Universität Innsbruck

Department of Mathematics

seminar with bachelor thesis

Simplification Methods for Sum of Squares Programs

Author
Mirco Berner

Supervisor
Univ.-Prof. Dr. Tim Netzer

June 22, 2018

Abstract

To check if a polynomial p is psd has many applications in optimization, real algebraic geometry, group theory and nonlinear analysis problems like Lyapunov stability analysis. The verification if $p \geq 0$ could be time consuming, since this problem is known to be a NP-hard problem. If p is a sum of squares, i.e. it can be expressed as sum of squares of other polynomials, then p is clearly psd. To check if p is a sum of squares is more feasible, because to determine if a sum of squares decomposition exists for a given polynomial is equivalent to a linear matrix inequality feasibility problem for the decomposition $p = z^T Qz$ and this could be done in polynomial time. The computation required to solve the feasibility problem depends on the number of monomials in z. The Newton polytope is a method to prune unnecessary monomials from the decomposition, but this method requires the construction of a convex hull and this can be time consuming for polynomials with many terms. This thesis discusses another method to prune z, called the zero diagonal algorithm. The algorithm is highly efficient, needs less computation and is more powerful than the Newton polytope method, since he is based on a single property of psd matrices and returns never a larger set of monomials for the vector z than the Newton polytope method. The algorithm is then extended to a more general simplification method for sum of squares programs and we show how problems could be formulated as sum of squares programs and therefore how to apply the zero diagonal algorithm to these problems.

I. Introduction

Note, this thesis largely follows [1]. In the interest to address a large reading public, even those without a strong mathematical background, we have tried to make this paper as self-contained as possible. This paper forgoes of the usual mathematical structure like definition followed by a sentence and the belonging proof. Instead, we have tried to inform the reader in an entertaining way by more telling a narration than to provide a dry article, although we do not spare on the necessary notations, theorems and proofs.

It is a fundamental question in mathematics and especially applied mathematics, if a function is nonnegative. In other words, if it holds for a function $f : \mathbb{R}^n \to \mathbb{R}$,

$$f(x_1, \ldots, x_n) \geq 0, \qquad \forall x_1, \ldots, x_n \in \mathbb{R}.$$

One interesting aspect behind this questioning is the fact, that numerous problems in optimization, real algebraic geometry, group theory and other areas of mathematics could be formally expressed using a finite number of multivariate polynomial equalities and inequalities [1],[3],[7],[10],[12].

So in this paper, we will confine us to the case $f \in \mathbb{R}[x_1, \ldots, x_n] := \mathbb{R}[x]$, where $\mathbb{R}[x] :=$

$\{\sum_{\alpha\in\mathbb{N}^n} c_\alpha x_1^{\alpha_1}\cdots x_n^{\alpha_n} |\ \alpha_i \in \mathbb{N},\ c_\alpha \in \mathbb{R},\ \text{finite}\, c_\alpha \neq 0\}$ denotes the set of all polynomials in the variables $x_1, \ldots, x_n$ and coefficients c_α in the field $\mathbb{R}$. For abbreviation we write for a monomial $x_1^{\alpha_1}\cdots x_n^{\alpha_n} = x^\alpha$ and for $p \in \mathbb{R}[x]$, $p = \sum_{\alpha\in\mathcal{A}} c_\alpha x^\alpha$ with $\mathcal{A} \subset \mathbb{N}^n$ and $c_\alpha \neq 0$. Furthermore for a monomial x^α is $\deg(x^\alpha) := \sum_{i=1}^n \alpha_i := |\alpha|$ the degree of the monomial and for $p \in \mathbb{R}[x]$ $\deg(p) := \max_{\alpha\in\mathcal{A}} \{\deg(x^\alpha)\}$.

Although the task to check if a polynomial $p \geq 0$ seems really easy to solve, reality has proven us wrong. Even geniuses from the last couple centuries had a tough time with this task and despite of the invention of the computer this problem is still NP-hard, even in really elementary cases like 4 dimensions ($x \in \mathbb{R}^4$) [7]. In the language of computers, this means the time to tackle the problem exceeds by far the desirable time to solve the task or sometimes the problem is actually intractable for the computer.

So therefore it would be really useful to have a 'tool', which certificates the nonnegativity of a polynomial (shortly p is positive semidefinite or psd [6]). Take for example the polynomial

$$p_1 = x_1^2 - 2x_1x_2 + x_2^2.$$

This polynomial is obviously nonnegative (psd), since $p_1 = (x_1 - x_2)^2$. But take for instance the polynomials [1],[10]:

1. $p_{motzkin} = x_1^4x_2^2 + x_1^2x_2^4 - 3x_1^2x_2^2 + 1$
2. $p_{SOS} = 3x_1^4 - 2x_1^2x_2 + 7x_1^2 - 4x_1x_2 + 4x_2^2 + 1$
3. $p_{negative} = 4x_1^2 - \frac{21}{10}x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$

In these cases it is not 'so' obvious if they are nonnegative. In the remainder of this article we will refer to this polynomials and give an answer concerning the nonnegativity and what distinguish them from the other ones.

Apparently every polynomial $p := \sum_{i\in\mathcal{I}\subseteq\mathbb{N}} f_i^2 \in \sum[x]$, where $\sum[x]$ denotes the set of all sum of squares polynomials (SOS), is nonnegative. In this case, there is no calculation necessary. We do not need to consider any local minimum, which could be a NP-hard problem. Instead, the problem has been fundamentally simplified. So if we could find a representation of p as SOS, the answer could be easily verified. That implication poses the fundamental question:

`Is every nonnegative polynomial a sum of squares (SOS)?`

If this question could be affirmed, the question of nonnegativity for a polynomial p would be equivalent to the question "Is it possible to find a SOS decomposition of p?". Hence an algorithm, what yields a possible SOS representation would be a powerful 'tool'. This algorithm should have two properties:

1. `efficiency`
2. `reliability`

What does efficiency and reliability in this context mean? Efficiency describes the property to get the result by relative few steps of calculation for the computer and reliability to get always the SOS representation of p, if p is SOS. 'Luckily', there exists an algorithm with these properties. This follows from the fact, that for each $p \in \mathbb{R}[x]$ exists a quadratic matrix Q over $\mathbb{R}$ and a vector z, [2] whose entries are monomials in the variables $x_1, \dots, x_n$ (for example $z_3 = x_1 x_2 x_n$) so that

$$p = z^T Q z.$$

The nonnegativity of a polynomial $p \in \sum[x]$ suggests that the matrix Q could be positive semidefinite (psd), i.e.

$$u^T Q u \geq 0,\ \forall u \in \mathbb{R}^{columns(Q)} \ \Leftrightarrow: Q \succeq 0.$$

This idea proposes the approach of finding a matrix $Q \succeq 0$ with $p = z^T Q z$. Indeed, it holds [1]

$$p \in \sum[x] \ \Leftrightarrow\ p = \sum_{i \in \mathcal{I}} f_i^2 \ \Leftrightarrow\ p = z^T Q z,\ Q \succeq 0.$$

In the subsequent chapters we will examine further the exact representation of the matrix $Q \succeq 0$ and the vector z.

This problem is much better tractable. It requires only to solve a **linear matrix inequality feasibility problem** [1]. Practically this means we could solve the problem in polynomial time and therefore if there exists a SOS representation. But this strategy has a fundamental pitfall. Consider the situation:

$$p_1 = x_1^2 - 2x_1x_2 + x_2^2 = (x_1 - x_2)^2 \ \Leftrightarrow\ p_1 = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}^T \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \ =:\ z^T Q z$$

This representation of p_1 as SOS is optimal. But take for instance the vector $z = z^* = [1, x_1, x_2, \dots, x_n^{100}]^T$, that choice would yield the same polynomial p_1, though an extremely large matrix Q^* and plenty steps of calculation for computing the entries of Q^*, although almost every entry is equal to zero. That is not really desirable. This example shows the complexity depends strongly on the size of the vector z respectively the number of his entries. In addition, this approach or algorithm does not really satisfy the property efficiency.

So it would be really useful to know in advance which monomials are necessarily an entry of z^*. In our example $p_1 = {z^*}^T Q^* z^*$, there were only the entries x_1, x_2 necessary, that means we could prune z^* to $z = [z_2^*, z_3^*]^T = [x_1, x_2]^T$. A quite helpful and common used method to prune z^* in advance is the **Newton polytope** [1], although this method has one big drawback, it requires the construction of a convex hull and this construction could be time comsuming for polynomials with many terms [1].

This paper refers to an alternative algorithm for pruning out unnecessary monomials from z^*, called the **zero diagonal algorithm**. This algorithm has two crucial benefits compared to the Newton polytope. He yields to better results, meaning the number of entries from $z_{diagonal}$ is never larger than z_{newton} [1]. Besides of this advantage, the algorithm forgoes of constructing a convex hull, instead he is really easy to implement with very little computational cost [1].

These both advantages result from a simple property of a positive semidefinite matrix. If the entry (i, i) from $Q \succeq 0$ is zero, than the entire i^{th} row and column must be zero [1].

Further we will extend the essential idea of the zero diagonal algorithm to a general algorithm for **sum of squares programs**, who include **free decision variables** $u_1, \ldots, u_r$. These programs [1], [12] consist of a cost function

$$p_c(u_1, \ldots, u_r) = c^T u := \langle c, u \rangle$$

who should be minimized respectively maximized and a set of constraining polynomials $\hat{p}_1, \ldots, \hat{p}_N$ of the shape

$$\hat{p}_k = a_{k,0}(x) + \textstyle\sum_{i=1}^{r} u_i a_{k,i}(x) \in \sum[x], \qquad a_{k,i} \in \mathbb{R}[x].$$

This simplification method removes next to the pruned monomials the free variables $u_1, \ldots, u_r$ who are implicitly constrained to be zero [1]. The reduction of the free decision variables can improve numerical conditioning and computational time to solve the sum of squares program [1].

Last but not least with the aid of a few samples we well show how to apply the idea of the zero diagonal algorithm to problems in the range of optimization, stability analysis, geometry and matrix copositivity [1], [4], [10], [12], i.e. we demonstrate how questions in these fields could be formulated as a sum of squares program.

II. Nonnegative Polynomials and SOS

In the previous chapter we have asked the question if each nonnegative polynomial is a sum of squares. This question is not only interesting for our application but rather from fundamental importance in mathematics.
Exactly this question proposed the german mathematician David Hilbert [6], but shortly afterwards his friend Hermann Minkowski convinced him 1885 by his doctoral dissertation that such a statement would be to strong [6] and therefore he proposed a "weaker" question:

```
Is every nonnegative polynomial a sum
of squares of rational functions?
```

What does this question mean? Instead of using polynomials as sum of squares, he proposed to take functions of the form $f = \dfrac{p}{q}$, with $p, q \in \mathbb{R}[x]$. Functions of this form are called rational functions [6] and the set of all rational functions is denoted by $\mathbb{R}(x_1, \ldots, x_n) := \mathbb{R}(x) = \{ \dfrac{p}{q} | \, p, q \in \mathbb{R}[x] \}$. Therefore we have the questionable implication:

$$p \geq 0 \;\Rightarrow\; p = \textstyle\sum_{i \in \mathcal{I} \subset \mathbb{N}} f_i^2, \quad f_i \in \mathbb{R}(x) \mathbf{?}$$

Due to the still importance of the problem he proposed this question among 22 other fundamental mathematical questions in 1900 at the International Congress of Mathematicians. This question is also known as Hilbert's 17^{th} problem and has demonstrated to be a real twister, considering that it needed 27 years until Artin finally gave an affirmative proof [2],[6]. Moreover Albrecht Pfister showed in 1967 [6] for a psd polynomial p that $p(x_1, \ldots, x_{\mathbf{n}}) = \sum_{i=1}^{m \leq 2^{\mathbf{n}}} \left(\dfrac{g_i}{h_i} \right)^2$ for $g_i, h_i \in \mathbb{R}[x]$. Hence follows [6]

$$p \geq 0 \;\Leftrightarrow\; p = \sum_{i=1}^{m \leq 2^{\mathbf{n}}} f_i^2, \quad f_i \in \mathbb{R}(x_1, \ldots, x_{\mathbf{n}}),$$

but does this also hold for $f_i \in \mathbb{R}[x]$? Shortly after Hilbert changed his assumption concerning nonnegativity and SOS, he proved in a seminal paper in 1888 by abstract means - who were not constructive - the existence of some counterexamples, i.e. Hilbert could not state exact presentations of some counterexamples but he could prove the existence of some polynomials who are psd (nonnegative) but not SOS [6]. Therefore:

$$p \geq 0 \;\not\Leftrightarrow\; p = \textstyle\sum_{i \in \mathcal{I}} f_i^2 \;\in \sum[x].$$

Further he gave a clear distinction for the different classes in the set of all psd polynomials.

