Point Distribution On The Sphere

Final edition

Name: **Richard Lu**

Report version: **1.5**

June 18, 2022

Contents

Part I

voronöi cell

1 Introduction

This is a research report regarding to the study of 'point distribution on the sphere'. this subject has been greatly studied by many researchers so far. Many important and beautiful results were presented over the years. While in this report, we try to follow the path of Dr. Han in his paper (see Section 2 in [15]). The goal is to investigate the existence of point distribution that satisfied those 3 sub-conditions as he (Dr.Han) mentioned in his paper (see Condition 4 in Section 2).

Many comments will be included and will be removed as the study progresses. One comments example is:

this is an example of the comment

A discussion of the Tammes's problem and his dual problems will be included in the first part of this report, followed by a detailed exploration of the 3 sub-conditions mentioned above. The discussion of voronöi cell will be excluded in the first few version of this report.

Here are some special notations used in this reports:

$\mathbb{T}_n$	n-points distribution generated by solution of the n-points Tammes's problem
dist(A, B)	the geodesic distance from point A to point B
$\because$	suppose that
$\lightning$	contradiction
$\mathbb{D}_n$	n-points distribution generated by group-theoretic construction method in [14]

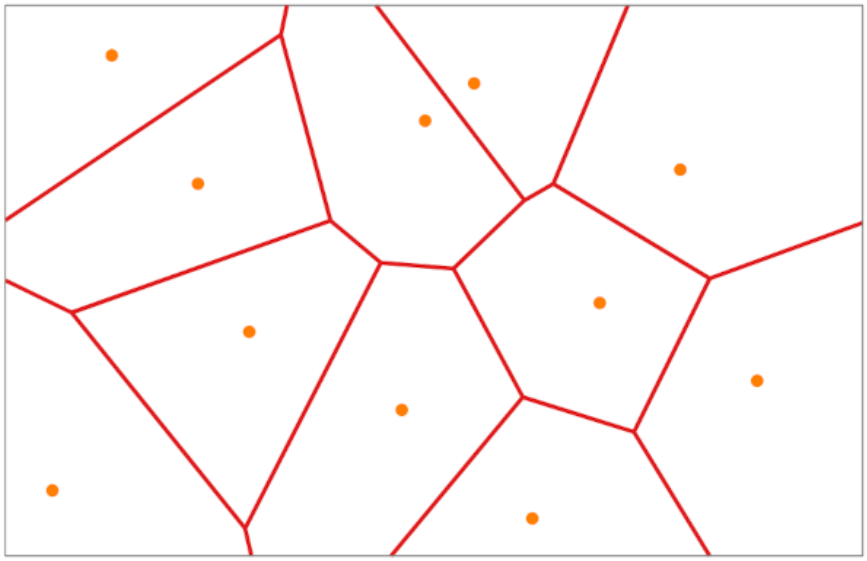

Figure 1: voronöi cell example

2 voronöi cell

In this section, a concise introduction of voronöi cell will be given. one of the origin of the voronöi cell is when one consider the post office problem :

Consider a town has multiple post offices, how can we assign each residence a post office that reduce the cost of delivering. That is, only assign the closest post office to one residence.

As shown in the figure 1, if we consider the each orange point as the post office. Then, the whole town could be partitioned into these small cells. Such that for any residence in the particular cell (i.e. any points in the cell), the closest post office to it is the cell site (the only post office in the cell). In this way, we preprocess the whole data points, and assign each data points to its closest cite. In the content of post office problem. We could pre-calculate the distance from every location on the map to the every post office and assign each residence with a particular location to its nearest post office.

One interesting application of the voronöi cell is that recent years, it has been used in the supervised classification algorithm in the area of machine learning. Many recommendation algorithms like nearest neighbour uses the voronöi network to make suggestions by finding the nearest (most similar) profile within a high dimensional space.

2.1 Definition

It is time to formally define the voronöi cell:

2.1: voronöi cell

For a given point set $\mathbb{P} = \{P_1, P_2, P_3\}$, The voronöi cell $\mathrm{V}(P_1) \coloneqq \{x \in \mathbb{R}^n : |xP_1| < |xQ| \quad \forall Q \in \mathbb{P} \backslash \{P_1\}\}$

On the other hand, if we define an auxiliary notion at first:

2.2: half plane

For 2 points P, Q in some domain $\mathbb{D}$, the half plane is define as:

$h(P, Q) \coloneqq \{x : |xP| < |xQ| \quad \forall x \in \mathbb{D}\}$

if we adopt the above definition, voronöi cell could also be defined as follows:

2.3: voronöi cell (alternative definition)

For a given point set $\mathbb{P} = \{P_1, P_2, P_3\}$, The voronöi cell $\mathrm{V}(P_1) \coloneqq \bigcap_{P_1 \neq Q, \forall Q \in \mathbb{P}} h(P_1, Q)$

