# HONORS PROJECT

YIXUAN ZHOU

Abstract. Short description

## 1. Introduction

*Polynomials* are very powerful in solving problems in applied math and engineering. And deciding the nonnegativity of a given polynomial become very important in many settings (polynomial optimizations and statistical regressions). Therefore, this decision problem become a central topic in the study of real algebraic geometry. In computational complexity terminology, it has been known that the deciding whether an arbitrary multivariate polynomial is a *nonnegative polynomial* 2.9 is a NP-hard problem (computationally infeasible) when the degree of the polynomial is greater than two.

However, if we redirect our focus to determine whether a given polynomial can be expressed in terms of *sum of squares (SOS)* 2.11, the problem is reduced to solving a *semidefinite programming (SDP)* 2.17 problem which has known algorithms that can solve the problem in polynomial times (efficiently) with respect to the input size. Though Hilbert has showed that the set of all *SOS* coincide with the set of all *nonnegative polynomials* only if the polynomials are univariate, quadratic, or bivariate with degree 4, the *SOS* is still the mostly well-known and well-used certificate in this decision problem due to its simplicity and the existence of efficient solvers. [BPT13]

When it comes to solving the *SDP*, the solver is actually dealing with a set of linear constraints, which can be compactly written in the form $Ax = b$, obtained by the process of *comparison of coefficients (COC)*. This process can be understood in terms of vectors after realizing that the set of n-variate real polynomials with degree less than or equals to $d$, denoted by $\mathbb{R}[\mathbf{x}]_{n,d}$ form a vector space. Then any polynomial $p(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]_{n,d}$ can be uniquely identified as $p(\mathbf{x}) = \sum_{j=0}^{k} c_j b_j$ given a basis $\mathcal{B} = \{b_0, ..., b_k\}$ of $\mathbb{R}[\mathbf{x}]_{n,d}$. Therefore, the process of *COC* is just formulating the equality constraints by identifying the constants $c_0, ..., c_k$. [Rec14]

Because the linear system that formed by *COC* depends on the basis $\mathcal{B}$, the stability of the linear system, measured by the *condition number* 2.6, is drastically fluctuated depends on the $\mathcal{B}$ that we chose. In this paper, I am going to run experiments, with the computer's aid, of different choices of the basis $\mathcal{B}$ to determine, in different scenarios, what is the best choice of basis to carry out the process of *COC* so that the resulted linear system is most stabled.

*Key words and phrases.* some keywords go here.

The paper is organized in the following: In section 2, I will introduce all the preliminary materials, with section 2.1 focus to introduce all the tools that will be needed, section 2.2 emphasized on introducing the detailed procedure of the process of $COC$ and the relations with the basis of the polynomial space, and section 2.3 listing some polynomial basis that will be considered in this paper and their properties. In section 3, the main numerical results of the *condition number* will be presented. And in section 4, I will present some discussion about the result that is obtained in section 3 and some thoughts on what are the further efforts that can be made to this problem.

## Acknowledgements

I would like to express my deepest gratefulness to my program advisor Mareike Dressler for offering this opportunity to do this honor project. During the past three quarters, she not only has been providing insightful ideas about the direction of this project should be going, but also has been guiding me through the process of doing research. The project could not have done without her support and help. Thank you.

## 2. Preliminaries

Here background definitions etc will be put.

### 2.1. **Linear Algebra And Semidefinite Programming.**

2.2. **Real Coefficient Polynomials.** Because we are going to use a lot of tools from lienar algebra, we first introduce some key definiations that we will use in this thesis.

**Definition 2.1.** Given a matrix $A \in \mathbb{R}^{n\times n}$, we say it is *symmetric* if $A^T = A$.

**Theorem 2.2.** *Spectrum Theorem*

*Given a symmetric matrix*$A \in \mathbb{R}^{n\times n}$*, it can be diagnoalized as*

$$A = P^{-1}DP \tag{2.1}$$

*where $D$ is the diagnoal matrix with all real values, and $P$ is an orthonomal matrix.*

*In other word, $A$ has all real eigenvalues, and their corresponding eigenvectors form an orthonomal basis of* $\mathbb{R}^n$*.* [Gol96]

**Definition 2.3.** Given a matrix $A \in \mathbb{R}^{n\times n}$, it is *positive semidifinate* if $A$ is symmetric and

$$x^TAx \geq 0 \quad \forall x \in \mathbb{R}^n \tag{2.2}$$

And we denote it as $A \succcurlyeq 0$.

