Bsc thesis applied mathematics

Using relaxations of sum of squares formulations for the kissing number problem

Jugo Rang

Delft University of Technology

Supervisor
Dr. F.M. de Oliveira Filho

Other committee members
Dr. K.P. Hart

June, 2020 Delft

Abstract

In recent years the importance of sum of squares and semidefinte programming has been seen in the field of combinatorial optimisation. All linear programs can be rewritten into a semidefinte one and by using hierarchies of semidefinite programs these can be solved for polynomial optimisation problems. Recently, in 2019, A.A. Ahmadi and A. Majumdar released a paper called "DSOS and SDSOS Optimization: More Tractable Alternatives to Sum of Squares and Semidefinite Optimization"[1] where they introduced the concept of sum of square polynomials obtained from diagonally dominant matrices. Since this concept is relatively new I am going to look at the viability of these sum of square polynomials in older known optimisation problems such as the kissing number problem. I will do this by writing a semidefinite program in which the sum of square polynomials are created by diagonally dominant matrices. I will then compare the newly found upper bounds with the upper bounds found by sampling and the volume bound.

Contents

1 Introduction

The kissing number problem asks for the maximum number of non-overlapping unit balls in $\mathbb{R}^n$ that can simultaneously touch a central unit ball. This maximum is denoted by τ_n. The problem owns its name to a billiard term: two balls "kiss" when they touch. Over the last few centuries the kissing number problem has become a well-known problem in the field of combinatorial geometry as it is one of the best-known packing problems. In dimension 1 the kissing number is 2. As can be seen in Figure 1, in this dimension the unit circles are actually intervals on the line segment. The only manner to put an interval next to a central interval one while being on the line is by placing one interval on the left and one the right of the central interval.

Figure 1: The kissing number problem in $\mathbb{R}^1$.

In dimension 2 the kissing number problem is 6, as illustrated in figure 2. A short proof is given.

Let C be the center of an unit circle which is touched by unit circles centred at $C_1, C_2, ..., C_n$. Now for the circle centered at C_i consider the line segments connecting C and C_i. Each of these line segments starts in C so the sum of angles between adjacent line segments is 2π.

Now assume by contradiction there are more than six circles touching the central circle C. This means at least two adjacent line segments are separated by an angle of less than $\frac{\pi}{3}$. Lets say these are the line segments CC_1 and CC_2. All segments CC_i have the same length of 2, which is twice the radius, for all i. Now the triangle CC_1C_2 is isosceles and the side C_1C_2 has a length of less than 2. This means that the circles constructed form centers C_1 and C_2 intersect, a contradiction.

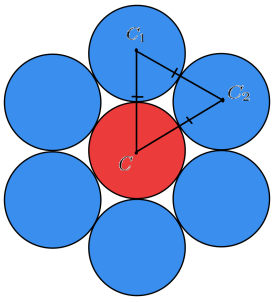

Figure 2: The kissing number problem in $\mathbb{R}^2$.

The first real disagreement about the kissing number problem arose when the mathematicians Isaac Newton (4 January 1643 – 31 March 1727) and David Gregory (3 June 1659 – 10 October 1708) tried to solve the kissing number problem for dimension 3 in 1694. Newton said the maximum to be 12 but Gregory argued it was 13 [3]. The discussion on this subject was raised because of the configuration of 12 spheres touching the middle sphere. To get this configuration we start with the case in dimension 2, One central unit sphere surrounded by six unit spheres. We can then put three unit spheres on top of the central unit sphere and three unit spheres on the bottom. As can be seen in Figure 3 when the 12 outer spheres touch the middle sphere they do not touch each other. This led to believe that by moving the outer spheres around another sphere could be placed touching the central sphere.

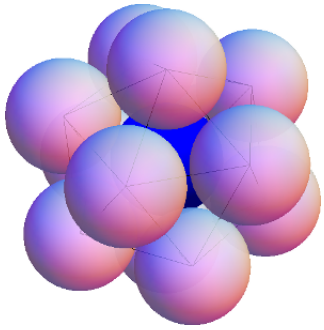

Figure 3: The kissing number problem in $\mathbb{R}^3$.

The answer to who was correct was only found in 1953 by Schütte and van der Waerden [11]. As it turns out the kissing number in dimension 3 is 12, so Newton was correct. However Gregory's assumption was not that far fetched. The lower bound of the kissing number problem in $\mathbb{R}^3$ should be 12 as can be seen from the configuration in Figure 3. As for an upper bound, the volume bound, see Section 2, is often seen as the simplest bound for the kissing number problem which givens an upper bound of 14. Later, Odlyzko and Sloane [8] determined the values for the $n = 8$ and $n = 24$, namely $\tau_8 = 240$ and $\tau_{24} = 196560$. After that Musin [7] determined that $\tau_4 = 24$. To this day these 6 dimensions, 1-4, 8 and 24, are the only dimensions for which the kissing number is known.