Now, we have the voronöi cell, we could use it to finish another two related definition:

2.4: voronöi edge

For a given point set $\mathbb{P} = \{P_1, P_2, P_3\}$, The voronöi edge $\mathrm{V}(\{P_1, P_2\}) \coloneqq$ relative-interior $(\partial V(P_1) \bigcap \partial V(P_2)) = \{x : |xP_1| = |xP_2| \ \& \ |xP_1| < |xQ| \quad \forall Q \in \mathbb{P} \backslash \{P_1, P_2\}\}$

and finally, we have:

2.5: voronöi vertex

For a given point set $\mathbb{P} = \{P_1, P_2, P_3\}$, The voronöi vertex $\mathrm{V}(\{P_1, P_2, P_3\}) \coloneqq \partial V(P_1) \bigcap \partial V(P_2) \bigcap \partial V(P_3)$

With all the finition above together (exclude Def 2.3 and Def 2.2), it creates a voronöi diagram system, denote as $\mathrm{Vor}(\mathbb{P})$ with a certain point set $\mathbb{P}$. For future reference, the point in this points set $\mathbb{P}$ are also called sites.

2.2 Basic Theorems

Theorem 1

Given a set $\mathbb{P} \subset \mathbb{R}^2$ of n sites, Vor($\mathbb{P}$) consists of at most (2n-5) vertices and (3n-6) edges.

Proof

we start the proof by creating a dummy point A as shown in figure 2

Applying Euler characteristic equation we have:

$(|V|+1) - |E| + |F| = 2$

$\Rightarrow (|V|+1) - |E| + n = 2 \ (\alpha)$

since every vertex has a degree ≥ 3

$\Rightarrow 2|E| \geq 3(|V|+1)$

$\Rightarrow 2 \leq (|V|+1) + n - \frac{3}{2}(|V|+1)$

$\Rightarrow |V| + 1 \leq 2n - 4$

$|V| \leq 2n - 5$ [plug in (α)]

$\Rightarrow |E| \leq 3n - 6$

□

You could access the planar graph used in the proof at:

https://www.geogebra.org/calculator/rwdptfr8

Theorem 2

Let $\mathbb{P} \subset \mathbb{R}^2$ be a n-sites sets. If all sites are collinear, Vor($\mathbb{P}$) cosists only of (n-1) parallel lines. Otherwise, Vor($\mathbb{P}$) is connected and its edges are line segments or half-lines.

