Computing Real Solutions of Polynomial Systems via Low-rank Moment Matrix Completion*

Yue Ma and Lihong Zhi
Key Laboratory of Mathematics Mechanization
AMSS, Beijing 100190, China
{yma, lzhi}@mmrc.iss.ac.cn

ABSTRACT

In this paper, we propose a new algorithm for computing real roots of polynomial equations or a subset of real roots in a given semi-algebraic set described by additional polynomial inequalities. The algorithm is based on using modified fixed point continuation method for solving Lasserre's hierarchy of moment relaxations. We establish convergence properties for our algorithm. For a large-scale polynomial system with only few real solutions in a given area, we can extract them quickly. Moreover, for a polynomial system with an infinite number of real solutions, our algorithm can also be used to find some isolated real solutions or real solutions on the manifolds.

Categories and Subject Descriptors: I.1.2 [Symbolic and Algebraic Manipulation]: Algorithms; G.1.6 [Numerical Analysis]: Global optimization

General Terms: algorithms, experimentation

Keywords: low-rank moment matrix completion, nuclear norm minimization, alternating direction method, fixed point continuation method, semi-algebraic set.

1. INTRODUCTION

There is a large literature on the problem of finding all solutions of a system of polynomial equations,

$$\{g_1(x) = 0, \ldots, g_{s_1}(x) = 0\}, \tag{1}$$

where $g_i \in \mathbb{R}[x] = \mathbb{R}[x_1, \ldots, x_n]$ for $i = 1, \ldots, s_1$, see, e.g., [5, 16, 35, 52, 53, 55] and excellent softwares, e.g., PHCpack developed by Verschelde [59]. In many practical applications, one is only interested in real solutions or solutions satisfying some additional inequality constraints,

$$\{g_{s_1+1}(x) \geq 0, \ldots, g_{s_2}(x) \geq 0\}, \tag{2}$$

where $g_i \in \mathbb{R}[x]$ for all $i = s_1 + 1, \ldots, s_2$. There are symbolic and numeric algorithms including interval methods, subdivision methods, homotopy continuations for finding only real solutions of multivariate polynomials, see, e.g., [6, 42, 46]. Some efficient and practical algorithms based on critical point method are proposed in [1, 2, 4, 5, 51] to compute one point on each semi-algebraically connected component of a real algebraic variety.

Recently, there is also an arising interest in using numerical semidefinite programming (SDP) based method, e.g., Lasserre et al. [24, 31, 32, 33, 34] and Chesi et al. [12, 13, 14] for characterizing and computing the real solutions of polynomial systems. As pointed out in [32], the great benefit of using SDP techniques is that it exploits the real algebraic nature of the problem right from the beginning and avoids the computation of complex components. For example, the moment-matrix algorithms in [32, 33, 34] solve a sequence of SDP problems

$$\begin{cases} \min & 1 \\ \text{s. t.} & y_0 = 1, \\ & M_t(y) \succeq 0, \\ & M_{t-d_j}(g_j\ y) = 0, \quad j = 1, \ldots, s_1, \\ & M_{t-d_j}(g_j\ y) \succeq 0, \quad j = s_1 + 1, \ldots, s_2, \end{cases} \tag{3}$$

where $d_j := \lceil \deg(g_j)/2 \rceil, j = 1, \ldots, s_2$.

Theorem 1 *[32, Proposition 4.5, 4.6] If there is only a finite number of real solutions of (1), for t large enough, one can find an optimal solution y^* of (3) satisfying the rank condition*

$$\operatorname{rank}\ M_k(y^*) = \operatorname{rank}\ M_{k-d}(y^*), \tag{4}$$

where $d = \max_{1 \leq j \leq s_2} d_j$ and $d \leq k \leq t$. If $M_t(y^*)$ has maximum rank, then the number of real solutions satisfying (1) and (2) is equal to $\operatorname{rank}(M_k(y^*))$ and they all can be extracted by computing formal multiplication matrices from $M_k(y^*)$ and a basis of the column space of $M_{k-1}(y^*)$.

The moment relaxations (3) can be solved efficiently by interior-point type algorithms. If the dimension of moment matrix is not too large, then interior-point SDP solvers, such as SeDuMi [54] or SDPT3 [56], will quickly find a moment matrix solving (3) and having maximum rank. However, it can prove to be quite challenging when the dimensions of the matrix $m > 1000$ and the number of constraints $p > 6000$ because the computational cost grows like $\mathcal{O}(pm^3 + p^2m^2 + p^3)$. Although the maximal rank property of most interior-point algorithms can guarantee us to find all real solutions of polynomial systems (1) and (2), the rank condition (4) may

*This material is based on work supported by a NKBRPC 2011CB302400, the Chinese National Natural Science Foundation under Grants 91118001, 60821002/F02, 60911130369 and 10871194.

be only satisfied for sufficient large order t, which results a large-scale SDP problem since the dimension of the moment matrix at order t is $m = \binom{n+t}{t}$ and the number of linear equality constraints is $p = \sum_{j=1}^{s_1} \frac{1}{2}\binom{n+t-d_j}{n}\left(\binom{n+t-d_j}{n}+1\right)$, both of them increase very fast with t. Furthermore, if there are an infinite number of real solutions, then the rank condition (4) will never be satisfied for moment matrices with maximal rank. Hence, in [24, 30], they replace the constant object function in (3) by the trace of the moment matrix and show that their software GloptiPoly is very efficient for finding a partial set of real solutions for a large set of polynomial systems [30, Table 6.3, 6.4]. We notice that the trace of a semidefinite moment matrix is equal to its nuclear norm defined as the sum of its singular values. Therefore, (3) can be transformed to the following nuclear norm minimization problem:

$$\begin{cases} \min & ||M_t(y)||_* \\ \text{s. t.} & y_0 = 1, \\ & M_t(y) \succeq 0, \\ & M_{t-d_j}(g_j\ y) = 0, \quad j = 1, \ldots, s_1, \\ & M_{t-d_j}(g_j\ y) \succeq 0, \quad j = s_1+1, \ldots, s_2. \end{cases} \tag{5}$$

The nuclear norm minimization problem has been well studied by many people, see e.g., [18, 48]. It can be directly solved by interior-point methods in [8, 9, 19, 36, 50] or projection methods in [25, 27, 40, 47, 61] for large-scale SDP problems. Since most of these methods also use second-order information, the memory requirement for computing descent directions quickly becomes too large as the problem size increases. See [30, Table 6.3, 6.4] for examples which run out of memory for a relatively small relaxation order t.

If we remove the positive semidefiniteness conditions in (5), then the problem becomes to find a symmetric matrix with minimum nuclear norm satisfying a set of linear equality constraints. Several fast algorithms using only first-order information have been developed in [10, 22, 37, 38]. Moreover, some accelerated gradient algorithms are also proposed in [7, 26, 43, 44, 45, 57, 58] which have an attractive convergence rate of $O(1/k^2)$, where k is the iteration counter. These first-order methods, based on function values and gradient evaluation, cannot yield as high accuracy as interior point methods, but much larger problems can be solved since no second-order information needs to be computed and stored.

