

# Computing the generators of the truncated real radical ideal by moment matrices and SDP facial reduction

Fei Wang<sup>a</sup>, Greg Reid<sup>a</sup>, Henry Wolkowicz<sup>b</sup>

<sup>a</sup>*Department of Applied Mathematics, Western University*

<sup>b</sup>*Department of Combinatorics & Optimization, University of Waterloo*

---

## Abstract

Recent breakthroughs have been made in the use of semidefinite programming and its application to real polynomial solving. For example, the real radical of a zero dimensional ideal, can be determined by such approaches as shown by Lasserre and collaborators. Some progress has been made on the determination of the real radical in positive dimension by Ma, Wang and Zhi. Such work involves the determination of maximal rank semidefinite moment matrices. Existing methods are computationally expensive and have poorer accuracy on larger examples.

This paper is motivated by problems in the numerical computation of the real radical ideal in the general positive case.

In this paper we give a method to compute the generators of the real radical for any given degree  $d$ . We combine the use of moment matrices and techniques from SDP optimization: facial reduction first developed by Borwein and Wolkowicz. In use of the semidefinite moment matrices to compute the real radical, the maximum rank property is very key, and with facial reduction, it can be guaranteed with very high accuracy. Our algorithm can be used to test the real radical membership of a given polynomial. In a special situation, we can determine the real radical ideal in the positive dimensional case.

*Keywords:* Real radical ideal, facial reduction, moment matrix, Douglas-Rachford.

---

## 1. Introduction

The breakthrough work of Lasserre and collaborators [26, 42] shows that the *real radical ideal*,  $RRI$ , of a real polynomial system with finitely many solutions can be determined by computing the kernel of so-called moment matrices arising from a *semidefinite programming (SDP)* feasibility problem. This RRI is generated by a system of real polynomials having only real roots that are free of multiplicities. The number of such real roots may be considerably less than the number of complex roots (see the paper [35] for examples and references). Global numerical solvers, such as homotopy continuation solvers typically compute all real roots by first computing all complex (including real) roots. And if the roots have multiplicity, then elaborate strategies are needed to avoid difficulties that arise as the paths from the homotopy solvers approach these singular roots [41]. A conjectured extension of such methods to positive dimensional polynomial systems has been given recently by Ma, Wang and Zhi [32, 31].

---

*Email addresses:* [fwang96@uwo.ca](mailto:fwang96@uwo.ca) (Fei Wang), [red@uwo.ca](mailto:red@uwo.ca) (Greg Reid), [hwolkowicz@uwaterloo.ca](mailto:hwolkowicz@uwaterloo.ca) (Henry Wolkowicz)

Our approach also builds on the method of moment matrices. A key step is to solve the problem of the following type for  $X$

$$\mathcal{A}(X) = b, \quad X \in \mathcal{S}_+^k, X \text{ is maximum rank}, \quad (1.1)$$

where  $\mathcal{S}_+^k$  denotes the convex cone of  $k \times k$  real symmetric positive semidefinite matrices, and  $\mathcal{A} : \mathcal{S}_+^k \rightarrow \mathbb{R}^l$  is a linear transformation which enforces the moment matrix structure for  $X$ .

The standard regularity assumption for (1.1) is the *Slater constraint qualification* or strict feasibility assumption:

$$\text{there exists } X \text{ with } \mathcal{A}X = b, \quad X \in \text{int } \mathcal{S}_+^k. \quad (1.2)$$

We let  $X \succeq 0, \succ 0$  denote  $X \in \mathcal{S}_+^k, \in \text{int } \mathcal{S}_+^k$ , respectively. It is well known that the Slater condition for SDP holds generically, e.g., [19]. Surprisingly, many SDP problems arising from particular applications, and in particular our polynomial system applications, are marginally infeasible, i.e., fail to satisfy strict feasibility. This means that the feasible set lies within the boundary of the cone, which creates difficulties with numerical algorithms such as interior point solvers and the maximum rank can not be computed accurately. To help regularize such SDP problems, facial reduction was introduced in 1982 by Borwein and Wolkowicz [8, 9]. However it was only much later that the power of facial reduction was exhibited in many applications, e.g., [51, 48, 1]. Developing algorithmic implementations of facial reduction that work for large classes of SDP problems and the connections with perturbation and convergence analysis has recently been achieved in e.g., [24, 16, 12, 17].

In this paper, we use facial reduction approach to effectively reduce the size of the SDP problem associated with the input polynomial system so that it is strictly feasible and then solve the reduced problem using the Douglas-Rachford method. We then use the *geometric involutive basis* to check if the kernel of the moment matrix is a truncated ideal (ideal-like). This leads to a method to compute the generators of real radicals up to any given degree  $d$ . Suppose given a subset  $S$  of the real solution set of the input polynomial system. The vanishing ideal of  $S$  denoted by  $I(S)$  contains the real radical. By our approach, we can determine if  $I(S)$  is contained in the real radical. If it is so, then  $I(S)$  is the real radical. If not, then  $S$  is not complete and a large  $S$  is needed. This leads to a method to numerically determine the *completeness* of the real solution subset  $S$ . See [10] for details of this approach. We compare the performance of our techniques with the popular SDP solver SeDuMi(CVX) which uses an interior point method. On our illustrative examples, our approach has better accuracy, and the maximum rank condition can be guaranteed without misleading small eigenvalues.

## 2. Real radical and moment matrices

### 2.1. real radical

Suppose that  $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$  and consider a system of  $m$  multivariate polynomials  $P = \{p_1(x), p_2(x), \dots, p_m(x)\} \subseteq \mathbb{R}[x_1, x_2, \dots, x_n]$  with real coefficients. Its solution set or variety is

$$V_{\mathbb{R}}(p_1, \dots, p_m) = \{x \in \mathbb{R}^n : p_j(x) = 0, 1 \leq j \leq m\}. \quad (2.1)$$

The ideal generated by  $P = \{p_1, \dots, p_m\} \subseteq \mathbb{R}$  is:

$$\langle P \rangle_{\mathbb{R}} = \langle p_1, \dots, p_m \rangle_{\mathbb{R}} = \{f_1 p_1 + \dots + f_m p_m : f_j \in \mathbb{R}[x], 1 \leq j \leq m\}. \quad (2.2)$$

and its associated radical ideal over  $\mathbb{R}$  is defined as

$$\sqrt[\mathbb{R}]{\langle P \rangle} = \{f \in \mathbb{R}[x] : f^{2t} + \sum_{j=1}^s q_j^2 \in \langle P \rangle \text{ for some } q_j \in \mathbb{R}[x], t \in \mathbb{N} \setminus \{0\}\}. \quad (2.3)$$

A fundamental result [3] is:

**Theorem 2.1.** [Real Nullstellensatz] For any ideal  $I \subseteq \mathbb{R}[x]$  we have  $\sqrt[\mathbb{R}]{I} = I(V_{\mathbb{R}}(I))$ .

Consequently

$$\sqrt[\mathbb{R}]{\langle P \rangle} = \{f(x) \in \mathbb{R}[x] : f(x) = 0 \text{ for all } x \in V_{\mathbb{R}}(P)\}. \quad (2.4)$$

**Remark 2.1.** An ideal  $I \subseteq \mathbb{R}[x]$  is real radical if and only if for all  $p_1, \dots, p_m \in \mathbb{R}[x]$ :

$$p_1^2 + \dots + p_m^2 \in I \implies p_1, \dots, p_m \in I. \quad (2.5)$$

For these and many other results see [3] and the references cited therein.

## 2.2. Moment matrix

**Definition 2.1** (Moment Matrix [28]). Given a linear form  $\lambda \in \mathbb{R}[x]^*$ ,  $x = (x_1 \cdots x_n)$  which maps a polynomial to a real number. A symmetric matrix

$$M(\lambda) = (\lambda(x^\alpha x^\beta))_{\alpha, \beta \in \mathbb{N}^n} \quad (2.6)$$

is called a moment matrix of  $\lambda$  where  $\mathbb{N} = \{0, 1, 2, \dots\}$ .

Similarly, we define the truncated moment matrix.

**Definition 2.2** (Truncated Moment Matrix [28]). Given a linear form  $\lambda_d \in (\mathbb{R}[x]_{2d})^*$ , the truncated moment matrix of  $\lambda_d$  is defined to be

$$M(\lambda_d) = (\lambda_d(x^\alpha x^\beta))_{\alpha, \beta \in \mathbb{N}_d^n} \quad (2.7)$$

where  $\mathbb{N}_d^n = \{\gamma \in \mathbb{N}^n : |\gamma| = \sum_{j=1}^n \gamma_j \leq d\}$ .

**Example 2.1.** Suppose  $\lambda_1 \in \mathbb{R}[x, y]_{2d}^*$  for  $d = 1$ . Then

$$M(\lambda_1) = \begin{bmatrix} u_{00} & u_{10} & u_{01} \\ u_{10} & u_{20} & u_{11} \\ u_{01} & u_{11} & u_{02} \end{bmatrix}. \quad (2.8)$$

Without loss, we assume  $u_{00} = 1$  throughout this paper.