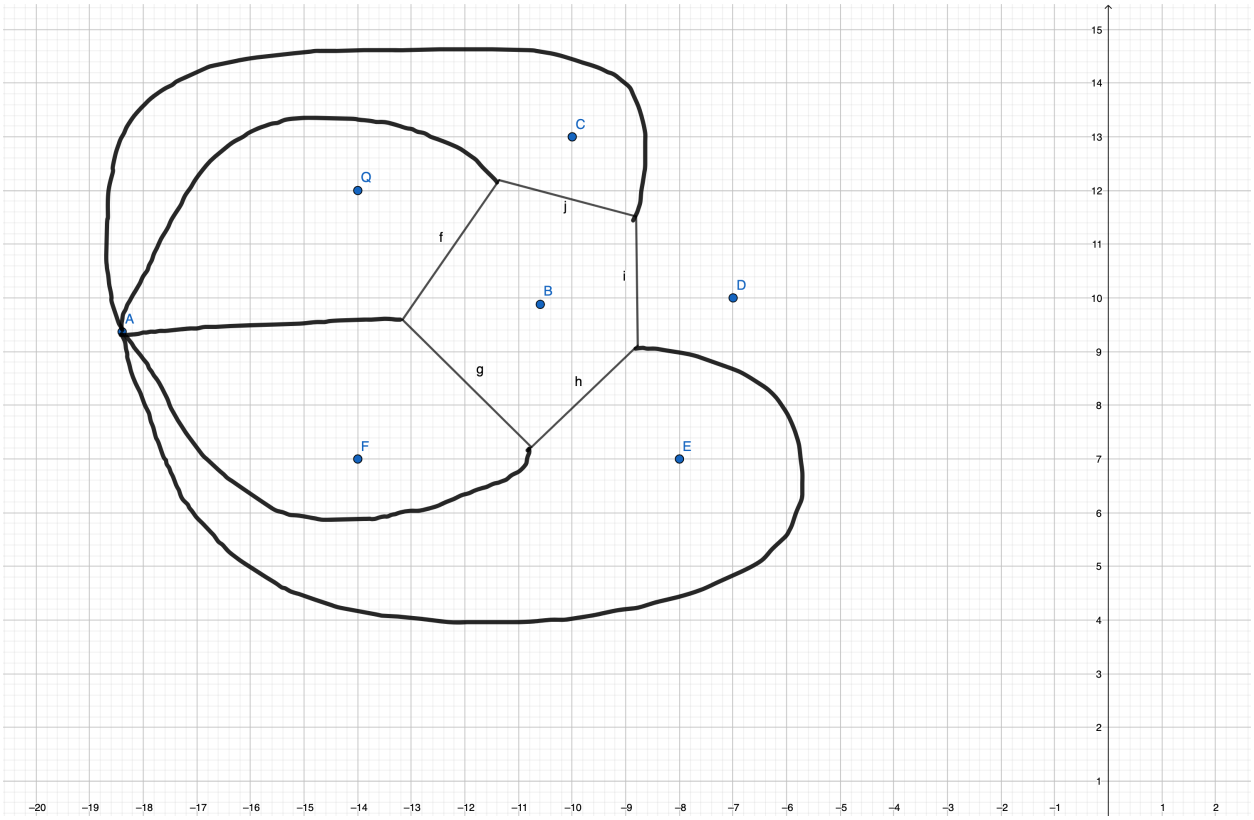

Figure 2: planar graph

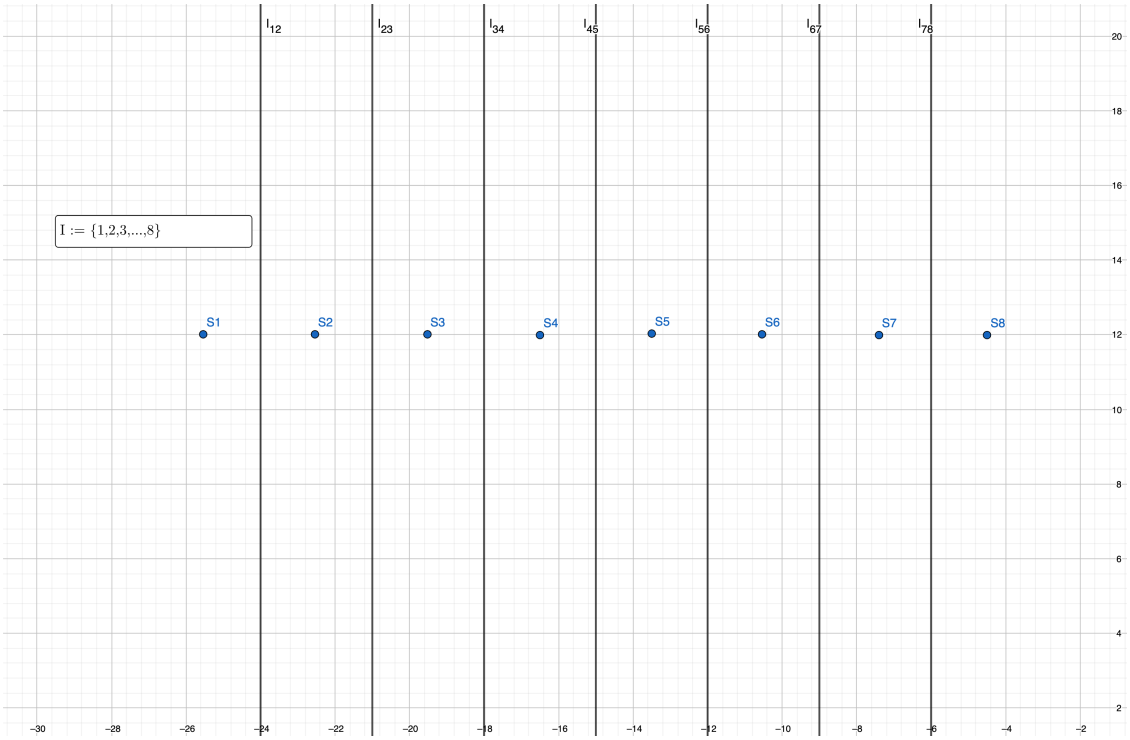

Figure 3: parallel lines

Proof

The main proof contains 2 parts.

1. If all the sites are collinear, refer to the figure 3. (This graph shows an example of 8 points, but it's also apply to n points)

   A first observation told us that there is no voronöi vertex in this graph. we shall proof this claim by contradiction.