Theorem 1: *Let $p \in \mathbb{R}[x_1]$ be an univariate polynomial of any degree then* [5]

$$p \geq 0 \;\Leftrightarrow\; p \in \sum[x_1] \;\Leftrightarrow\; p = \sum_{i=1}^{m \leq 2^1} f_i^2,\; f_i \in \mathbb{R}[x_1] \;\Leftrightarrow\; p = f_1^2 + f_2^2$$

Proof. Clearly each psd polynomial p has pairs of roots who are either complex in conjugation or real. This follows from the fact that the graph of a psd polynomial p is above the x axis and if $p = 0$, then p touches the x axis but is not intersecting the axis. Additionally $0 = p(w) = \overline{p(w)} = \overline{\sum_{i=0}^{2d} c_i w^i} = \sum_{i=0}^{2d} c_i \overline{w^i} = p(\overline{w}) = 0$. Therefore the roots have the form $a_j \pm ib_j$ and we can write p as

$$\begin{aligned} p(x_1) &= c_{2d} \cdot [(x_1 - [a_1 + ib_1]) \cdots (x_1 - [a_d + ib_d])] \cdot [(x_1 - [a_1 - ib_1]) \cdots (x_1 - [a_d - ib_d])] \\ &= c_{2d} \cdot [g_1(x_1) + ig_2(x_1)] \cdot [g_1(x_1) - ig_2(x_1)] \\ &= f_1^2(x_1) + f_2^2(x_1). \end{aligned}$$

□

Theorem 2: *Let $p \in \mathbb{R}[x]$ be a multivariate polynomial of degree 2 then* [2]

$$p \geq 0 \;\Leftrightarrow\; \sum_{i \in \mathcal{I} \subset \mathbb{N}} f_i^2, \quad \deg(f_i) = 1$$

Proof. Primarily we could assume that p is homogenous of degree 2. That means every monomial of p has degree 2. That follows from the fact, if p is psd, then the homogenous polynomial p_h of degree 2, which is obtained by multiplicating each monomial of p with the new variable x_0 as often that each monomial has degree 2, is also psd. Therefore is $p_h(x_0 = 1, x_1, \ldots, x_n) = p(x_1, \ldots, x_n)$. Furthermore holds for every homogeneous quadratic polynomial

$$p_h = x^T M x = \sum_{i,j=0}^{n} m_{ij} x_i x_j$$

for a symmetric matrix $M = (m_{ij})_{i,j} \in \mathbb{R}^{(n+1)\times(n+1)}$. The nonnegativity of p_h implicates that the matrix M is psd and therefore is each eigenvalue $\lambda_i \geq 0$ and the number of eigenvalues $\lambda_j > 0$ is equal to rank(M). Thus is according to the spectral theorem for a symmetric psd matrix M,

$$\begin{aligned} &U^T M U = D := \operatorname{diag}(\lambda_1, \ldots, \lambda_{rank(M)}, 0_{rank(M)+1}, \ldots, 0_{n+1}) \;\Leftrightarrow \\ &M = UDU^T = UD^{\frac{1}{2}} D^{\frac{1}{2}} U^T = AA^T, \quad U^T U = I. \end{aligned}$$

Let be $M = A^\star A^{\star T}$ with $A \in \mathbb{R}^{(n+1)\times(n+1)}$ pruned to the matrix $A^\star \in \mathbb{R}^{(n+1)\times rank(M)}$, since the last columns are equal to zero and denote $A^\star_{-i} := v_i$, then it follows

$$M = \sum_{j=1}^{rank(M)} v_j v_j^T$$

and consequently

$$p_h = x^T \left(\sum_{k=1}^{rank(M)} v_k v_k^T \right) x = \sum_{k=1}^{rank(M)} \left(v_k^T x \right)^2 .$$

$\square$

Theorem 3: *Let* $p \in \mathbb{R}[x_1, x_2]$ *be a bivariate polynomial of degree* 4 *then* [6]

$$p \geq 0 \; \Leftrightarrow \sum_{i=1}^{m \leq 2^{\mathbf{2}}-1} f_i^2 = \sum_{i=1}^{m \leq \mathbf{3}} f_i^2, \quad f_i \in \mathbb{R}[x_1, x_2]$$

Remark: Hilbert's proof is beautiful and short but difficult and incomprehensible for the modern reader, thus we waive on this proof. We refer to [14] or [15] for modern accounts of Hilbert's proof.

In any other case Hilbert has shown in 1888 the existence of a psd polynomial $p \in \mathbb{R}[x]$ which is not a sum of squares in $\mathbb{R}[x]$ [6]. Hence we get the implications

1. $p(x_1) \geq 0 \quad \wedge \; (n = 1) \quad \Leftrightarrow \; p(x_1) = f_1^2(x_1) + f_2^2(x_1)$

2. $p(x) \geq 0 \quad \wedge \; (d = 2) \quad \Leftrightarrow \; p(x) = \sum_{i \in \mathcal{I} \in \mathbb{N}} f_i^2(x), \quad \deg(f_i) = 1$

3. $p(x_1, x_2) \geq 0 \quad \wedge \; (n = 2, d = 4) \quad \Leftrightarrow \; p(x_1, x_2) = f_1^2(x_1, x_2) + \ldots + f_{m \leq 3}^2(x_1, x_2)$

where d denotes the degree of p and n the number of different variables of p.

At first glance you would think this is a big damper because our question "If psd is equivalent to SOS" was denied and so therefore our entire considerations concerning of finding a good 'tool', which certificates the nonnegativity of a polynomial p, are futile. This means our algorithm is useless in the case if $p \in \mathbb{R}[x]$ is nonnegative but not a sum of squares.
However, if 'many' psd polynomials are as well SOS, our algorithm would be still a good 'tool' for many cases and is therefore more than adequate for most applications. This assumption is bolstered by the following two implications [7]

$$p \geq 0 \text{ on } \mathbb{R}^n \; \Rightarrow \; \forall \varepsilon > 0 \exists \kappa \in \mathbb{N} : \left[p + \varepsilon \left(\sum_{h=0}^{\kappa} \sum_{i=1}^{n} \frac{x_i^{2h}}{h!} \right) \right] \in \sum [x],$$

$$p \geq 0 \text{ on } \mathbb{R}^n \; \Rightarrow \; p \geq 0 \text{ on } [-1, 1]^n \; \Rightarrow \; \forall \varepsilon > 0 \exists \kappa \in \mathbb{N} : \left[p + \varepsilon \left(1 + \sum_{i=1}^{n} x_i^{2\kappa} \right) \right] \in \sum [x]$$

from Lasserre and Netzer. Consequently, the SOS cone is dense in the cone of nonnegative polynomials.

Besides these two theoretical aspects, Parrilo and Sturmfels have shown in a practical way by computing up to degree 15, that the global minimum $p^* \in \mathbb{R}$ of a lower bounded polynomial $p \in \mathbb{R}[x]$ almost always coincides with the largest value $\lambda \in \mathbb{R}$, so that $[p(x) - \lambda] \in \sum[x]$ [8]. Practically, this means, that $(p(x) - p^*) \geq 0$ and since $[p(x) - \lambda] \in \sum[x]$ is $\lambda \leq p^*$ and for $\lambda = p^* \geq 0$ follows that $p \in \sum[x]$, since $p(x) = p(x) - p^* + (\sqrt[2]{p^*})^2 = p(x) - \lambda + (\sqrt[2]{\lambda})^2 \in \sum[x]$.

That suggests, a nonnegative polynomial p is 'mostly' a sum of squares [16] and therefore is the distinction between nonnegativity and SOS in most applications negligible. This supposition is also supported by the fact, that mathematicians needed over 80 years to construct a polynomial $p_{motzkin}$, called the Motzkin polynomial, which is psd but not SOS [2], [6].

Theorem 4: [Motzkin, 1967] *The polynomial* $p_{motzkin} = x_1^4 x_2^2 + x_1^2 x_2^4 - 3x_1^2 x_2^2 + 1$ *is nonnegative.* [2]

Proof. We will present this fact by 2 different ways, since each way is a useful instrument to prove the nonnegativity of a polynomial p.

1. For $0 \leq a, b, c \in \mathbb{R}$ is the arithmetic mean always bigger than the geometrical [2], [5], hence

$$(abc)^{\frac{1}{3}} \leq \frac{1}{3}(a + b + c)$$

and for $a = 1,\ b = x_1^4 x_2^2,\ c = x_1^2 x_2^4$ follows

$$x_1^4 x_2^2 + x_1^2 x_2^4 + 1 \geq 3x_1^2 x_2^2$$

and consequently is $p_{motzkin} \geq 0$.

2. According to Artins proof, each psd polynomial is a sum of squares of rational functions and therefore should exist a decomposition with no more than 2^2 squares [6]. And indeed it holds [5]

$$\begin{aligned} p_{motzkin} = &\left(\frac{x_1^2 - x_2^2}{x_1^2 + x_2^2}\right)^2 + \left(\frac{x_1^2 x_2 (x_1^2 + x_2^2 - 2)}{x_1^2 + x_2^2}\right)^2 + \\ &\left(\frac{x_1 x_2^2 (x_1^2 + x_2^2 - 2)}{x_1^2 + x_2^2}\right)^2 + \left(\frac{x_1 x_2 (x_1^2 + x_2^2 - 2)}{x_1^2 + x_2^2}\right)^2. \end{aligned}$$

Therefore $[(x_1^2 + x_2^2)^2 \cdot p_{motzkin}] \in \sum[x]$ and thus $p_{motzkin} \geq 0$. $\square$

Remark: Reznick proved for a homogeneous polynomial p_h [9]:

$$p_h > 0 \text{ on } \mathbb{R}^n \setminus \{0\} \ \Rightarrow \left[\left(\sum_{i=1}^{n} x_i^2\right)^{\mathbf{r}} \cdot p_h\right] \in \sum[x] \text{ for some } r \in \mathbb{N}.$$

Astonishingly the result of the lemma from Reznick is also valid for the motzkin polynomial with $r = 2$, although $p_{motzkin}(1, 1) = 0$.

This example proposes the strategy to check if p is psd: If p is not SOS, then check if $\left[\left(\sum_{i=1}^{n} x_i^2\right)^r \cdot p\right] \in \sum[x]$ for some $r \in \mathbb{N}$.

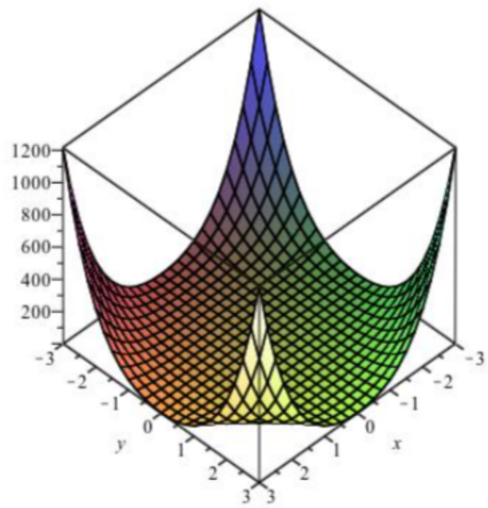


Figure 1: graph Motzkin polynomial [2]

Remark: Although we do not know exactly 'how many' psd polynomials are as well SOS, the lemmas of Lasserre, Netzer and the work of Parrilo, Sturmfels besides the long construction process of the motzkin polynomial suggest most psd polynomials are likewise SOS [16] and therefore it is more than sufficient to use the constraint SOS instead of nonnegativity. This assumption is important for the applications in the following chapters.

III. Zero Diagonal Algorithm

In this chapter we show how to approach the problem if a polynomial p is a sum of squares and therefore we introduce the Newton polytope [1] and as well the zero diagonal algorithm [1]. Furthermore we explain in detail the benefits of the zero diagonal algorithm.

As mentioned before in the introduction, a good algorithm possesses the property efficiency. Furthermore would it be helpful to decide beforehand, if the polynomial p is not a sum of squares and thus we could spare us the algorithm and consequently the costs.

Two useful facts concerning this consideration:

1. If p is a sum of squares then is p nonnegative and particularly lower bounded and must therefore have even degree.
2. If p is a sum of squares and has only one leading coefficient c_m (p has only one monomial m with coefficient c_m and $\deg(m) = \deg(p)$), then is c_m positive.

Besides these two obvious facts, there exist a range of features for SOS polynomials, which are not so apparent. Therefore we introduce the **Gram matrix** [2], who is a helpful aid to better understand the other features of SOS polynomials.