In [39], adding back the positive semidefiniteness condition by introducing a thresholding operator based on Schur decompositions, we provide an accelerated fixed point continuation algorithm with Barzilai-Borwein technique (AFPC-BB) for solving the minimum-rank Gram matrix completion problems and giving sum-of-squares (SOS) representations of polynomials. The algorithm has been used successfully to compute exact rational SOS representations of nonnegative polynomials with millions of monomials, which correspond to solving SDP problems with millions of constrains. This motivates us to investigate how to extend AFPC-BB algorithm to find a low-rank moment matrix satisfying not only the equality constraints but also the positive semidefiniteness constraints.

Main results. We define a basic closed semi-algebraic set

$$K := \{x \in \mathbb{R}^n \mid g_1(x) = 0, \ldots, g_{s_1}(x) = 0; \\ g_{s_1+1}(x) \geq 0, \ldots, g_{s_2}(x) \geq 0\}. \tag{6}$$

In this paper, we propose a novel algorithm for computing at least one point in K. Our algorithm starts with transforming the problem into solving a sequence of nuclear norm minimization problems (5), which can be recast as moment matrix completion problems (12). Then, we extend the AFPC-BB algorithm in [39] to solve (12). Finally, suppose at the relaxation order t, we find a low-rank moment matrix whose principal submatrix $M_k(y^*)$ satisfies the rank condition (4), then real solutions can be extracted by computing formal multiplication matrices from the columns of $M_k(y^*)$ and a basis of the column space of $M_{k-1}(y^*)$.

There is no guarantee for our algorithm to find all real roots. However, if there is only one or few real roots of a large-scale polynomial system, we can extract them quickly. Moreover, when the number of real solutions in K is infinite, it is still possible for our algorithm to find some isolated real solutions or real solutions on the manifolds. Numerical experiments demonstrate that new algorithm can find real solutions quickly for some hard examples in [30, Table 6.3, 6.4].

Structure of the paper. In Section 2, we introduce some notations and recast the moment relaxations into finding symmetric positive semidefinite matrices of minimum nuclear norm subject to linear equality constraints. We propose an alternating direction method and modified fixed point iterations for solving a sequence of moment matrix completion problems. In Section 3, we establish the convergence result for the iterations given in Section 2. In Section 4, we present an algorithm and demonstrate the effectiveness of the algorithm for computing real roots of a set of benchmark systems.

2. COMPUTING REAL ROOTS OF POLYNOMIAL SYSTEMS

Let $\mathbb{N}$ denote the set of nonnegative integers and we set $\mathbb{N}_t^n := \{\alpha \in \mathbb{N}^n \mid |\alpha| := \Sigma_{i=1}^n \alpha_i \leq t\}$ for $t \in \mathbb{N}$. For $\alpha \in \mathbb{N}^n$, x^α denotes the monomial $x_1^{\alpha_1} \cdots x_n^{\alpha_n}$. Let $y_\alpha := \int x^\alpha d\mu$ for a given Borel measure μ on $\mathbb{R}^n$, then the sequence $y = (y_\alpha)_{\alpha \in \mathbb{N}^n}$ is called *the sequence of moments of the measure* μ.

Given a sequence $y = (y_\alpha)_{\alpha \in \mathbb{N}^n} \in \mathbb{R}^{\mathbb{N}^n}$, its *real moment matrix* is defined as

$$M(y) := (y_{\alpha+\beta})_{\alpha,\beta \in \mathbb{N}^n}.$$

Similarly, given a truncated sequence $y = (y_\alpha)_{\alpha \in \mathbb{N}_{2t}^n} \in \mathbb{R}^{\mathbb{N}_{2t}^n}$ for an integer $t \geq 1$, its truncated *moment matrix of order* t is the matrix $M_t(y)$ with (α, β)th entry $y_{\alpha+\beta}$, for $\alpha, \beta \in \mathbb{N}_t^n$. For a given monomial basis $(x^\alpha)_{\alpha \in \mathbb{N}^n}$, we define the matrix B_α as follows (see [29])

$$B_\alpha(\zeta, \eta) := \begin{cases} 1, & \text{if } \zeta + \eta = \alpha; \\ 0, & \text{otherwise.} \end{cases}$$

Then the truncated moment matrix can be written as

$$M_t(y) = \Sigma_{\alpha \in \mathbb{N}_t^n} B_\alpha y_\alpha.$$

Let $m = \binom{n+t}{t}$ be the dimension of the moment matrix $M_t(y)$. Suppose $g_j(x) = \Sigma_{\alpha \in \mathbb{N}^n} g_{j,\alpha} x^\alpha \in \mathbb{R}[x]$ with finitely many nonzero coefficients $g_{j,\alpha} \in \mathbb{R}$. If the (k,l)th entry of $M_t(y)$ is y_β, then the (k,l)th entry of the $(t-d_j)$th *localizing matrix* $M_{t-d_j}(g_j y)$ with dimension $m_j = \binom{n+t-d_j}{t-d_j}$ is defined by

$$M_{t-d_j}(g_jy)(k,l) := \Sigma_\alpha\ g_{j,\alpha}y_{\alpha+\beta}.$$

For $j = 1, \ldots, s_1, 1 \leq k \leq m_j, k \leq l \leq m_j$, we define the matrix

$$A^{(j)}_{(k,l)} := \Sigma_\alpha\ g_{j,\alpha}B_{\alpha+\beta}, \quad b^{(j)}_{(k,l)} := g_{j,0},$$

and reorder $A^{(j)}_{(k,l)}$, $b^{(j)}_{(k,l)}$ as $A_1, \ldots, A_p$ and $b_1, \ldots, b_p$ respectively, where $p = \sum_{j=1}^{s_1}(m_j^2 + m_j)/2$. We define the linear operator $\mathcal{A} : \mathbb{S}^m \to \mathbb{R}^p$ as

$$\mathcal{A}(M_t(y)) := (\langle A_1, M_t(y)\rangle, \ldots, \langle A_p, M_t(y)\rangle)^T, \tag{7}$$

where $\mathbb{S}^m$ denotes the set of semidefinite matrices and the inner product $\langle A_i, M_t(y)\rangle := \mathrm{Tr}(A_i^T M_t(y))$.