2 The volume bound

A sphere is the set of all points equidistant from a single point in space. Therefore a unit sphere is defined as $S^{n-1} = \{x \in \mathbb{R}^n : \|x\| = 1\}$. A *spherical cap* is the region of a sphere which lies above a given plane, see Figure 4. The *spherical distance* between two points P and Q on a sphere is the distance of the shortest path along the surface of the sphere. The *minimum angular distance* is the angle these two points make with the center of the sphere. A *spherical code* is

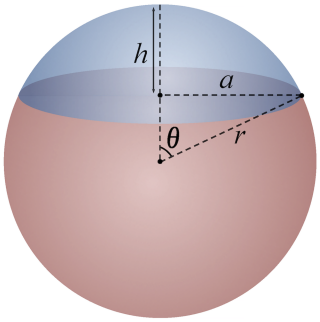

Figure 4: A spherical cap.

a finite set of points on the surface of a sphere in n dimensional space. The volume bound is based on a spherical code which covers the entire surface of a sphere. Now we can define a spherical cap with a centre in point x and angle θ as $S(x,\theta) = \{y \in R^n : x \cdot y \geq \cos\theta\}$, which is the set containing all points at spherical distance at most θ from x. The parameter $A(n,\theta)$ denotes the maximum size of a spherical code in S^{n-1} with minimum angular distance θ. If two unit spheres touch a central unit sphere, in the same manner as in the case of $\mathbb{R}^2$, the angle between two contact points is at least $\frac{\pi}{3}$. Therefore arranging the unit spheres such that they do not overlap and all touch a central unit sphere is the same as finding a spherical code in S^{n-1} with minimum angular distance $\frac{\pi}{3}$. This means that finding the kissing number in $\mathbb{R}^n$ is equal to finding $A(n, \frac{\pi}{3})$.

A spherical code in S^{n-1} with minimum angle distance θ can form a set of interior-disjoint spherical caps of radius $\frac{\theta}{2}$ by placing such cap on every point of the spherical code. Let ω be surface measure of S^{n-1} so that

$$\omega(S^{n-1}) = \frac{(2\pi)^{\frac{n}{2}}}{\Gamma(\frac{n}{2})} \tag{1}$$

where $\Gamma(\frac{n}{2})$ is the Euler gamma function. The area of spherical cap $S(n,\alpha)$ can be computed as following:

$$\omega(S^{n-2}) \int_{\cos\alpha}^{1} (1-u^2)^{\frac{n-3}{2}}\, du \tag{2}$$

So because a spherical code corresponds to a packing of spherical caps on the sphere we have the upper bound.

$$A(n,\theta) \leq \left\lfloor \frac{\omega(S^{n-1})}{\omega(S(n,\frac{\theta}{2}))} \right\rfloor \tag{3}$$

This is an upper bound because the surface of the sphere is covered by spherical caps which is greater than the maximum size of a spherical code with minimum angular distance θ. By filling in this upper bound for the kissing number problem for $\mathbb{R}^2$, $\theta = \frac{\pi}{3}$, the upper bound is 6, which as seen before is exactly the

kissing number for $\mathbb{R}^2$. However filling this in for $\mathbb{R}^3$ gives a right hand side of approximately 14.9282 which means that $\tau_3 \leq 14$.

3 Semidefinite programming

Important for semidefinite programming is the *trace inner product* of two matrices $A, B \in \mathbb{R}^{n\times n}$ which is

$$\langle A, B\rangle = Tr(A^T B) = \sum_{i,j=1}^{n} A_{i,j}B_{i,j}$$

We will denote by S^n the set of symmetric $n \times n$ matrices and write $A \succeq 0$ when A is positive semidefinite. Now we let $C \in \mathbb{R}^{n\times n}$ and $A_i \in \mathbb{R}^{n\times n}$ for $i = 1, ..., n$ be symmetric matrices. A semidefinite program is a special case of a linear program in which these C and $A_1, ..., A_n$ are diagonal. Furthermore let $b \in \mathbb{R}^m$ be a given vector with real numbers for b_i with $i = 1, ..., m$. A semidefinite program is thus a maximization problem in the form of

$$p^* = sup_X\{\langle C, X\rangle : \langle A_i, X\rangle = b_i \text{ for } i \in [m], X \succeq 0\} \tag{4}$$

where the symmetric $n \times n$ matrices $A_1, ..., A_m \in S^n$ and the vector $b \in \mathbb{R}^m$ are the date of the semidefinite program and the matrix X is the variable, constraind to lie in a subspace and to be positive semidefinite. Writing (4) down as an optimisation problem it looks like this:

$$\begin{aligned} &\textbf{max} && \langle C, X\rangle \\ &\textbf{subject to} && \langle A_i, X\rangle = b_i \quad \forall i = 1, ..., n \\ &&& X \succeq 0 \end{aligned} \tag{5}$$

So this function maximises the linear function $X \to \langle C, X\rangle$ where $X \in \mathbb{R}^{n\times n}$ ranges over all positive semidefinite matrices while satisfying the given linear constraints.

The efficient algorithms for semidefinite programming makes it favourable as a generalization of linear programming. This is because semidefinite programs can solve moderately-sized problems more efficiently.