   ϟ there exists a voronöi vertex and this vertex is formed by $S_a, S_b, S_c \in I$.

   if we denote this vertex as V_0, then it must satisfy: $|S_aV_0| = |S_bV_0| = |S_cV_0|$

   Denote $V_{mn} \coloneqq \{V_0 \in \mathbb{R}^2 : |S_mV_0| = |S_nV_0|\}$ for $m, n \in I$

   $\Rightarrow V_0 \in V_{ab} \bigcap V_{bc}$

   however, 3 points already determines the center of the circle, which means:

   $\Rightarrow V_0 = V_{ab} \bigcap V_{bc}$

   by definition, V_ab is the bisection of points S_a and S_b

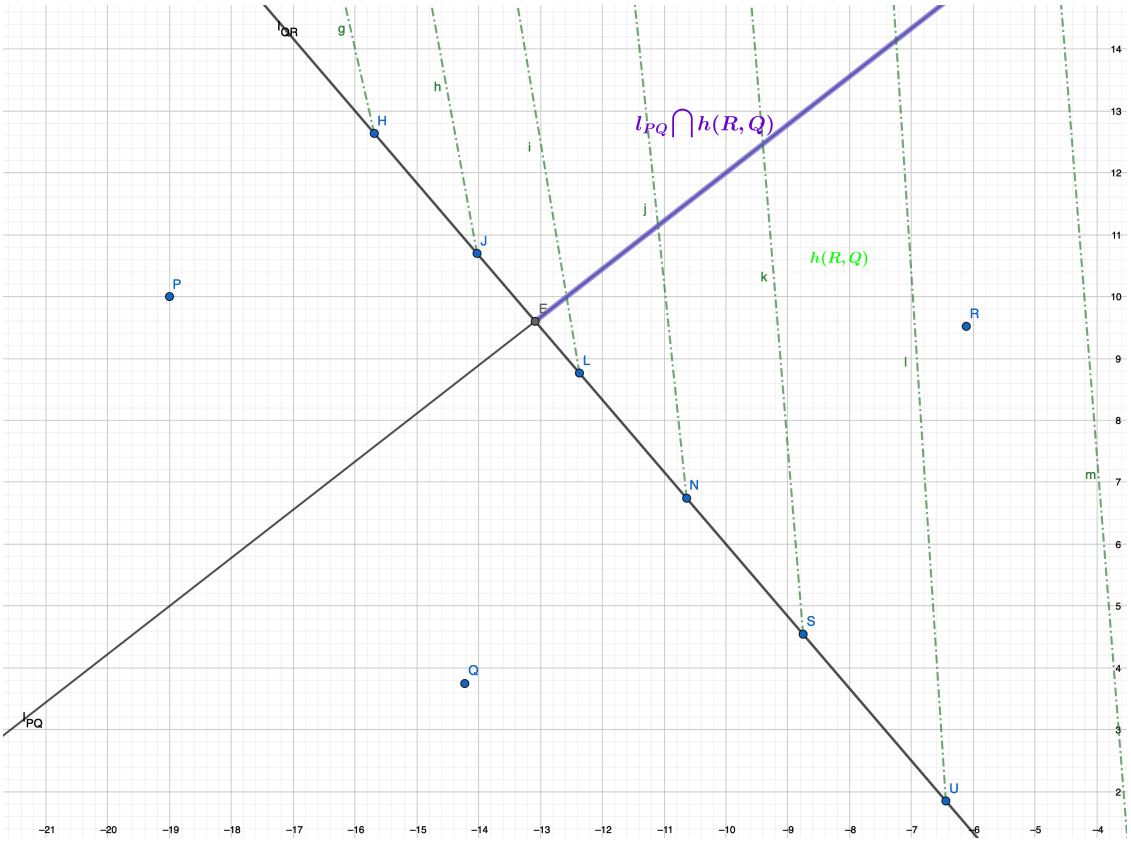

Figure 4: Non collinear case

$\Rightarrow V_a b = l_a b$

Similarly:

$\Rightarrow V_b c = l_b c$

$\Rightarrow V_a b \parallel V_b c$

$\Rightarrow V_0 = V_{ab} \bigcap V_{bc} = \emptyset$

By ruling out the possibility of voronöi vertices. It could only contain voronöi edges, with n cites, it generates (n-1) parallel voronöi edges.

2. If not all the sites are collinear

   now refer to the figure 4

   $\wr \exists R \in \mathbb{P}$, R is not collinear with P and Q

   first by the definition of voronöi system, Vor($\mathbb{P}$) is connected.

   now $\wr$ $\exists$ a voronöi edge l_{PQ} in Vor($\mathbb{S}$) where $\mathbb{S}$ denotes the current sites.

   Assume that l_{PQ} is a whole line (i.e. not a line segment)

   $\Rightarrow \forall x \in h(R, Q)$, we have: $|xP| > |xR|, |xQ| > |xR|$

   Thus by the definition of the half plane $h(R, Q)$

   Some part of the line l_{PQ} shouldn't be the edge of Vor($\mathbb{S}$)

   $\Rightarrow$ edge l_{PQ} is not a whole line. It's bounded by l_{QR} and it's either a line-segment or a half-line.

□

you can access the figure 3 at https://www.geogebra.org/calculator/kxdwsqry and access figure 4 at https://www.geogebra.org/calculator/ge9pgtub

Before proceeding to the Theorem 3, we need another definition about voronöi vertex.

2.6: Charaterization of voronöi vertex

$C_p(x) :=$ The largest circle centered at x without sites in its interior.