The kernel of a positive semidefinite truncated moment matrix has the following ‘‘real radical-like’’ property:

**Lemma 2.1.** [28] Assume  $M(\lambda_d) \succeq 0$  and let  $p, q_j \in \mathbb{R}[x]$ ,  $f := p^{2m} + \sum_j q_j^2$  with  $m \in \mathbb{N}$ ,  $m \geq 1$ . Then,  $f \in \ker M(\lambda_d) \implies p \in \ker M(\lambda_d)$ .

We also have the following theorems which are known:

**Theorem 2.2.** [27, Lemma 3.1] Suppose that the ideal  $I = \langle f_1, \dots, f_m \rangle_{\mathbb{R}}$  with  $\max_i(\deg(f_i)) = d$  and let  $B$  be the coefficient matrix of  $\{f_1, \dots, f_m\} \subseteq \mathbb{R}[x]$ . Let  $M(\lambda_d)$  be a truncated moment matrix such that  $B \cdot M(\lambda_d) = 0$  and  $M(\lambda_d) \succeq 0$ . If the rank of  $M(\lambda_d)$  is maximum then

$$\mathbb{P} \ker M(\lambda_d) \subseteq \sqrt[\mathbb{R}]{I}. \quad (2.9)$$

**Theorem 2.3.** (Flat extension theorem [14]) Assume  $M(\lambda_d) \succeq 0$ . The following statements are equivalent:

- (i) There exists an extension  $M(\lambda_{d+1}) \succeq 0$  and  $\text{rank } M(\lambda_d) = \text{rank } M(\lambda_{d+1})$ .
- (ii)  $\ker M(\lambda_d)$  is ideal-like.

**Lemma 2.2.** [27, Theorem 3.4, Corollary 3.8] Assume  $M(\lambda) \succeq 0$  and  $\text{rank } M(\lambda_d) = \text{rank } M(\lambda_{d-1}) = r$ . Then  $J = \langle \mathbb{P} \ker M(\lambda_d) \rangle_{\mathbb{R}}$  is real radical and zero-dimensional. One can extend  $\lambda_d$  to  $\lambda = \sum_{i=1}^r \alpha_i \lambda_{v_i} \in \mathbb{R}[x]^*$  where  $\alpha_i > 0$  and  $\{v_1, \dots, v_r\} = V_{\mathbb{R}}(\mathbb{P} \ker M(\lambda_d))$ . Furthermore  $\lambda = \lambda_d$  when  $\lambda$  is restricted to  $\mathbb{R}[x]_{2d}$ .

### 3. Computation of generators of the real radical up to a given degree

Based on the maximum rank moment matrix, the geometric involutive form [35], the results of Curto and Fialkow [14] and Lasserre et al. [27] we give an algorithm for computing the real radical up to a given degree  $d$ .

Throughout this section we consider a system of multivariate polynomials  $\{f_1, \dots, f_m\} \subseteq \mathbb{R}[x_1, x_2, \dots, x_n]$  of degree  $d = \max_i(\deg(f_i))$ . The associated real ideal is denoted

$$I := \langle f_1, f_2, \dots, f_m \rangle_{\mathbb{R}} \quad (3.1)$$

and its associated real radical ideal is denoted by  $\sqrt[\mathbb{R}]{I}$ .

In particular we solve the following problem:

**Problem 3.1.** *Given a system of polynomials  $\{f_1, \dots, f_m\} \subseteq \mathbb{R}[x_1, x_2, \dots, x_n]$  with associated ideal  $I$  and an integer  $d$  we give an algorithm to compute:*

$$\left(\sqrt[\mathbb{R}]{I}\right)_{(\leq d)} := \{f \in \sqrt[\mathbb{R}]{I} : \deg(f) \leq d\}. \quad (3.2)$$

We will represent  $\left(\sqrt[\mathbb{R}]{I}\right)_{(\leq d)}$  by polynomials corresponding to vectors in  $\ker M(\lambda_d)$  where  $M(\lambda_d)$  is the truncated moment matrix to degree  $d$  as defined in Definition 2.2.

In order to obtain our main result we will require that  $\ker M(\lambda_d)$  is *ideal-like* as defined by Curto and Fialkow [14]. We note that there is a bijective correspondence between vectors  $v \in \ker M(\lambda_d)$  and polynomials given by  $v \mapsto \mathbb{P}(v) = v^T(x^\alpha)_{\alpha \in \mathbb{N}^n}$  where  $(x^\alpha)_{\alpha \in \mathbb{N}^n}$  is the vector of all monomials of degree  $\leq d$  ordered in the same way as the rows of the moment matrix. Conversely each polynomial  $g$  used to form the coefficient matrix  $B$ , is mapped to a vector  $\text{vec}(g)$  in  $\ker M(\lambda_d)$ .

**Definition 3.1** (Ideal-Like truncated moment matrix [14]). *The kernel of a truncated moment matrix  $M(\lambda_d)$  is ideal-like of degree  $d$  if the following two conditions are satisfied:*

- If  $f_1, f_2 \in \mathbb{P} \ker M(\lambda_d)$  then  $f_1 + f_2 \in \mathbb{P} \ker M(\lambda_d)$ .
- If  $f \in \mathbb{P} \ker M(\lambda_d)$  and  $g \in \mathbb{R}[x]$  has  $\deg(fg) \leq d$ , then  $fg \in \mathbb{P} \ker M(\lambda_d)$ .

The ideal-like property is denoted as *RG* in [14].

Our main result is:

**Theorem 3.1.** *Suppose that  $I = \langle f_1, \dots, f_m \rangle_{\mathbb{R}}$  with  $\max_i(\deg(f_i)) = d$  and let  $B$  be the coefficient matrix of  $\{f_1, \dots, f_m\} \subseteq \mathbb{R}[x]$ . Let  $M(\lambda_d)$  be a truncated moment matrix such that  $B \cdot M(\lambda_d) = 0$  and  $M(\lambda_d) \succeq 0$ . If the rank of  $M(\lambda_d)$  is maximum and  $\ker M(\lambda_d)$  is ideal-like then*

$$\mathbb{P} \ker M(\lambda_d) = \left(\sqrt[\mathbb{R}]{I}\right)_{(\leq d)}. \quad (3.3)$$

To prove the above theorem, we will need Theorem 2.2, Theorem 2.3 and Lemma 2.2.

We now prove Theorem 3.1.

PROOF. Suppose  $\ker M(\lambda_d)$  is *ideal-like*,  $M(\lambda_d) \succeq 0$  and  $M(\lambda_d)$  has maximum rank together with the other assumptions in Theorem 3.1.

Our goal is to show that

$$\mathbb{P} \ker M(\lambda_d) = \left(\sqrt[\mathbb{R}]{I}\right)_{(\leq d)}.$$

First by Theorem 2.2, the following direction is obvious:

$$\mathbb{P} \ker M(\lambda_d) \subseteq \left( \sqrt[\mathbb{R}]{I} \right)_{(\leq d)}.$$

So we only need to show

$$\mathbb{P} \ker M(\lambda_d) \supseteq \left( \sqrt[\mathbb{R}]{I} \right)_{(\leq d)}.$$

By Theorems 2.3 and 2.2,  $\lambda_d$  can be extended to  $\lambda_{d+1}$  such that  $J = \langle \mathbb{P} \ker M(\lambda_{d+1}) \rangle_{\mathbb{R}}$  is real radical and zero-dimensional. Since  $I \subseteq J$ , we have  $\sqrt[\mathbb{R}]{I} \subseteq J$ . By Theorem 2.2, one can extend  $\lambda_d$  to  $\lambda = \sum_{i=1}^r \alpha_i \lambda_{v_i} \in \mathbb{R}[x]^*$  where  $\alpha_i > 0$  and  $\{v_1, \dots, v_r\} = V_{\mathbb{R}}(\mathbb{P} \ker M(\lambda_{d+1})) = V_{\mathbb{R}}(J)$  and  $\lambda_{v_i}$  is an evaluation mapping at  $v_i$  such that  $\lambda_{v_i}(f) = f(v_i)$ . Thus  $\lambda_d = \sum_{i=1}^r \alpha_i \lambda_{v_i}^{(d)}$  where  $\lambda_{v_i}^{(d)}$  is the truncated linear form of  $\lambda_{v_i}$ . Since  $\sqrt[\mathbb{R}]{I} \subseteq J$ , we have  $\{v_1, \dots, v_r\} \subseteq V_{\mathbb{R}}(\sqrt[\mathbb{R}]{I})$ .

Now we can prove the other inclusion:

$$\mathbb{P} \ker M(\lambda_d) \supseteq \left( \sqrt[\mathbb{R}]{I} \right)_{(\leq d)}.$$

So we let  $g \in \left( \sqrt[\mathbb{R}]{I} \right)_{(\leq d)}$  and we want to show that  $g \in \mathbb{P} \ker M(\lambda_d)$ , that is to show that  $\text{vec}(g)^T M(\lambda_d) = 0$ .