All of equality constraints $M_{t-d_j}(g_jy) = 0, j = 1, \ldots, s_1$ in (5) can be recast by one formula:

$$\mathcal{A}(M_t(y)) = b, \tag{8}$$

where $b = (b_1, \ldots, b_p)^T$. For $z \in \mathbb{R}^p$, the adjoint operator $\mathcal{A}^* : \mathbb{R}^p \longrightarrow \mathbb{S}^m$ is defined as

$$\mathcal{A}^*(z) := A_1z_1 + \cdots + A_pz_p. \tag{9}$$

Remark 1 *In view of the memory requirement and computational efficiency, in our implementation, the action of the linear operator $\mathcal{A}$ on $M_t(y)$ is formulated in a more suitable way. Note that the moment matrix $M_t(y)$ is symmetric, we stack only columns of the upper triangular part of $M_t(y)$ in a single vector, denoted as $\mathbf{svec}(M_t(y)) \in \mathbb{R}^{(m^2+m)/2}$. We transform symmetric matrices $A_i \in \mathbb{R}^{m\times m}$ in the same way to vectors $\mathbf{svec}(A_i) \in \mathbb{R}^{(m^2+m)/2}$ for $i = 1, \ldots, p$. Then (8) is equivalent to*

$$A\ \mathbf{svec}(M_t(y)) = b,$$

where

$$A = \begin{pmatrix} \mathbf{svec}(A_1)^T \\ \vdots \\ \mathbf{svec}(A_p)^T \end{pmatrix}.$$

The matrix A is super sparse, in our implementation, we only store nonzero entries of A and their locations (i, j).

For the positive semidefiniteness constraints $M_{t-d_j}(g_jy) \succeq 0$, $j = s_1+1, \ldots, s_2$, we convert them to equality constraints by introducing slack matrix variables $Z_j \in \mathbb{S}^{m_j}$ satisfying

$$M_{t-d_j}(g_jy) = Z_j, \quad Z_j \succeq 0. \tag{10}$$

Following the definition of the operator $\mathcal{A}$, each equality constraint in (10) can be written as

$$\mathcal{C}_j(M_t(y)) = \mathbf{svec}(Z_j),\ j = s_1 + 1, \ldots, s_2, \tag{11}$$

where $\mathcal{C}_j : \mathbb{S}^m \to \mathbb{R}^{(m_j^2+m_j)/2}$, $j = s_1 + 1, \ldots, s_2$.

The above transformations give rise to a reformulation of the semidefinite relaxation (5) to a moment matrix completion problem

$$\begin{cases} \min & ||X||_* \\ \text{s. t.} & \mathcal{A}(X) = b, \\ & X = X^T, X \succeq 0, \\ & \mathcal{C}_j(X) = \mathbf{svec}(Z_j), \quad j = s_1 + 1, \ldots, s_2, \\ & Z_j = Z_j^T, Z_j \succeq 0, \quad j = s_1 + 1, \ldots, s_2. \end{cases} \tag{12}$$

The linear equality constraints can be relaxed, resulting in a Lagrangian version regularized minimization problem:

$$\min_{X\in\mathbb{S}^m_+, Z_j\in\mathbb{S}^{m_j}_+, j=s_1+1,\ldots,s_2} \mu\|X\|_* + \frac{1}{2}\|\mathcal{A}(X) - b\|_2^2 + \frac{1}{2}\sum_{j=s_1+1}^{s_2} \|\mathcal{C}_j(X) - \mathbf{svec}(Z_j)\|_2^2, \tag{13}$$

where $\mathbb{S}^m_+, \mathbb{S}^{m_j}_+$ denote the sets of symmetric positive semidefinite matrices and $\mu > 0$ is a parameter.

We minimize the objective function in (13) with respect to X and $Z_j, j = s_1 + 1, \ldots, s_2$ alternatively. The alternating direction method is an effective approach for solving the linearly constrained convex programming problem with a separate structure whose objective function is in the form of the sum of individual functions without crossed variables. It was first considered by Gabay [20] and Gabay and Mercier [21], see also [11, 17, 40, 60] for some recent results. For a fixed $X = \hat{X} \in \mathbb{S}^m_+$, the first two terms in (13) are constant and all of variables Z_j are separated, the optimal solution of (13) can be obtained by solving

$$\min_{Z_j\in\mathbb{S}^{m_j}_+} \|\mathcal{C}_j(\hat{X}) - \mathbf{svec}(Z_j)\|_2^2, \tag{14}$$

for $j = s_1 + 1, \ldots, s_2$.

Let $Y = \Sigma_i\lambda_iq_iq_i^T$ be the spectral decomposition of a symmetric matrix $Y \in \mathbb{S}^m$ with eigenvalues λ_i and orthogonal eigenvectors q_i. Then the projection of $Y \in \mathbb{S}^m$ onto the positive semidefinite cone $\mathbb{S}^m_+$ and its polar cone $\mathbb{S}^m_-$ are denoted, respectively, by

$$Y_+ = \sum_{\lambda_i>0}\lambda_iq_iq_i^T, \quad Y_- = \sum_{\lambda_i<0}\lambda_iq_iq_i^T. \tag{15}$$

Obviously, we have

$$Y = Y_+ + Y_-. \tag{16}$$

Therefore, the unique minimizer of (14) is given by the following projection

$$\hat{Z}_j = \mathbf{smat}(\mathcal{C}_j(\hat{X}))_+, \tag{17}$$

where **smat** maps a vector back to a symmetric matrix. It is an inverse function of **svec**.

On the other hand, for $Z_j = \hat{Z}_j$, $j = s_1 + 1, \ldots, s_2$, the problem (13) can be written as

$$\min_{X\in\mathbb{S}^m_+} \mu\|X\|_* + \frac{1}{2}\|\mathcal{A}(X) - b\|_2^2 + \frac{1}{2}\sum_{j=s_1+1}^{s_2}\|\mathcal{C}_j(X) - \mathbf{svec}(\hat{Z}_j)\|_2^2. \tag{18}$$

We can use AFPC-BB algorithm introduced in [39] to solve this problem.

Definition 1 *[39] Suppose $W = Q\Lambda Q^T$ is a Schur decomposition of a matrix $W \in \mathbb{S}^m$, where $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_m)$ and Q is a real orthogonal matrix. For any $\nu \geq 0$, the matrix thresholding operator $\mathcal{T}_\nu(\cdot)$ is defined as*

$$\mathcal{T}_\nu(W) := Q\mathcal{T}_\nu(\Lambda)Q^T,$$

where

$$\mathcal{T}_\nu(\Lambda) = \mathrm{diag}(\{\lambda_1 - \nu\}_+, \ldots, \{\lambda_m - \nu\}_+),$$

and $\{t\}_+ := \max(0, t)$.

For a given $\tau > 0$, we define the operator

$$h(X, Z_{s_1+1}, \dots, Z_{s_2}) := X - \tau \mathcal{A}^*(\mathcal{A}(X) - b) - \Sigma_{j=s_1+1}^{s_2} \tau \mathcal{C}_j^*(\mathcal{C}_j(X) - \mathbf{svec}(Z_j)). \quad (19)$$

Let μ be a positive real number and X_0 be an initial starting matrix. Our iterative shrinkage procedure for solving problem (18) is given as follows, for $k = 0, 1, 2, \dots$

$$X^{k+1} = \mathcal{T}_{\tau\mu}(h(X^k, \hat{Z}_{s_1+1}, \dots, \hat{Z}_{s_2})). \quad (20)$$

It has been proved in [39] that under some mild assumptions on the operators $\mathcal{A}$ and $\mathcal{C}_j$, $j = s_1 + 1, \dots, s_2$, the sequence $\{X^k\}$ obtained by (20) converges to the unique optimal solution of problem (18).