4 The linear programming bound of Delsarte, Goethals, and Seidel

As seen in Section 2 it is important to find an accurate upper bound for $A(n, \theta)$ in order to solve the kissing number problem. One such upper bound is the linear programming bound found by Delsarte, Goethals, and Seidel [4]. In order to get to this upper bound we model spherical codes as independent sets in a certain graph.

Let $G = (V, E)$ be a graph without loops or parallel edges. A set of vertices $V' \subseteq V$ is said to be *independent* if x, y are nonadjacent for all x, $y \in V'$. The independence number of G, which is denoted by $\alpha(G)$, is the maximum cardiality of any independent set of G. Given $n \geq 1$,$\theta \in (0, \pi]$, consider the graph $G(n, \theta)$ where we use S^{n-1} as the set of vertices. In $G(n, \theta)$ distinct vertices x, $y \in S^{n-1}$ are adjacent if $\cos\theta < x \cdot y < 1$. Now the subset of vertices $V' \subseteq S^{n-1}$ is independent in $G(n, \theta)$ if and only if V' is a spherical code with minimum angular distance θ. Therefore it follows that $A(n, \theta) = \alpha(G(n, \theta))$. An important thing to note however is that the graph $G(n, \theta)$ is infinite, though the independence number is finite.

Since we know $A(n, \theta) = \alpha(G(n, \theta))$ we need to find an upper bound for the Independence number $\alpha(G(n, \theta))$. One such upper bound is the theta prime number, $\vartheta'(G)$, which was introduced by McElliece, Rodemich, and Rumsey [6] and Schrijver [10]. This number was based on the Lovász theta number, $\vartheta(G)$, introduced by László Lovász [5] in 1979. One way to define $\vartheta'(G)$ for a finite graph G is as the optimal value of the following semidefinite programming problem:

$$\begin{array}{lll} \textbf{inf} & \lambda & \\ \textbf{subject to} & Z(x,x) = \lambda & \text{for all } x \in V, \\ & Z(x,y) \leq 0 & \text{if } x \neq y \text{ and } xy \notin E, \\ & Z : V \times V \to \mathbb{R} & \text{is symmetric and } Z - J \text{ is positive semidefinite.} \end{array} \tag{6}$$

Here J is the all-ones matrix.

The graph used to define ϑ' was a finite graph while the graph $G(n, \theta)$ is infinite as stated before. So in order to get an upper bound for $A(n, \theta)$ using ϑ' the definition should be extended so it holds for infinite graphs. Let V be a compact space, A *kernel* is a continuous function $K : V \times V \to \mathbb{R}$. A kernel K is *positive* if $(K(x, y)_{x,y \in U}$ is positive semidefinite for all finite sets $U \subseteq V$. For $G = (V, E)$ where V is compact, ϑ' is the optimal value of:

$$\begin{array}{lll} \textbf{inf} & \lambda & \\ \textbf{subject to} & Z(x,x) = \lambda & \text{for all } x \in V, \\ & Z(x,y) \leq 0 & \text{if } x \neq y \text{ and } xy \notin E, \\ & Z : V \times V \to \mathbb{R} & \text{is a symmetric kernel} \\ & & \text{and } Z - J \text{ is positive.} \end{array} \tag{7}$$

This time J is the constant 1 kernel. The following theorem now holds.

Theorem 1. *Let $G(V, E)$ be a graph where V is a compact topological space. If (Z, λ) is feasible for (7) then $\alpha(G) \leq \lambda$. In particular $\alpha(G) \leq \vartheta'(G)$*

Proof. Because of the assumptions we made on V we can use the following observation of Bochner [2]: a continuous and symmetric kernel $K : V \times V \to \mathbb{R}$ is positive if and only if $((K(x_i, x_j))_{i,j=1}^N$ is positive semidefinite for every choice

$x_1, ..., x_N$ of infinitely many points in V. So if (Z, λ) is feasible for (7) and $C \subseteq V$ is a nonempty independent set then,

$$0 \leq \sum_{x,y \in C} (Z - J)(x, y) = \sum_{x,y \in C} Z(x, y) - |C|^2 \leq \lambda |C| - |C|^2 \tag{8}$$

and thus $|C| \leq \lambda$. Therefore $\alpha(G) \leq \lambda$ and in particular $\alpha(G) \leq \vartheta'(G)$ □

Since S^{n-1} is compact it holds that $\alpha(G(n, \theta)) \leq \vartheta'(G(n, \theta))$. This leaves the final problem, computing $\vartheta'(G(n, \theta))$. As seen from the definition this means finding the optimal solution of (7). In order to do this we will use the fact that $G(n, \theta)$ is symmetric. We start by introducing the orthogonal group $O(n) = \{A \in \mathbb{R}^{n \times n} : A^T A = I\}$ which acts on S^{n-1} by sending x to Ax which preserves the inner product. This means that every $A \in O(n)$ gives an automorphism of $G(n, \theta)$. For $A \in O(n)$ the orthogonal group $O(n)$ also acts on the kernels $K : S^{n-1} \times S^{n-1} \to \mathbb{R}$ as follows.

$$(A \cdot K)(x, y) = K(A^{-1}x, A^{-1}y) \quad \forall x, y \in S^{n-1} \tag{9}$$

Further more we say that the kernel K is *invariant* if for all $A \in O(n)$ we have $A \cdot K = K$. The reason we want to restrict ourselves to invariant kernels in (7) is because we can then use the following theorem of Schoenberg [9] to rewrite (7).