Since  $g \in \sqrt[\mathbb{R}]{I}$  with  $\deg(g) \leq d$ , we have  $g(v_i) = 0, i = 1, \dots, r$ . Therefore, we have  $g^2(v_i) = \text{vec}(g)^T M(\lambda_{v_i}^{(d)}) \text{vec}(g) = 0$ . Since  $M(\lambda_{v_i}^{(d)}) \succeq 0$ , we have  $\text{vec}(g)^T M(\lambda_{v_i}^{(d)}) = 0$  for  $i = 1, \dots, r$ . Hence  $\sum_{i=1}^r \alpha_i \text{vec}(g)^T M(\lambda_{v_i}^{(d)}) = 0$ , so  $\text{vec}(g)^T M(\lambda_d) = 0$  and  $g \in \mathbb{P} \ker M(\lambda_d)$  which is what we wanted to show.  $\square$

By Theorem 3.1, we now have a complete algorithm to Problem 3.1:

<b>Algorithm 1:</b> RealRadical( $F, d$ )
<p><b>Input</b>(<math>F = \{f_1, \dots, f_m\} \subseteq \mathbb{R}[x]</math>, <math>x \in \mathbb{R}^n</math>, an integer <math>d \geq \deg(F)</math>.)  Set <math>F'</math> to be the prolongation of <math>F</math> to degree <math>d</math>.  <b>repeat</b>      <math>B := \text{CoeffMtx}(F')</math>.      Solve for maximum rank moment matrix <math>M(\lambda_d)</math> such that <math>B^T M(\lambda_d) = 0, M(\lambda_d) \succeq 0</math>.      (This can be done by Algorithm 2 in the next section.)      <math>F'' := \mathbb{P}(\ker M(\lambda_d))</math>.      Compute <math>\text{GIF}(F'')</math>.      Project/ Prolong <math>\text{GIF}(F'')</math> to degree <math>d</math>: <math>F' := \text{GIF}(F'')_{(\leq d)}</math>.  <b>until</b> <math>\dim F' = \dim F''</math>;  <b>Output</b>(<math>F'</math>, a basis for <math>\{f \in \sqrt[\mathbb{R}]{I} : \deg(f) \leq d\}</math>.)</p>

In Algorithm 1, CoeffMtx computes the coefficients in the monomial basis, although potentially other bases could be used. It exploits the property that the the GIF algorithm obtains polynomials in a form that satisfies the ideal-like property. In particular note that for a given  $f$  in Definition 3.1,  $fg = \sum_{\alpha} a_{\alpha} x^{\alpha} f$  is expanded in term of so-called prolongations by monomials  $x^{\alpha}$ . The invariance of geometric involutive bases under prolongation-projection implies that each  $x^{\alpha} f$  is in the basis, and by superposition  $fg$  is also in the basis. We note that Pommaret involutive bases don't necessarily satisfy the ideal-like property but can be extended easily by an explicit algorithm to such basis [22, 40]. Groebner bases can also be extended, by essentially reformulating them as involutive basis [22].

Involutivity originates in the geometry of differential equations. See Kuranishi [25] for a famous proof of termination of Cartan’s prolongation algorithm for nonlinear partial differential equations. A by-product of these methods has been their implementation for linear homogeneous partial differential equations with constant coefficients, and consequently for polynomial algebraic systems. See [22] for applications and symbolic algorithms for polynomial systems. The symbolic-numeric version of a *geometric involutive form*, *GIF*, was first described and implemented in Wittkopf and Reid [46]. It was applied to approximate symmetries of differential equations in [6] and to polynomial solving in [38, 36, 39]. See [50] where it is applied to the deflation of multiplicities in multivariate polynomial solving. For more details and examples see [37, 6]. The details of the GIF algorithm, including, prolongations and projections, can be found in our earlier work [35].

#### 4. SDP and facial reduction

A symmetric matrix  $M$  of sizes  $k \times k$  is called positive semidefinite, denoted as  $M \succeq 0$ , if one of the following two criteria is satisfied:

1.  $x^T M x \geq 0$  for all  $x \in \mathbb{R}^k$ .
2. All eigenvalues of  $M$  are non-negative.

Similarly, a symmetric matrix  $M$  of sizes  $k \times k$  is called positive definite, denoted as  $M \succ 0$ , if one of the following two criteria is satisfied:

1.  $x^T M x > 0$  for all nonzero  $x \in \mathbb{R}^k$ .
2. All eigenvalues of  $M$  are strictly positive.

The set of all  $k \times k$  symmetric matrices are denoted as  $\mathcal{S}^k$ . The cone of  $k \times k$  all positive semidefinite matrices is denoted as  $\mathcal{S}_+^k$ . The cone of  $k \times k$  all positive definite matrices is denoted as  $\mathcal{S}_{++}^k$ .

**Definition 4.1** (Trace product). *Given two symmetric matrices  $A, B$ , we define the trace inner product  $\langle A, B \rangle = \text{trace}(A^T B) = \sum_{ij} A_{ij} B_{ij}$ .*

**Definition 4.2.** *Suppose  $A_1, \dots, A_l \in \mathbb{R}^{k \times k}$ , the linear operator  $\mathcal{A}$  from  $\mathbb{R}^{k \times k}$  to  $\mathbb{R}^l$  is defined as:*

$$\mathcal{A}(X) = [\langle A_1, X \rangle, \dots, \langle A_l, X \rangle]^T, X \in \mathbb{R}^{k \times k}. \quad (4.1)$$

*The adjoint operator of  $\mathcal{A}$  from  $\mathbb{R}^l$  to  $\mathbb{R}^{k \times k}$ , denoted as  $\mathcal{A}^*$ , is defined as:*

$$\mathcal{A}^* y = \sum_{i=1}^l A_i y_i, y \in \mathbb{R}^l. \quad (4.2)$$

**Definition 4.3.** *Given a matrix  $H = (a_{ij})_{1 \leq i, j \leq k} \in \mathbb{R}^{k \times k}$ , define  $\text{vec}(H)$  to be the vectorization of  $H$ , i.e.,*

$$\text{vec}(H) = [a_{11}, a_{12}, \dots, a_{1k}, a_{21}, a_{22}, \dots, a_{k1}, \dots, a_{kk}]^T.$$

*The matrix representation of the linear operator  $\mathcal{A}$ , denoted as  $A$ , is  $A = [\text{vec}(A_1), \dots, \text{vec}(A_l)]^T$ .*

#### 4.1. Face, minimal face and facial structure

We give a brief introduction to faces, minimal faces, and lemmas about facial structure. The definitions below can be found in [8, 9, 11, 18, 34].

**Definition 4.4.** Given convex cones  $F, K$  and  $F \subseteq K$ , we call  $F$  a face of  $K$ ,  $F \trianglelefteq K$  if

$$x, y \in K, x + y \in F \implies x, y \in F.$$

Given a nonempty convex subset  $S$  of  $K$ , the minimal face of  $K$  containing  $S$  is defined to be the intersection of all faces of  $K$  containing  $S$ .

**Definition 4.5.** Suppose  $F$  is a face of  $\mathcal{S}_+^k$ , the orthogonal complement of  $F$  denoted as  $F^\perp$ , is defined to be  $F^\perp = \{Z \in \mathcal{S}^k : Z \cdot X = 0, \forall X \in F\}$ . The dual cone of  $F$ , denoted as  $F^*$ , is defined to be  $F^* = \{Z \in \mathcal{S}^k : Z \cdot X \succeq 0, \forall X \in F\}$ .

The following lemmas about the facial structure of the semidefinite cone  $\mathcal{S}_+^k$  are well known, see e.g. [47].

**Lemma 4.1.** Any face  $F$  of  $\mathcal{S}_+^k$  is either  $0$ ,  $\mathcal{S}_+^k$  or

$$F = \{X \in \mathcal{S}^k : X = U M U^T, M \in \mathcal{S}_+^r\} \quad (4.3)$$

where  $U$  is an  $k \times r$  matrix.

**Lemma 4.2.** Suppose  $F$  is a face of  $\mathcal{S}_+^k$  and  $W \in \mathcal{S}_+^k$ . Then  $\mathcal{S}_+^k \cap \{W\}^\perp$  and  $F \cap \{W\}^\perp$  are faces of  $\mathcal{S}_+^k$ , where  $\{W\}^\perp = \{X \in \mathcal{S}^k : X \cdot W = 0\}$ .

#### 4.2. Facial reduction

The idea of facial reduction was originally developed by Borwein and Wolkowicz [8, 9] in the 1980s. However it has been nontrivial to develop practical algorithms implementing facial reduction. Only recently have practical algorithms been developed. For example it was recently applied to solve the large sensor network localization problems [24, 16].

We consider the set  $F_P = \{X \in \mathcal{S}^k : \mathcal{A}(X) = b, X \succeq 0\}$ , clearly  $F_P$  is a convex subset of  $\mathcal{S}^k$ . The following theorem gives information on the facial structure of  $F_P$ :

**Theorem 4.1** ([34, SDP version of Lemma 28.4]). Define  $F_{\min}$  to be the minimal face containing  $F_P$ .  $\mathcal{A}^*$  is the adjoint of  $\mathcal{A}$  defined before. For a face  $F \trianglelefteq \mathcal{S}_+^k$  containing  $F_P$ , the following holds :

$$\left\{ \begin{array}{l} (I) \quad \mathcal{A}(X) = b, X \in F \\ (II) \quad b^T y = 0, Z = \mathcal{A}^* y \in F^* \setminus F^\perp \end{array} \right\} \implies X \in \{Z\}^\perp \cap F \subset F. \quad (4.4)$$

In addition,  $F = F_{\min}$  if and only if (II) has no solution.