**Proposition 2.4.** *A matrix is* positive semidifinate *if and only if all its eigenvalues are greater than or equals to 0*

*Proof.* If a matrix $A$ has eigenvalue $\lambda < 0$, then if $x$ is the correspdoning eigenvector, we have $x^TAx = \lambda x^Tx < 0$. On the other hand, if $A$ has all positive eignevalues, because $A$ being a symmetric matrix, its eigenvalue form a basis. Thus for any $x \in \mathbb{R}^n$, we have $x = \sum_{i=1}^n c_iv_i$ where $v_i$ are the eigenvectors of $A$, that are also orthnormal to each other. Hence, $x^TAx = \sum_{i=1}^n \lambda_i$. Since all $\lambda_i \geq 0$, we have that $x^TAx \geq 0$.

□

**Definition 2.5.** Given a matrix $A \in \mathbb{R}^{n\times m}$, the *pesudo-inverse*, which is also knows as the *Moore-Penrose* inverse of $A$, is the matrix $A^+$ satisfying:

- $AA^+A = A$
- $A^+AA^+ = A^+$
- $(AA^+)^T = AA^+$
- $(A^+A)^T = AA^+$

Every matrix has its pesudo-inverse, and when $A \in \mathbb{R}^{n\times m}$ is *full rank*, that is $rank(A) = min\{n, m\}$, $A$ can be expressed in simple algebric form.

In particular, when $A$ has linearly independent columns, $A^+$ can be computed as

$$A^+ = (A^TA)^{-1}A^T \tag{2.3}$$

In this case, the pesudo-inverse is called the *left inverse* since $A^+A = I$.

And when $A$ has linearly independent rows, $A^+$ can be be computed as

$$A^+ = A^T(AA^T)^{-1} \tag{2.4}$$

In this case, the pesudo-inverse is called the *right inverse* since $AA^+ = I$.

**Definition 2.6.** Given a matrix $A \in \mathbb{R}^{n\times m}$, the condition number of $A$, $\kappa(A)$ is defined as

$$\kappa(A) = ||A|| \cdot ||A^+|| \tag{2.5}$$

for any norm imposed on $A$, for instance *Frobenius norm.*

**Remark 2.7.** *The condition number measure how stable the system is. It can be alternatively defined as*

$$\kappa(A) = \frac{\sigma_{max}(A)}{\sigma_{min}(A)} \tag{2.6}$$

*where the $\sigma$ denotes the singular values of $A$.*

*Thus, it can be understand as how stale our system is. Intuiatively, when the condition number is large, some error in the input along the max direction of the singular value, our result would largly flacuate because the error, maginified by the singular value, will dominate the input that is along the direction of the minimum singular value. Therefore, the smaller the condition number is, the more stable our system is under flcuations caused by noises. The rigirous explaination of the condition numebr can be found in* [CK07]

Now with the tools from linear algebra, we are ready to proceed to the realm of real coefficients polynomials.

**Definition 2.8.** Let $\mathbb{R}[x]_{n,d}$ denotes the set of real coefficient polynmials with $n$ variables and at most $d$ degree.

**Definition 2.9.** Let $P_{n,2d}$ denotes the set of nonegative polynomials with $n$ variables and at most $2d$ degree, that is

$$P_{n,2d} = \{p \in \mathbb{R}[x]_{n,2d} : p(x) \geq 0, \forall x \in \mathbb{R}^d\} \tag{2.7}$$

**Remark 2.10.** *When trying to determine the nonnegativity of a polynomial, there is no reason to consider the set $P_{n,d}$ when d is odd, since if the degree of a polynomial is odd, then it will always be nonegative at some point.*

**Definition 2.11.** Let $\Sigma_{n,2d}$ denotes the set of polynomials with $n$ variables and at most $d$ degree that are *Sum of Suqares*, that is

$$\Sigma_{n,2d} = \{p \in \mathbb{R}[x]_{n,2d} : \exists\ q_1(x), ..., q_k(x) \in \mathbb{R}[x]_{n,d}\ s.t.\ p(x) = \sum_{i=1}^{k} q_i^2(x)\} \tag{2.8}$$

**Remark 2.12.** *It is easy to check that $P_{n,2d}$ form an vector space. There are a lot of choices of basis, and the most canonical one is the monomial basis.*

**Example 2.13.** $P_(2,2)$ is a vector space, then $\mathcal{B}_{2,2} = x^2, xy, y^2, x, y, 1$ is a basis of the vector space.