With $\mathbb{R}[x_1, \ldots, x_n]_d := \mathbb{R}[x]_d$ and $\mathrm{Sym}_d(\mathbb{R})$ we denote the set of all polynomials p with $\deg\ p \leq d$ respectively the set of all symmetric matrices $G \in \mathbb{R}^{d \times d}$. The $\mathbb{R}$ vector space $\mathbb{R}[x]_d$ has the dimension $\Delta_d = \binom{n+d}{d}$ and possesses for instance the monomial base $X_d := (x^\alpha)_{\alpha \in \mathbb{N}^n, |\alpha| \leq d} = (1, x_1, x_2, \ldots, x_1^2, x_1 x_2, \ldots)$ [2].

Now we consider the linear mapping

$$G : \mathrm{Sym}_{\Delta_d}\mathbb{R} \rightarrow \mathbb{R}[x]_{2d} : \quad M \mapsto X_d^T M X_d.$$

For $M = (m_{\alpha\beta})_{|\alpha|,|\beta| \leq d}$ is (α, β are vectors)

$$G(M) = \sum_{|\alpha|,|\beta| \leq d} m_{\alpha\beta} x^\alpha x^\beta = \sum_{|\gamma| \leq 2d} \left(\sum_{\alpha+\beta=\gamma} m_{\alpha\beta} \right) x^\gamma.$$

Obviously G is surjective and for $p \in \mathbb{R}[x]_{2d}$ is [2]

$$G^{-1}(p) = \{Q + \sum_{i=1}^{h} \lambda_i N_i \mid \lambda_i \in \mathbb{R}\}$$

whereby $\{N_i\}_{i=1}^h$ is a base of $ker(G)$, Q is one possible solution for $p = X_d^T Q X_d$ and $h = \left(\frac{l_{X_d}(l_{X_d}+1)}{2} - l_w\right)$ for $l_{X_d} = \binom{n+d}{d}$ respectively $l_w = \binom{n+2d}{2d}$. h follows from the dimension formula, since M is symmetric and has therefore $\frac{l_{X_d}(l_{X_d}+1)}{2}$ linear independent entries, moreover is the dimension of $\mathbb{R}_{2d} = l_w$ and therefore is $h = \frac{l_{X_d}(l_{X_d}+1)}{2} - l_w$. For this reason is $G^{-1}(p)$ a not empty affine subspace of $\mathrm{Sym}_{\Delta_d}\mathbb{R}$ [2]. An element of $G^{-1}(p)$ is called **Gram matrix**.

Now we have all the tools to prove the pivotal point of this paper, which is building the cornerstone for the following considerations.

Theorem 6: *For a polynomial* $p \in \mathbb{R}[x]_{2d}$ *holds* [2]

$$p = \left(\sum_{i=1}^{r \leq \binom{n+d}{d}} f_i^2 \right) \in \sum[x] \Leftrightarrow p = X_d^T M X_d,\ 0 \preceq M \in \mathrm{Sym}_{\Delta_d}(\mathbb{R}).$$

Proof. Let be $0 \preceq M \in \mathrm{Sym}_{\Delta_d}(\mathbb{R})$ a Gram matrix of polynomial p. According to Theorem 2 do we find a decomposition

$$M = \sum_{i=1}^{r=rank(M)} v_i v_i^T, \quad v \in \mathbb{R}^{\Delta_d}.$$

Therefore is

$$p = X_d^T M X_d = \sum_{i=1}^{rank(M)} X_d^T v_i v_i^T X_d = \sum_{i=1}^{rank(M)} \left(v_i^T X_d\right)^2 = \sum_{i=1}^{rank(M)} f_i^2 \in \sum[x].$$

Conversely let

$$p = \sum_{i \in \mathcal{I} \subset \mathbb{N}} f_i^2, \quad f_i \in \mathbb{R}[x],$$

then is $p \in \mathbb{R}[x]_{2d}$ with $f_i \in \mathbb{R}[x]_d$ and thus we can find a vector $v \in \mathbb{R}^{\Delta_d}$ for $f_i = v_i^T X_d$. Therefore is

$$p = \sum_{i \in \mathcal{I} \subset \mathbb{N}} f_i^2 = \sum_{i \in \mathcal{I} \subset \mathbb{N}} \left(v_i^T X_d\right)^2 = \sum_{i \in \mathcal{I} \subset \mathbb{N}} X_d^T v_i v_i^T X_d = X_d^T \left(\sum_{i \in \mathcal{I} \subset \mathbb{N}} v_i v_i^T \right) X_d,$$

and p has the positive semidefinite Gram matrix

$$\left(\sum_{i \in \mathcal{I} \subset \mathbb{N}} v_i v_i^T \right) \succeq 0.$$

$\square$

The theorem shows that the question "Is $p \in \sum[x]$?" could be equivalently formulated to the feasible problem [1]

$$\text{"Find matrix } M \succeq 0 \text{ such that } p = X_d^T M X_d\text{"}. \tag{1}$$

Since

$$p = \sum_{|\alpha|,|\beta| \leq d} m_{\alpha\beta} x^\alpha x^\beta = \sum_{|\gamma| \leq 2d} \left(\sum_{\alpha+\beta=\gamma} m_{\alpha\beta} \right) x^\gamma$$

and the matrix M is constrained do be positive semidefinite, we do have linear equality constraints on the entries of M. Problem (1) is also called a linear matrix inequality (LMI) feasibility problem [1].

As we have mentioned and seen in the introduction, the complexity to solve this problem grows extremely with the dimension of the Gram matrix M of p [1]. Our general approach for a polynomial p with deg p = 2d and n different variables $x_1, \ldots, x_n$ would lead to $l_{X_d} := \binom{n+d}{d}$ entries of X_d and therefore is the Gram matrix $M \in \mathbb{R}^{l_{X_d} \times l_{X_d}}$. The number of entries l_{X_d} from X_d also increases rapidly with the number of variables and the degree of the polynomial p [1]. As already shown in the introduction, the decomposition of $p = zQz$ is not necessarily unique, this means it would be desirable to get a vector z with preferably less entries.

The Newton polytope is an algorithm, who reduces the dimension l_{X_d} of the vector X_d by pruning out unnecessary monomials of X_d [1]. For convenience we will give at first some terminology concerning polytopes.

Let $\mathcal{A} \subset \mathbb{R}^n$, then is convhull($\mathcal{A}$) the convex hull of $\mathcal{A}$. Further let $C \subset \mathbb{R}^n$ be a convex set. A point $\alpha \in C$ is called an extreme point of C, if for $\alpha_1, \alpha_2 \in C$ and $0 < \lambda < 1$ holds [1]

$$\alpha = \lambda\alpha_1 + (1-\lambda)\alpha_2 \;\Rightarrow\; \alpha_1 = \alpha_2 = \alpha.$$

Practically this means α is a corner point of the set C. A polytope is the convex hull of a non empty, finite set $\{\alpha_1, \ldots, \alpha_p\} \subset \mathbb{R}^n$. The extreme points of a polytope are called vertices [1]. Let C be a polytope and $\mathcal{V}$ the finite set of vertices of C, then is C = convhull($\mathcal{V}$) and $\mathcal{V}$ is a minimal vertex representation of C.
Besides the vertices representation of C = convhull($\mathcal{V}$), there exists another representation of C with the intersection of a finite collection of halfspaces, i.e. there exists a matrix $H \in \mathbb{R}^{N \times n}$ and a vector $g \in \mathbb{R}^N$ such that $C = \{\alpha \in \mathbb{R}^n \mid H\alpha \leq g\}$ [1]. This representation of C is called a facet or half-space representation of C.

Since we have now introduced the conceptualities of a polytope and half-space, we can face to the Newton polytope and extend this concept a little bit further.

The Newton polytope of a polynomial $p = \sum_{\alpha \in \mathcal{A} \subset \mathbb{N}^n} c_\alpha x^\alpha$ is defined as [1]

$$N(p) := \text{convhull}(\mathcal{A}) \subset \mathbb{R}^n$$

and the reduced Newton polytope is defined by [1]

$$\tfrac{1}{2}N(p) := \{\tfrac{1}{2}\alpha \mid \alpha \in N(p) \}.$$

For example the Newton polytope of the motzkin polynomial $p_{motzkin} = x_1^4 x_2^2 + x_1^2 x_2^4 - 3x_1^2 x_2^2 + 1$ is the convex hull of the points {(0,0),(2,2),(4,2),(2,4)} and is therefore [2]:

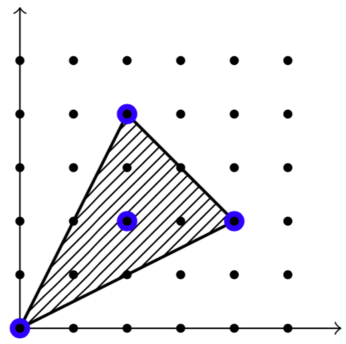


Figure 2: Newton polytope Motzkin polynomial [2]

Soon we will prove that the vertices of the Newton polytope $N(p = \sum_1^m f_i^2 \in \sum[x])$ are vectors whose entries are even numbers and $N(f_i) \subset \frac{1}{2}N(p)$. This is a key result for monomial reduction. But before we could demonstrate that fact, we need to do some preparations.
Consider for $v \in \mathbb{R}^n$ and $r \in \mathbb{R}$ the half-space

$$H_{v,r} := \{\alpha \in \mathbb{R}^n \mid \alpha^T v := \langle \alpha, v \rangle \geq r\}.$$

Remark: A polytope P is the intersection of all half-spaces $H \supset P$ [2].

Theorem 7: *For $p = \sum_{\alpha \in \mathcal{A} \subset \mathbb{N}^n} c_\alpha x^\alpha \in \mathbb{R}[x], v \in \mathbb{Q}^n$ and $r \in \mathbb{Q}$ holds* [2]

$$N(p) \subseteq H_{v,r} \Leftrightarrow \forall a \in \mathbb{R}^n : \lim_{t \searrow 0} |t^{-r} \cdot p(a_1 t^{v_1}, \ldots, a_n t^{v_n})| < \infty.$$

Proof. Since $\langle \alpha, v \rangle \geq r$ for all $\alpha \in \mathbb{N}^n$ with $c_\alpha \neq 0$, follows

$$\lim_{t \searrow 0} |t^{-r} \cdot p(a_1 t^{v_1}, \ldots, a_n t^{v_n})| = \lim_{t \searrow 0} |\sum_{\alpha \in \mathcal{A}} c_\alpha \cdot a^\alpha \cdot t^{\langle \alpha, v \rangle - r}| \ \leq \ |\sum_{\alpha \in \mathcal{A}} c_\alpha \cdot a^\alpha| < \infty.$$

Conversely let assume there exists an exponent $\alpha \in \mathbb{N}^n$ with $c_\alpha \neq 0$ and $\langle \alpha, v \rangle = s < r$. Let be s minimal and $\{\alpha^1, \ldots, \alpha^m\}$ the set of all α^i who satisfy $\langle \alpha^i, v \rangle = s$. Then exists $a \in \mathbb{R}^n$ with $\gamma := \sum_{i=1}^m c_{\alpha^i} a^{\alpha^i} \neq 0$ and consequently is

$$\lim_{t \searrow 0} |t^{-r} \cdot p(a_1 t^{v_1}, \ldots, a_n t^{v_n})| = \lim_{t \searrow 0} |\gamma \cdot t^{s-r} + \left(\sum_{\alpha \in \mathcal{A} \setminus \{\alpha^1, \ldots, \alpha^m\}} c_\alpha \cdot a^\alpha \cdot t^{\langle \alpha, v \rangle - r} \right) | = \infty$$

since

$$\lim_{t \searrow 0} t^{s-r} = \infty \quad \text{and} \quad \min_{\alpha \in \mathcal{A} \setminus \{\alpha^1, \ldots, \alpha^m\}} \{\deg t^{\langle \alpha, v \rangle - r}\} > s - r.$$

$\square$

Theorem 8: *For* $p, f_1, \ldots, f_m \in \mathbb{R}[x]$ *holds* [2]

$$\begin{aligned}
&(i) \quad N(p^2) = 2N(p) := \{2a \mid a \in N(p)\} \\
&(ii) \quad f_i, f_j \geq 0 \;\Rightarrow\; N(f_i) \subseteq N(f_i + f_j) \\
&(iii) \quad p = \left(f_1^2 + \ldots + f_m^2\right) \in \sum[x] \;\Rightarrow\; N(f_i) \subseteq \frac{1}{2}N(p),\ \forall i \in \{1, \ldots, m\}.
\end{aligned}$$

Proof. (i): For $v \in \mathbb{Q}^n$, $a \in \mathbb{R}^n$ and $r \in \mathbb{Q}$ is according to Theorem 7

$$N(p^2) \subseteq H_{v,r} \;\Leftrightarrow\; \forall a \in \mathbb{R}^n : \lim_{t \searrow 0} |t^{-r} \cdot p(a_1 t^{v_1}, \ldots, a_n t^{v_n})^2| < \infty \Leftrightarrow$$

$$\lim_{t \searrow 0} |t^{-\frac{r}{2}} \cdot p(a_1 t^{v_1}, \ldots, a_n t^{v_n})| < \infty \;\Leftrightarrow\; N(p) \subseteq H_{v, \frac{r}{2}} = \frac{1}{2} H_{v,r} \;\Leftrightarrow\; 2N(p) \subseteq H_{v,r}$$

and two polytopes, who are included in the same set of (rational) half-spaces, are equal.