The matrix  $Z$  is called the *exposing vector* of  $F$ . Each time (II) is solved, an exposing vector  $Z$  is obtained and can be used to update  $F \leftarrow \{Z\}^\perp \cap F$ . Repeating this process until (II) is infeasible ((II) admits no solution), we get a sequence of faces containing  $F_P$ :  $F_0 \supset F_1 \supset F_2 \supset \dots \supset F_{\min} \supset F_P$  where  $F_0 = \mathcal{S}_+^k$  and  $F_{i+1} = F_i \cap \{Z_i\}^\perp$ . This iteration process to find the minimal face  $F_{\min}$  is called *facial reduction* on the primal form and is guaranteed to terminate in at most  $n - 1$  iterations [45]. The minimal number of facial reductions is called the *singularity degree*.

The correctness of Theorem 4.1 in the SDP case is due to the following theorem:

**Theorem 4.2** (Primal Theorem of Alternative [11, 18]). *Suppose  $\mathcal{A} : \mathcal{S}_+^k \rightarrow \mathbb{R}^l$  is a linear transformation,  $b \in \mathbb{R}^l$ ,  $P \in \mathcal{S}^k$  and  $Z \in \mathcal{S}^k$ . Then exactly one of the following alternative systems is consistent:*

$$(I) \quad 0 \prec P \in F := \{P \in \mathcal{S}^k : \mathcal{A}(P) = b, P \succeq 0\}. \quad (\text{Slater}) \quad (4.5a)$$

$$(II) \quad 0 \neq Z \in D := \{Z \in \mathcal{S}^k : Z = \mathcal{A}^*y \succeq 0, b^T y = 0\}. \quad (\text{Auxiliary}) \quad (4.5b)$$

PROOF. Note that if (II) is consistent, then  $Z$  exposes a face of  $\mathcal{S}_+^n$  that contains the minimal face  $(F, \mathcal{S}_+^n)$ . That is, for  $P \in F$  we have

$$\text{trace } ZP = \text{trace}(\mathcal{A}^*y)P = y^T b = 0.$$

The remainder of the proof can be found in [11, 18].

Equation (4.5a) is called the *primal problem* and equation (4.5b) is called the *auxiliary problem*.

### 4.3. Facial reduction maximum rank algorithm

Our facial reduction algorithm follows from Theorem 4.1. We use the following Lemmas to convert (I), (II) of Theorem 4.1 to equivalent problems which are easier and more practical to solve. The proofs of these Lemmas can be found in the Appendix.

**Lemma 4.3.** *Suppose a face is given as  $F = \{X \in \mathcal{S}^k : X = UMU^T, M \in \mathcal{S}_+^r\}$ . Then*

$$\exists X \in F, \mathcal{A}(X) = b \iff \exists \bar{X} \in \mathcal{S}_+^r, U^T \mathcal{A}U(\bar{X}) = b, \quad (4.6)$$

where  $U^T \mathcal{A}U$  is a linear operator from  $\mathcal{S}^r$  to  $\mathbb{R}^l$  defined as

$$U^T \mathcal{A}U(\bar{X}) = [\langle U^T A_1 U, X \rangle, \dots, \langle U^T A_l U, X \rangle]^T, \bar{X} \in \mathcal{S}^r. \quad (4.7)$$

**Lemma 4.4.** *Suppose  $F = \{X \in \mathcal{S}^k : X = UMU^T, M \in \mathcal{S}_+^r\}$ . Then*

$$\exists Z = \sum_{i=1}^l A_i y \in F^* \setminus F^\perp, b^T y = 0 \quad (4.8)$$

$$\iff \exists \bar{Z} = \sum_{i=1}^l U^T A_i U y \succeq 0 \neq 0, b^T y = 0. \quad (4.9)$$

**Lemma 4.5.** *Suppose  $Z$  is an exposing vector satisfying (4.8) and  $\bar{Z}$  satisfying (4.9) with  $V = \text{null}(\bar{Z})$ ,  $F = \{X \in \mathcal{S}^k : X = UMU^T, M \in \mathcal{S}_+^r\}$  is the face. Then*

$$\{Z\}^\perp \cap F = \{X \in \mathcal{S}^k : X = UV\bar{M}V^T U^T, \bar{M} \in \mathcal{S}_+^r\}. \quad (4.10)$$

Recall in Algorithm 1, we need to find  $M(\lambda_d)$  such that  $B^T M(\lambda_d) = 0, M(\lambda_d) \succeq 0$ . All such moment matrices form a convex subset of  $\mathbb{R}^{k \times k}$ . Also in general, all the moment matrix  $M(\lambda_d)$  form an affine subspace  $\mathcal{A}(X) = b$ . The construction of  $\mathcal{A}$  is described in [35]. So the set  $\{M(\lambda_d) : B^T M(\lambda_d) = 0, M(\lambda_d) \succeq 0\}$  can be converted to a convex set  $\mathcal{F}_p := \{X \in \mathcal{S}^k : \mathcal{A}(X) =$



$b, B^T X = 0, X \succeq 0$ . The algorithm to find maximum rank solutions of  $\mathcal{F}_p$  needed in Algorithm 1 is summarized as follows:

<p><b>Algorithm 2:</b> Facial reduction on the primal. Compute the minimal face <math>F_{min} := US_+^d U^T</math> of <math>S_+^k</math> containing <math>\mathcal{F}_p</math>, where <math>\mathcal{F}_p := \{X \in S^k : \mathcal{A}(X) = b, B^T X = 0, X \succeq 0\}</math>. Obtain the maximum rank solution of <math>\mathcal{F}_p</math>.</p>
<p>Input(<math>\mathcal{A} : S^k \rightarrow \mathbb{R}^l, b \in \mathbb{R}^l, B \in \mathbb{R}^{k \times m}</math>, set <math>j = 1, U = I</math>.)</p> <p><b>repeat</b></p> <p style="padding-left: 2em;">If <math>j = 1</math>, set <math>Z = BB^T</math>.</p> <p style="padding-left: 2em;">If <math>j &gt; 1</math>,</p> <p style="padding-left: 4em;">find <math>Z \succeq 0</math></p> <p style="padding-left: 4em;">subject to <math>Z = \sum_{i=1}^l A_i y_i, b^T y = 0 : y \in \mathbb{R}^l</math>. <span style="float: right;">(<math>\diamond</math>)</span></p> <p style="padding-left: 2em;">Find a basis <math>V</math> for <math>\text{null}(Z)</math>.</p> <p style="padding-left: 2em;">Update <math>\mathcal{A}</math> by setting <math>A_i \leftarrow V^T A_i V, i = 1 \dots l</math>.</p> <p style="padding-left: 2em;">Update <math>U</math> by setting <math>U \leftarrow U \cdot V</math>.</p> <p style="padding-left: 2em;"><math>j = j + 1</math>.</p> <p><b>until</b> (<math>\diamond</math>) only has zero solution;</p> <p>Solve <math>\mathcal{A}(P) = b, P \succ 0</math>. Solution of <math>\mathcal{F}_p</math> is <math>X := UPU^T</math>.</p> <p>Output(<math>X</math> which is maximum rank solution.)</p>

**Theorem 4.3** (Maximum rank). *Algorithm 2 returns a maximum rank solution of  $\mathcal{F}_p$ .*

PROOF. At step  $j$ , when an exposing vector  $Z \succeq 0$  is found ( $Z = BB^T$  when  $j = 1$  or  $Z$  satisfies ( $\diamond$ ) when  $j > 1$ ), we can reduce the problem to an equivalent smaller problem without loss of information by Lemma 4.3, 4.4, 4.5 and Theorem 4.1. When ( $\diamond$ ) only has zero solution, we have reduced the problem to a minimal face with no further facial reductions can be done according to Theorem 4.1 and all the feasible solutions of  $\mathcal{F}_p$  has the form  $X := UPU^T$ . By Theorem 4.2 when ( $\diamond$ ) only has zero solution, there exists  $P \succ 0$  such that  $\mathcal{A}(P) = b, P \succ 0$ . As a result,  $X := UPU^T$  is the maximum rank solution of  $\mathcal{F}_p$  if we can find  $P$  which is positive definite.  $\square$

**Remark 4.1** (Singularity degree). *The minimal number of facial reduction steps is called singularity degree. The examples in Section 7 show that some examples with singularity more than 1 can be accurately solved by Facial reduction heuristics. For more details, see [44, 17].*

## 5. Projection method

In Algorithm 2, we need to solve two problems: the auxiliary problem to solve is (5.4) and the primal problem after facial reduction to solve is  $\mathcal{A}(P) = b, P \succ 0$ . Essentially, we need to find the intersection between an affine subspace (linear constraints) and a positive semidefinite cone. We consider the *Douglas-Rachford reflection-projection* (DR) method which involves projections and reflections between two convex sets. These two convex sets are the affine subspace and the positive semidefinite cone in our case. There are also other projection-based methods, such as method of *alternating projection* [21]. We prefer the DR method as it displays better convergence properties in our tests. Also, unlike the alternating projection method, which is likely to converge to the boundary of cone, the DR method is likely to converge to the interior of the cone which is needed in Algorithm 2 for solving  $\mathcal{A}(P) = b, P \succ 0$ .