**Remark 2.14.** *Given $\mathcal{B}_{n,d}$ be a basis of $P_{n,d}$. If we list the elements of $\mathcal{B}_{n,d}$ in a column vector, write as $b$, then $bb^T$ form a matrix whose upper triangle entries can be collected to form a basis of $P_{n,2d}$.*

**Example 2.15.** Let $\mathcal{B}_{2,1} = x, y, 1$, be a basis of $P_{2,1}$, then

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \begin{bmatrix} x & y & 1 \end{bmatrix} = \begin{bmatrix} x^2 & xy & x \\ xy & y^2 & y \\ x & y & 1 \end{bmatrix} \tag{2.9}$$

Then the upper triangle of the result form a basis of $P_{2,2}$.

Then the question that we are interested in is given any polynomial $p(x) \in P_{n,2d}$, decide whether it is in $\Sigma_{n,2d}$. To help solving this decision problem, the following proposition will become very helpful.

**Proposition 2.16.** *Given $p(x) \in P_{n,2d}$, if $p(x) \in \Sigma_{n,2d}$, then for any basis $\mathcal{B}_{n,d}$ of $P_n n, d$, there exists a matrix $\mathcal{Q} \succcurlyeq 0$ such that*

$$\mathcal{B}_{n,d}^T \mathcal{Q} \mathcal{B}_{n,d} = p(x) \tag{2.10}$$

*Proof.* For any $p(x) \in P_{n,2d}$, if $p(x) \in \Sigma_{n,2d}$, then we can write

$$p(x) = \sum_{i=1}^{k} q^2(x) = \begin{bmatrix} q_1(x), ..., q_k(x) \end{bmatrix} \begin{bmatrix} q_1 \\ \vdots \\ q_k \end{bmatrix} \tag{2.11}$$

Notice that $q_j(x) \in P_{n,d}$.

Now given $\mathcal{B}_{n,d} = \{b_1, ..., b_{\binom{n+d}{d}}\}$ be a basis of $P_{n,d}$, we have

$$q_j(x) = \sum_{i=1}^{\binom{n+d}{d}} c_j b_j = \begin{bmatrix} c_1, ..., c_{\binom{n+d}{d}} \end{bmatrix} \begin{bmatrix} b_1 \\ \vdots \\ b_{\binom{n+d}{d}} \end{bmatrix} \tag{2.12}$$

By subsituting the section equation into the first, we have

$$p(x) = \begin{bmatrix} b_1 & \cdots & b_{\binom{n+d}{d}} \end{bmatrix} \begin{bmatrix} c_{1,1} & \cdots & c_{1,k} \\ \vdots & & \\ c_{\binom{n+d}{d},1} & \cdots & c_{\binom{n+d}{d},k} \end{bmatrix} \begin{bmatrix} c_{1,1} & \cdots & c_{1,\binom{n+d}{d}} \\ \vdots & & \\ c_{k,1} & \cdots & c_{k,\binom{n+d}{d}} \end{bmatrix} \begin{bmatrix} b_1 \\ \vdots \\ b_{\binom{n+d}{d}} \end{bmatrix} \tag{2.13}$$

Now the matrices in the middle is $C^T C = \mathcal{Q}$ a positive semidefinite matrix, which proofs the foward direction of this theorem.

On the other hand, if we know $p(x) = \mathcal{B}_{n,d}^T \mathcal{Q} \mathcal{B}_{n,d}$, where $\mathcal{Q}$ is a positive semidefinite matrix, we can just apply the Cholesky decomposition to get $\mathcal{Q} = L^T L$, and recover the SOS form of $p(x)$. [BPT13] □

Therefore, we have redueced our problem decision problem to finding this positive semidefinite matrix. It turns out that this can be done via *Semidefinite Programming* [BPT13]

**Definition 2.17.** An *Semidefinite Problem (SDP)* in standard primal form is written as

$$minimize\langle C, X\rangle \quad subject\ to\langle A_i, X\rangle = b_i, i = 1, ..., k \quad X \succcurlyeq 0 \tag{2.14}$$

For more detail on semidefinite programming, refer to [BPT13].