(ii): Let be $N(f_i + f_j) \subseteq H_{v,r}$ and $a \in \mathbb{R}^n$, then is

$$\lim_{t \searrow 0} |t^{-r} \cdot (f_i(a_1 t^{v_1}, \ldots, a_n t^{v_n}) + f_j(a_1 t^{v_1}, \ldots, a_n t^{v_n}))| < \infty \;\Leftrightarrow\; N(f_i + f_j) \subseteq H_{v,r}$$

and

$$|t^{-r} \cdot f_i(a_1 t^{v_1}, \ldots, a_n t^{v_n})| \leq |t^{-r} \cdot (f_i(a_1 t^{v_1}, \ldots, a_n t^{v_n}) + f_j(a_1 t^{v_1}, \ldots, a_n t^{v_n}))|.$$

Furthermore holds

$$N(f_i) \subseteq H_{v,r} \;\Leftrightarrow\; \lim_{t \searrow 0} |t^{-r} \cdot f_i(a_1 t^{v_1}, \ldots, a_n t^{v_n})| < \infty$$

and therefore $N(f_i) \subseteq N(f_i + f_j)$.

(iii): $N(p = f_1^2 + \ldots + f_m^2) \supseteq N(f_i^2) = 2N(f_i) \;\Leftrightarrow\; N(f_i) \subseteq \frac{1}{2}N(p)$. □

Now we can prove the key result for monomial reduction.

Theorem 9: *If* $p = \sum_{\alpha \in \mathcal{A}} c_\alpha x^\alpha = \sum_{i=1}^m f_i^2 \in \sum[x]$ *then the vertices* $\{\alpha_{v_1}, \ldots, \alpha_{v_\alpha}\} = \mathcal{V}$ *of* $N(p)$ *are vectors whose entries are even numbers and* $N(f_i) \subseteq \frac{1}{2}N(p)$*. Further are the coefficients* $\{c_{\alpha_{v_1}}, \ldots, c_{\alpha_{v_\alpha}}\}$ *positive numbers.* [1]

Proof. $N(f_i) \subseteq \frac{1}{2}N(p)$ follows from Theorem 8 and

$$N(p) = \text{convhull}(\bigcup_{i=1}^m N(f_i^2)) = \text{convhull}(\bigcup_{i=1}^m 2N(f_i)) = 2 \cdot \text{convhull}(\bigcup_{i=1}^m N(f_i)).$$

The vertices $\{v_{f_i}^1, \ldots, v_{f_i}^{\tilde{f}_i}\} = \mathcal{V}_{f_i}$ of each $N(f_i)$ have apparently entries of positive integers and therefore has each vertex of $\text{convhull}(\bigcup_{i=1}^m N(f_i))$ entries with positive integers.

Consequently has $N(p) = 2 \cdot \text{convhull}(\bigcup_{i=1}^{m} N(f_i))$ only vertices with even numbers.

Since $N(p) = \text{convhull}(\bigcup_{i=1}^{m} N(f_i^2))$, follows for a vertex $\alpha_{v_i} := \sum_{k,j} 2v_{f_j}^k \in \mathcal{V}$ for appropriate k, j. Thus is

$$c_{\alpha_{v_i}} = \sum_{k,j} c_{v_{f_j}^k} \cdot c_{v_{f_j}^k} > 0.$$

$\square$

Finally Theorem 9 enables us to show that the Motzkin polynomial is not a sum of squares, that means $0 \leq p_{motzkin} \notin \sum[x]$.

Theorem 10: *The motzkin polynomial* $p_{motzkin} = x_1^4x_2^2 + x_1^2x_2^4 - 3x_1^2x_2^2 + 1$ *is not a sum of squares.* [2]

Proof. Assume $p_{motzkin} = f_1^2 + \ldots + f_m^2 \in \sum[x]$, then $N(f_i) \subseteq \frac{1}{2}N(p_{motzkin})$ according to Theorem 9. This means f_i includes only monomials of $\mathcal{M}_{mot} = \{1, x_1x_2, x_1^2x_2, x_1x_2^2\}$ since

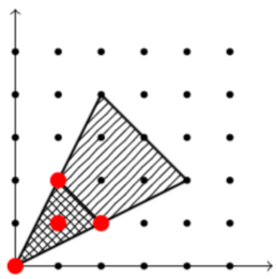


Figure 3: $\frac{1}{2}N(p_{motzkin})$ and $N(p_{motzkin})$ [2]

The monomial $x_{neg} = x_1^2x_2^2$ of a term f_j^2, who includes $x_1^2x_2^2$, has a unique decomposition, i.e. for $a, b \in \mathcal{M}_{mot}$ follows $x_{neg} = x_1^2x_2^2 = a \cdot b \Leftrightarrow a = b = x_1x_2$ and therefore has the monomial x_{neg} a positive coefficient in f_j^2. Though the motzkin polynomial has the coefficient $c_{2,2} = -3$ for the monomial $x_{neg} = x_1^2x_2^2$. Consequently is $p_{motzkin} \notin \sum[x]$. $\square$

Remark: $\hat{p}_{motzkin} = x_1^4x_2^2 + x_1^2x_2^4 - 3x_1^2x_2^2 + \gamma \notin \sum[x]$ for all $\gamma \in \mathbb{R}$, since our proof is unconnected of the the monomial $x_1^0x_2^0 = 1 \cdot 1 = 1$ or in other words, there exist some positive polynomials with large minimums, who are not SOS. For example is $\min \hat{p}_{motzkin} = \gamma - 1$. [2]

Another interesting point of the reduced Newton polytope is, that it empowers us to state an upper bound for the squares of a decomposition, if their exists a partition.

Theorem 11: *Let* $p \in \sum[x]$ *and* $r := |\frac{1}{2}N(p) \cap \mathbb{N}^n|$ *the number of lattice points in*

$\dfrac{1}{2}N(p)$, *then is* [2]

$$p = \sum_{i=1}^{m \le r} f_i^2.$$

Proof. Let be $p = f_1^2 + \ldots + f_m^2$, then each f_i includes only monomials with exponents from $\dfrac{1}{2}N(p) \cap \mathbb{N}^n$ and therefore could be for each v_i only the same r entries $v_i^{r_1}, \ldots, v_i^{r_r} \neq 0$, whereby $f_i = v_i^T X_d$. Hence we get the decomposition $p = X_d^T M X_d$ with $0 \preceq M = \sum_{i=1}^{m} v_i v_i^T$ and rank$(M) \le r$. Then Theorem 11 follows with Theorem 6. □

Remark: If a polynomial $p \in \sum[x]$ possesses the Newton polytope $N(p) = N(p_{motzkin})$, then follows with Theorem 11, that $p = \sum_{i=1}^{m \le 4} f_i^2$. Our general approach would lead to the estimation $m \le \binom{2+3}{3} = 10$. [2]

Theorem 9 implies that any monomial x^α appearing in the vector X_d respectively z of a SOS decomposition $p = X_d^T M X_d$ or $p = z^T Q z$ must satisfy $\alpha \in \frac{1}{2}N(p) \cap \mathbb{N}^n$.

That fact forms the basis for the Newton polytope method for pruning monomials. This method could be implemented as follow [1]:
Construct the vector X_d in the variables $x_1, \ldots, x_n$, who includes all monomials deg $x^\alpha \le d$ and has therefore $\binom{d+n}{d}$ entries. Then compute a half-space representation $\{\alpha \in \mathbb{R}^n \mid H\alpha \le g\}$ for the reduced Newton polytope $\frac{1}{2}N(p)$ and prune out any monomials in X_d who are not elements of $\frac{1}{2}N(p)$.

To prune out any monomials equals checking each monomial in X_d to see if the corresponding degree vector satisfies the half-plane constraints $H\alpha \le g$. This step is computationally very fast, though the computation of the half-space representation of the convex hull $\frac{1}{2}\mathcal{A}$ could be time consuming for polynomials p with many terms, i.e. $\mathcal{A}$ has many elements. Hence we present soon an alternative implementation of the Newton polytope method, that avoids the construction of a half-space representation of the reduced Newton polytope $\frac{1}{2}N(p)$.

But first a short example regarding the Newton polytope method [1].

Example: Consider the following polynomial from the introduction [1]

$$p_{SOS} = 3x_1^4 - 2x_1^2 x_2 + 7x_1^2 - 4x_1 x_2 + 4x_2^2 + 1$$

with degree $p_{SOS} = 2 \cdot d = 4$ in two variables x_1, x_2. Therefore we get the vector

$$X_{d=2} = [1 \quad x_1 \quad x_2 \quad x_1^2 \quad x_1 x_2 \quad x_2^2]^T$$

with length $l_{X_d} = 6$. An SOS decomposition $p = X_d^T M X_d$ of a 4 degree polynomial with our general approach would include all 6 monomials of X_d.

We use the Newton polytope method to prune some unnecessary monomials of this list out . Furthermore is $\mathcal{A} := \{[4,0],[2,1],[2,0],[1,1],[0,2],[0,0]\}$ the set of the monomial degree vectors for the polynomial p_{SOS}. These vectors are shown as circles in the left figure *'Newton Polytope'* [1]. The Newton polytope $N(p_{SOS})$ is the large triangle with vertices $\{[4,0],[0,0],[0,2]\}$. The right figure *'Reduced Newton Polytope'* [1] shows the degree vectors for the six monomials in X_d (circles) and the reduced Newton polytope (large triangle) [1].

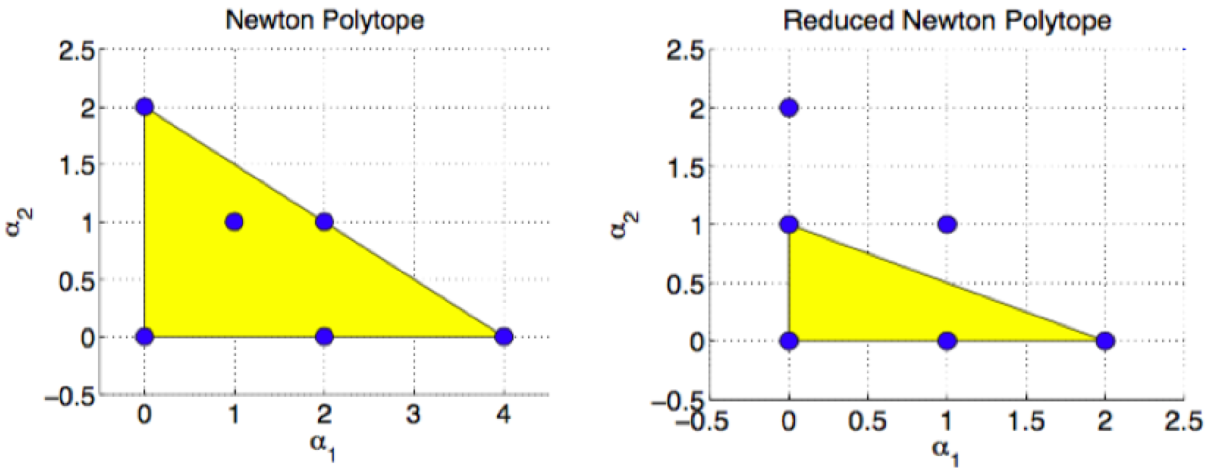
The reduced Newton polytope $\frac{1}{2}N(p_{SOS})$ is the triangle with the vertices $\{[2,0],[0,0],[0,1]\}$. By Theorem 9, x_1x_2 and x_2^2 can not appear in any SOS decomposition of p_{SOS} because $[1,1],[0,2] \notin \frac{1}{2}N(p_{SOS})$. These monomials could be pruned from X_d and the search for an SOS decomposition can be performed using only the four monomials in the reduced Newton polytope:

$$\hat{X}_d = [1 \quad x_1 \quad x_2 \quad x_1^2]^T$$

The length of the reduced vector $\hat{X}_d$ is $l_{\hat{X}_d} = 4$. The SOS feasibility problem with this reduced vector $\hat{X}_d$ is feasible since one possible solution for $p_{SOS} = \hat{X}_d^{\,T} M \hat{X}_d$ is

$$M := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 7 & -2 & 0 \\ 0 & -2 & 4 & -1 \\ 0 & 0 & -1 & 3 \end{pmatrix}$$

with $M \succeq 0$ because $M \in \mathrm{Sym}_4\mathbb{R}$ has the eigenvalues $\lambda_{1,2,3,4} \approx \{1, 2.08, 3.87, 8.04\} \geq 0$. [1]



In our example the construction of the convex hull was straightforward, since we had only two variables x_1, x_2 and therefore was it more a case of 'color by numbers'. But take for example a polynomial with 15 different variables and many terms, at first glance, this seems insoluble or is at least really cumbersome, since there does not exist an 'efficient' algorithm to construct the convex hull for such high dimensionens.