### 5.1. Projection to the positive semidefinite cone

Given  $X \in \mathcal{S}^k$ , denote  $\mathcal{P}_{\mathcal{S}_+^k}(X, r)$  as the projection of  $X$  to  $\mathcal{S}_+^k$  such that the projected matrix has rank  $r$ , we have the following well-known theorem:

**Theorem 5.1** (Eckart-Young [20]). *Suppose  $X \in \mathcal{S}^k$ , the projection of  $\mathcal{P}_{\mathcal{S}_+^k}(X, r)$  with  $r \leq k$  is:  $\mathcal{P}_{\mathcal{S}_+^k}(X, r) = V\mathcal{P}_{\mathcal{S}_+^k}(D, r)V^T$  and  $X = VDV^T$  is the eigenvalue decomposition of  $X$  and  $D$  is a diagonal matrix with all the eigenvalues of  $X$ .  $\mathcal{P}_{\mathcal{S}_+^k}(D, r)$  is obtained by keeping the first  $r$  largest positive eigenvalues unchanged while setting all the other eigenvalues to zero.*

### 5.2. Projection to an affine subspace

Suppose an affine subspace is given as follows:

$$\{X \in \mathcal{S}^k, \mathcal{A}(X) = b\}. \quad (5.1)$$

To project  $X$  from  $\mathcal{S}^k$  onto the affine subspace (5.1), we have the following well-known theorem:

**Theorem 5.2.** [33] *Given a matrix  $\bar{X} \in \mathcal{S}^k$ , and  $\mathcal{A}, b$  as in (5.1). Let  $A$  be the matrix representation of  $\mathcal{A}$  as defined in Definition (4.3) and  $A^\dagger$  be the Moore-Penrose pseudoinverse of  $A$ , i.e.,  $A^\dagger = A^T(AA^T)^{-1}$ .*

$$\begin{aligned} \text{Suppose } X^* &:= \operatorname{argmin}\{\|X - \bar{X}\| : \mathcal{A}(X) = b\} \\ \text{Then } X^* &= \bar{X} + A^\dagger(b - A\bar{X}). \end{aligned} \quad (5.2)$$

We denote  $X^* = \mathcal{P}_{\mathcal{A}}(X)$ .

### 5.3. Transform of the auxiliary problem

The auxiliary problem ( $\diamond$ ) can be solved by CVX or other SDP solvers, but in order to get higher accuracy, we use Douglas-Rachford iteration. To do that, we need to reformulate the auxiliary problem ( $\diamond$ ). First, it is easy to see problem ( $\diamond$ ) can be converted to the form:

$$\begin{aligned} \text{Find } y \in \mathbb{R}^l : b^T y = 0, A^T y - \operatorname{vec}(Z) = 0, \\ Z \succeq 0, \operatorname{trace}(Z) = 1. \end{aligned} \quad (5.3)$$

We add the trace constraint to make sure  $Z \neq 0$ . If (5.3) is infeasible then ( $\diamond$ ) only has zero solution.

In addition, the following theorem shows how to transform problem (5.3) into a simpler form that is suitable for applying the Douglas-Rachford method.

**Theorem 5.3.** *Suppose  $A$  is the matrix representation of the linear operator  $\mathcal{A}$  and  $(A^T)^\dagger$  is the Moore-Penrose pseudoinverse of  $A^T$ . Let  $L = [b^T \cdot (A^T)^\dagger; I - A^T \cdot (A^T)^\dagger; \operatorname{vec}(I)]$  and  $R = [0; 0; 1]$ . Then problem (5.3) is equivalent to the following:*

$$\text{Find } Z \in \mathcal{S}^k : L \cdot \operatorname{vec}(Z) = R, Z \succeq 0. \quad (5.4)$$

PROOF. Let's assume  $\operatorname{vec}(Z) = A^T y$ , then we have  $A^T(A^T)^\dagger \operatorname{vec}(Z) = A^T(A^T)^\dagger A^T y = A^T y = \operatorname{vec}(Z)$  since  $(A^T)^\dagger A^T = I$ . Also  $(A^T)^\dagger \operatorname{vec}(Z) = (A^T)^\dagger A^T y = y$ .

It is easy to verify the other direction, by making the substitution  $y = (A^T)^\dagger \operatorname{vec}(Z)$ .  $\square$

By our experiments, we found this formulation has the best performance when coupled with the Douglas-Rachford methods. So we use (5.4) for solving problem ( $\diamond$ ) in Algorithm 2.

#### 5.4. The Douglas-Rachford method

In Sections 5.1 and 5.2 we showed how to project a matrix to a positive semidefinite cone and an affine subspace. Briefly speaking, the DR methods first project a matrix  $X$  to the positive semidefinite cone, then reflect it by multiplying the projected matrix by 2 and subtracting  $X$  from it. Similarly, the resulting matrix is projected and reflected over an affine subspace as well. Finally the average of the original matrix and the reflected matrix is taken to update  $X$  to  $X_{new}$ . The convergence rate of DR method is studied by Bauschke et al [4, 5]. The original idea about the Douglas-Rachford method came from solving partial differential equations [15]. Then later Lions and Mercier brought the Douglas-Rachford method to light by connecting it to convex analysis [29]. (More details about the DR method can be found in e.g., [2, 7].) We apply Douglas-Rachford to solve both the primal problem and the auxiliary problem. One step of the Douglas-Rachford method is the following:

$$\begin{aligned} Y &= 2\mathcal{P}_{\mathcal{S}_+^k}(X, r) - X, \\ Z &= 2\mathcal{P}_{\mathcal{A}}(Y) - Y, \\ X_{new} &= (X + Z)/2. \end{aligned} \tag{5.5}$$

At each step, we calculate the residual  $Res := \|\mathcal{A}(Y) - b\|$ , which is the residual after projecting onto the positive semidefinite cone. If the residual is less than the given tolerance, we stop and return  $Y$ . According to the basic theorem on the convergence of the sequence, [7, Thm 3.3, Page 11], the residuals of the projections of the iterates on one of the sets have to be used for the stopping criteria. We use the residual after the projection onto the SDP cone since we want our final matrix to be positive semidefinite.

#### 5.5. Choosing the appropriate rank for the projections

In practice, some problems appear to be very ill-conditioned. One example is the geometric polynomial in Section 7. Those examples have eigenvalue decomposition of the solutions from problem  $(\diamond)$  with some eigenvalues that are very small compared to the others, and the DR iterations converge very slowly. This indicates the rank  $r$  used in the projection  $\mathcal{P}_{\mathcal{S}_+^k}(X, r)$  can not be maximum.

To deal with such problems, we would have to project the matrix to a good rank  $r$  matrix as described in Theorem 5.1 when applying the DR method to (5.4) for solving problem  $(\diamond)$ . In other words, at each step of facial reduction, we are not computing the smallest possible face. Instead, we try to find a bigger but much more accurate face. So we may need more facial reductions but we can obtain more accurate results.

The strategy we used to get this good matrix is to look at the eigenvalues of  $Z$  in (5.4). We drop the eigenvalues which are significantly smaller than the other eigenvalues and  $r$  is chosen to be the number of eigenvalues which are well conditioned. For example, if the eigenvalues are 0.7, 0.2, 0.00002, 0, 0, 0, we will choose  $r = 2$  instead of 3 or 6. After this, we will resolve (5.4) with the updated  $r$  to obtain a more accurate face.

## 6. A special case for determining positive dimensional real radical

Our theorem on the determination of the real radical up to finite degree is illustrated graphically in Figure B.1. Here suppose  $F = \{f_1, \dots, f_m\} \subset \mathbb{R}[x]$  and we applied Algorithm RealRadical( $F, d$ ) for a given  $d$ , and that the resulting system has leading monomials shown as the corners of the black monomial staircase. See [13] for the description of such diagrams. Then the system is prolonged

and the kernel of its moment matrix is examined for new generators at degrees  $d+1, d+2, \dots$ . The only way that this is not a complete generating set for the real radical (and that our conjecture fails), is that there is a minimum degree  $d' > d$  where after prolongation to  $d'$  new generators are determined that lie outside simple prolongations of the black leading generators. These have leading monomials shown in red. Some times the completeness of the generating set at degree  $d$  can be checked by a critical point calculation. For example, if the critical point method shows that the variety is real positive dimensional, then this could rule out the existence of the red staircase predicting a 0-dimensional real variety. In particular, if the number of red circles in Figure B.1 is 1 and the variety of  $F$  is real positive dimensional, then  $\text{RealRadical}(F, d)$  returns the generators of  $\sqrt[\mathbb{R}]{\langle F \rangle_{\mathbb{R}}}$ . So we have the following theorem:

**Theorem 6.1.** *Given a system of polynomials  $F = \{f_1, \dots, f_m\} \subseteq \mathbb{R}[x_1, x_2, \dots, x_n]$  with associated ideal  $I$  and an integer  $d$ . Let  $G = \{g_1, \dots, g_k\} \subset \mathbb{R}[x]$  be the output of the  $\text{RealRadical}(F, d)$  algorithm applied to  $F$  and  $s$  is the number of different polynomials of degree  $d$  in  $G$ . If  $s = \binom{d+n-1}{n-1} - 1$  and the variety of  $F$  is real positive dimensional. Then*

$$\sqrt[\mathbb{R}]{\langle F \rangle_{\mathbb{R}}} = \langle G \rangle_{\mathbb{R}}. \quad (6.1)$$

PROOF. By Theorem 3.1,  $\left(\sqrt[\mathbb{R}]{\langle F \rangle_{\mathbb{R}}}\right)_{(\leq d)} = \text{span}_{\mathbb{R}} G$ . Suppose in contradiction  $\sqrt[\mathbb{R}]{\langle F \rangle_{\mathbb{R}}} \supset \langle G \rangle_{\mathbb{R}}$ , then there exists a  $d' > d$  such that  $(\langle H \rangle_{\mathbb{R}})_{(\leq d')} \subset \left(\sqrt[\mathbb{R}]{\langle F \rangle_{\mathbb{R}}}\right)_{(\leq d')}$  where  $H$  is the prolongation of  $G$  to degree  $d'$ . Therefore there exists a polynomial  $\tilde{g} \in \text{span}_{\mathbb{R}} \tilde{G}$  but  $\tilde{g} \notin \text{span}_{\mathbb{R}} H$  with  $\deg(\tilde{g}) = d' > d$  where  $\tilde{G} = \{\tilde{g}_1, \dots, \tilde{g}_t\}$  spans  $\left(\sqrt[\mathbb{R}]{\langle F \rangle_{\mathbb{R}}}\right)_{(\leq d')}$ .