Returning to our problem, we require $\mathcal{Q} \succcurlyeq 0$, and the set of constraints $\langle A_i, X\rangle = b_i$ is the same as satisfying $p(x) = \mathcal{B}_{n,d}^T \mathcal{Q} \mathcal{B}_{n,d}$, which can be done via comparing the coefficients of each element in the basis that we use. [Rec14] Therefore, when we plug in the constraints into a SDP solver, if the result that we get is not no solution, then we know that the $p(x) \in \Sigma_{n,2d}$

**Example 2.18.** We can do an example here

Now, we further examine the constraints. Using the inner product of matrices, there is a natural representation of the constraints using the inner products $\langle \cdot, \cdot \rangle$ between matrices. That is $\langle A, B\rangle = tr(A^T B)$

**Proposition 2.19.** *We pick a basis of $\mathcal{B}_{n,d} = \{b_1, ..., b_{\binom{n+d}{d}}\}$ of $P_{n,d}$, and list it in a vector form $b = \begin{bmatrix} b_1 & \cdots & b_{\binom{n+d}{d}} \end{bmatrix}^T$. Then by remark 2.14, we can form a basis $\mathcal{B}_{n,2d} = \{b'_1, ..., b'_{\binom{n+2d}{2d}}\}$ based on the vector $b$. Suppose the $p(x)$ that we are interested in is written in the form $\sum_{i=1}^{\binom{n+2d}{2d}} c_i b'_i$. Then, we can reformulate $p(x) = b^T \mathcal{Q} b = \langle Q, b \cdots b^T \rangle$*

Because the constrants are linear with respect to $\mathcal{Q}$, we can re-write it into a system on linear equations with respect to the entries of the matrix $\mathcal{Q}$. Thus we can reformulate the constrants as $Aq = b$, where $q = [q_{1,1}, ..., q_{1,k}, q_{2,2}, ..., q_{k,k}]$. And we are interested in the numerical stability of $A$.

The stability of $A$ can be measured by the *condition number*. Therefore, we shall seek for the base that minimize $\kappa(A)$. [Rec14]

**Definition 2.20.** We call the matrix $b \cdot b^T$ in Proposition 2.19 the *Moment Matrix.*

**Example 2.21.** Here is another place to do an example to illustrate Moment Matrix.

Utilizing the *Moment Matrix*, we can already measure the numerical stability for the different bases $\mathcal{B}_{n,d}$ that one can choose to carry out the comparing coefficient process to generate the constraints in the *SDP*. However, what if the polynomial $p(x)$ used a different base comparing to the *moment matrix*? Given the polynomial $p(x)$ in a basis $\mathcal{B}'{}_{n,2d}$, one can find a change of basis matrix to convert write $p(x)$ in $\mathcal{B}_{n,2d}$ that we will use to compare coefficients. This change of coefficients process has its own numerical properties as well. So it would be desired if we can take that into account when analyzing the numerical properties of our constrants. Therefore, we shall introduce the *Coefficient Moment Matrix.* In the remaining part of this section, we shall build our way to it.

Suppose the *moment matrix* is constructed with the basis $\mathcal{B}_{n,d}$ and the polynomial $p(x)$ is written in the basis $\mathcal{B}'_{n,2d}$. That is suppose $\mathcal{B}'_(n, 2d) = \{b'_0, ..., b'_k\}$ where $k = \binom{n+2d}{2d} - 1$, we have $p(x) = c_0 b'_0 + ... + c_k b'_k$.