Hence we take another 'way' to reduce the monomial vector X_d. This way is called the zero diagonal algorithm [1]. The algorithm is easy to implement, highly efficient and more powerful than the Newton polytope method. [1]

The reason behind these desirable properties is the characteristic of a psd matrix $0 \preceq M \in \mathrm{Sym}(\mathbb{R})$. If $M_{i,i}$ (in this case is i a scalar and not a vector) is constrained to be zero, then the entire i^{th} row and column is 0 and the associated monomial $(X_d)_i$ could be pruned from the vector X_d [1].

Let $W := \{\alpha_1, \dots, \alpha_{l_{X_d}}\} \subseteq \mathbb{N}^n$ be the set of all degree vectors of X_d (for example is $\alpha_1 = \{0, \dots, 0\}$ and $\alpha_{l_{X_d}} = \{0, \dots, 0, d\}$), then could we write p as [1]

$$p = \sum_{\alpha \in \mathcal{A} \subseteq \mathbb{N}^n} c_\alpha x^\alpha = X_d^T M X_d = \sum_{i=1}^{l_{X_d}} \sum_{j=1}^{l_{X_d}} M_{i,j} x^{\alpha_i + \alpha_j} \tag{2}$$

and therefore do we have linear equality constraints on the entries of the Gram matrix M. The structure of this equations plays an important role in the zero diagonal algorithm. The entries of W are not independent [1], i.e. there exists $a, b, c, d \in \{1, \dots, l_{X_d}\}$ so that $\alpha_a + \alpha_b = \alpha_c + \alpha_d$ and therefore is $(X_d)_a \cdot (X_d)_b = (X_d)_c \cdot (X_d)_d$. This fact shows, that the representation of p from equation (2) is not really tractable. We get a better representation of p by introducing the set [1]

$$W + W := \{\alpha \in \mathbb{N}^n \mid \exists\, \alpha_i, \alpha_j \in W \text{ with } \alpha = \alpha_i + \alpha_j\}$$

of the unique degree vectors. With this new set we can write the polynomial p as [1]

$$p = \sum_{\alpha \in \mathcal{A} \subseteq \mathbb{N}^n} c_\alpha x^\alpha = X_d^T M X_d = \sum_{\alpha \in W+W} \left(\sum_{(i,j) \in S_\alpha} M_{i,j} \right) x^\alpha \tag{3}$$

whereby $S_\alpha := \{(i,j) \mid \alpha_i + \alpha_j = \alpha\}$. This representation has the advantage of the better handiness because we can directly compaire the coefficients c_α of p with the entries of M, or in other words we get the following linear equality constraints [1]

$$\sum_{(i,j) \in S_\alpha} M_{i,j} = \begin{cases} c_\alpha, & \alpha \in \mathcal{A} \\ 0, & \alpha \notin \mathcal{A}. \end{cases} \tag{4}$$

Equations (3) and (4) propose the construction of a matrix $A \in \mathbb{R}^{l_w \times l_X^2}$ (for instance $A_{1-} = [1, 0, \dots, 0]$) and a vector $b \in \mathbb{R}^{l_w}$ who includes the coefficients c_α of p and zeroize [1] (for example is $b_1 = c_{\alpha_1} = c_{(0,\dots,0)}$) so that

$$Aq = b \tag{5}$$

whereby $q = vec(M) \in \mathbb{R}^{l_X^2}$ is the vector obtained by vertically stacking the columns of M [1]. Furthermore is $l_{X_d} = \binom{n+d}{d}$ and $l_w = \binom{n+2d}{2d}$ the dimension of $\mathbb{R}[x]_d$ respectively $\mathbb{R}[x]_{2d}$ (or the number of elements of $W + W$) [1]. Hence we get the implication

$$p = \sum_{\alpha \in \mathcal{A}} c_\alpha x^\alpha = X_d^T M X_d \in \sum[x] \;\Leftrightarrow\; \exists\, q = vec(M) : \; Aq = b \,\wedge\, M \succeq 0. \tag{6}$$

As we have seen in the Newton polytope method, we could reduce the lenght of X_d in some cases and therefore the size of M and consequently the complexity to compute the Gram matrix $M \succeq 0$. This could be also done by the zero diagonal algorithm. The reduction respectively the algorithm bases on the following theorem.

Theorem 12: *If $S_{2\alpha_i} = \{(i,i)\}$ then*

$$M_{i,i} = \begin{cases} c_{2\alpha_i} & 2\alpha_i \in \mathcal{A}, \\ 0 & 2\alpha_i \notin \mathcal{A}. \end{cases}$$

Furthermore if $p = X_d^T M X_d$ with $M \succeq 0$ and $M_{i,i} = 0$ then is $p = \hat{X}_d^T \hat{M} \hat{X}_d$ where $\hat{X}_d \in \mathbb{R}^{l_{X_d}-1}$ is the vector obtained by deleting the i^{th} element of X_d and $0 \preceq \hat{M} \in \mathbb{R}^{(l_{X_d}-1)\times(l_{X_d}-1)}$ is the matrix obtained by deleting the i^{th} row and column from M [1].

Proof. Since $S_{2\alpha_i} = \{(i,i)\}$ is according to equation (4)

$$\sum_{(i,j)\in S_{2\alpha_i}} M_{i,j} = M_{i,i} = \begin{cases} c_{2\alpha_i}, & 2\alpha_i \in \mathcal{A}, \\ 0, & 2\alpha_i \notin \mathcal{A}. \end{cases}$$

Furthermore is for $0 \preceq M \in \mathrm{Sym}_{l_{X_d}}(\mathbb{R})$

$$M_{i,i} = 0 \;\Rightarrow\; M_{-i} = M_{i-} = 0 \tag{7}$$

Assume $M_{i,i} = 0$ but $M_{i,k} = M_{k,i} \neq 0$. Then we construct the new matrix

$$\hat{A}_{ik} := \begin{pmatrix} M_{ii} & M_{ik} \\ M_{ki} & M_{kk} \end{pmatrix} = \begin{pmatrix} 0 & M_{ik} \\ M_{ik} & M_{kk} \end{pmatrix} \in \mathbb{R}^{2\times 2}$$

who is not psd, since $\det(\hat{A}_{ik}) = -M_{ik}^2 < 0$ and the determinant is likewise the product of the eigenvalues and therefore exists eigenvalue $\lambda < 0$, but this is not possible for a psd matrix. Thus exists $\hat{x} \in \mathbb{R}^2$ with

$$\hat{x}^T \hat{A}_{ik} \hat{x} < 0.$$

Furthermore we set the vector $X := [0, \ldots, 0, \hat{x}_1, 0, \ldots, 0, \hat{x}_2, 0, \ldots, 0]$ whereby $X_i = \hat{x}_1$ and $X_k = \hat{x}_2$. Thereby follows

$$X^T M X = \hat{x}^T \hat{A}_{ik} \hat{x} < 0$$

and consequently M is not psd, though this is in contrast to the constraint $M \succeq 0$.

According to equation (3) and implication (7) is for $M_{k,k} = 0$

$$p = \sum_{i=1}^{l_{X_d}} \sum_{j=1}^{l_{X_d}} M_{i,j} x^{\alpha_i + \alpha_j} = \left(\sum_{i \in \{1,\dots,l_{X_d}\} \setminus \{k\}} \right) \left(\sum_{j \in \{1,\dots,l_{X_d}\} \setminus \{k\}} \right) M_{i,j} x^{\alpha_i + \alpha_j}$$

and therefore the monomial $(X_d)_k$ is not included in any term of p and consequently redundant [1]. $\square$

In the following we will give a short overview how to implement the zero diagonal algorithm and some explanations concerning the single steps [1].

1. **Given:** A polynomial $p = \sum_{\alpha \in \mathcal{A}} c_\alpha x^\alpha$
2. **Initialization:** Set $k = 0$ and $G_0 := \{\alpha_i\}_{i=1}^{l_{X_d}} \subseteq \mathbb{N}^n$
3. **Form** $Aq = b$: Construct the equality constraint data, $A \in \mathbb{R}^{l_w \times l_{X_d}^2}$ and $b \in \mathbb{R}^{l_w}$, obtained by equating coefficients of $p = X_d^T Q X_d$.
4. **Iteration:**
5. Set $Z = \emptyset$, $k := k + 1$, $G_k := G_{k-1}$
6. Search $Aq = b$: If there is an equation of the form $Q_{i,i} = 0$ then set $G_k := G_k \setminus \{\alpha_i\}$ and $Z = Z \cup \mathcal{I}$ where $\mathcal{I}$ are the entries of q corresponding to the i^{th} row and column of Q.
7. For each $j \in Z$ set j^{th} column of A equal to zero.
8. Terminate if $Z = \emptyset$ otherwise return to step 5.
9. **Return:** G_k, A, b

The set G_k denotes the pruned list of monomial degree vectors at the k^{th} iterate. Further is $l_{X_d} = \binom{n+d}{d}$ and $l_w = \binom{n+2d}{2d}$. The main step in the iteration is the search for equations that directly constrain a diagonal entry $Q_{i,i}$ to be zero (Step 6). Based on Theorem 12 and implication (8), if $Q_{i,i} = 0$ then the monomial z_i and the i^{th} row and column of Q can be removed. This is equivalent to zeroing out the corresponding columns of A (Step 7)(For example if $Q_{1,1} = 0$, then we zeroing out the first l_{X_d} columns and the $(1 + k \cdot l_{X_d})^{th}$ column of A for $k \in \{1, \dots, l_{X_d} - 1\}$).

This implementation has the advantage that A and b do not need to be recomputed for each updated set G_k. Zeroing out columns of A in Step 7 also means that new equations of the form $Q_{i,i} = 0$ may be uncovered during the next iteration. The iteration continues until no new zero diagonal entries of Q are discovered.

Remark: The columns of the set Z can/should be deleted of A prior to passing the data to a semi-definite programming solver (a solver, who is solving the equations $Aq = b$ with the constraint that $0 \preceq Q$ for $q = vec(Q)$), since this improves the efficiency and stability [1].

Theorem 13: *The zero diagonal algorithm terminates in a finite number of steps k_f, and $G_{k_f} \subseteq \left(\frac{1}{2}N(p) \cap \mathbb{N}^n\right)$ is the resulting set of monomials by the zero diagonal algorithm, which are not necessarily redundant. Moreover if $p = \sum_{i=1}^m f_i^2 \in \sum[x]$, then $(N(f_i) \cap \mathbb{N}^n) \subseteq G_{k_f}$.* [1]

Proof. G_0 (the initial set with all monomials of degree $\leq d$) has l_{X_d} elements. The algorithm terminates unless at least one point is removed from G_k. Thus the algorithm must terminate after $k_f \leq l_{X_d} + 1$ steps.

To show $G_{k_f} \subseteq \left(\frac{1}{2}N(p) \cap \mathbb{N}^n\right)$ consider a vertex α_i of convhull(G_{k_f}). If there exists $u, v \in$ convhull(G_{k_f}) such that $2\alpha_i = u + v$ then is $\alpha_i = \frac{1}{2}(u + v)$ and therefore follows $u = v = \alpha_i$ for a vertex α_i. Consequently is $S_{2\alpha_i} = \{(i, i)\}$ (for the set G_{k_f}) and by Theorem 12

$$M_{i,i} = \begin{cases} c_{2\alpha_i} & 2\alpha_i \in \mathcal{A} \\ 0 & 2\alpha_i \notin \mathcal{A} \end{cases}$$

follows $M_{i,i} \neq 0$ (not necessarily) since $\alpha_i \in G_{k_f}$, that means α_i was not removed during the final iteration of the zero diagonal algorithm and therefore is $M_{i,i}$ not constrained to be zero. Consequently is $2\alpha_i \in \mathcal{A} \subseteq N(p)$ and this implies $\alpha_i \in \frac{1}{2}N(p)$. Therefore $\frac{1}{2}N(p)$ contains all vertices of convhull(G_{k_f}) and we get the implication

$$\left(\mathbb{N}^n \cap G_{k_f}\right) = G_{k_f} \subseteq \text{convhull}(G_{k_f}) \subseteq \frac{1}{2}N(p),$$

which implicates $G_{k_f} \subseteq \left(\frac{1}{2}N(p) \cap \mathbb{N}^n\right)$.