Theorem 2. *A kernel $K : S^{n-1} \times S^{n-1} \to \mathbb{R}$ is positive and invariant if and only if*

$$K(x, y) = \sum_{k=0}^{\infty} f_k P_k^n(x \cdot y) \tag{10}$$

for some non-negative numbers $f_0, f_1 ...$ such that $\sum_{k=0}^{\infty} f_k$ converges, in which case the series in (10) converges absolutely and uniformly over $S^{n-1} \times S^{n-1}$. The numbers f_k are uniquely determined by and uniquely determine K.

Here P_k^n is the Jacobi polynomial of degree k and parameters $\alpha = \beta = \frac{n-3}{2}$. In order to compute these polynomials the Gram-Schmidt orthogonalisation can be applied to the inner product of the sequence of polynomials $1, t, t^2$. With inner product:

$$(\varphi, \psi) = \int_{-1}^{1} \varphi(t)\psi(t)(1 - t^2)^{\frac{n-3}{2}} dt \tag{11}$$

for $\varphi, \psi : [-1, 1] \to \mathbb{R}$. The Jacobi polynomials can be computed with recursion [12] as following:

$$\begin{aligned}
&2n(n + \alpha + \beta)(2n + \alpha + \beta - 2)P_n^{(\alpha,\beta)}(x) \\
&= (2n + \alpha + \beta - 1)((2n + \alpha + \beta)(2n + \alpha + \beta - 2)x + \alpha^2 - \beta^2)P_{n-1}^{(\alpha,\beta)}(x) \\
&- 2(n + \alpha - 1)(n + \beta - 1)(2n + \alpha + \beta)P_{n-2}^{(\alpha,\beta)}(x), \ n = 2, 3, 4.... \\
&P_0^{(\alpha,\beta)}(x) = 1 \quad P_1^{(\alpha,\beta)}(x) = \frac{1}{2}(\alpha + \beta + 2)x + \frac{1}{2}(\alpha - \beta)
\end{aligned} \tag{12}$$

By filling in $\alpha = \beta = \frac{n-3}{2}$ this will give:

$$P_0^n(x) = 1 \quad P_1^n(x) = \frac{1}{2}(n-1) \tag{13}$$

It is also important to normalise the Jacobi polynomials such that $P_k^n(1) = 1$. Now since the Jacobi polynomial P_0^n is the polynomial 1, using Theorem 2, it holds that a symmetric kernel $Z : S^{n-1} \times S^{n-1} \to \mathbb{R}$ is such that $Z - J$ is positive if and only if

$$Z(x,y) = \sum_{k=0}^{\infty} f_k P_k^n(x \cdot y), \quad f_0 \geq 1, \quad f_1, f_2 ... \text{ are non-negative with} \sum_{k=0}^{\infty} f_k < \infty \tag{14}$$

If we use this fact combined with the fact that the vertices $x, y \in S^{n-1}$ are adjacent in $G(n,\theta)$ if $\cos|\theta| \leq x \cdot y \leq 1$ then (7) can be rewritten in a more tractable form:

$$\begin{aligned} \textbf{min} \quad & \sum_{k=0}^{\infty} f_k \\ \textbf{subject to} \quad & \sum_{k=0}^{\infty} f_k P_k^n(t) \leq 0 \quad \text{if} -1 \leq t \leq \cos|\theta| \\ & f_0 \geq 1 \text{ and } f_1, f_2 ... \text{are non-negative with} \sum_{k=0}^{\infty} f_k < \infty \end{aligned} \tag{15}$$

This minimisation problem is known as the linear programming bound of Delsarte, Goethals, and Seidel [4]. This linear program however is one with still infinitely many variables and infinitely many constraints since every k introduces a new variable and each $t \in [-1, \cos\theta]$ introduces a new constraint. Luckily this problem is easily solved by either sampling or using sums of squares. In the next Section I will elaborate more on these two methods.

5 Modeling the linear programming bound as semidefinite program.

As seen in section 4 the linear programming bound (15) of Delsarte, Goethals, and Seidel [4] still has infinitely many variables and constraints. There are two ways to deal with this. One method is by sampling. The other method is to adjusting the the linear programming problem, such that (15) can be modeled as a semidefinite program. In order to do so the sum of squares restriction is used.