**Definition 2.22.** We define the *coefficient extraction map* as the following map,

$$\begin{aligned} \mathcal{C} : \mathbb{R}[x]_{\leq,2d} \times \mathbb{R}[x]_{\leq,2d} &\to \mathbb{R} \\ \mathcal{C}(p, b'_j) &\mapsto c_j \end{aligned} \tag{2.15}$$

[Rec14]

When fixing $s_j$, we have the *coefficient extraction map* being a linear map with respect to the polynomial $p(x)$. Indeed, we have

$$\begin{aligned} \mathcal{C}(\lambda p, b'_j) &= \lambda \mathcal{C}(p, b'_j) \quad \lambda \in \mathbb{R} \\ \mathcal{C}(p_1 + p_2, b'_j) &= \mathcal{C}(p_1, b'_j) + \mathcal{C}(p_2, b_j) \end{aligned} \tag{2.16}$$

**Remark 2.23.** *When the $\mathcal{B}'{}_{n,2d} = \{b'_1, ..., b'_k\}$ is a orthonomal basis, i.e. $\langle b_i, b_j \rangle \delta_{i,j}$ (the Dirac delta function), where the inner product is defined as*

$$\langle p, q \rangle = \int_a^b p(x)q(x)d\alpha(x) \tag{2.17}$$

*. There is a natural concretely definiation for the* coefficients extraction map*. That is*

$$\mathcal{C}(p(x), b'_j) = \langle p(x), b'_j \rangle \tag{2.18}$$

*When the basis is only orthognal, we can still define the* coefficients extration map *concretely as,*

$$\mathcal{C}(p(x), b'_j) = \frac{1}{||b'_j||} \langle p(x), b'_j \rangle \tag{2.19}$$

*where the norm $|| \cdot ||$ is induced by the corresponding inner product.*

**Remark 2.24.** *We should actually write the* coefficient extraction map *as $\mathcal{C}_{\mathcal{B}'{}_{n,2d}}$, since it depends on the base itself. However, when the base is clear, we just write it as $\mathcal{C}$.*

We can generalize the *coefficients extraction map* to take in a matrix as the first argument, and just entry-wise apply the map. With an abuse of notation, we have

**Definition 2.25.** Given a matrix of polyomials $(p_{i,j}(x))_{i,j}$ all in the bases Let the *coefficient extration map* be defined as

$$\begin{aligned} \mathcal{C} : \mathbb{R}[x]^{m \times n}_{\leq n,2d} \times \mathbb{R}[x]_{\leq n,2d} &\to \mathbb{R}^{m \times n} \\ \mathcal{C}(\begin{bmatrix} p_{1,1} & ... & p_{1,n} \\ & \vdots & \\ p_{m,1} & ... & p_{m,n} \end{bmatrix}, b'_j) &= \begin{bmatrix} \mathcal{C}(p_{1,1}, b'_j) & ... & \mathcal{C}(p_{1,n}, b'_j) \\ & \vdots & \\ \mathcal{C}(p_{m,1}, b'_j) & ... & \mathcal{C}(p_{m,n}, b'_j) \end{bmatrix} \end{aligned} \tag{2.20}$$

**Remark 2.26.** *And immediate result from the above definition is that, given $Q \in \mathbb{R}^{m \times m}$, $M \in \mathbb{R}[x]^{m \times m}_{n,2d}$, and a basis $\mathcal{B}'_{n,2d} = \{b'_1, ..., b'_k\}$, the matrix inner product provides the following relation,*

$$\mathcal{C}(\langle Q, M \rangle, b'_j) = \langle Q, \mathcal{C}(M, b'_j) \rangle \tag{2.21}$$

Notice that, let $\mathcal{B}_{n,d} = \{b_0, ..., b_l\}$, where $l = \binom{n+d}{d} - 1$, let $\vec{b} = [b_1, ..., b_l]^T$, $M = \vec{b} \cdot \vec{b}^T \in \mathbb{R}^{l,l}$. Then given $p(x)$ in $\mathcal{B}'_{n,2d} = \{b'_0, ..., b'_k\}$, $p = c_0, b'_0 + ... + c_k b'_k$, given $Q \in \mathbb{R}^{l,l}$ be the change of basis matrix from $\mathcal{B}_{n,2d}$ to $\mathcal{B}'_{n,2d}$, where $\mathcal{B}_{n,2d}$ is generated by $\mathcal{B}_{n,d}$ using remark 2.14, we have

$$c_j = \mathcal{C}(p, b'_j) = \mathcal{C}(\langle Q, M\rangle, b'_j) = \langle Q, \mathcal{C}(M, b'_j)\rangle \tag{2.22}$$

**Definition 2.27.** Define the matrix $\mathcal{A}_j = \mathcal{C}(M, b'_j)$ be the *coefficient moment matrix* of $b'_j$.

