvalue (i.e.,  $\lambda_-$  in (3.7)) of  $(z_i, t_i)$  vanishes. Therefore, the rank of  $(z_i, t_i)$  in the algebra  $\mathbb{R}^{n_i+1}$  must be 0 or 1. Since the rank of  $z$  is the sum of ranks of the individual blocks, (5.4b) implies that  $\text{rank}(z) \leq \ell + 2(m - \ell) = 2m - \ell$  holds.

Conversely, suppose that  $x$  satisfies (5.5). Then, in particular,  $Ax + b \in \mathcal{K}$  holds, which, by definition implies that (5.4a) is satisfied. Under the assumption that  $Ax + b \in \mathcal{K}$  and  $(A_i x + b_i, \langle c_i, x \rangle + d_i)$  is never zero, we have that the rank of  $(A_i x + b_i, \langle c_i, x \rangle + d_i)$  is either 1 or 2 and the former happens if and only if  $\|A_i x + b_i\|_2 = \langle c_i, x \rangle + d_i$ . Therefore, if  $\text{rank}(Ax + b) \leq 2m - \ell$ , then at least  $\ell$  of the inequalities are tight, since otherwise we would have  $\text{rank}(Ax + b) \geq 2(m - (\ell + 1)) + (\ell + 1) > 2m - \ell$ .  $\square$

If  $\ell \geq 1$ , we have equivalently that  $x$  satisfies (5.5) if and only if

$$\lambda_{2m}^\downarrow(Ax + b) = \dots = \lambda_{2m-\ell+1}^\downarrow(Ax + b) = 0.$$

The final step to reformulate the problem in the format described in Section 4.1 is to perform the variable transformation  $y = Ax + b$ . This leads to the following feasibility problem

$$\text{find } y \in (\text{range}(A) + b) \cap (\mathcal{K} \cap \{y \mid \text{rank}(y) \leq 2m - \ell\}). \quad (5.6)$$

Finally, if we let  $\bar{\mathcal{C}} := \text{range}(A) + b$ ,  $\mathcal{C} := \{u \in \mathbb{R}_+^{2m} \mid u_{2m} = \dots = u_{2m-\ell+1} = 0\}$ , we get the following equivalent problem

$$\text{find } y \in \bar{\mathcal{C}} \quad \text{and} \quad \lambda^\downarrow(y) \in \mathcal{C}, \quad (5.7)$$

which can be solved as described in Section 4.1.

We remark that the projection of  $v \in \mathbb{R}_+^\uparrow$  onto  $\mathcal{C}$  is easily computable as

$$\text{proj}_{\mathcal{C}}(v) = (\max(v_1, 0), \dots, \max(v_{2m-\ell}, 0), 0, \dots, 0), \quad \forall v \in \mathbb{R}_+^\uparrow.$$

Therefore, using Corollary 3.4, the iterative scheme (4.6) reduces to

$$z_k = (1 - \alpha_k)y_k + \alpha_k \text{proj}_{\bar{\mathcal{C}}}(y_k), \quad y_{k+1} \in U(z_k, \text{proj}_{\mathcal{C}}(\lambda^\downarrow(z_k))).$$

By its turn, an element of  $U(z_k, \text{proj}_{\mathcal{C}}(\lambda^\downarrow(z_k)))$  can be obtained explicitly through (3.12) and the spectral decomposition in the algebra of second-order cones as described in Example 3.7.

A final observation is that in order to obtain a “if and only if” correspondence between (5.6) and the pair (5.4a) and (5.4b), Theorem 5.1 requires the assumption that  $(A_i x + b_i, \langle c_i, x \rangle + d_i) \neq 0$  holds for all  $i$  whenever  $x$  satisfies (5.4a) and (5.4b). An important case where this assumption is satisfied is when all the  $c_i$ ’s are zero and the  $d_i$ ’s are nonzero as in the next subsection.