Now assume the number of different polynomials of degree  $d'$  in  $H$  is  $t$  and the number of different polynomials of degree  $d'$  in  $\tilde{G}$  is  $\bar{t}$ , then  $t < \bar{t}$  because the existence of  $\tilde{g}$ . From combinatorics, the number of different monomials of degree  $d$  in  $n$  variables is  $\binom{d+n-1}{n-1}$ . Since  $G$  is already involutive and  $s = \binom{d+n-1}{n-1} - 1$ , we have  $t = \binom{d+n-1}{n-1} - 1$  as well. Also clearly  $\bar{t} \leq \binom{d'+n-1}{n-1}$ , so we have  $\bar{t} = \binom{d'+n-1}{n-1}$  which means  $\sqrt[\mathbb{R}]{\langle F \rangle_{\mathbb{R}}}$  is a 0-dimensional real variety, a contradiction with the assumption that the variety of  $F$  is real positive dimensional. So the theorem is proved.  $\square$

## 7. Examples

In this section, we give some examples. We used MATLAB version 2015a. The computations were carried out on a desktop with ubuntu 12.04 LTS, Intel Core™2 Quad CPU Q9550 @ 2.83 GHz  $\times$  4, 8GB RAM, 64-bit OS, x64-based processor.

We give the first examples (Ex.7.2 and Ex.7.3) showing additional facial reductions for polynomials, that can be accurately approximated in practice. Our previous attempts [35] were not accurate.

**Example 7.1** (Reducible cubic).

$$(x + y)(x^2 + y^2 + 2). \quad (7.1)$$

*Note that the second factor has no real roots, so it is discarded and the real radical is generated by  $(x + y)$ . The moment matrix corresponding to (7.1) is a  $10 \times 10$  matrix. The coefficient matrix  $B$  is  $[0, 2, 2, 0, 0, 0, 1, 1, 1, 1]^T$ . Using Algorithm 1, after two facial reductions, we obtained a maximum rank 4 moment matrix with residual less than  $10^{-14}$  in less than 200 DR iterations and the generators of real radical is computed to degree 3.*

**Example 7.2** (Reducible quintic).

$$(1 + x + y)(x^4 + y^4 + 2). \quad (7.2)$$

The moment matrix corresponding to (7.2) is a  $21 \times 21$  matrix. We solve this problem using Algorithm 1. Algorithm 1 can get 14 decimal accuracy and a maximum rank moment matrix of rank 6 in about 1300 DR iterations with 2 facial reductions. The output approximates the real radical ideal generated by  $\langle 1 + x + y \rangle$  and its prolongations to degree 5.

**Example 7.3** (Two variable geometric polynomial with 3 facial reductions).

$$1 + (x + y) + (x + y)^2 + (x + y)^3. \quad (7.3)$$

The moment matrix corresponding to (7.3) is a  $10 \times 10$  matrix. The coefficient matrix  $B$  is  $[2, 2, 2, 1, 0, 1, 1, 1, 1, 1]^T$ .

This example is a demonstration of the ill-conditioned case discussed in Section 5.5. We first solve it using Algorithm 2 with rank  $r$  to be maximum in  $\mathcal{P}_{\mathcal{S}_+^k}(X, r)$ , which returns solution of rank 5 with residual  $10^{-7}$  after 2 facial reductions. However, the DR method for solving the auxiliary problem (4.5b) converges very slowly. So we check the eigenvalues of solution of the auxiliary problem (4.5b). After the first facial reduction, the eigenvalues are 0.5, 0.2, 0.18, 0.08, 0, 0, 0, 0, 0. So we drop the fourth one and set  $r = 3$ . We resolve (4.5b) using the DR method, which again is quite slow. So we check the eigenvalues and they are now 0.709, 0.29, 0.00002, 0, 0, 0, 0, 0, 0. The third one is very small so we drop it and set  $r = 2$ . Then we resolve (4.5b) with  $r = 2$ . This time the auxiliary problem is solved with residual  $10^{-15}$ . Then a third facial reduction is done by setting  $r = 3$  and the residual is  $10^{-14}$ .

After 3 facial reductions, the face is reduced to dimension 4 and the moment matrix is obtained with residual  $10^{-13}$ . The eigenvalues of the final moment matrix are 4.70, 3.48, 0.89, 0.59, 0, 0, 0, 0, 0, 0 which gives the correct maximum rank of 4.

Application of Algorithm 1 yields the correct generators of the real radical up to degree 3.

**Example 7.4.** [10]

$$f = \{2yz - y, 2y^2 + y, xy, 4x^2z + 4z^3 + y\}. \quad (7.4)$$

The real radical of this polynomial system is [10]:

$$\{z^2 + y/2, yz - y/2, y^2 + y/2, xz, xy, y + z\}.$$

The moment matrix of this problem is  $20 \times 20$ . We use Algorithm 2 to solve for maximum rank moment matrix. The sizes of the SDP problem are [20, 16, 14, 8] after 3 facial reductions. The residual of the auxiliary problem at each facial reduction is  $10^{-15}, 10^{-14}$ . (The first facial reduction is done by Matlab eigenvalue decomposition so we don't put its residual here.) The moment matrix is solved with residual  $10^{-13}$  and the maximum rank is 8.

We then compare algorithm 2 with traditional interior point solver SeDuMi(CVX). As the computations in the above examples and Table B.1, B.2 demonstrate, the traditional interior point SDP solver SeDuMi(CVX) is not the right choice for computing the maximum rank moment matrices as it usually has poor performance when it is trying to maximize rank. With facial reductions and the DR method, we can get much better accuracy and also the correct maximum rank.

## 8. Perturbed examples

In this section, we study how small perturbations affect our algorithm 2. The computational results are shown in Table B.3.

**Example 8.1** (Perturbed Reducible cubic).

$$(x + y)(x^2 + y^2 + 0.000001xy + 2). \quad (8.1)$$

**Example 8.2** (Perturbed Reducible quintic).

$$(1 + x + y)(x^4 + y^4 + 0.000001xy + 2). \quad (8.2)$$

**Example 8.3.**

$$1 + 1.000001(x + y) + 0.999999(x + y)^2 + 1.000001(x + y)^3. \quad (8.3)$$

**Example 8.4.**

$$f = \{2yz - y + \epsilon, 2y^2 + y - \epsilon, xy + \epsilon, 4x^2z + 4z^3 + y - \epsilon, \epsilon = 1 \times 10^{-14}\}. \quad (8.4)$$

In examples 8.1 8.2 and 8.1 where the coefficients of the real radical ideal change continuously with respect to the changes of the input polynomial system, algorithm 2 has the same performance on the examples without perturbation. For example 8.4 where theoretically the real radical ideal can be very different under small perturbations, algorithm 2 still works very well if the perturbation is smaller than the residual of the final moment matrix.

## 9. Conclusion

SDP feasibility problems typically involve the intersection of the convex cone of semi-definite matrices with a linear manifold. Their importance in applications has led to the development of many specific algorithms. However these feasibility problems are often marginally infeasible, i.e., they do not satisfy strict feasibility as is the case for our polynomial applications. Such problems are *ill-posed* and *ill-conditioned*.

This paper is part of a series in which we exploit facial reduction and its application systems of real polynomial and differential equations for real solutions. The current work is directed at guaranteeing the maximal rank property and the ideal-like condition to ensure all the generators of the real radical up to a given degree are captured. It also establishes the first examples of additional facial reduction that are effective in practice for polynomial systems.

This builds on our work in [35] in which we introduced facial reduction, for the class of SDP problems arising from analysis and solution of systems of real polynomial equations for real solutions. Facial reduction yields an equivalent smaller problem for which there are strictly feasible generic points. Facial reduction also reduces the size of the moment matrices occurring in the application of SDP methods. For example the determination of a  $k \times k$  moment matrix for a problem with  $m$  linearly independent constraints is reduced to a  $(k - m) \times (k - m)$  moment matrix by one facial reduction. The high accuracy required by facial reduction and also the ill-conditioning commonly encountered in numerical polynomial algebra [43] motivated us to implement Douglas-Rachford iteration in [35].