**Proposition 2.28.** *$\mathcal{A}_j$ is symmetric, because $M$ is symmetric.*

Recall, the *SOS* problem is to decide, given a polynomial $p(x)$, whether there exists a $Q \preccurlyeq 0$ such that $p = \vec{b}^T Q \vec{b}$, where $\vec{b}$ be the vector generated by $\mathcal{B}_{n,d}$. Suppose $p(x)$ is given in $\mathcal{B}'{}_{n,2d}$, we can then reformulate the constraints $\vec{b}^T Q \vec{b}$ using the *coefficient moment matrix* as

$$\langle Q, \mathcal{A}_j\rangle = c_j \quad \forall j = 0, ..., \binom{n+2d}{2d} - 1 \tag{2.23}$$

Let $\mathcal{A}_j = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,l} \\ a_{0,1} & a_{1,1} & \cdots & a_{1,l} \\ & \vdots & & \\ a_{0,l} & a_{1,l} & \cdots & a_{l,l} \end{bmatrix}, Q = \begin{bmatrix} q_{0,0} & q_{0,1} & \cdots & q_{0,l} \\ q_{0,1} & q_{1,1} & \cdots & q_{1,l} \\ & \vdots & & \\ q_{0,l} & q_{1,l} & \cdots & q_{l,l} \end{bmatrix}$

Set $\vec{a_j} = [a_{0,0}, 2a_{0,1}..., 2a_{0,l}, a_{1,1}, 2a_{1,2}, ..., a_{l,l}]^T \in \mathbb{R}^{l+1}$, and $\vec{q} = [q_{0,0}, ..., q_{l,l}]$. We can re-write the inner product $\langle Q, \mathcal{A}_j\rangle$ using the fact that both $\mathcal{A}_j$ and $Q$ are symmetric.

$$\langle Q, \mathcal{A}_j\rangle = q^T \cdot a_k = a_k^T \cdot q \tag{2.24}$$

Then, finally, we can re-write the constraint $p(x) = \vec{b}^T Q \vec{b}$, as the system of linear equations that

$$\mathcal{A} = \begin{bmatrix} a_0^T \\ \vdots \\ a_l^T \end{bmatrix} q = \begin{bmatrix} c_0 \\ \vdots \\ c_l \end{bmatrix} \tag{2.25}$$

Thus, the numerical property of the *SDP* problem is completely captured by the *condition number* of $A$. Therefore, we will write python code to examine different combinations of bases under different degrees and number of variates in the Preliminaries sections. We will list the polynomial bases that we are interested in the following sections, and will briefly touch upon *Semidifinite Programing* before we present our results.

2.3. **Polynomial Basis.**

2.4. **Solving Semidefinite Program.** Toy examples maybe

## 3. Numerical Results

**Proposition 3.1.**

*Proof.* □

maybe a theorem

**Theorem 3.2.**

or an example...

**Example 3.3.**

pictures are always a good idea...

## 4. Maybe Some Proofs

## 5. Resume, Outlook, or/and Open Problems

what did you do, what questions are still open, natural next steps etc.

## References

[BPT13] G. Blekherman, P.A. Parrilo, and R.R. Thomas, *Semidefinite optimization and convex algebraic geometry*, MOS-SIAM Series on Optimization, vol. 13, SIAM and the Mathematical Optimization Society, Philadelphia, 2013.

[CK07] E. Cheney and D. Kincaid, *Numerical mathematics and computing*, International student edition, Cengage Learning, 2007.

[Gol96] Gene Golub, *Matrix computations*, Johns Hopkins University Press, Baltimore, 1996.

[Rec14] M. Recher, *Zur numerischen Auswirkung von Basiswahlen in der polynomiellen Optimierung*, 2014, Master Thesis, Universität zu Köln.

8514 Villa La Jolla Drive # 114, La Jolla, CA, 92037

*Email address*: yiz044@ucsd.edu

9500 Gilmam Drive # 0112, La Jolla, CA, 92093

*Email address*: mdressler@ucsd.edu