According to Theorem 9 is $N(f_i) \subseteq \frac{1}{2}N(p)$ and $\frac{1}{2}N(p) \subseteq$ convhull(G_0) since each vertex of $\frac{1}{2}N(p)$ is included in G_0. This implies $(N(f_i) \cap \mathbb{N}^n) \subseteq (\text{convhull}(G_0) \cap \mathbb{N}^n) = G_0$. If $p = \sum_{i=1}^m f_i^2$ then there exists a matrix $0 \preceq M \in \text{Sym}_{l_{X_d}}(\mathbb{R})$ with $p = X_d^T M X_d$. If the iteration removes no degree vectors then $G_{k_f} = G_0$ and the proof is complete.

Assume the iteration removes at least on degree vector and let α_i be the first removed degree vector. Then is $M_{i,i} = 0$. By Theorem 9 the monomial $(X_d)_i$ cannot appear in any f_j. Hence $(N(f_j) \cap \mathbb{N}^n) \subseteq G_0 \setminus \{\alpha_i\}$. Therefore is $(N(f_j) \cap \mathbb{N}^n) \subseteq G_{k_f}$, because for each $y \in \{0, \ldots, k_f\}$ is $(N(f_j) \cap \mathbb{N}^n) \subseteq G_y$. [1] □

Remark: In our decomposition of $p = z^T M z = X_d^T M X_d$ in Theorem 13 is z not constrained to be $z = X_d$, this means we could z initialize with less monomials (if we would know in advance, that some monomials are redundant), i.e. $l_z < l_X = \binom{n+d}{d}$. The proof of Theorem 13 for $z \neq X_d$ with $\left(\frac{1}{2}N(p) \cap \mathbb{N}^n\right) \subseteq G_0$ is exactly the same apart from some adaptions.

Theorem 13 shows that the zero diagonal algorithm is more powerful than the Newton polytope method, since $G_{k_f} \subseteq \left(\frac{1}{2}N(p) \cap \mathbb{N}^n\right)$ [1]. We will show the scheme of the zero diagonal algorithm and the fact of Theorem 13 by two short examples.

Example:[1] Consider again the polynomial $p_{SOS} = 3x_1^4 - 2x_1^2 x_2 + 7x_1^2 - 4x_1 x_2 + 4x_2^2 + 1$ [1]. For the decomposition of $p_{SOS} = z^T Q z = X_d^T M X_d$ we initialize the vector $z := X_2$ with all momonials of degree ≤ 2 and $l_z = l_{X_2} = \binom{2+2}{2} = 6$. Equating the coefficients of p_{SOS} and $z^T Q z$ yields the following linear equality constraints on the entries of Q:

$$
\begin{aligned}
Q_{2,1} + Q_{1,2} &= 0, & Q_{4,1} + Q_{1,4} + Q_{2,2} &= 7 \\
Q_{4,2} + Q_{2,4} &= 0, & Q_{6,4} + Q_{4,6} + Q_{5,5} &= 0 \\
Q_{3,1} + Q_{1,3} &= 0, & Q_{6,1} + Q_{1,6} + Q_{3,3} &= 4 \\
Q_{5,4} + Q_{4,5} &= 0, & Q_{5,2} + Q_{2,5} + Q_{4,3} + Q_{3,4} &= -2 \\
Q_{6,3} + Q_{3,6} &= 0, & Q_{6,2} + Q_{2,6} + Q_{5,3} + Q_{3,5} &= 0 \\
Q_{6,5} + Q_{5,6} &= 0, & Q_{5,1} + Q_{1,5} + Q_{3,2} + Q_{2,3} &= -4 \\
Q_{1,1} &= 1 & Q_{4,4} &= 3 \\
Q_{6,6} &= 0
\end{aligned}
$$

As we have mentioned a matrix $A \in \mathbb{R}^{15 \times 6^2}$ and vector $b \in \mathbb{R}^{15}$ can be constructed to represent these equations in the form $Aq = b$ with $q = vec(M)$. Note that $Q_{6,6} = 0$ and this implies that $Q_{i,6} = Q_{6,i} = 0$ with $i \in \{1, \ldots, 6\}$ for any SOS decomposition of p_{SOS}. Thus the monomial $z_6 = x_2^2$ can not appear in any SOS decomposition and it can be removed from the list. After eliminating x_2^2 and removing the 6^{th} row and column of Q, the equality constraints reduce to

$$
\begin{aligned}
Q_{2,1} + Q_{1,2} &= 0, & Q_{4,1} + Q_{1,4} + Q_{2,2} &= 7 \\
Q_{4,2} + Q_{2,4} &= 0, & Q_{5,5} &= 0 \\
Q_{3,1} + Q_{1,3} &= 0, & Q_{3,3} &= 4 \\
Q_{5,4} + Q_{4,5} &= 0, & Q_{5,2} + Q_{2,5} + Q_{4,3} + Q_{3,4} &= -2 \\
Q_{5,3} + Q_{3,5} &= 0 & Q_{5,1} + Q_{1,5} + Q_{3,2} + Q_{2,3} &= -4 \\
Q_{1,1} &= 1 & Q_{4,4} &= 3
\end{aligned}
$$

Removing the 6^{th} row and column of Q is equivalent to zeroing out the appropriate columns of the matrix A. This uncovers the new constraint $Q_{5,5} = 0$ which implies that the monomial $z_5 = x_1 x_2$ can be pruned from the list. After eliminating $x_1 x_2$, the procedure can be repeated once again after removing the 5^{th} row and column of Q. No new diagonal entries of Q are constrained to be zero and hence no additional monomials can be pruned from z. The final list G_3 of monomials consists of four monomials.

$$z = [1 \quad x_1 \quad x_2 \quad x_1^2]^T$$

The Newton polytope method returned the same list.

Example: [1] Consider the polynomial $p = x_1^2 + x_2^2 + x_1^4 x_2^4$ [1] in two variables and degree 8. The Newton polytope is $N(p) = \text{convhull}(\{[2, 0], [0, 2], [4, 4]\})$ and the reduced

Newton polytope is $\frac{1}{2}N(p) = \text{convhull}(\{[1,0],[0,1],[2,2]\})$. Further the monomial vector corresponding to $\frac{1}{2}N(p) \cap \mathbb{N}^n$ is

$$z := [x_1 \quad x_2 \quad x_1x_2 \quad x_1^2x_2^2]^T.$$

There are $l_z = l_{X_4} = \binom{2+4}{4} = 15$ monomials in two variables with degree ≤ 4. For simplicity, assume the zero diagonal algorithm is initialized with $G_0 := \frac{1}{2}N(p) \cap \mathbb{N}^n$. Equating the coefficients of p and z^TQz yields the contraint $Q_{3,3} = 0$ in the first iteration of the zero diagonal algorithm. The monomial $z_3 = x_1x_2$ is pruned and no additional monomials are removed at the next iteration.

The zero diagonal algorithm returns $G_2 = \{[1,0],[0,1],[2,2]\}$. G_2 is a proper subset of $\frac{1}{2}N(p) \cap \mathbb{N}^n$. The same set of monomials is returned by the zero diagonal algorithm after 13 steps if G_0 is initialized with the $l_z = 15$ degree vectors corresponding to all possible monomials in two variables with degree ≤ 4. This example demonstrates that the zero diagonal algorithm can return a strictly smaller set of monomials than the Newton polytope method.

As we have seen, the zero diagonal algorithm yields the same list, or sometimes a smaller list, like the Newton polytope method [1]. But is this list always optimal? That means it is not possible to prune some monomials of G_{k_f} so that still $p = z^TQz$ for $Q \succeq 0$ and a vector z, whose entries include less monomials than the list G_{k_f}. Or in other words

$$z_i = 0 \Leftrightarrow Q_{i,i} = 0.$$

Consider the polynomial $p_{nop} = 1 + x_1^2 + x_1^2x_2^2 + x_1^4 + x_2^4 \in \sum[x]$ who is obviously a sum of squares. We initialize the vector $z := X_2 = [1, x_1, x_2, x_1x_2, x_1^2, x_2^2]^T$ with the set $G_0 = \{1, x_1, x_2, x_1x_2, x_1^2, x_2^2\}$ for the decomposition $p_{nop} = z^TQz$ for $0 \preceq Q \in \mathbb{R}^{6\times 6}$. Since

$$p_{nop} = \sum_{\alpha \in \{(0,0),(2,0),(2,2),(4,0),(0,4)\}} c_\alpha x^\alpha = X_2^TQX_2 = \sum_{x^\alpha \in \{(X_4)_1,\ldots,(X_4)_{15}\}} \left(\sum_{(i,j)\in S_\alpha} Q_{i,j} \right) x^\alpha$$

we get the constraints (we forgo to note all equations, because these 3 equations are more then sufficient to show the result)

$$Q_{1,1} = c_{0,0} = 1, \qquad Q_{3,3} + Q_{6,1} + Q_{1,6} = 0, \qquad Q_{6,6} = c_{0,4} = 1.$$

Thus is $Q_{1,1} = Q_{6,6} = 1$ and $Q_{6,1}$, $Q_{1,6}$ are not constrained to be zero, since the zero diagonal algorithm is not erasing the first and sixth column respectively row. Moreover is $Q_{3,3}$ not necessarily zero and consequently terminates the zero diagonal algorithm with the set $G_{k_f} = G_0$. One possible decomposition of p_{nop} is

$$p_{nop} = X_2^T \cdot \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot X_2$$

and for $Q_{3,3} = Q_{6,1} = Q_{1,6} = 0$ we get the pruned decomposition

$$p_{nop} = [1, x_1, x_1x_2, x_1^2, x_2^2] \cdot \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot [1, x_1, x_1x_2, x_1^2, x_2^2]^T$$

Both matrices are psd, since their eigenvalues ≥ 0. Moreover the monomial x_2 could pruned from the resulting list $G_{k_f} = G_0$ and as a consequence is the zero diagonal algorithm not optimal. Besides of this fact, our example shows if the polynomial p has a 'certain' structure, then the zero diagonal algorithm is even futile, since no element was removed from $G_0 = \{(X_d)_1, \ldots, (X_d)_{l_{X_d}}\}$.

Remark: The property of not beeing optimal is a direct corollary of equation (3), since the monomials in X_d are not linear independent [1] and therefore is $z_i = 0 \not\Leftrightarrow Q_{i,i} = 0$.

Now, in the following chapter we will understand why we stressed so much the connection of nonnegativity and SOS and show further the usefulness of the zero diagonal algorithm.

IV. Simplification Method For Sum Of Squares Programs

As we have mentioned in the introduction, to check if a polynomial $p \geq 0$ is really interesting in optimization [12]. Consider the following situation:
We have a function p_c, who describes for example the cost of a product. It would be desirable to produce a cheap product and therefore we are looking for the minimum of p_c. Furthermore we have some constraints on the cost function p_c. These constraints can be expressed as polynomial inequalities and hence as a set of nonnegative polynomials. This situation looks as follow:

$$\begin{aligned} &\min_{u \in \mathbb{R}^r} p_c(u) \\ &\text{subject to: } \tilde{p}_1 - p_1 \geq 0, \ldots, \tilde{p}_N - p_N \geq 0 \end{aligned}$$

whereby $\tilde{p}_k - p_k := \hat{p}_k := a_k(x,u) = a_{k,0}(x) + \sum_{i=1}^r u_i a_{k,i}(x)$. Assume that $p_c := c^T u$ and therefore we get the optimization problem

$$\begin{aligned} &\min_{u \in \mathbb{R}^r} p_c(u) = c^T u \\ &\text{subject to: } \hat{p}_k := a_k(x,u) = a_{k,0}(x) + \sum_{i=1}^r u_i a_{k,i}(x) \geq 0, \qquad k \in \{1, \ldots, N\}. \end{aligned}$$

To check if $\hat{p}_k \geq 0$ could be NP-hard [7] and hence the optimization problem is not really tractable. This problem 'suggests' another approach. If we substitue the constraint $\hat{p}_k \geq 0$ by $\hat{p}_k \in \sum[x]$, the problem is feasible but there remains a question if this substitution is not to strong, i.e. we do not get the same minimum or at least a useful one.