### 5.2.1 Intersection of ellipsoids and a numerical experiment

As a special case of (5.4), suppose we are given  $m$  ellipsoids and we would like to find an intersection point that lies in the boundary of at least  $\ell$  of them. Formally, given positive definite matrices  $Q_i \in \mathcal{S}^n$  and  $p_i \in \mathbb{R}^n$  for  $i = 1, \dots, m$ , our problem is to find  $x \in \mathbb{R}^n$  satisfying

$$\begin{cases} (x - p_i)^T Q_i (x - p_i) \leq 1 & i = 1, \dots, m, \\ \text{At least } \ell \text{ of these inequalities are tight.} \end{cases} \quad (5.8a)$$

$$(5.8b)$$

The corresponding  $A$  and  $b$  are given by

$$b := (-Q_1^{1/2} p_1, 1, \dots, -Q_m^{1/2} p_m, 1), \quad Ax := (Q_1^{1/2} x, 0, \dots, Q_m^{1/2} x, 0). \quad (5.9)$$

We can rewrite (5.8) as

$$\begin{cases} Ax + b \in \mathcal{K} := \underbrace{\mathcal{L}_2^{n+1} \times \dots \times \mathcal{L}_2^{n+1}}_{m \text{ times}} \\ \text{rank}(Ax + b) \leq 2m - \ell \end{cases}$$

which is equivalent to

$$\text{find } y \in \bar{\mathcal{C}} \text{ and } \lambda^\downarrow(y) \in \mathcal{C}, \quad (5.10)$$

where  $\bar{\mathcal{C}} := \text{range}(A) + b$  and  $\mathcal{C} := \{u \in \mathbb{R}_+^{2m} \mid u_{2m} = \dots = u_{2m-\ell+1} = 0\}$ . We observe that in this case, the “ $c_i$ ’s” and “ $d_i$ ’s” in Theorem 5.1 are all 0 and 1, respectively. So, indeed, (5.10) and the pair (5.8a), (5.8b) are equivalent.

**Recovering  $x$  from  $y$**  Here we remark that a solution to (5.8) can be easily recovered from a solution to (5.10). Let  $y = (\bar{y}_1, y_{10}, \dots, \bar{y}_m, y_{m0}) \in \bar{\mathcal{C}}$ . Let us check that there is a unique  $x$  satisfying  $y = Ax + b$ . By (5.9), the relation  $y = Ax + b$  is equivalent to

$$\bar{y}_i = Q_i^{1/2} (x - p_i), \quad y_{i0} = 1, \quad i = 1, \dots, m.$$

The former equation is equivalent to

$$x = Q_1^{-1/2} \bar{y}_1 + p_1 = \dots = Q_m^{-1/2} \bar{y}_m + p_m.$$

Numerically, we can also average the different expressions to  $x$  which leads to the following formula.

$$x = \frac{1}{m} \sum_{i=1}^m (Q_i^{-1/2} \bar{y}_i + p_i). \quad (5.11)$$

**Numerical results** Here, we show numerical results of applying Algorithm 1 to (4.5) in order to solve the problem (5.10) in the case  $(n, m, \ell) = (2, 3, 1), (2, 3, 2)$ . We generated randomly three ellipsoids centered at the origin as illustrated in Figure 1. We use constant step-size  $\alpha_k = 0.99$ . Let  $\{y^k\}$  be the generated sequence and  $x^k$  be the approximate solution recovered by  $y^k$  using the formula (5.11). We terminate the algorithm if

$$\text{dist}(y^k, \bar{\mathcal{C}}) \leq 10^{-3}$$

(Note that  $\text{dist}(\lambda^\downarrow(y^k), \mathcal{C}) = 0$  holds for  $k \geq 1$  by construction). In Figure 1 we demonstrate the numerical results.

- The case  $\ell = 1$  in (5.8) corresponds to the problem of finding a boundary point on the intersection of given  $m$  ellipsoids. Figure 1 shows the behavior of  $\{x^k\}$  from various initial points. One can see that the trajectory  $\{x^k\}$  successfully converges to a point on the boundary.
- The case  $\ell = 2$  in (5.8) corresponds to finding a “degenerate” boundary point of the intersection of given  $m$  ellipsoids. Figure 1 illustrates that  $\{x^k\}$  successfully converges to a desired point except the one starting from the origin. We remark that the origin yields a stationary point for the corresponding problem (4.5) in this case.