Now we are ready for the Theorem 3:

Theorem 3

Let $\mathbb{P} \subset \mathbb{R}^2$ be a n-sites sets.

if x is a voronöi vertex $\Leftrightarrow |C_p(x) \bigcap \mathbb{P}| \geq 3$.

Proof

1. if x is a voronöi vertex $\Rightarrow |C_p(x) \bigcap \mathbb{P}| \geq 3$

   if x is a voronöi vertex, by the definition. It must have 3 or more sites around it.

   i.e. $x = V(\{P_1, P_2, P_3, ..., P_n\})$ with $P_n \in \mathbb{P}$, now use the definition 2.5:

   We get:

   $|xP_1| = |xP_2| = |xP_3| = ... = |xP_n|$

   This is an indication that $P_1, ..., P_n$ are on the circle $\odot A$ which centered at x with radius $|xP_1|$

   Now we have to proof $\odot A$ is the largest circle centered at x without sites in its interior.

   ↾ $\exists Q \in \mathbb{P} \backslash \{P_1, ..., P_n\}$, and assume it sits inside the $\odot A$, it means:

   $|xQ| < |xP_1|$

   However, by the definition of the voronöi vertex of x regarding to site P_1, it must satisfy:

   $|xP_1| < |xM| \quad \forall M \in \mathbb{P} \backslash \{P_1, ..., P_n\} \qquad (\lightning)$

2. if x is a voronöi vertex $\Leftarrow |C_p(x) \bigcap \mathbb{P}| \geq 3$

   By the definition of $C_p(x)$, there is no sites in its interior, which means $C_p(x) \bigcap \mathbb{P}$ contains only the sites on the boundary of the circle $C_p(x)$.

   ↾ $|C_p(x) \bigcap \mathbb{P}| = n \geq 3$, i.e. there are n points around the center x

   explicitly, assume these points are $P_1, P_2, ..., P_n$. Since these points are on the boundary of the circle $C_p(x)$, we have:

   $|xP_1| = |xP_2| = |xP_3| = ... = |xP_n|$

also notice, there are no other sites in this circle, means:

$|xP_1| < |xM| \quad \forall M \in \mathbb{P} \backslash \{P_1, ..., P_n\}$

together, we have: x is a voronöi vertex of Vor($\mathbb{P}$)

□

2.3 Fortune's Algorithm

The Fortune's algorithm is invented to generate the voronöi diagrams in 2 dimensional space. Explicitly, the Fortune's algorithm inputs n sites with a region D, It outputs the voronöi diagrams or deivided region D' with multiple voronöi cells.

Before starting the introduction of this algorithm, two concepts must be introduced first. The whole algorithm is based on the sweeping line algorithm. it means that in this algorithm, it has a horizontal line (sweep line) moves from top to bottom, each time it hits a cite in the point distribution, the system triggers some events. As shown in Figure 5, The Sweep line L is the horizontal line that controls the whole process while the beach line is a monotone curve formed from the closet pieces of parabolic arcs (closest to the sweep line L)

2.7: fortune's algorithm

Given a region D and n sites.

1. Placing the Sweep line L above all the sites and start to move down slowly.
2. Each time when L passes over a new site, trigger the point events.
3. Additionally, when one parabolic arc shrinks to zero, trigger the circle events.
4. As the sweep line L moving down to a point where no events can be triggered, stops the algorithm.

2.3.1 Point events

As shown in figure 6 whenever the point event are triggered, we first shoot a vertical ray up to the parabola that lies above this point in the beach line.Then split the arc into two parts by inserting a new small arc at this points. The newly created parabolic arc has the directrix L (same as the sweep line, we assume that the equation for L is simply: y = L) and focus $P \in \mathbb{P}$ ($P(P_x, P_y)$ is the site that L passes through.) And the arc has the equation: $(x - P_x)^2 = 2(P_y - L)[y - \frac{1}{2}(L + P_y)]$. Finally, as the sweep line goes down, the new arc will expand. Trace the intersection of different parabolic arcs on the beach line, we get the voronöi edges.

2.3.2 Circle events

Refer to figure 7, as the length of a parabolic arc shrinks to zero, then one of the parabolic arc on the beach line will disappear and a new voronöi vertex will be created at this special point.

A small proof to this events is that when one parabolic arc disappears as sweep line goes down as in the second graph in figure 7, if we denote this center point as x, then:
$|xP_i| = |xP_j| = |xP_k|$, hence all three points lies on the same circle centered at x. Denote this circle by $C_p(x)$ and notice that there is no other cites in this circle. Thus, $C_p(x) \bigcap \mathbb{P} = 3$, Finally by the Theorem 3 in section 2.2, x is a voronöi vertex.