Let us fix the number of iterations (20) to be one and define an iteration as follows

$$\begin{cases} Z_j^{k+1} = & \mathbf{smat}(\mathcal{C}_j(X^k))_+, \quad j = s_1 + 1, \dots, s_2, \\ X^{k+1} = & \mathcal{T}_{\tau\mu}(h(X^k, Z_{s_1+1}^{k+1}, \dots, Z_{s_2}^{k+1})). \end{cases} \quad (21)$$

Theorem 2 *Given a small positive number μ, suppose $X^* \in \mathbb{S}_+^m$ and $Z_j^* \in \mathbb{S}_+^{m_j}$, $j = s_1 + 1, \dots, s_2$ satisfy*

$$\begin{cases} Z_j^* = \mathbf{smat}(\mathcal{C}_j(X^*))_+, \quad j = s_1 + 1, \dots, s_2, \\ X^* = \mathcal{T}_{\tau\mu}(h(X^*, Z_{s_1+1}^*, \dots, Z_{s_2}^*)), \end{cases} \quad (22)$$

and

$$\|\mathcal{A}(X^*) - b\|_2^2 + \sum_{j=s_1+1}^{s_2} \|\mathcal{C}_j(X^*) - \mathbf{svec}(Z_j^*)\|_2^2 < \frac{\mu^2}{m \max_{1 \le j \le s_2} \|g_j\|_2^2}. \quad (23)$$

Then $(X^*, Z_{s_1+1}^*, \dots, Z_{s_2}^*)$ is the unique optimal solution of the problem (13).

Before we prove Theorem 2, let us recall the definition of the subgradient of the nuclear norm at a symmetric matrix $X \in \mathbb{S}^m$ (see [39, Theorem 1]):

$$\partial\|X\|_* = \{Q_1 Q_1^T - Q_2 Q_2^T + Z : Q_i^T Z = 0, i = 1, 2, \text{ and } \|Z\|_2 \le 1\}, \quad (24)$$

where Q_1 and Q_2 are orthogonal eigenvectors associated with the positive and negative eigenvalues of X, respectively.

Proof. Given an $X^* \in \mathbb{S}_+^m$, as in (17), the minimizer of the problem (13) with respect to Z_j is

$$\mathbf{smat}(\mathcal{C}_j(X^*))_+, \quad j = s_1 + 1, \dots, s_2.$$

Given $Z_j^* \in \mathbb{S}_+^{m_j}, j = s_1 + 1, \dots, s_2$, since the objective function in (13) is strictly convex, there exists a unique minimizer. Let $\nu = \tau\mu$ and

$$Y^* = h(X^*, Z_{s_1+1}^*, \dots, Z_{s_2}^*) = X^* + E \in \mathbb{S}^m,$$

where

$$E = -\tau \mathcal{A}^*(\mathcal{A}(X^*) - b) - \sum_{j=s_1+1}^{s_2} \tau \mathcal{C}_j^*(\mathcal{C}_j(X^*) - \mathbf{svec}(Z_j^*)).$$

Without loss of generality, we assume that the eigenvalues of Y^* can be ordered as

$$\lambda_1(Y^*) \ge \dots \ge \lambda_{k_1}(Y^*) \ge \nu > \lambda_{k_1+1}(Y^*) \ge \dots > 0 > \dots \ge \lambda_k(Y^*), \lambda_{k+1}(Y^*) = \dots = \lambda_m(Y^*) = 0.$$

We compute a Schur decomposition of Y^* as

$$Y^* = Q_1 \Lambda_1 Q_1^T + Q_2 \Lambda_2 Q_2^T,$$

where $\Lambda_1 = \mathrm{diag}(\lambda_1, \dots, \lambda_{k_1})$, $\Lambda_2 = \mathrm{diag}(\lambda_{k_1+1}, \dots, \lambda_k)$, Q_1 and Q_2 are block matrices corresponding to Λ_1 and Λ_2 respectively. Then we have

$$\mathcal{T}_\nu(Y^*) = Q_1(\Lambda_1 - \nu I)Q_1^T,$$

and

$$Y^* - \mathcal{T}_\nu(Y^*) = \nu(Q_1 Q_1^T + Z), \quad Z = \nu^{-1} Q_2 \Lambda_2 Q_2^T.$$

By definition, $Q_1^T Z = 0$.

- If $\lambda_{k_1+1}(Y^*) \ge |\lambda_k(Y^*)|$, then $\|Z\|_2 = \lambda_{k_1+1}(Y^*)/\nu < 1$.
- Otherwise, we have

$$\begin{aligned} \|E\|_F^2 \le & \tau^2 \|\mathcal{A}^*(\mathcal{A}(X^*) - b)\|_F \\ & + \tau^2 \sum_{j=s_1+1}^{s_2} \|\mathcal{C}_j^*(\mathcal{C}_j(X^*) - \mathbf{svec}(Z_j^*)\|_F^2 \\ \le & \tau^2 m \max_{1 \le j \le s_2} \|g_j\|_2^2 \big(\|\mathcal{A}(X^*) - b\|_2^2 \\ & + \sum_{j=s_1+1}^{s_2} \|\mathcal{C}_j(X^*) - \mathbf{svec}(Z_j^*)\|_2^2\big) \\ < & \tau^2 \mu^2. \end{aligned}$$

Notice that $E \in \mathbb{S}^m$ and $W^* \in \mathbb{S}_+^m$, by [23, Theorem 8.1.5], we have

$$\|Z\|_2 = \frac{|\lambda_k(Y^*)|}{\nu} = \frac{\max\{|\lambda_1(E)|, |\lambda_m(E)|\}}{\nu} \le \frac{\|E\|_F}{\nu} < 1.$$

Hence, according to (24), we have $Y^* - \mathcal{T}_\nu(Y^*) \in \nu \partial \|\mathcal{T}_\nu(Y^*)\|_*$, which means that $0 \in \nu \partial \|\mathcal{T}_\nu(Y^*)\|_* + \mathcal{T}_\nu(Y^*) - Y^*$. Therefore, we have

$$0 \in \mu \partial \|X^*\|_* - \mathcal{A}^*(\mathcal{A}(X^*) - b) - \sum_{j=s_1+1}^{s_2} \mathcal{C}_j^*(\mathcal{C}_j(X^*) - \mathbf{svec}(Z_j^*)).$$

Hence, $(X^*, \mathbf{smat}(\mathcal{C}_{s_1+1}(X^*))_+, \dots, \mathbf{smat}(\mathcal{C}_{s_2}(X^*))_+)$ is an optimal solution of the problem (13). $\square$

3. CONVERGENCE ANALYSIS

In this section, we analyze the convergence properties of iterations defined in (21). The nonexpansive property of the thresholding operator $\mathcal{T}_\nu$ is given in [39], i.e. for any $X_1, X_2 \in \mathbb{S}^m$,

$$\|\mathcal{T}_\nu(X_1) - \mathcal{T}_\nu(X_2)\|_F \le \|X_1 - X_2\|_F.$$

Moreover,

$$\begin{aligned} & \|X_1 - X_2\|_F = \|\mathcal{T}_\nu(X_1) - \mathcal{T}_\nu(X_2)\|_F \\ \iff & X_1 - X_2 = \mathcal{T}_\nu(X_1) - \mathcal{T}_\nu(X_2). \end{aligned}$$

Let us set $Z_j = \mathbf{smat}(\mathcal{C}_j(X))_+, \quad j = s_1 + 1, \dots, s_2$, and still use h to denote the function in X only:

$$h(X) := h\left(X, \mathbf{smat}(\mathcal{C}_{s_1+1}(X))_+, \dots, \mathbf{smat}(\mathcal{C}_{s_2}(X))_+\right). \quad (25)$$

The following lemma shows that the operator $h(\cdot)$ is nonexpansive.