5.1 Sampling the linear programming bound.

In order to turn (15) into a tractable optimisation problem, there are two problems to tackle. The first is that of the infinitely many variables generated by

each new k. In order to solve this, set an integer $d > 0$ and set $f_k = 0$ if $d > k$. This way the series $\sum_{k=0}^{\infty} f_k P_k^n$ is shortened such that there is a finite amount of variables added. Setting $k = 2d$ results in the following optimisation problem:

$$\begin{aligned} \textbf{min} \quad & \sum_{k=0}^{2d} f_k \\ \textbf{subject to} \quad & \sum_{k=0}^{2d} f_k P_k^n(t) \leq 0 \quad \text{if} -1 \leq t \leq \cos|\theta| \\ & f_0 \geq 1 \text{ and } f_1, f_2 ... \text{are non-negative} \end{aligned} \tag{16}$$

The second problem is that of the infinitely many constraints generated by $t \in [-1, \cos\theta]$. In order to solve this problem sampling is necessary. For this choose a finite set $S \subseteq [-1, \cos\theta]$ and only consider the constraint for $t \in S$ in (15). If the new linear program does not have a feasible solution a larger value for d should be chosen. It then has to be checked whether or not the new linear program has an optimal solution that is feasible for the original program. After restricting to a sample, the optimal solution may not be feasible for the original program. Then it is possible to fix this by rescaling the solution and changing the variable f_0. This however may result in a bound that is much worse if the sample is not fine enough.

5.2 Modeling the linear programming bound as a semidefinite program using sums of squares

One of the important constraints in the linear programming bound (16) is that the polynomials $\sum_{k=0}^{2d} f_k P_k^n(t)$ with $d < \infty$ are non-negative. This is assured because $f_0, f_1, ...$ are all non-negative. A polynomial $p \in \mathbb{R}[x_1, ...x_n]$ is a *sum of squares* if there are polynomials $q_1, ..., q_m$ such that $p = q_1^2 + ... + q_m^2$. It is thus easily seen that if p can be written as a sum of squares that p is non-negative everywhere. What this sum of squares has to do with semidefinite programming is made clear by the following Theorem. In this theorem $\mathbb{R}[x_1, ...x_n]_{\leq d}$ is the set of polynomials with degree at most d for $d \geq 0$. Furthermore, for any finite set of polynomials B it holds that $v_B : B \to B$ such that $v_B(p) = p$ for all $p \in B$

Theorem 3. *Let $p \in \mathbb{R}[x_1, ...x_n]$ be a polynomial of degree $2d$ and let B be a basis of $\mathbb{R}[x_1, ...x_n]_{\leq d}$. Then p is a sum of squares if and only if there is a positive semidefinite matrix $Q : B \times B \to R$ such that $p = v_B^T Q v_B$*

Proof. Assume Q is a positive semidefinite matrix such that $p = v_B^T Q V_B$ Then for some vectors $u_i : B \to \mathbb{R}$ we have $Q = u_1 u_1^T + ... + u_m u_m^T$. Writing $q_i = u_i^T v_B$ we have that each u_i is a polynomial and

$$p = v_B^T Q v_B = v_B^T u_1 u_1^T + ... + v_B^T u_m u_m^T v_B = q_1^2 + ... + q_m^2$$

so p is a sum of squares.

On the contrary assume $p = q_1^2 + ... + q_m^2$ for some $q_1, ..., q_m$. The degree of each q_i is at most d and thus can be expressed as linear combination of the polynomials in B. So for $i = 1, ..., m$ let $u_i : B \to \mathbb{R}$ be such that $q_i = u_i^T v_B$. Then $p = v_B^T Q V_B$ and $Q = u_1 u_1^T + ... + u_m u_m^T$ positive semidefinite. □

So in order to set up a tractable semidefinite programming problem the given polynomial $p \in \mathbb{R}[x_1, ...x_n]$ should be of degree $2d$ and a sum of squares. For it to be a sum of squares a basis B of $\mathbb{R}[x_1, ...x_n]_{\leq d}$ is chosen. This is because Theorem 3 now tells us that p is a sum of squares if and only if here is a positive semidefinite matrix $Q : B \times B \to R$ such that $p = v_B^T Q v_B = \langle v_B v_B^T, Q \rangle$. Since this is an identity between polynomials where both sides are of degree $2d$ the identity should be checked by expanding on a basis $B_=$ of $\mathbb{R}[x_1, ...x_n]_{\leq 2d}$ and comparing the coefficients. This can be done by letting $\mathbf{coef}(r, q)$ be the coefficient of $r \in B_=$ in the expansion of $q \in \mathbb{R}[x_1, ...x_n]_{\leq 2d}$ on the basis $B_=$. By applying $\mathbf{coef}(r, \cdot)$ to $v_B v_B^T$ entrywise we obtain a real matrix. We will denote this matrix as $\mathbf{coef}(r, v_B v_B^T)$. It follows that $p = \langle v_B v_B^T, Q \rangle$ if and only if

$$\langle \mathbf{coef}(r, v_B v_B^T), Q \rangle = \mathbf{coef}(r, p) \text{ for each } r \in B_= \tag{17}$$

We now reduced the requirement of p beeing a sum of squares to the existance of a positive semidefinite matrix $Q : B \times B \to R$ satisfying the linear constraints from (17). Since we want the polynomials $\sum_{k=0}^{2d} f_k P_k^n(t)$ to be non-negative inside a certain domain $t \in [-1, \cos\theta]$ and because the polynomials are univariate a theorem of Lukács [12] can be used. A polynomial $p \in \mathbb{R}[x]$ is non-negative on the interval $[a, b]$ if and only if there are polynomials q, q' such that:

$$\begin{aligned} p(x) &= q(x)^2 + (x-a)(b-x)q'(x)^2 \qquad \text{if } p \text{ has an even degree} \\ p(x) &= (x-a)q(x)^2 - (b-x)q'(x)^2 \quad \text{if } p \text{ has an uneven degree} \end{aligned} \tag{18}$$