2.3.3 Live graph

You could also access this live version of Fortune's Algorithm and play around with it:
https://www.desmos.com/calculator/ejatebvup4?lang=pl

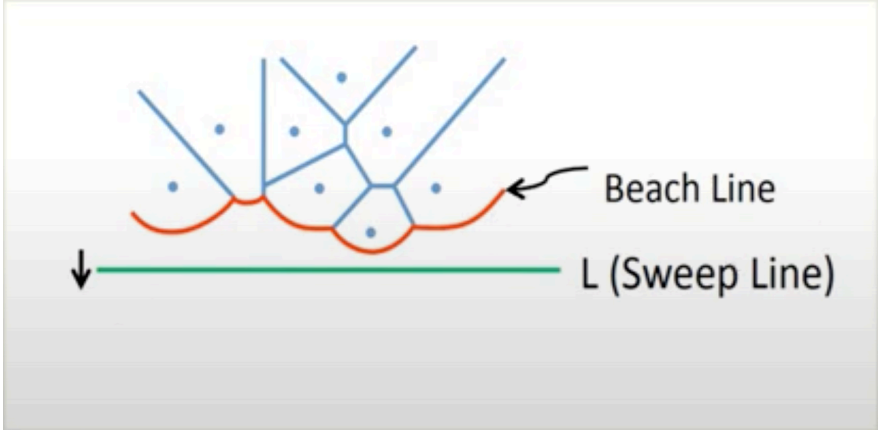

Figure 5: The sweep line and the beach line

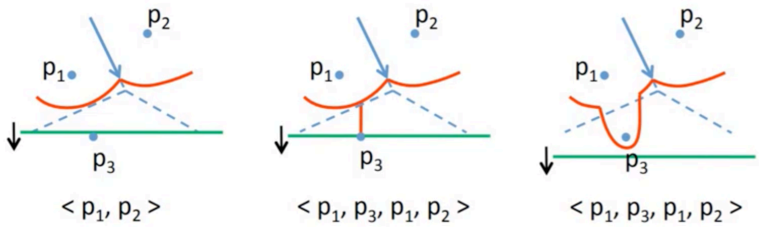

Figure 6: Point events

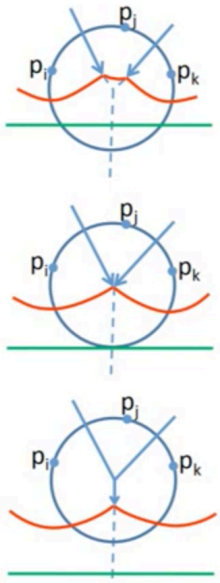

Figure 7: Circle events

Part II

Tammes' problem and point distribution on the sphere

3 Tammes's Problem

Here is an introduction to Tammes's problem and its dual problems. It turns out that Tammes's problem has several different dual problem and may lead to many different ways of calculation.

The Tammes's Problem is named after a Dutch botanist R.M.L. Tammes. He posed this problem when he observed arrangements of exit places on the surface of spherical pollen grains in 1930. The problem is to packing a given number of circles n on the surface of a sphere such that the minimum distance between circles is maximized.

One of the dual problem to the Tammes's problem is to find the the n-point distribution $\{\mathbb{P}\}_1^n$ (point configuration) that maximize the minimal distance between any different points in the distribution. If we denote such maximized distance as d_n, then:

$d_n = max\{min\{P_iP_j : 1 \leq i < j \leq n\}\}$

The connection between these two problems are considering the n-points are the centers of the n circles in Tammes's prblem.

Unfortunately, after passing nearly a decades, although many researches are conducted regarding to this subject. Only few small n has a full solution. For larger n, there is still no solution to it.

4 Condition 1: Well-separation

4.1: Well-separation

There exist absolute constant c > 0 such that for any positive integer n, there is a collection of points $\{P_j^n\}_{j=1}^n$ such that for any j, k $= 1, \ldots,$ n and j $\neq$ k,

$\text{dist}(P_j, P_k) \geq \frac{c}{\sqrt{n}}$

For some constant c and $P_j, P_k \in \mathbb{P}_n$

If we wish to verify the $\mathbb{T}_n$ $\forall n \in \mathbb{N}$ for condition 1, it's sufficient to find the lower bound of the geodesic version of $d_n = max\{min\{P_iP_j : 1 \leq i < j \leq n\}\}$ in the Tammes's Problem. Since they both represent the distance of any two points in the distribution.

If we could successfully find its lower bound, and call it as L_t, then: (Assuming everything is in geodesic distance)

$\Rightarrow d_n = dist(P_j, P_k) \geq L_t \geq \frac{c}{\sqrt{n}}$

$\Rightarrow \sqrt{n}L_t \geq c$

since $\sqrt{n} \geq 1$

$\Rightarrow \sqrt{n}L_t \geq L_t,$

Thus we only have to make $c \leq L_t$, this condition will be satisfied.