Lemma 1 *Suppose that the step size $\tau \in (a,b)$ where*

$$\begin{cases} a = \frac{1}{2\left(\|\mathcal{A}\|_2^2+\sum_{j=s_1+1}^{s_2}\|\mathcal{C}_j\|_2^2\right)}, \\ b = \min\left(3a, \frac{1}{2\sum_{j=s_1+1}^{s_2}\|\mathcal{C}_j\|_2^2}\right). \end{cases} \quad (26)$$

Then $h(\cdot)$ defined in (25) is non-expansive, i.e., for any $X_1, X_2 \in \mathbb{S}^m$,

$$\|h(X_1) - h(X_2)\|_F \le \|X_1 - X_2\|_F.$$

Moreover, we have

$$\begin{aligned} &\|h(X_1) - h(X_2)\|_F = \|X_1 - X_2\|_F \\ &\Longleftrightarrow h(X_1) - h(X_2) = X_1 - X_2. \end{aligned}$$

Proof. According to (16), we have

$$\begin{aligned} &\|X_1 - X_2\|_F^2 - \|X_{1+} - X_{2+}\|_F^2 \\ &= \langle X_1 - X_2, X_1 - X_2\rangle - \langle X_{1+} - X_{2+}, X_{1+} - X_{2+}\rangle \\ &= \|X_{1-} - X_{2-}\|_F^2 + 2\mathrm{Tr}(X_{1+}^T X_{2-} + X_{1-}^T X_{2+}) \\ &\ge 0. \end{aligned}$$

Thus, for any $j = s_1+1, \ldots, s_2$,

$$\begin{aligned} \|Z_{1,j} - Z_{2,j}\|_F &= \|\mathbf{smat}(\mathcal{C}_j(X_1))_+ - \mathbf{smat}(\mathcal{C}_j(X_2))_+\|_F \\ &\le \|\mathbf{smat}(C_j(X_1)) - \mathbf{smat}(C_j(X_2))\|_F \\ &\le \|\mathcal{C}_j\|_2\ \|X_1 - X_2\|_F. \end{aligned} \quad (27)$$

We obtain

$$\begin{aligned} &\|h(X_1) - h(X_2)\|_F \\ &\le \|I - \tau\mathcal{A}^*\mathcal{A} - \sum_{j=s_1+1}^{s_2} \tau\mathcal{C}_j^*\mathcal{C}_j\|_2\ \|X_1 - X_2\|_F \\ &\quad + \sum_{j=s_1+1}^{s_2} \tau\|\mathcal{C}_j\|_2\ \|Z_{1,j} - Z_{2,j}\|_F \\ &\le \left(\|I - \tau\mathcal{A}^*\mathcal{A} - \sum_{j=s_1+1}^{s_2}\tau\mathcal{C}_j^*\mathcal{C}_j\|_2 + \sum_{j=s_1+1}^{s_2}\tau\|\mathcal{C}_j\|_2^2\right) \\ &\quad \|X_1 - X_2\|_F. \end{aligned}$$

For a $\tau \in (a,b)$ with a, b defined by (26), we have

$$\|I - \tau\mathcal{A}^*\mathcal{A} - \sum_{j=s_1+1}^{s_2}\tau\mathcal{C}_j^*\mathcal{C}_j\|_2 + \sum_{j=s_1+1}^{s_2}\tau\|\mathcal{C}_j\|_2^2 \le 1.$$

Therefore, $\|h(X_1) - h(X_2)\|_F \le \|X_1 - X_2\|_F$. □

We now claim that the iterations defined in (21) converge to the optimal solution of the problem (13).

Theorem 3 *Suppose that the step size $\tau \in (a,b)$ with a, b defined by (26). Then a sequence of solutions*

$$(X^k, Z_{s_1+1}^k, \ldots, Z_{s_2}^k)$$

generated by (21) converges to the unique optimal solution

$$(X^*, Z_{s_1+1}^*, \ldots, Z_{s_2}^*)$$

of the problem (13) satisfying conditions (22) and (23).

Proof. The proof uses the same reasoning as in [39, Theorem 4]. Let $\nu = \tau\mu$. Since both $\mathcal{T}_\nu(\cdot)$ and $h(\cdot)$ are non-expansive, $\mathcal{T}_\nu(h(\cdot))$ is also non-expansive. Therefore, $\{X^k\}$ lies in a compact set and have a limit.

Suppose $\widetilde{X} = \lim_{j\longrightarrow\infty} X^{k_j}$ satisfying the condition (23) in Theorem 2. Since $X^* = \mathcal{T}_\nu(h(X^*))$, we have

$$\begin{aligned} \|X^{k+1} - X^*\|_F &= \|\mathcal{T}_\nu(h(X^k)) - \mathcal{T}_\nu(h(X^*))\|_F \\ &\le \|h(X^k) - h(X^*)\|_F \le \|X^k - X^*\|_F. \end{aligned}$$

Thus, the sequence $\{\|X^k - X^*\|_F\}$ is monotonically non-increasing and converges to $\|\widetilde{X} - X^*\|_F$. Moreover, the function $\mathcal{T}_\nu(h(\cdot))$ is continuous, we have

$$\mathcal{T}_\nu(h(\widetilde{X})) = \lim_{j\longrightarrow\infty}\mathcal{T}_\nu(h(X^{k_j})) = \lim_{j\longrightarrow\infty} X^{k_j+1},$$

which means that $\mathcal{T}_\nu(h(\widetilde{X}))$ is also a limit of $\{X^k\}$. Therefore, we have

$$\begin{aligned} \|\mathcal{T}_\nu(h(\widetilde{X})) - \mathcal{T}_\nu(h(X^*))\|_F &= \|\mathcal{T}_\nu(h(\widetilde{X})) - X^*\|_F \\ &= \|\widetilde{X} - X^*\|_F. \end{aligned}$$

Using Lemma 1 we obtain

$$\mathcal{T}_\nu(h(\widetilde{X})) - \mathcal{T}_\nu(h(X^*)) = h(\widetilde{X}) - h(X^*) = \widetilde{X} - X^*,$$

which implies $\mathcal{T}_\nu(h(\widetilde{X})) = \widetilde{X}$. By Theorem 2, $\widetilde{X}$ is the optimal solution to the problem (13), i.e., $\widetilde{X} = X^*$. Hence, we have

$$\lim_{k\longrightarrow\infty}\|X^k - X^*\|_F = 0,$$

i.e., $\{X^k\}$ converges to its unique limit point X^*. By (27), we have

$$\|Z_j^k - Z_j^*\|_F \le \|\mathcal{C}_j\|_2\|X^k - X^*\|_F.$$

Therefore,

$$\lim_{k\longrightarrow\infty}\|Z_j^k - Z_j^*\|_F = 0,$$

for $j = s_1+1, \ldots, s_2$, i.e., $(Z_{s_1+1}^k, \ldots, Z_{s_2}^k)$ converges to its unique limit point $(Z_{s_1+1}^*, \ldots, Z_{s_2}^*)$. □

4. IMPLEMENTATION AND NUMERICAL EXPERIMENTS

In this section we give a more detailed exposition of the proposed moment matrix completion algorithm for finding at least one point in the semi-algebraic set K defined by (6).