Because we are dealing with sums of squares p is always of even degree and the interval is $[-1, \cos\theta]$. Therefore p can be written as:

$$p(x) = q(x)^2 - (x+1)(\cos\theta - x)q'(x)^2$$

The next step is to use this in order to write the linear program bound (15) as a semidefinite program in order to find an upper bound for $A(n, \theta)$. The first step we take is the same as it was with the sampling method in section 5.1, limiting the amount of variables by setting $k = 2d$. Since we want $\sum_{k=0}^{2d} f_k P_k^n$ to be non-positive on $t \in [-1, \cos\theta]$ we can require the sum of squares polynomials q, with degree up to $2d$, and q',with degree up to $2(d-1)$, to be such that

$$\sum_{k=0}^{2d} f_k P_k^n(t) = -q(t) - (t+1)(\cos\theta - t)q'(t) \tag{19}$$

Now q is a sum of squares if and only if there is a positive semidefinite matrix $Q : B \times B \to \mathbb{R}$ such that $q = \langle v_B v_B^T, Q \rangle$ with B the basis of $\mathbb{R}[t]_{\leq d}$. This same statement also holds for q' and B' the basis of $\mathbb{R}[t]_{\leq d-1}$. So q' is a sum of squares

if and only if there is a positive semidefinite matrix $Q' : B' \times B' \to \mathbb{R}$ such that $q' = \langle v_{B'} v_{B'}^T, Q' \rangle$. Therefore an upper bound for $A(n, \theta)$ is any feasible solution of the semidefinite optimisation problem.

$$\begin{aligned} \textbf{min} \quad & \sum_{k=0}^{\infty} f_k \\ \textbf{subject to} \quad & \sum_{k=0}^{\infty} f_k P_k^n(t) + \langle v_B v_B^T, Q \rangle + \langle (t+1)(\cos\theta - t) v_{B'} v_{B'}^T, Q' \rangle = 0 \\ & f_0 \geq 1 \text{ and } f_1, ..., f_{2d} \geq 0 \\ & Q : B \times B \to \mathbb{R} \succeq 0 \text{ and } Q' : B' \times B' \to \mathbb{R} \succeq 0 \end{aligned} \tag{20}$$

6 Sums of squares from diagonally dominant matrices

As we have seen using sum of squares gives us a semidefinite optimisation problem which establishes an upper bound for $A(n, \cos\theta)$. We wanted p to be a sum of squares so we focused on the matrix Q to be positive semidefinite such that $p = v_B^T Q v_B$. So p is a sum of squares constructed from a positive semidefinite matrix Q. However in the hope more efficient inner approximations are obtained the condition of Q being a positive semidefinite matrix can be changed to Q being a diagonally dominant matrix. This is a stronger condition but might effect the accuracy. This idea was recently introduced by A.A. Ahmadi and A. Majumdar [1]. A symmetric matrix A is *diagonally dominant* (DD) if $a_{ii} \geq \sum_{j \neq i} |a_{ij}|$ for all i. Therefor it follows that p is a *diagonally-dominant-sum-of-squares* (DSOS) if it admits a representation as $p(x) = v_B^T Q v_B = \langle v_B v_B^T, Q \rangle$, where Q is a diagonal dominant matrix and B the basis of $\mathbb{R}[t]_{\leq d}$. It is also important to note that a diagonal dominant matrix is positive semidefinite. If A is a diagonal dominant matrix then:

$$\begin{aligned} x^T A x = \sum_{i=1}^{n} a_{i,i} x_i^2 + \sum_{i \neq j} a_{i,j} x_i x_j &\geq \sum_{i=1}^{n} (\sum_{i \neq j} |a_{i,j}|) x_i^2 - \sum_{i \neq j} |a_{i,j}||x_i||x_j| \\ &= \sum_{j>i} (|a_{i,j}|(x_i^2 + x_j^2 - 2|x_i||x_j|)) \geq 0 \end{aligned} \tag{21}$$

So A is positive semidefinite. Finding an upper bound for $A(n, \theta)$ using dsos was the main focus of my research. The optimisation problem I used in order

to do so resembled (20) closely but was in essence very different.

$$\begin{aligned} \textbf{min} \quad & \sum_{k=0}^{\infty} f_k \\ \textbf{subject to} \quad & \sum_{k=0}^{\infty} f_k P_k^n + \langle v_B v_B^T, Q \rangle + \langle (t+1)(\cos\theta - t) v_{B'} v_{B'}^T, Q' \rangle = 0 \\ & f_0 \geq 1 \text{ and } f_1, ..., f_{2d} \geq 0 \\ & Q : B \times B \to \mathbb{R} \text{ and } Q' : B' \times B' \to \mathbb{R} \text{ are diagonally dominant} \end{aligned} \tag{22}$$

The two optimisation problems are different because we use DSOS for which linear programming is used instead of semidefinite programming which was used for the positive semidefinite matrices in (20). A diagonal dominant matrix A can be modeled in a linear program by introducing a second matrix B. First the diagonals of A are required to be non-negative. Because the absolute value of every other entry in A is required, B is modeled such that every $b_{i,j} \in B$ is the absolute value of $a_{i,j} \in A$ for $j \geq i$ since A is symmetric. This is done by adding the constraints $B[i,j] >= A[i,j]$ and $B[i,j] >= -A[i,j]$).