A fundamental open problem is to generalize the work of [26, 42] to positive dimensional ideals. In essence, this requires to determine a degree bound  $d$  such that  $\left(\sqrt[\mathbb{R}]{I}\right)_{(\leq d)}$  actually generates the

whole real radical  $\sqrt[\mathbb{R}]{I}$ . At the current stage, there is no practical degree bound when  $I$  is real positive dimensional. In section 6, we only give an answer for a special case.

An important open problem is the following: *Give an numerical algorithm, capable in principle of determining an approximate real point on each component of a real variety.* We note that the methods of Wu and Reid [49] and Hauenstein [23] only answer this question under certain conditions, say that the ideal is real radical and defined by a regular sequence. Also see [30], which gives an alternative extension of complex numerical algebraic geometry to the reals, in the complex curve case.

Recently, Hauenstein et al [10] have made progress on this problem by using sample points determined by Hauenstein's critical point algorithm which is able to certify the generators of the real radical ideal in some cases. Our results Theorem 3.1 enables the determination of the generators up to a given degree. Thus gives an answer to the open problem of real radical ideal membership test left in [10]. Potentially, the efficiency for computing the sample points can also be improved which will be described in a subsequent work.

## References

- [1] A. Alfakih and H. Wolkowicz. Matrix completion problems. In *Handbook of semidefinite programming*, volume 27 of *Internat. Ser. Oper. Res. Management Sci.*, pages 533–545. Kluwer Acad. Publ., Boston, MA, 2000. 2
- [2] F.J.A. Artacho, J.M. Borwein, and M.K. Tam. Recent results on Douglas-Rachford methods. *Serdica Mathematical Journal*, 39:313–330, 2013. 11
- [3] S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in Real Algebraic Geometry*, volume 10 of *Algorithms and Computation in Math.* Springer-Verlag, 2 edition, 2006. 2, 3
- [4] H.H. Bauschke, J.Y. Bello Cruz, T.T.A. Nghia, H.M. Phan, and X. Wang. The rate of linear convergence of the Douglas-Rachford algorithm for subspaces is the cosine of the Friedrichs angle. *J. Approx. Theory*, 185:63–79, 2014. 11
- [5] H.H. Bauschke and D. Noll. On the local convergence of the Douglas-Rachford algorithm. *Arch. Math. (Basel)*, 102(6):589–600, 2014. 11
- [6] J. Bonasia, F. Lemaire, G.J. Reid, and L. Zhi. Determination of approximate symmetries of differential equations. *Group Theory and Numerical Analysis*, 39:249, 2005. 6
- [7] J.M. Borwein and M.K. Tam. A Cyclic Douglas–Rachford Iteration Scheme. *J. Optim. Theory Appl.*, 160(1):1–29, 2014. 11
- [8] J.M. Borwein and H. Wolkowicz. Facial reduction for a cone-convex programming problem. *J. Austral. Math. Soc. Ser. A*, 30(3):369–380, 1980/81. 2, 7
- [9] J.M. Borwein and H. Wolkowicz. Regularizing the abstract convex program. *J. Math. Anal. Appl.*, 83(2):495–530, 1981. 2, 7
- [10] D. Brake, J. Hauenstein, and A. Liddell. Numerically validating the completeness of the real solution set of a system of polynomial equations. *Proceedings of the 41th International Symposium on Symbolic and Algebraic Computation*, 2016. 2, 13, 15

- [11] Y.-L. Cheung, S. Schurr, and H. Wolkowicz. Preprocessing and regularization for degenerate semidefinite programs. In D.H. Bailey, H.H. Bauschke, P. Borwein, F. Garvan, M. Thera, J. Vanderwerff, and H. Wolkowicz, editors, *Computational and Analytical Mathematics, In Honor of Jonathan Borwein's 60th Birthday*, volume 50 of *Springer Proceedings in Mathematics & Statistics*, pages 225–276. Springer, 2013. 7, 8
- [12] Y.-L. Cheung and H. Wolkowicz. Sensitivity analysis of semidefinite programs without strong duality. Technical report, University of Waterloo, Waterloo, Ontario, 2014. submitted June 2014, 37 pages. 2
- [13] David Cox, John Little, and Donal O'shea. *Ideals, varieties, and algorithms*, volume 3. Springer, 1992. 11
- [14] RE Curto and LA Fialkow. Solution of the truncated complex moment problem for flat data-introduction. *Memoirs of the American Mathematical Society*, 119(568):1, 1996. 3, 4
- [15] Jim Douglas, JR and H.H. Rachford, JR. On the numerical solution of heat conduction problems in two and three space variables. *Trans. Amer. Math. Soc.*, 82:421–439, 1956. 11
- [16] D. Drusvyatskiy, N. Krislock, Y.-L. Cheung Voronin, and H. Wolkowicz. Noisy sensor network localization: robust facial reduction and the Pareto frontier. Technical report, University of Waterloo, Waterloo, Ontario, 2014. arXiv:1410.6852, 20 pages. 2, 7
- [17] D. Drusvyatskiy, G. Li, and H. Wolkowicz. Alternating projections for ill-posed semidefinite feasibility problems. Technical report, University of Waterloo, Waterloo, Ontario, 2014. submitted Sept. 2014, 12 pages. 2, 9
- [18] D. Drusvyatskiy, G. Pataki, and H. Wolkowicz. Coordinate shadows of semi-definite and euclidean distance matrices. *Math. Programming*, 25(2):1160–1178, 2015. ArXiv:1405.2037.v1. 7, 8
- [19] M. Dür, B. Jargalsaikhan, and G. Still. The Slater condition is generic in linear conic programming. Technical report, University of Trier, Trier, Germany, 2012. 2
- [20] C. Eckart and G. Young. A principal axis transformation for non-Hermitian matrices. *Bull. Amer. Math. Soc.*, 45:118–121, 1939. 10
- [21] R. Escalante and M. Raydan. *Alternating projection methods*, volume 8 of *Fundamentals of Algorithms*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2011. 9
- [22] V.P. Gerdt and Y.A. Blinkov. Involutive bases of polynomial ideals. *Mathematics and Computers in Simulation*, 45(5):519–541, 1998. 5, 6
- [23] Jonathan D Hauenstein. Numerically computing real points on algebraic sets. *Acta applicandae mathematicae*, 125(1):105–119, 2013. 15
- [24] N. Krislock and H. Wolkowicz. Explicit sensor network localization using semidefinite representations and facial reductions. *SIAM Journal on Optimization*, 20(5):2679–2708, 2010. 2, 7
- [25] M. Kuranishi. On e. cartan's prolongation theorem of exterior differential systems. *American Journal of Mathematics*, pages 1–47, 1957. 6



- [26] J.B. Lasserre, M. Laurent, and P. Rostalski. A prolongation–projection algorithm for computing the finite real variety of an ideal. *Theoretical Computer Science*, 410(27):2685–2700, 2009. 1, 14
- [27] Jean Bernard Lasserre, Monique Laurent, and Philipp Rostalski. Semidefinite characterization and computation of zero-dimensional real radical ideals. *Foundations of Computational Mathematics*, 8(5):607–647, 2008. 3, 4
- [28] M. Laurent and P. Rostalski. The approach of moments for polynomial equations. In Miguel F. Anjos and Jean B. Lasserre, editors, *Handbook on semidefinite, conic and polynomial optimization*, International Series in Operations Research & Management Science, 166, pages 25–60. Springer, New York, 2012. 3
- [29] P. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979, 1979. 11
- [30] Y. Lu, D.J. Bates, A.J. Sommese, and C.W. Wampler. Finding all real points of a complex curve. In *Algebra, geometry and their interactions*, volume 448 of *Contemp. Math.*, pages 183–205. Amer. Math. Soc., Providence, RI, 2007. 15
- [31] Y. Ma. *Polynomial Optimization via Low-rank Matrix Completion and Semidefinite Programming*. PhD thesis, Academy of Mathematics and Systems Science, Chinese Academy of Science, 2012. 1
- [32] Y. Ma, C. Wang, and L. Zhi. A certificate for semidefinite relaxations in computing positive dimensional real varieties. *Journal of Symbolic Computation*, 72:1 – 20, 2016. 1
- [33] Carl D Meyer. *Matrix analysis and applied linear algebra*, volume 2. Siam, 2000. 10
- [34] G. Pataki. Strong duality in conic linear programming: facial reduction and extended duals. In David Bailey, Heinz H. Bauschke, Frank Garvan, Michel Thera, Jon D. Vanderwerff, and Henry Wolkowicz, editors, *Computational and analytical mathematics*, volume 50 of *Springer Proc. Math. Stat.*, pages 613–634. Springer, New York, 2013. 7
- [35] G. Reid, F. Wang, H. Wolkowicz, and W. Wu. Semidefinite Programming and facial reduction for Systems of Polynomial Equations. *Preprint arXiv:1504.00931v1*, 2015. 1, 4, 6, 8, 12, 14
- [36] G.J. Reid, J. Tang, and L. Zhi. A complete symbolic-numeric linear method for camera pose determination. In *Proceedings of the 2003 international symposium on Symbolic and algebraic computation*, pages 215–223. ACM, 2003. 6
- [37] G.J. Reid, F. Wang, and W. Wu. Geometric involutive bases for positive dimensional polynomial ideals and sdp methods. Technical report, Department of Appl. Math., University of Western Ontario, 2014. 6
- [38] G.J. Reid and L. Zhi. Solving polynomial systems via symbolic-numeric reduction to geometric involutive form. *Journal of Symbolic Computation*, 44(3):280–291, 2009. 6
- [39] R. Scott, G.J. Reid, W. Wu, and L. Zhi. Geometric involutive bases and applications to approximate commutative algebra. In Lorenzo Robbiano and John Abbott, editors, *Approximate Commutative Algebra*, pages 99–124. Springer, 2010. 6