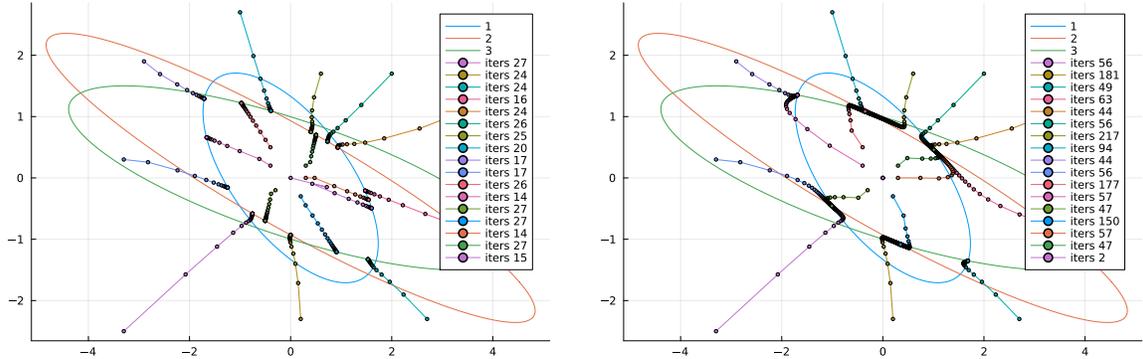


Figure 1: Illustration of the trajectory  $\{x^k\}$  from various initial points for the problem (5.8) with  $n = 2$  and  $m = 3$ . The left and right figures correspond to the cases for  $\ell = 1$  and  $\ell = 2$ , respectively. Each line except the ellipsoid is a trajectory of the proposed method where ‘iters’ shows the number of iterations.

## 6 Concluding remarks

In this work, using the FTvN system framework developed by Gowda [17], we discussed eigenvalue programs beyond the usual matrix setting. We also analyzed and implemented a simple projected gradient algorithm, see Algorithm 1. Finally, we also showed some applications such as general inverse eigenvalue problems. In the particular case where the FTvN system comes from the Euclidean Jordan algebra of second order cones, we showed that it is possible to express the so-called *vanishing quadratic constraints*, which is useful to obtain boundary points of the intersection of ellipsoids with certain degeneracy properties.

The topics we discussed in this paper are still in their infancy and much remains to be done from both modelling and algorithmic perspectives. One particular direction where many fruitful results may yet be found is related to rank problems involving second-order cones. As problems over second order cones are typically more scalable than matrix problems, fitting a particular modelling application that requires some rank-like constraint into this setting may lead to better performance when compared with a model that uses matrices.

While the writing of this paper was in the final stages, Garner, Lerman and Zhang [16] released a preprint that is very close to the spirit of what we propose in this paper. There too they consider problems with general constraints on eigenvalues and analyze methods based on the idea of solving subproblems on the space of eigenvalues. That said, one important difference between the work [16] and our approach is that our discussion uses the FTvN framework while [16] only consider problems over matrices. In particular, the applications involving the second-order cone discussed here seem to be out of the scope of [16].

## Acknowledgements

The first author was supported partly by the JSPS Grant-in-Aid for Early-Career Scientists 21K17711. The second author was supported partly by the JSPS Grant-in-Aid for Early-Career Scientists 23K16844.