4.1 Algorithm

The following algorithm computes real solutions of a system of polynomial equations and inequations.

Algorithm *MMCRSolver*

Input: ▸ Polynomials $g_1, \ldots, g_{s_1}, g_{s_1+1}, \ldots, g_{s_2} \in \mathbb{R}[x]$, $\mu_1 > \bar{\mu} > 0$, an integer $L > 0$.

Output: ▸ Real solutions x in the semi-algebraic set K.

1. Set $t = \max_{1\le j\le s_2}\lceil\deg(g_j)/2\rceil$, $a_0 = 1$, X^0 is a matrix with only one nonzero entry $X^0(1,1) = 1$.
2. For the relaxation order t, compute operators $\mathcal{A}$ and $\mathcal{C}_j$, $j = s_1+1, \ldots, s_2$.
3. For $\mu = \mu_1, \ldots, \mu_L$, do
   (a) choose a step size τ_k via the BB technique;
   (b) compute $Y^k = X^k + \frac{a_{k-1}-1}{a_k}(X^k - X^{k-1})$;

(c) compute $Z_j^{k+1} = \mathbf{smat}(\mathcal{C}_j(Y^k))_+$, $j = s_1+1, \dots, s_2$;

(d) compute $X^{k+1} = \mathcal{T}_{\tau_k\mu_k}(Y^k - \tau_k\mathcal{A}^*(\mathcal{A}Y^k - b) - \sum_{j=s_1+1}^{s_2} \tau_k\mathcal{C}_j^*(\mathcal{C}_j(Y^k) - \mathbf{svec}(Z_j^{k+1})))$;

(e) compute $a_{k+1} = \frac{1+\sqrt{1+4a_k^2}}{2}$;

(f) if the stop criterions (30) or (31) are true, then return a solution X_{opt}.

4. If the condition (4) holds for X_{opt} then

(a) compute multiplication matrices by (28);

(b) return real solutions extracted by (29).

5. Otherwise, replace t by $t+1$ and go back to step 2.

For the inner loop step 3, we incorporate an accelerating technique used in the AFPC-BB algorithm [39] to our alternating minimization scheme (21) for solving (13). They rely on computing the next iterate X^{k+1} based not only on the previous one X^k, but on the combination of the two previously computed iterates X^k and X^{k-1}. The main computational cost of step 3 is computing the Schur decompositions. Since the moment matrices returned from MMCRSolver usually have low rank. We are hence interested in numerical methods for computing the dominant singular values and singular vectors. Following the strategies in [10, 39], we use PROPACK [28] in Matlab to compute a partial Schur decomposition of a symmetric matrix. The number of singular values computed by PROPACK will directly affect the number of real solutions obtained. If we aim to find only very few number of real roots, then PROPACK is very efficient for computing the first few dominant singular values. This can be seen clearly from the example "puma" in Table 1. We also adopt a continuation strategy to accelerate the convergence. For the problem (13) with a target parameter $\bar{\mu}$ being a moderately small number, we solve a sequence of problems (13) defined by a decreasing sequence μ_k. The parameter η determines the rate of reduction of the consecutive μ_k, i.e.,

$$\mu_{k+1} = \max(\eta\mu_k, \bar{\mu}), \quad k = 1, \dots, L-1.$$

From Theorem 3 the convergence of the inner loop is guaranteed provided that $\tau \in (a, b)$. However, this choice is too conservative and the convergence is typically slow. In our experiments, we use the Barzilai-Borwein technique to chose the step size τ_k (see [3, 39]).

In step 4, in order to check the rank condition (4), we compute ranks of the principal submatrices $M_k(y)$ of $M_t(y)$ for $k \le t$ by the singular value decomposition (SVD). Following [24, 32], the approximate rank of a matrix is defined as the number of singular values bigger than a fixed tolerance 10^{-8} or with a decay of more than 10^{-3} lying between two singular values. If the truncated moment matrix $M_k(y)$ satisfies (4), following [49], we compute the SVD of $M_{k-1}(y) = U\Sigma V^T$, where U and V are unitary matrices and Σ is a diagonal matrix whose diagonal entries are real decreasing non-negative numbers. The set $\{u_1, \dots, u_r\}$ of columns of U corresponding to the nonzero diagonal entries of Σ forms an orthogonal basis of the column space of $M_{k-1}(y)$. For $j = 1, \dots, r$, let

$$b_j = u_j^T (x^\alpha)_{\alpha\in\mathbb{N}_{k-1}^n},$$

then we get a polynomial set $\mathcal{B} = \{b_1, \dots, b_r\}$. The multiplication matrices of x_j with respect to $\mathcal{B}$ can be formed stably as

$$M_{x_j} = U_r^T \cdot N_{x_j} \cdot V^T \cdot S, \tag{28}$$

where $U_r = (u_1, \dots, u_r)$, N_{x_j} are rows of $M_k(y)$ corresponding to $x_j\mathcal{B}$ respectively, and S is a diagonal matrix with elements which are reciprocals of the first r elements of Σ. The matrix S is well-conditioned since all elements are bounded by the reciprocal of the fixed tolerance.

Finally, the solutions can be obtained by computing common eigenvalues of the multiplication matrices M_{x_j}, $j = 1, \dots, n$. Following [15, 24], we build a random combination of multiplication matrices

$$M' = \sum_{j=1}^{n} \omega_j M_{x_j},$$

where $\omega_j \ge 0$ and $\Sigma_{j=1}^n \omega_j = 1$ and compute a Schur decomposition

$$M' = QRQ^T,$$

where $Q = (q_1, \dots, q_r)$ is an orthogonal matrix and R is an upper triangular matrix with eigenvalues in the diagonal. Then, we extract r real solutions:

$$(q_j^T M_{x_1} q_j, \dots, q_j^T M_{x_n} q_j), \quad j = 1, \dots, r. \tag{29}$$

If the precision of an extracted real solutions is not enough, then we use Newton's method to refine it.