- [40] Werner M Seiler. *Involution: The formal theory of differential equations and its applications in computer algebra*, volume 24 of *Algorithms and Computation in Mathematics*. Springer, 2010. 5
- [41] A.J. Sommese and C.W. Wampler. *The Numerical solution of systems of polynomials arising in engineering and science*, volume 99. World Scientific, 2005. 1
- [42] F. Sottile. *Real solutions to equations from geometry*, volume 57 of *University Lecture Series*. American Mathematical Society, Providence, RI, 2011. 1, 14
- [43] Hans J. Stetter. *Numerical polynomial algebra*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2004. 14
- [44] J.F. Sturm. Error bounds for linear matrix inequalities. *SIAM J. Optim.*, 10(4):1228–1248 (electronic), 2000. 9
- [45] Levent Tunçel. *Polyhedral and semidefinite programming methods in combinatorial optimization*, volume 27 of *Fields Institute Monographs*. American Mathematical Society, Providence, RI, 2010. 7
- [46] A.D. Wittkopf and G.J. Reid. Fast differential elimination in c: The cdiffelim environment. *Computer Physics Communications*, 139(2):192–217, 2001. 6
- [47] H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors. *Handbook of semidefinite programming*. International Series in Operations Research & Management Science, 27. Kluwer Academic Publishers, Boston, MA, 2000. Theory, algorithms, and applications. 7
- [48] H. Wolkowicz and Q. Zhao. Semidefinite programming relaxations for the graph partitioning problem. *Discrete Appl. Math.*, 96/97:461–479, 1999. Selected for the special Editors’ Choice, Edition 1999. 2
- [49] W. Wu and G.J. Reid. Finding points on real solution components and applications to differential polynomial systems. In *Proceedings of the 38th international symposium on International symposium on symbolic and algebraic computation*, pages 339–346. ACM, 2013. 15
- [50] X. Wu and L. Zhi. Determining singular solutions of polynomial systems via symbolic–numeric reduction to geometric involutive forms. *Journal of Symbolic Computation*, 47(3):227–238, 2012. 6
- [51] Q. Zhao, S.E. Karisch, F. Rendl, and H. Wolkowicz. Semidefinite programming relaxations for the quadratic assignment problem. *J. Comb. Optim.*, 2(1):71–109, 1998. Semidefinite programming and interior-point approaches for combinatorial optimization problems (Fields Institute, Toronto, ON, 1996). 2

## Appendix A. Proofs of Lemma 4.3, 4.4, 4.5.

### Appendix A.1. Proof of Lemma 4.3

First suppose there exists  $X = UMU^T$  satisfying  $\mathcal{A}(X) = b$ , then we have  $U^T \mathcal{A}U(M) = \mathcal{A}(UMU^T) = b$  due to the cyclic property of the trace product.

For the other direction, suppose there exists  $\bar{X}$  satisfying  $U^T \mathcal{A}U(\bar{X}) = b$ , let  $X = U\bar{X}^T U^T$  then it is easy to see  $\mathcal{A}(X) = b$  as well. □

*Appendix A.2. Proof of Lemma 4.4*

Suppose (4.8) holds, there exists  $Z = \sum_{i=1}^l A_i y \in F^*$  which means  $\langle Z, U M U^T \rangle \succeq 0$  for all  $M \in \mathcal{S}_+^r$  and  $\langle U^T Z U, M \rangle \succeq 0$  for all  $M \in \mathcal{S}_+^r$ . Also  $Z \notin F^\perp$  which means  $\langle U^T Z U, M \rangle \neq 0$  for some  $M \in \mathcal{S}_+^r$  which indicates  $U^T Z U \neq 0$ .

Now suppose (4.9) holds, since  $\bar{Z} = U^T Z U \succeq 0$ , we have  $\langle Z, U M U^T \rangle = \langle U^T Z U, M \rangle \succeq 0$  for all  $M \in \mathcal{S}_+^r$ . Hence  $Z \in F^*$ . Since  $\bar{Z} \neq 0$ , we have  $Z \notin \text{null}(U^T)$  so  $Z \notin F^\perp$ .  $\square$

*Appendix A.3. Proof of Lemma 4.5*

First, suppose  $X = U V \bar{M} V^T U^T$ , then  $\langle Z, X \rangle = \langle U^T Z U, V \bar{M} V^T \rangle = 0$  which means  $Z X = 0$  since  $Z \succeq 0, X \succeq 0$ . So  $X \in \{Z\}^\perp$  and  $X \in F$ .

For the other direction, if  $X \in F$ , then  $X = U M U^T$  for some  $M \in \mathcal{S}_+^r$ . If  $X \in \{Z\}^\perp$ , then  $X Z = 0$  which means  $\langle X, Z \rangle = \langle M, U^T Z U \rangle = \langle M, \bar{Z} \rangle = 0 \Rightarrow M \bar{Z} = 0$ . Hence  $M = V \bar{M} V^T$  for  $V = \text{null}(\bar{Z})$  and  $X = U V \bar{M} V^T U$  for some  $\bar{M} \in \mathcal{S}_+^r$ .  $\square$

**Appendix B. Tables and figures**

	min # FR	rank (FR)	Singlty deg	Res(FR)	Res(CVX)
Ex 7.1	2	10, 9, 4	1	$10^{-14}$	$10^{-9}$
Ex 7.2	2	21, 20, 6	1	$10^{-14}$	$10^{-9}$
Ex 7.3	3	10, 9, 7, 4	2	$10^{-13}$	$10^{-9}$
Ex 7.4	3	20, 16, 14, 8	2	$10^{-13}$	$10^{-9}$

Table B.1: **Comparison between facial reduction and SeDuMi (1)** Here: min (max) # FR means minimal (maximum) number of facial reductions in our tests; rank(FR) means the size of the problem after each facial reduction, the first one is the size of the original problem; Singlty degree is the singularity degree of the SDP problem after the 1st facial reduction; Res(FR) is the residual of the final moment matrix using facial reduction and DR iterations (Algorithm 2); Res(CVX) is the residual of the final moment matrix using CVX(SeDuMi).

	max rank	res each FR	# DR each FR	toler FR	toler CVX
Ex 7.1	4	$10^{-15}, 10^{-15}$	120, 7	$10^{-16}$	$10^{-12}$
Ex 7.2	6	$10^{-15}, 10^{-14}$	267, 6	$10^{-16}$	$10^{-9}$
Ex 7.3	4	$10^{-15}, 10^{-14}, 10^{-15}$	260, 143, 1	$10^{-16}$	$10^{-5}$
Ex 7.4	8	$10^{-15}, 10^{-14}, 10^{-14}$	625, 192, 29	$10^{-16}$	$10^{-7}$

Table B.2: **Comparison between facial reduction and SeDuMi (2)** max rank is the maximum rank of the moment matrix; res each FR is the residual of solving the corresponding SDP problem by DR after each facial reduction; # DR each FR is the number of DR iterations to solve the corresponding SDP problem after each facial reduction; toler FR is the tolerance of eigenvalues to obtain the correct maximum rank using facial reductions (Algorithm 2); toler CVX is the tolerance of eigenvalues to obtain the correct maximum rank using CVX(SeDuMi);

	min # FR	rank (FR)	Singlty deg	# DR each FR	Res(FR)
Ex 8.1	2	10, 9, 4	1	120, 7	$10^{-14}$
Ex 8.2	2	21, 20, 6	1	245, 6	$10^{-13}$
Ex 8.3	3	10, 9, 7, 4	2	260, 146, 1	$10^{-13}$
Ex 8.4	3	20, 16, 14, 8	2	616, 452, 29	$10^{-13}$

Table B.3: **Perturbations**

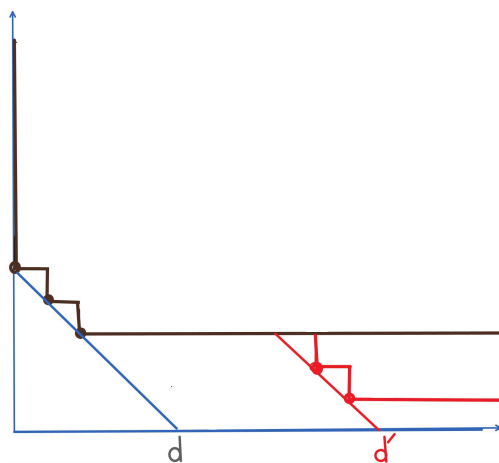


Figure B.1: In the Figure, the black monomial staircase represents the leading monomials of the generators of the real radical determined to degree  $d$  by  $\text{RealRadical}(F, d)$ . The only way these can fail to be a complete set of generators for the real radical is that there is a minimum degree  $d' > d$  where additional generators with leading monomials of exactly degree  $d'$  shown in red are found outside black monomial staircase.