According to the lemmas of Lasserre and Netzer the cone of the SOS polynomials is dense in the cone of the nonnegative polynomials [7] and besides of these two lemmas Parillo and Sturmfels have shown by some computation [9], that most nonnegative polynomials are sum of squares. These two facts propose the idea, that we get in most cases a useful minimum, i.e. our new solution u_{SOS} coincides with the solution $u_{nonnegative}$ from the origin problem or is not far 'away'. This idea motivates the following optimization problem

$$\begin{aligned} &\min_{u \in \mathbb{R}^r} p_c(u) = c^T u \\ &\text{subject to: } \hat{p}_k = a_k(x,u) = a_{k,0}(x) + \sum_{i=1}^r u_i a_{k,i}(x) \in \sum[x], \qquad k \in \{1, \ldots, N\} \end{aligned}$$

Problems of this structure are called <u>sum of squares programs</u> [1], [12]. The variables $u_1, \ldots, u_r \in \mathbb{R}$ were denoted as <u>decision variables</u> and the polynomials $\{a_k\}_{k=1}^N$ are given problem data and are affine in u [1]. According to Theorem 6 we get another presentation of sum of squares programs [1]

$$\begin{aligned} &\min_{u \in \mathbb{R}^r} p_c(u) = c^T u \\ &\text{subject to: } \hat{p}_k = a_k(x,u) = z_k^T Q_k z_k, \qquad 0 \preceq Q_k \in \mathrm{Sym}(\mathbb{R}),\ k \in \{1, \ldots, N\} \end{aligned}$$

for an appropriate monomial vector z_k. Clearly, Q_k is a new matrix of decision variables which is obtained by equating the coefficients of $a_k(x,u)$ and $z_k^T Q_k z_k$ [1].

Furthermore exists a matrix $A_u \in \mathbb{R}^{l_w \times m}$ and a vector $b \in \mathbb{R}^{l_w}$ so that [1]

$$A_u y_u = b,$$

whereby $y_u := [u^T, vec(Q_1)^T, \ldots, vec(Q_N)^T]^T$ [1] (this is similar to the case $Aq = b$, though a little bit more complex because of decision variables and the N different constraints). The dimension m is equal to $r + \sum_{i=1}^{N} m_k^2$ where Q_k is $m_k \times m_k$. After introducing a Gram matrix for each constraint the SOS program can be expressed as [1]

$$\begin{aligned} &\min_{u \in \mathbb{R}^r} p_c(u) = c^T u \\ &\text{subject to: } A_u y_u = b, \qquad Q_k \succeq 0,\ k \in \{1, \ldots, N\}. \end{aligned}$$

Specifically, the constraints in some SOS programs imply both $u_i \geq 0$ and $u_i \leq 0$, i.e. this is an implicit constraint that $u_i = 0$ for some $i \in \{1, \ldots, r\}$ and consequently $a_{k,i}$ for $k \in \{1, \ldots, N\}$ could be pruned out [1]. This reduction 'simplificates' our problem and therefore is this procedure called simplification method for SOS programs.

This method is a generalization of the zero diagonal algorithm, since it removes next to the pruned monomials also the free decision variables, who are implicitly constrained to be zero [1]. To better understand this method we will give a short example (although it does not include simplification).

Example:[Parrilo] Consider the one dimensional polynomial $f_{\alpha,\beta} = (x^4+1) + \alpha(x^3 - x) + \beta(3x^3 + 2x^2) = x^4 + (\alpha+3\beta)x^3 + 2\beta x^2 - \alpha x + 1$. We want to find a SOS decompositon and get therefore the equation

$$\begin{aligned} f_{\alpha,\beta}(x) &= 1 - \alpha x + 2\beta x^2 + (\alpha+3\beta)x^3 + x^4 \\ &= \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}^T \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{12} & q_{22} & q_{23} \\ q_{13} & q_{23} & q_{33} \end{bmatrix} \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix} \\ &= q_{11} + 2q_{12}x + (q_{22} + 2q_{13})x^2 + 2q_{23}x^3 + q_{33}x^4 \end{aligned}$$

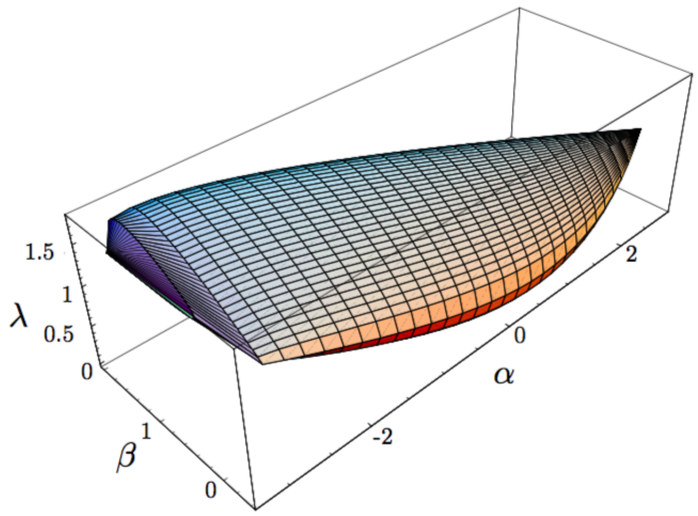
with the constraint $0 \preceq Q \in \mathrm{Sym}_3(\mathbb{R})$. Hence we get the feasible set

$$\left\{ (\alpha, \beta) \quad | \quad \exists \lambda \text{ s.t.} \begin{bmatrix} 1 & -\frac{1}{2}\alpha & \beta - \lambda \\ -\frac{1}{2}\alpha & 2\lambda & \frac{1}{2}(\alpha+3\beta) \\ \beta - \lambda & \frac{1}{2}(\alpha+3\beta) & 1 \end{bmatrix} \succeq 0 \right\}$$

with a possible matrix $A_{\alpha,\beta}$ and vector b:

$$A_{\alpha,\beta} = \begin{bmatrix} A^1_{\alpha,\beta} & A^2_{\alpha,\beta} \end{bmatrix}, \; A^1_{\alpha,\beta} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & -2 \\ -1 & -3 \\ 0 & 0 \end{bmatrix}, \; A^2_{\alpha,\beta} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \; b = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

and as a graphical solution



You should recall in the univariat case holds for a polynomial $p \geq 0 \Leftrightarrow p \in \sum[x]$ [5].

The implementation could look as follow (to ease the notation we set $N = 1$, for $N > 1$ is the implementation straightforward)[$a_{i,j}$ is the coefficient from the monomial x^{α_i} of the polynomial a_j][1]:

1. **Given:** Polynomials $\{a_j\}_{j=1}^r$ in variables x. Define
   $a(x,u) := a_0(x) + a_1(x)u_1 + \ldots + a_r(x)u_r$
2. **Initialization:** Set $k = 0$ and choose a finite set $G_0 := \{\alpha_i\}_{i=1}^m \subseteq \mathbb{N}^n$ such that
   $\left[\cup_{u \in \mathbb{R}^r} \frac{1}{2} N(a(x,u))\right] \cap \mathbb{N}^n \subseteq G_0$.
3. **Form $A_u y_u = b$:** Construct the equality constraint data, $A_u \in \mathbb{R}^{l_w \times (r+m^2)}$ and
   $b \in \mathbb{R}^{l_w}$, obtained by equating coefficients of $a(x,u) = z^T Q z$, where
   $z := [x^{\alpha_1}, \ldots, x^{\alpha_m}]^T$ and $y_u = \left[u^T, vec(Q)^T\right]^T$.
4. **Sign Data:** Initialize the $r + m^2$ vector s to be $s_i = +1$ if $(y_u)_i$ corresponds to a
   diagonal entry of Q. Otherwise set $s_i = \textbf{NaN}$.
5. **Iteration:**
6. Set $Z = \emptyset$, $S = \emptyset$, $k := k + 1$, $G_k := G_{k-1}$
7. Process equality constraints of the form $a_{i,j}(y_u)_j = b_i$ where $a_{i,j} \neq 0$

7a. If $b_i = 0$ then set $s_j = 0$ and $Z = Z \cup j$
7b. Else if $s_j = \mathbf{NaN}$ then set $s_j = \text{sign}(a_{i,j}b_i)$ and $S = S \cup j$
7c. Else if $s_j = -1$ and $\text{sign}(a_{i,j}b_i)$=$+1$ then set $s_j = 0$ and $S = S \cup j$
7d. Else if $s_j = +1$ and $\text{sign}(a_{i,j}b_i)$= -1 then set $s_j = 0$ and $S = S \cup j$
8. If for any $j \in Z$, $(y_u)_j$ corresponds to a diagonal entry $Q_{i,i}$ then set $G_k := G_k \setminus \{\alpha_i\}$ and $Z = Z \cup \mathcal{I}$ where $\mathcal{I}$ are the entries of y_u. corresponding to the i^{th} row and column of Q.
9. For each $j \in Z$ set the j^{th} column of A_u equal to zero.
10. Terminate if $Z = \emptyset$ and $S = \emptyset$ otherwise return to step 6.
11. **Return:** G_k, A_u, b, s

The algorithm is initialized with a finite set of vectors $G_0 \subseteq \mathbb{N}^n$, whereby G_0 must be chosen so that it contains all possible reduced Newton polytopes, because the Newton polytope of $a(x, u)$ depends on the choice of u [1]. One choice is to initialize G_0 corresponding to the degree vectors of all monomials in n variables and degree $\leq 2d := \max_u[\deg a_k(x, u)]$ [1].

Since A_u and b need to be computed when formulating the semidefinite program constraints, this step does not require additional computation associated with the simplification procedure. The last pre-processing step is the initialization of the sign vector s. The entries of s_i are $+1$, -1, or 0 if it can be determined from the constraints that $(y_u)_i$ is ≥ 0, ≤ 0 or $= 0$, respectively [1]. $s_i = \mathbf{NaN}$ if no sign information can be determined for $(y_u)_i$. If $(y_u)_i$ corresponds to a diagonal entry of Q then s_i can be initialized to $+1$, since the diagonal entries $q_{i,i} \geq 0$ of a psd matrix $0 \preceq Q \in \text{Sym}(\mathbb{R})$ [1].

The main iteration step is the search for equations that directly constrain any decision variable to be zero (Step 7a). This is similar to the zero diagonal algorithm. The iteration also attempts to determine sign information about the decision variables. Steps 7b-7d update the sign vector based on equality constraints involving a single decision variable. For example, a decision variable must be zero if the decision variable has been previously determined to be ≤ 0 and the current equality constraint implies that it must be ≥ 0 (Step 7c).

These decision variables can be removed from the optimization. Steps 8 and 9 prune monomials and zero out appropriate columns of A_u. The iteration continues until no additional information can be determined about the sign of the decision variables [1].

This SOS simplification procedure automatically uncovers some implicitly constrained u_i to be zero and removes these decision variables from the optimization. This is important because implicit constraints can cause numerical issues for SDP solvers. A significant reduction in computation time and improvement in numerical accuracy has been observed when implicitly constrained variables are removed prior to calling a solver[1].

In the next chapter we will give some examples of interesting applications of the zero diagonal algorithm

V. Applications

Global bounds for polynomial functions: Given a multivariate polynomial function $f \in \mathbb{R}[x]$ which is bounded below on $\mathbb{R}^n$. We want to find the global minimum f^* and a point x^* attaining it:

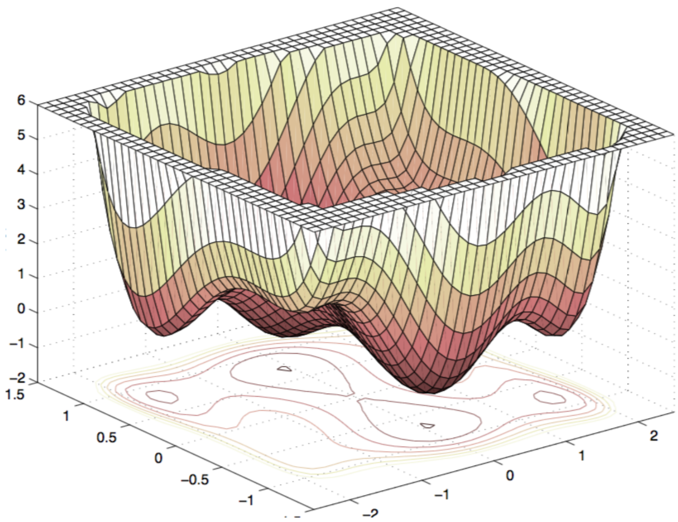
$$f^* = f(x^*) = \min\{f(x) : x \in \mathbb{R}^n\}.$$

This problem is NP-hard (at least for polynomials with degree ≥ 4). As mentioned Parrillo and Sturmfels haven taken another way[9]. They have searched for the largest value $\gamma \in \mathbb{R}$ such that $f(x) - \gamma$ is a sum of squares in $\mathbb{R}[x]$. In their experiments was in almost any case $\gamma = f^*$. Clearly, γ is a lower bound for the optimal value f^* [9]. The condition

$$\begin{aligned} &\min \gamma \\ &\text{subject to: } f(x) + \gamma \in \sum[x] \end{aligned}$$

is a sum of squares program, since we have the cost function $p_c(u) = \gamma$ with $c = 1$, $u = \gamma$ and $f(x) = a_0(x)$, $a_1(x) = 1$. Hence we get for $a(x, u) = a_0(x) + ua_1(x) = f(x) + \gamma$ (obviously the problem of finding the minimum for a polynomial must be a sum of squares program, because we have taken the same approach to motivate sum of squares programs).