## References

- [1] H. Attouch, J. Bolte, and B. F. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods. *Mathematical Programming*, 137(1):91–129, Feb 2013.
- [2] M. Baes. Convexity and differentiability properties of spectral functions and spectral mappings on euclidean jordan algebras. *Linear Algebra and its Applications*, 422:664–700, 2007.
- [3] H. H. Bauschke and J. M. Borwein. On projection algorithms for solving convex feasibility problems. *SIAM Review*, 38(3):367–426, 1996.
- [4] J. M. Borwein, G. Li, and M. K. Tam. Convergence rate analysis for averaged fixed point iterations in common fixed point problems. *SIAM Journal on Optimization*, 27(1):1–33, 2017.
- [5] C. Chen, T. K. Pong, L. Tan, and L. Zeng. A difference-of-convex approach for split feasibility with applications to matrix factorizations and outlier detection. *Journal of Global Optimization*, 78(1):107–136, Mar. 2020.
- [6] X. Chen and M. T. Chu. On the least squares solution of inverse eigenvalue problems. *SIAM Journal on Numerical Analysis*, 33(6):2417–2430, 1996.
- [7] M. Chu and G. Golub. *Inverse Eigenvalue Problems, Theory, Algorithms, and Applications*. Oxford University Press, Oxford, 2005.
- [8] M. T. Chu. Inverse eigenvalue problems. *SIAM Review*, 40(1):1–39, 1998.
- [9] M. T. Chu and G. H. Golub. Structured inverse eigenvalue problems. *Acta Numerica*, 11:1–71, 2002.
- [10] D. Drusvyatskiy, A. D. Ioffe, and A. S. Lewis. Transversality and alternating projections for nonconvex sets. *Foundations of Computational Mathematics*, 15(6):1637–1651, Aug. 2015.
- [11] J. Faraut and A. Korányi. *Analysis on symmetric cones*. Oxford mathematical monographs. Clarendon Press, Oxford, 1994.
- [12] L. Faybusovich. Several Jordan-algebraic aspects of optimization. *Optimization*, 57(3):379–393, 2008.
- [13] P. Frankel, G. Garrigos, and J. Peypouquet. Splitting methods with variable metric for kurdyka–łojasiewicz functions and general convergence rates. *Journal of Optimization Theory and Applications*, 165(3):874–900, Jun 2015.
- [14] S. Friedland, J. Nocedal, and M. L. Overton. The formulation and analysis of numerical methods for inverse eigenvalue problems. *SIAM Journal on Numerical Analysis*, 24(3):634–667, 1987.
- [15] M. Fukushima, Z.-Q. Luo, and P. Tseng. Smoothing functions for second-order-cone complementarity problems. *SIAM Journal on Optimization*, 12(2):436–460, 2002.
- [16] C. Garner, G. Lerman, and S. Zhang. Spectrally constrained optimization. *ArXiv e-prints*, 2023. [arXiv:2307.04069](https://arxiv.org/abs/2307.04069).

- [17] M. S. Gowda. Optimizing certain combinations of spectral and linear/distance functions over spectral sets. *ArXiv e-prints*, 2019. [arXiv:1902.06640](https://arxiv.org/abs/1902.06640).
- [18] M. S. Gowda and J. Jeong. Commutativity, majorization, and reduction in Fan–Theobald–von Neumann systems. *Results in Mathematics*, 78(3), 2023. Article Number: 72.
- [19] M.-F. R. Jacek Bochnak, Michel Coste. *Real Algebraic Geometry*. Springer-Verlag, Berlin, Heidelberg, 1998.
- [20] J. Jeong and M. S. Gowda. Spectral sets and functions on Euclidean Jordan algebras. *Linear Algebra and its Applications*, 518:31 – 56, 2017.
- [21] J. Jeong and M. S. Gowda. Transfer principles, Fenchel conjugate and subdifferential formulas in Fan-Theobald-von Neumann systems. *ArXiv e-prints*, 2023. [arXiv:2307.08478](https://arxiv.org/abs/2307.08478).
- [22] A. S. Lewis. Convex analysis on the Hermitian matrices. *SIAM Journal on Optimization*, 6(1):164–177, 1996.
- [23] A. S. Lewis and J. Malick. Alternating projections on manifolds. *Mathematics of Operations Research*, 33(1):216–234, Feb. 2008.
- [24] A. S. Lewis and H. S. Sendov. Nonsmooth analysis of singular values. part I: Theory. *Set-Valued Analysis*, 13:213–241, 2005.
- [25] G. Li and T. K. Pong. Calculus of the exponent of Kurdyka-Lojasiewicz inequality and its applications to linear convergence of first-order methods. *Foundations of Computational Mathematics*, 18:1199 – 1232, 2018.
- [26] T. Liu and B. F. Lourenço. Convergence analysis under consistent error bounds. *Foundations of Computational Mathematics*, Oct. 2022.
- [27] B. F. Lourenço and A. Takeda. Generalized subdifferentials of spectral functions over Euclidean Jordan Algebras. *SIAM Journal on Optimization*, 30(4):3387–3414, 2020.
- [28] B. Mordukhovich. *Variational Analysis and Applications*. Springer Monographs in Mathematics. Springer International Publishing, 2018.
- [29] D. Noll and A. Rondepierre. On local convergence of the method of alternating projections. *Foundations of Computational Mathematics*, 16(2):425–455, Feb. 2015.
- [30] R. T. Rockafellar and R. J.-B. Wets. *Variational Analysis*. Springer Berlin Heidelberg, 2009.
- [31] J. F. Sturm. Similarity and other spectral relations for symmetric cones. *Linear Algebra and Its Applications*, 312(1-3):135–154, 2000.