7 Results

In order to see whether or not using DSOS for the kissing number problem provides a good upper bound for $A(n, \theta)$ I compared the upper bounds I found with the ones found when using the volume bound and the sampling method on the linear programming bound. I compared these different bounds for up to 24 dimensions and 5 different angles. As seen in chapter 2.1, the angle in order to solve kissing number problem is $\frac{\pi}{3}$. An important thing to note however is that my sampling bound is slightly off due to a not fine enough sample. This is clearly seen in the dimensions $1 - 4, 8$ and 24 for which the kissing number is known. The results I found are shown in the tables below.

As stated before the angle θ determines the size of the spheres touching. Therefor $\theta = \frac{\pi}{3}$ results in spheres as large as the center one. As can be seen in Table 2 the DSOS method gives for $n = 2$ an upper bound which is a little higher than the sample and volume bound method give, however rounded it is still a correct upper bound. For $n = 3$ the DSOS method gives out an upper bound which is better than that of the volume bound however worse than the sample method but it is still close. For $n = 4$ to $n = 24$ the DSOS method starts getting higher and higher than the actual upper bound is. It performs worse than the sample bound the higher the dimension, n, gets. However the DSOS method is still closer to the actual upper bound than the volume method is.

n	DSOS	sample	volume
2	4	4	4
3	6	6	6.828427
4	8	8	11.00775
5	10	10	17.22408
6	12	12	26.45964
7	14	14	40.14028
8	16	16	60.34002
9	18	18	90.07425
10	20	20	133.7223
11	22	22	197.6371
12	24	24	291.0232
13	26	26	427.2039
14	28	28	625.4436
15	30	30	913.5721
16	32	32	1331.759
17	34	34	1937.938
18	36	36	2815.604
19	38	38	4085.01
20	40	40	5919.233
21	42	42	8567.239
22	44	44	12386.96
23	46	46	17892.71
24	48	48	25823.15

Table 1: Different bounds for $\theta = \frac{\pi}{2}$

n	DSOS	sample	volume
2	6.207238	5.998485	6
3	14.24833	13.15824	14.9282
4	28.42756	25.55626	34.68075
5	52.5	46.33255	77.7562
6	96	82.61664	170.5784
7	184.8	140.154	368.7349
8	312.0515	239.8798	788.6448
9	514.8	380.0353	1673.205
10	833.4884	595.7408	3527.559
11	1389.143	915.0566	7399.303
12	2481.231	1415.472	15455.75
13	4081.586	2231.624	32171.21
14	6373.574	3489.68	66765.38
15	9905.333	5424.385	138204.5
16	15902.12	8309.671	285446.9
17	27370	12210.26	588408.1
18	44988.68	17871.02	1210822
19	68051.03	25883.95	2487778
20	104594.3	37897.33	5104370
21	164199.7	56812.3	10459975
22	275636.7	84687.58	21410546
23	477161.5	127903.6	43780148
24	695205.9	195679.3	89437026

Table 2: Different bounds for $\theta = \frac{\pi}{3}$

As we increase the size of the angle to $\theta = \frac{\pi}{2}$ it means that the spheres touching the center ball are greater than the center ball itself. If we look at Table 1, we see that the DSOS method performs as good as the sample method and thus outperforms the volume bound for every dimension $n = 3, ..., 24$.

It is also possible for the spheres connected to the central ball to be smaller than the central one. This happens when we set the angle $\theta \geq \frac{\pi}{3}$.

n	DSOS	sample	volume
2	8.123189	7.997757	8
3	24.60889	23.43172	26.27414
4	67.05047	54.51365	80.25386
5	168.1612	149.9185	236.1009
6	415.5058	339.8278	678.8385
7	964.9874	753.6535	1921.832
8	2291.226	1674.368	5380.519
9	5085.93	3566.616	14937.55
10	11858.51	7340.546	41198.21
11	25488.9	15521.48	113027.1
12	58992.93	32473.28	308746.7
13	123570.2	64613.24	840324.7
14	281533.4	127805.7	2280109
15	585184.4	260947	6170436
16	1357337	531172.6	16660155
17	4342620	1047376	44891909
18		2097213	1.21E+08
19		4309659	3.24E+08
20		8828220	8.7E+08
21		17303388	2.33E+09
22		33667448	6.23E+09
23		64571768	1.67E+10
24		1.27E+08	4.45E+10