Remark 2 *If the polynomial system (1) has a finite number of real solutions, as shown in Theorem 1, the condition (4) holds for a large enough relaxation order t. Therefore the algorithm terminates in a finite number of steps. For positive-dimensional system, without asking the computed moment matrix having maximal rank, MMCRSolver may still find some isolated real solutions or real solutions on the manifolds.*

4.2 Numerical experiments

In this section we illustrate the effectiveness of MMCRSolver for finding at least one real solutions of polynomial systems taken from literature. The stopping criterions for the inner loop (step 3) of MMCRSolver in our numerical experiments are given as follows:

$$\frac{\|\mathcal{A}(X_{\text{opt}}) - b\|_2}{\|b\|_2} < 0.005, \tag{30}$$

or when

$$\frac{\|X^{k+1} - X^k\|_F}{\max(1, \|X^k\|_F)} < 10^{-4}. \tag{31}$$

The experiments are carried out by running MMCRSolver in MATLAB (Version 7.7.0.471) on a desktop computer with an Intel(R) Core(TM) i3-2100 CPU @ 3.10GHz and 2.00 GB of RAM. The codes can be downloaded from http://www.mmrc.iss.ac.cn/~lzhi/Research/hybrid/MMCRSolver.

Table 1 reports the performance of MMCRSolver on a series of benchmarks. We show the number of variables and the maximal degree of the polynomial system in the second and third columns. The moment relaxation order and the number of constraints are listed in column 4 and 5. We also show the CPU time in column 6 for computing the low-rank moment matrix satisfying the rank condition (4) and

Table 1: CPU times for finding real solutions by MMCRSolver and GloptiPoly

problem	var	deg	t	p	CPU	sol	CPU	sol
boon	6	4	4	21841	31.75	8	1220	8
eco8	8	3	3	11953	1.37	1	1310	1
heart	8	4	3	12853	53.09	2	1532	2
puma	8	2	3	14653	3.96	4	1136	4
puma	8	2	3	14653	6.61	13	1136	4
butcher	7	4	4	51877	214.38	1	-	-
d1	12	3	3	103559	76.55	4	-	-
kin1	12	2	3	103559	94.71	11	-	-
reimer5	5	6	6	107267	128.70	1	-	-

extracting real solutions. The number of solutions successfully extracted by MMCRSolver is shown in column 7. The last two columns are taken from Table 6.3 and 6.4 in [30] which show the CPU time for using the software GloptiPoly to compute real solutions and the number of extracted solutions.

As can be seen from this table, for the first four examples, we can extract the same number of real solutions in much less time than GloptiPoly. The example "puma" has 16 real solutions [41]. Although we have not yet been able to find all of them by MMCRSolver, 13 real solutions can be computed by our algorithm in a few more seconds after computing more dominant singular values by PROPACK. The last four examples involve more than 50000 constraints, which could not be handled by the current version of GloptiPoly.The example "butcher" is a positive-dimensional system, the rank condition (4) will never hold for the moment matrix with maximal rank. MMCRSolver can successfully extract one real root on the manifold defined by $x_1 = x_3 = 0, x_5 = x_6 = -1$. However, there are still some examples such as: cassou, des18_3 and rabmo which can be solved by PHCpack efficiently (http://homepages.math.uic.edu/~jan/) but MMCRSolver fails to find real solutions. We are working on improving the efficiency of our algorithm by considering structures of polynomial systems.

Example 1 *The example "puma" has only four real solutions satisfying the additional two inequality constraints:*

$$\{x_5 \geq 0,\, x_6 \geq 0\}.$$

After solving the moment relaxation (5) with order $t = 3$, we find

$$\mathrm{rank} M_1(y^*) = \mathrm{rank} M_3(y^*) = 4.$$

All four real solutions can be extracted by MMCRSolver within 36.93 *seconds.*

5. REFERENCES

[1] Aubry, P., Rouillier, F., and Safey El Din, M. Real solving for positive dimensional systems. *J. Symbolic Comput. 34*, 6 (2002), 543–560.

[2] Bank, B., Giusti, M., Heintz, J., and Mbakop, G. Polar varieties and efficient real elimination. *Math. Z. 238*, 1 (2001), 115–144.

[3] Barzilai, J., and Borwein, J. Two-point step size gradient methods. *IMA J. Numer. Anal. 8* (1988), 141–148.

[4] Basu, S., Pollack, R., and Roy, M.-F. On computing a set of points meeting every cell defined by a family of polynomials on a variety. *J. Complexity 13*, 1 (1997), 28–37.

[5] Basu, S., Pollack, R., and Roy, M.-F. *Algorithms in real algebraic geometry*, vol. 10 of *Algorithms and Computation in Mathematics*. Springer-Verlag, Berlin, 2003.

[6] Bates, D., and Sottile, F. Khovanskii-rolle continuation for real solutions. A version of this article will appear in Foundations of Computational Mathematics, 2011.

[7] Beck, A., and Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci. 2*, 1 (2009), 183–202.

[8] Burer, S., and Monteiro, R. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Math. Program. 95*, 2 (2003), 329–357.

[9] Burer, S., and Monteiro, R. Local minima and convergence in low-rank semidefinite programming. *Math. Program. 103*, 3 (2005), 427–444.

[10] Cai, J.-F., Candès, E. J., and Shen, Z. A singular value thresholding algorithm for matrix completion. *SIAM J. Optim. 20*, 4 (2010), 1956–1982.

[11] Chen, G., and Teboulle, M. A proximal-based decomposition method for convex minimization problems. *Math. Programming 64*, 1, Ser. A (1994), 81–101.

[12] Chesi, G., Garulli, A., Tesi, A., and Vicino, A. An LMI-based approach for characterizing the solution set of polynomial systems. In *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on* (2000), vol. 2, pp. 1501–1506.

[13] Chesi, G., Garulli, A., Tesi, A., and Vicino, A. Characterizing the solution set of polynomial systems in terms of homogeneous forms: an LMI approach. *International Journal of Robust and Nonlinear Control 13*, 13 (2003), 1239–1257.

[14] Chesi, G., and Hung, Y. Solving polynomial systems: an LMI-based approach. In *Decision and Control, 2006 45th IEEE Conference on* (dec. 2006), pp. 5132 –5137.

[15] Corless, R. M., Gianni, P. M., and Trager, B. M. A reordered Schur factorization method for zero-dimensional polynomial systems with multiple roots. In *Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation (Kihei, HI)* (New York, 1997), ACM, pp. 133–140 (electronic).

[16] Dickenstein, A., and Emiris, I. Z. *Solving Polynomial Equations*, vol. 14 of *Algorithms and Computation in Mathematics*. Springer, 2005.

[17] Eckstein, J., and Bertsekas, D. P. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Programming 55*, 3, Ser. A (1992), 293–318.

[18] Fazel, M. *Matrix rank minimization with applications.* PhD thesis, Stanford University, 2002.

[19] Fazel, M., Hindi, H., and Boyd, S. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings of the 2001 American Control Conference* (2001), pp. 4734–4739.

[20] Gabay, D. Applications of the method of multipliers to variational inequalities. In *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems*, M. Fortin and R. Glowinski, Eds., vol. 15. Elsevier, 1983, pp. 299–331.