Consider again the polynomial $p_{negative} = 4x_1^2 - \frac{21}{10}x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$ of the introduction [10]. We want to find the global minimum, but this task could become highly difficult, since it has many local minimums (see figure below). Therefore we apply the sum of squares program method and get for $\gamma \approx 1.03162845$ [10]. This turns out to be the exact global minimum, since that value is achieved for $x_1 \approx 0.089842$, $x_2 \approx -0.7126564$ [10]. Since $\gamma = f^*$ and $0 < \gamma \approx 1.03162845$, is $p_{negative}$ not psd.



Although we have stressed, that γ mostly coincides with f^*, though we want to give two counter examples [10]. Besides the bivariate motzkin polynomial, there exists a 3 dimensional motzkin polynomial $M(x, y, z) = x^4y^2 + x^2y^4 + z^6 - 3x^2y^2z^2$ [1]. This polynomial is also nonnegative since it attains it minimum $0 = M(1, 1, 1)$ [10]. Solving the corresponding SDPs, the best lower bound that can be obtained this way can be shown to be $-\frac{729}{4096} \approx -0.177978$, and follows from the decomposition [10]

$$M(x, 1, z) = x^4 + x^2 + z^6 - 3x^2z^2 + \frac{729}{4096} = (-\frac{9}{8}z + z^3)^2 + (\frac{27}{64} + x^2 - \frac{3}{2}z^2)^2 + \frac{5}{32}x^2.$$

This is a significant gap between $f^* = 0$ and $\gamma = \frac{729}{4096}$. But it could come worse. Take for instance the motzkin polynomial $p_{motzkin} = x_1^4x_2^2 + x_1^2x_2^4 - 3x_1^2x_2^2 + 1$. There does not exist any γ, so that $p_{motzkin} + \gamma \in \sum[x]$. This follows from Theorem 10 (motzkin polynomial is not a sum of squares) and the remark below Theorem 10 [2]. This example shows, that the gap could be infinite, although these examples are sparsely distributed.

Geometry: [10] In this problem, we compute a lower bound on the distance between a given point (x_0, y_0) and an algebraic curve $C(x, y) = 0$. Take $(x_0, y_0) = (1, 1)$, and let the algebraic curve be

$$C(x, y) := x^3 - 8x - 2y = 0.$$

In this case, we can formulate the optimization problem

$$\min_{C(x,y)=0} (x - 1)^2 + (y - 1)^2$$

and define the sum of squares program as follow

$$\min -\gamma^2 \tag{8}$$

$$\text{subject to: } (x - 1)^2 + (y - 1)^2 - \gamma^2 + (\alpha + \beta x)(x^3 - 8x - 2y) \in \sum[x, y]. \tag{9}$$

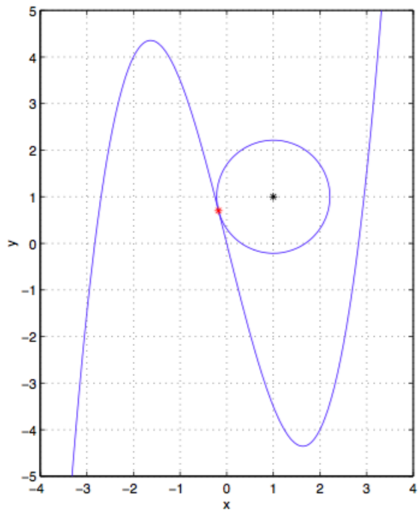
It should be clear that if condition (9) holds, then each point (x, y) in the curve are at a distance at least equal to γ from (x_0, y_0). To see this, note that if the point (x, y) is in the curve $C(x, y) = 0$, then the last term in (9) vanishes, and therefore $(x - 1)^2 + (y - 1)^2 \geq \gamma^2 \geq \gamma$. The expression is affine in α, β and γ^2 and so the problem can be directly solved using SDP [10]. (We need the polynomial $(\alpha+\beta x)$ to lower bound $(x^3 - 8x - 2y)$).

Furthermore we have $a_0(x, y) = (x - 1)^2 + (y - 1)^2$, $a_1 = -1$, $a_2 = x^3 - 8x - 2y$, $a_3 = x(x^3 - 8x - 2y)$ and $u_1 = \gamma^2$, $u_2 = \alpha$, $u_3 = \beta$ next to $c_1 = -1$, $c_2 = 0$, $c_3 = 0$.
The optimal solution of the SDPs is [10]:

$$\alpha \approx -0.28466411, \beta \approx 0.07305057, \ \gamma \approx 1.47221165.$$

The obtained bound γ is sharp, since it is achieved by the values [10]

$$x \approx -0.176299246, \ y \approx 0.702457168, \qquad (x, y) \in C = 0$$



The curve $C(x, y) = 0$ and the minimum distance circle.

Matrix copositivity: [10] A symmetric matrix $J \in \mathbb{R}^{n \times n}$ is said to be copositive if the associated quadratic form takes only nonnegative values on the nonnegative orthant, or in other words:

$$x_i \geq 0, \ i \in \{1, \ldots, n\} \Rightarrow x^T J x \geq 0.$$

As opposed to positive definiteness, which can be efficiently verified, checking if a given matrix is not copositive is an NP-complete problem [10]. The main difficulty in obtaining conditions for copositivity is dealing with the constraints in the variables, since each x_i has to be nonnegative. Therefore we set $x_i = z_i^2$ and study the global nonnegativity of the fourth order form given by [10]:

$$P(z) := z^T J z = \sum_{i,k} j_{i,k} z_i^2 z_k^2$$

where $z = [z_1^2, z_2^2, \ldots, z_n^2]^T$. It is easy to verify that J is copositive if and only if the form $P(z)$ is positive semidefinite [10]. Therefore, sufficient conditions for $P(z)$ to be nonnegative will translate into sufficient conditions for J being copositive. Consider the matrix [10]

$$J = \begin{bmatrix} 1 & -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 \end{bmatrix}$$

Unfortunately, P is not a sum of squares. But we can try to use the same strategy of the Motzkin polynomial, i.e. we use the result of the lemma from Reznick and check if

$$\left[\left(\sum_{i=1}^{n} z_i^2\right)^r \cdot P\right] \in \sum[z] \text{ for some } r \in \mathbb{N}. \tag{10}$$

And indeed, for $r = r_{min} = 1$ holds condition (10). Hence J is copositive [10]. Obviously for $r > r_{min}$ (10) is also satisfied.

Lyapunov function search:[4] The Lyapunov stability theorem has been a cornerstone of nonlinear system analysis for several decades. In principle, the theorem states that a system $\dot{x} = f(x)$, whereby $x := x(t)$, with equilibrium at the origin is stable if there exists a positive definite function $V(x)$ such that the derivative of V along the system trajectories is non-positive. We will now show how the search for a Lyapunov function can be formulated as a sum of squares program [4]. For our example, consider the system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -x_1^3 - x_1 x_3^2 \\ -x_2 - x_1^2 x_2 \\ -x_3 - \frac{3x_3}{x_3^2+1} + 3x_1^2 x_3 \end{bmatrix}$$

which has an equilibrium at the origin. Now assume that we are interested in a quadratic Lyapunov function $V(x)$ for proving stability of the system. Then $V(x)$ must satisfy [4]

$$V - \epsilon(x_1^2 + x_2^2 + x_3^2) \geq 0,$$
$$-\frac{\partial V}{\partial x_1}\dot{x}_1 - \frac{\partial V}{\partial x_2}\dot{x}_2 - \frac{\partial V}{\partial x_3}\dot{x}_3 \geq 0.$$

The first inequality, with ϵ being any constant greater than zero (in what follows we will choose $\epsilon = 1$), is needed to guarantee positive definiteness of $V(x)$. We will formulate a SOS program that computes a Lyapunov function for this system, by replacing the above nonnegativity conditions with SOS conditions.

However, notice that $\dot{x}_3$ is a rational function, and therefore is the second condition not a polynomial expression. But since $x_3^2 + 1 > 0$ for any x_3, we can just reformulate [4]

$$(x_3^2 + 1)\left(-\frac{\partial V}{\partial x_1}\dot{x}_1 - \frac{\partial V}{\partial x_2}\dot{x}_2 - \frac{\partial V}{\partial x_3}\dot{x}_3\right) \geq 0.$$

Next, we parameterize the candidate quadratic Lyapunov function $V(x)$ by some unknown real coefficients $c_1, \ldots, c_6$ and get

$$V(x) = c_1 x_1^2 + c_2 x_1 x_2 + \ldots + c_6 x_3^2,$$

and the following SOS program (with no cost function) can be formulated as [4]:
Find a polynomial $V(x)$, (equivalently, find $c_1, \ldots, c_6$) such that

$$\begin{aligned} &V(x) - (x_1^2 + x_2^2 + x_3^2) \text{ is SOS} \\ &(x_3^2 + 1)\left(-\frac{\partial V}{\partial x_1}\dot{x_1} - \frac{\partial V}{\partial x_2}\dot{x_2} - \frac{\partial V}{\partial x_3}\dot{x_3}\right) \text{ is SOS.} \end{aligned}$$

The SDP solver gets the solution $V(x) = 5.5489x_1^2 + 4.1068x_2^2 + 1.7945x_3^2$ as a Lyapunov function and that proves the stability of the system [4].

Conclusion

To check if a polynomial $p \geq 0$ could be NP-hard, whereas checking if $p \in \sum[x]$ can be done in polynomial time, since $p \in \sum[x] \Leftrightarrow p = z^T Q z$. Moreover each polynomial $p \in \sum[x]$ is a psd polynomial and therefore we can check if $p \geq 0$ by checking if $p \in \sum[x]$.

This idea even enables us in many cases to find the global minimum of p or even at least a useful one and hence we can check if p is psd by finding a SOS decomposition or a global minimum. This approach made the problem if p_{SOS}, $p_{negative}$ is psd feasible, since p_{SOS} is a sum of squares and $p_{negative}$ has the global minimum -1.03162845.

Furthermore the problem could be even more simplified by the Newton polytop, i.e. we can prune the vector z in the decomposition $p = z^T Q z$. The Newton polytope requires the construction of a convex hull and this could be time consuming. The zero diagonal algorithm forgoes on this construction and is based on a simple property of positive semidefinite matrices - if $Q_{i,i} = 0$ then is the i^{th} row and column equal to zero of Q.

The algorithm is fast since it only requires searching a set of linear equality constraints for those having certain properties and the set of monomials returned by the algorithm is a subset of the set returned by the Newton polytope method.

Furthermore the zero diagonal algorithm was extended to a more general reduction method for sum of squares programs and we have shown how to formulate problems in optimization as sum of squares programs.

References

[1] P. Seiler, Q. Zheng and G. J. Balas. Simplification Methods for Sum-of-Squares Programs

[2] T. Netzer. reelle Algebra, p. 36 - 45

[3] G. Blekherman. nonnegative polynomials and sums of squares

[4] A. Papachristodoulou. SOSTOOLS: control applications and new developments

[5] O. Davidek. example of applications of sum of squares problem in geometry

[6] O. Benoist. Writing Positive Polynomials as Sums of (Few) Squares

[7] J.B. Lasserre, T. Netzer. sos approximations of nonnegative polynomials via simple high degree perturbations

[8] P.A. Parrilo, B. Sturmfels. Minimizing Polynomial Functions

[9] B. Reznick. Uniform denominators in Hilbert's 17th problem, Math. Z. 220 (1995), no. 1, 75–97.

[10] P.A. Parrilo. Semidefinite programming relaxations for semialgebraic problems

[11] A.A. Ahmadi, P.A. Parrilo. Sum of Squares and Polynomial Convexity

[12] P. Seiler. SOSOPT: A Toolbox for Polynomial Optimization

[13] A. Papachristodoulou, J. Anderson, G. Valmorbida, S. Prajna, P. Seiler, P.A. Parrilo. Sum of Squares Optimization Toolbox for MATLAB

[14] W. Rudin. Sums of squares of polynomials. Amer. Math. Monthly, 107(9):813–821, 2000.

[15] R. G. Swan. Hilbert's theorem on positive ternary quartics. In Quadratic forms and their applications (Dublin, 1999), Volume 272 of Contemp. Math., pages 287–292. Amer. Math. Soc., Providence, RI, 2000.

[16] V. Powers Positive Polynomials and Sums of Squares: Theory and Practice