However, there are plenty of upper bound for $\mathbb{T}_n$ with the famous Fejes-Toth upper bound found by Dr.L. Fejes-Toth in his early paper.(see [12], 2) But we didn't find many things about its lower bound. The lower paper that mentions a little about his lower bound is in [10]. But we connot verify its validity at this moment due to the lack of resources. Some further study need to be done related to this subject.

5 Condition 2: No clustering around great circles

5.1: No Clustering around great circles

There exist absolute constant C > 0 such that for any positive integer n, there is a collection of points $\{P_j^n\}_{j=1}^n$ such that for any great circle G $\subset \mathbb{S}^2$, The cardinality of $\{j : dist_{\mathbb{S}^2}(G, P_j) \leq \frac{1}{n}\}$ is smaller or equal to a constant C.

$\forall\ P_j \in \mathbb{P}_n$

5.1 Problem Analysis

In order to test this hypothesis with some point distribution, A rudimentary algorithm could be formed which consists the following 3 general steps: (Here we use the point distribution $\mathbb{T}_n$ as an example

1. generate the representing coordinates of each points in $\mathbb{T}_n$
2. Given a specific great circle G, find $dist_{\mathbb{S}^2}(G, P)\ \forall P \in \mathbb{T}_n$ and compare the every one of them with $\frac{1}{n}$,then calculate how many of those geodesic distance is less than $\frac{1}{n}$ and denote this number as l_G
3. By varying the equation of great circle G, find M $:=$ max$\{l_G : \forall$ great circle G inside the unit sphere$\}$

I'll first analyze the feasibility of the above 3 steps one by one in the following paragraph:

5.1.1 Analysis of Step 1:

1. $\mathbb{T}_2$

   It is hard to obtain the coordinates of each points in $\mathbb{T}_n$ even knowing the solution for n-Tammes's problem.

   Try to start with the simplest case with n=2. Since there are only 2 points, the minimum distance between any 2 points in the distribution is the distance of this two

points. Mathematically, if we denote an arbitrary 2-point distribution as $\mathbb{P}_2$, then:

$$Min\{dist(P_i, P_j)\,\forall P_i, P_j \in \mathbb{P}_2\} = dist(P_1, P_2) \text{ for distinct } P_1, P_2 \in \mathbb{P}_2$$

Hence, we only need to maximize the distance of P_1 and P_2 to solve the Tammes's problem with 2 points.

obviously:

$\mathbb{T}_2$ is the configuration where the 2 points are placed in the antipodal position on the sphere.

WLOG, choose these 2 points to be the north and south pole on the sphere, we have the Cartesian coordinates for $\mathbb{T}_2$:

$$\mathbb{T}_2 : (0, 0, 1)\,\&\,(0, 0, -1) \tag{5.1}$$

2. $\mathbb{T}_3$

Things start to become complicated starting from n=3, Dr.L.Fejes-Toth has first found and proved these configuration in 1945 and many other modern proves are formulated by other researchers later on. One of the geometrical prove can be found in this paper (see [3], 76). They proved that $\mathbb{T}_3$ has the configuration such that the 3 vertices form an equilateral triangle and lie in 1 great circle in the sphere.

WLOG, choose 1 point to be the north pole and together with the other 2 points, they form a equilateral triangle in the x-z plane on the sphere:

$$\mathbb{T}_3 : (0, 0, 1), (-\frac{\sqrt{3}}{2}, 0, -\frac{1}{2}), (\frac{\sqrt{3}}{2}, 0, -\frac{1}{2}) \tag{5.2}$$

The following graph is the graphical configuration for $\mathbb{T}_2$,$\mathbb{T}_3$,$\mathbb{T}_4$,$\mathbb{T}_5$,$\mathbb{T}_6$ respectively.

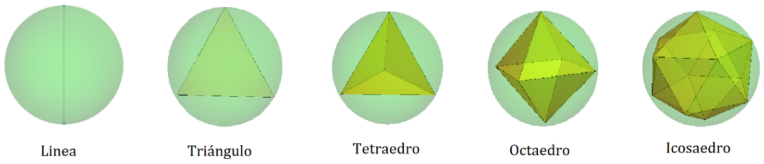

Figure 8: Solutions for $\mathbb{T}_n$ (Antoniano, 2019)