[21] Gabay, D., and Mercier, B. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers Mathematics with Applications 2* (1976), 1416–1438.

[22] Goldfarb, D., and Ma, S. Convergence of fixed point continuation algorithms for matrix rank minimization. *Foundations of Computational Mathematics 11*, 2 (2011), 183–210.

[23] Golub, G. H., and Van Loan, C. F. *Matrix computations*, third ed. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, 1996.

[24] Henrion, D., and Lasserre, J. Detecting global optimality and extracting solutions in GloptiPoly. In *Positive polynomials in control*, vol. 312 of *Lecture Notes in Control and Inform. Sci.* Springer, Berlin, 2005, pp. 293–310.

[25] Henrion, D., and Malick, J. Projection methods for conic feasibility problems; application to sum-of-squares decompositions. *Optimization Methods and Software 26*, 1 (2011), 23–46.

[26] Ji, S., and Ye, J. An accelerated gradient method for trace norm minimization. In *Proceedings of the 26th Annual International Conference on Machine Learning* (New York, NY, USA, 2009), ICML '09, ACM, pp. 457–464.

[27] Kočvara, M., and Stingl, M. On the solution of large-scale SDP problems by the modified barrier method using iterative solvers. *Math. Program. 109*, 2-3, Ser. B (2007), 413–444.

[28] Larsen, R. PROPACK - software for large and sparse SVD calculations. Available from: http://soi.stanford.edu/~rmunk/PROPACK/.

[29] Lasserre, J. Global optimization with polynomials and the problem of moments. *SIAM J. Optim. 11*, 3 (2001), 796–817 (electronic).

[30] Lasserre, J. *Moments, Positive Polynomials and Their Applications*. Imperial College Press, 2009.

[31] Lasserre, J., Laurent, M., Mourrain, B., Trébuchet, P., and Rostalski, P. Moment matrices, border bases and radical computation. Preprint, 2011.

[32] Lasserre, J., Laurent, M., and Rostalski, P. Semidefinite characterization and computation of zero-dimensional real radical ideals. *Foundations of Computational Mathematics 8* (2008), 607–647.

[33] Lasserre, J., Laurent, M., and Rostalski, P. A prolongation-projection algorithm for computing the finite real variety of an ideal. *Theoretical Computer Science 410*, 27-29 (2009), 2685–2700.

[34] Lasserre, J., Laurent, M., and Rostalski, P. A unified approach to computing real and complex zeros of zero-dimensional ideals. In *Emerging applications of algebraic geometry*, vol. 149 of *IMA Vol. Math. Appl.* Springer, New York, 2009, pp. 125–155.

[35] Lazard, D. Thirty years of polynomial system solving, and now? *J. Symbolic Comput. 44*, 3 (2009), 222–231.

[36] Liu, Z., and Vandenberghe, L. Interior-point method for nuclear norm approximation with application to system identification. *SIAM J. Matrix Anal. Appl. 31* (2009), 1235–1256.

[37] Ma, S., Goldfarb, D., and Chen, L. Fixed point and bregman iterative methods for matrix rank minimization. *Math. Program.* (2009), 1–33.

[38] Ma, Y. The minimum-rank Gram matrix completion via fixed point continuation method (in Chinese). *Journal of Systems Science and Mathematical Sciences 30*, 11 (2010), 1501–1511.

[39] Ma, Y., and Zhi, L. The minimum-rank Gram matrix completion via modified fixed point continuation method. In *ISSAC 2011: Proceedings of the 36th international symposium on Symbolic and algebraic computation* (New York, NY, USA, 2011), ACM, pp. 241–248.

[40] Malick, J., Povh, J., Rendl, F., and Wiegele, A. Regularization methods for semidefinite programming. *SIAM J. Optim. 20*, 1 (2009), 336–356.

[41] Morgan, A., and Shapiro, V. Box-bisection for solving second-degree systems and the problem of clustering. *ACM Trans. Math. Software 13*, 2 (1987), 152–167.

[42] Mourrain, B., and Pavone, J. P. Subdivision methods for solving polynomial equations. *J. Symb. Comput. 44* (March 2009), 292–306.

[43] Nesterov, Y. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady 27* (1983), 372–376.

[44] Nesterov, Y. Smooth minimization of non-smooth functions. *Math. Program. 103*, 1 (2005), 127–152.

[45] Nesterov, Y. Gradient methods for minimizing composite objective function. Tech. rep., 2007.

[46] Neumaier, A. *Interval Methods for Systems of Equations*, vol. 37 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 1991.

[47] Nie, J. Regularization methods for sum of squares relaxations in large scale polynomial optimization. Tech. rep., 2009. Available: http://arxiv.org/abs/0909.3551.

[48] Recht, B., Fazel, M., and Parrilo, P. A. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Rev. 52*, 3 (2010), 471–501.

[49] Reid, G., and Zhi, L. Solving polynomial systems via symbolic-numeric reduction to geometric involutive form. *J. Symbolic Comput. 44*, 3 (2009), 280–291.

[50] Rennie, J., and Srebro, N. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd international conference on Machine learning* (2005), ICML '05, pp. 713–719.

[51] Safey El Din, M., and Schost, E. Polar varieties and computation of one point in each connected component of a smooth real algebraic set. In *Proceedings of the 16th international symposium on Symbolic and algebraic computation* (New York, NY, USA, 2003), ISSAC '03, ACM, pp. 224–231.

[52] Sommese, A., and Wampler, C. *The Numerical Solution of Systems of Polynomials Arising in Engineering and Science*. World Scientific Press, Singapore, 2005.

[53] Stetter, H. *Numerical Polynomial Algebra*. SIAM, 2004.

[54] Sturm, J. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software 11/12* (1999), 625–653.

[55] Sturmfels, B. *Solving systems of polynomial equations*, vol. 97 of *CBMS Regional Conference Series in Mathematics*. Published for the Conference Board of the Mathematical Sciences, Washington, DC, 2002.

[56] Toh, K.-C., Todd, M., and Tütüncü, R. SDPT3 - a matlab software package for semidefinite programming. *Optimization Methods and Software 11* (1998), 545–581.

[57] Toh, K.-C., and Yun, S. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. Tech. rep., 2009. Available: http://www.optimization-online.org/DBHTML/2009/03/2268.html.

[58] Tseng, P. On accelerated proximal gradient methods for convex-concave optimization. *Submitted to SIAM J. Optim* (2008).

[59] Verschelde, J. Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation. *ACM Transactions on Mathematical Software 25*, 2 (1999), 251–276.

[60] Wen, Z., Goldfarb, D., and Yin, W. Alternating direction augmented Lagrangian methods for semidefinite programming. *Math. Program. Comput. 2*, 3-4 (2010), 203–230.

[61] Zhao, X.-Y., Sun, D., and Toh, K.-C. A Newton-CG augmented Lagrangian method for semidefinite programming. *SIAM J. Optim. 20*, 4 (2010), 1737–1765.