Table 3: Different bounds for $\theta = \frac{\pi}{4}$

n	DSOS	sample	volume
2	10.18972	9.928458	10
3	39.05725	36.61617	40.86345819
4	133.0509	104.5787	155.0130054
5	423.5922	355.7884	565.8387787
6	1294.949	1034.855	2017.540024
7	3849.977	2849.126	7080.847757
8	11216.41	7761.113	24570.25043
9	38406.72	20615.85	84529.9863
10	414779.3	53121.31	288871.5206
11		140436.6	981895.7451
12		348027.2	3322859.149
13		888744.7	11203658.15
14		2222068	37657586.87
15		5412463	126235506.8
16		13696133	422182913.1
17		32906121	1409077838
18		79540043	4694481448
19		1.96E+08	15615089723
20		4.65E+08	51865548610
21		1.13E+09	1.72049E+11
22		2.69E+09	5.70058E+11
23		6.28E+09	1.88678E+12
24		1.51E+10	6.23896E+12

Table 4: Different bounds for $\theta = \frac{\pi}{5}$

we fist have a look at what happens when we set the angle to be $\theta = \frac{\pi}{4}$. The fist thing to notice is that in Table 3 the DSOS method does not work for all the $n = 24$ dimensions it worked for before. This is because for higher dimensions computing this becomes to hard for the program. Another thing to notice is that it gives out higher upper bounds than the sampling method. However, except for $n = 2$, the DSOS method provides lower upper bounds than the volume bound does which is a good thing. For $\theta = \frac{\pi}{5}$ the DSOS method produces for even fewer dimensions results. As before however the DSOS method provides a lower upper bound than that of the volume bound, except for $n = 2$. The values of the sampling method are however even lower.

Lastly we have a look at Table 5 for which $\theta = \frac{\pi}{6}$ has been set. Again the DSOS method produces for even fewer dimension results. Again the sampling method produces the lowest upper bound, than the DSOS method and lastly the volume bound.

n	DSOS	sample	volume
2	12.22214	11.90256	12
3	56.40706	52.87139	58.69548054
4	230.9418	205.1827	266.250479
5	900.9846	742.3813	1161.579769
6	4362.751	2533.932	4948.651604
7		8502.9	20748.1474
8		27275.96	85996.34402
9		87164.64	353362.313
10		270971.3	1442201.389
11		834523.6	5854337.592
12		2544858	23659177.73
13		7600980	95259955.58
14		22785184	382345317.5
15		67622510	1530486091
16		2E+08	6112042203
17		5.75E+08	24358638798
18		1.68E+09	96901986251
19		4.85E+09	3.84869E+11
20		1.41E+10	1.52638E+12
21		4.08E+10	6.04572E+12
22		1.20E+11	2.39206E+13
23		3.40E+11	9.40569E+13
24		9.79E+11	3.73652E+14

Table 5: different bounds for $\theta = \frac{\pi}{6}$

So if we look at the results for all different dimensions and different angles for θ it is clear that the DSOS method does not provide a better upper bound for $A(n, \theta)$ than the sampling method does. However the DSOS method does provide us with a lower, and thus better, upper bound for $A(n, \theta)$ than the most arbitrary bound, the volume bound.

References

[1] Amir Ali Ahmadi and Anirudha Majumdar. Dsos and sdsos optimization: more tractable alternatives to sum of squares and semidefinite optimization. *SIAM Journal on Applied Algebra and Geometry*, 3(2):193–230, 2019.

[2] M Lashkarizadeh Bami. Bochner's theorem and the hausdorff moment theorem on foundation topological semigroups. *Canadian Journal of Mathematics*, 37(5):785–809, 1985.

[3] HSM Coxeter. An upper bound for the number of equal nonoverlapping spheres that can touch another of the same size. In *Convexity: Proceedings of the Seventh Symposium in Pure Mathematics of the American Mathematical Society*, volume 7, page 53. American Mathematical Soc., 1963.

[4] Philippe Delsarte, Jean-Marie Goethals, and Johan Jacob Seidel. Spherical codes and designs. In *Geometry and Combinatorics*, pages 68–93. Elsevier, 1991.

[5] László Lovász. On the shannon capacity of a graph. *IEEE Transactions on Information theory*, 25(1):1–7, 1979.

[6] Robert J McEliece, Eugene R Rodemich, and Howard C Rumsey Jr. The lovász bound and some generalizations. *J. Combin. Inform. System Sci*, 3(3):134–152, 1978.

[7] Oleg R Musin. The kissing number in four dimensions. *Annals of Mathematics*, pages 1–32, 2008.

[8] Andrew M Odlyzko and Neil JA Sloane. New bounds on the number of unit spheres that can touch a unit sphere in n dimensions. *Journal of Combinatorial Theory, Series A*, 26(2):210–214, 1979.

[9] Isaac J Schoenberg. Metric spaces and completely monotone functions. *Annals of Mathematics*, pages 811–841, 1938.

[10] Alexander Schrijver. A comparison of the delsarte and lovász bounds. *IEEE Transactions on Information Theory*, 25(4):425–429, 1979.

[11] Kurt Schütte and Bartel Leendert van der Waerden. Das problem der dreizehn kugeln. *Mathematische Annalen*, 125(1):325–334, 1952.

[12] Gabor Szeg. *Orthogonal polynomials*, volume 23. American Mathematical Soc., 1939.