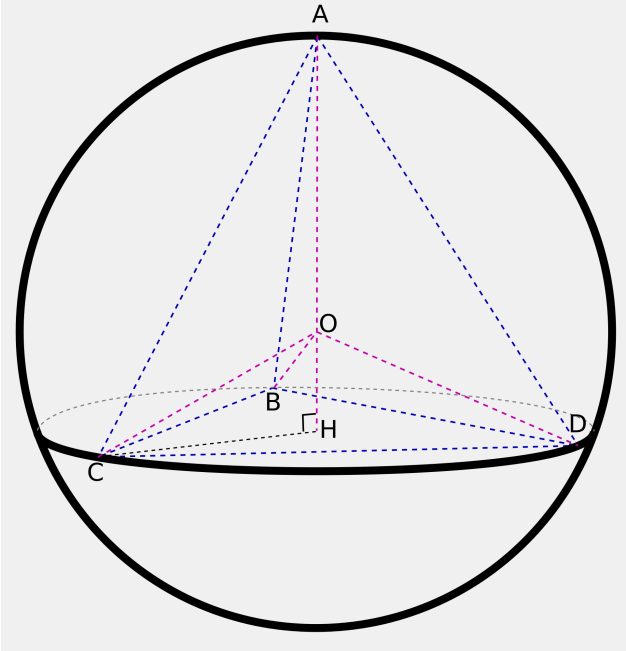

Figure 9: Tetrahedral

3. $\mathbb{T}_4$

The solution of 4-point Tammes's problem is more complicated but similar to $\mathbb{T}_3$ (see [3] and [12]). They found the general configuration for $\mathbb{T}_4$ is a regular tetrahedral.

WLOG, choose 1 point of the tetrahedral to be the north pole, we could calculate the coordinates of rest of the points according to the Figure 9 and get one of the possible representation:

$$\mathbb{T}_4 : (0,0,1), (0, \frac{\sqrt{3}}{2}, -\frac{1}{2}), (\frac{3}{4}, -\frac{\sqrt{3}}{4}, -\frac{1}{2}), (-\frac{3}{4}, -\frac{\sqrt{3}}{4}, -\frac{1}{2}) \tag{5.3}$$

For higher n points, the solution is either non-regular or extremely hard to calculate. See [7], they spent most of their paper in calculating the exact coordinates of $\mathbb{T}_{10}$. there is only a numerical result in this paper and extremely hard to solve.

There is a way that in general could give us a numerical solution for any $\mathbb{T}_n$, but the calculation takes days even weeks just for calculating $\mathbb{T}_n$, $\forall n \leq 100$, see [2] and [1] for detail.

5.1.2 Analysis of Step 2:

To achieve the goal in step 2, all we need to do is calculating the geodesic distance from one point in to G (G is not varying). For example, if we wish to calculate the distance from point A to the great circle lying in x-y plane in the Figure 10. First, we have to find the orthogonal great circle to G. If the projection of A is lying on the diameter of the great circle G as shown in the Figure 10 (Projection of A is B). The orthogonal great circle is $\odot$ OCD lying in x-z plane. Generally, the orthogonal circle is formed by a great circle going through the point itself and across through the projection of the point. Then the desired geodesic distance is the geodesic distance from points to the intersection point. As an example, the desired geodesic distance in Figure 10 is simply the arc AEC. In order to calculate it, we could connect the line between point A and its projection B. Hence, it's easy to see:

$$\begin{aligned}
\text{dist}(A, G) &= \widehat{AEC} \\
&= r \times \angle AOB \\
&= r \times \arcsin \frac{AB}{r} \\
&= \arcsin AB \quad \text{(unit sphere, r=1)}
\end{aligned}$$

The above calculation applies to the general case as well.

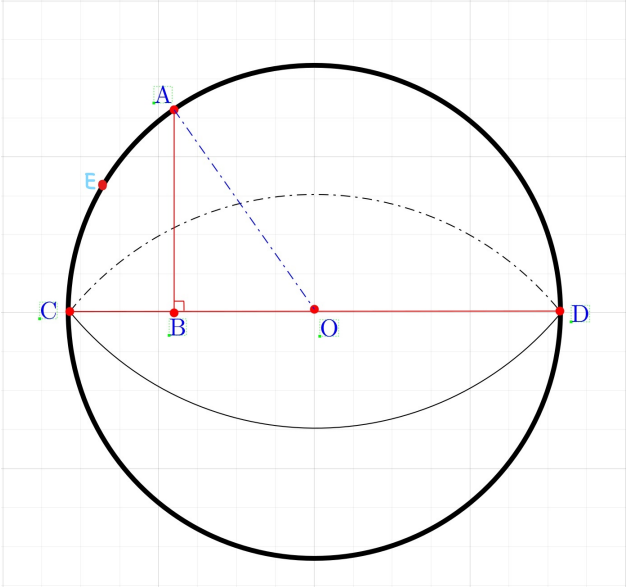

Figure 10: Great circle G is $\odot$ OCD lying in x-y plane

Also notice, for some arbitrary great circle G_a which lies in the plane $P : ax+by+cz = 0$ (across through point$(0, 0, 0)$), we have: (with the same convention as above regarding to point A and B

$$\begin{aligned} \text{dist}(A, G_a) &= \arcsin AB && (5.4)\\ &= \arcsin(dist_c(A, P)) && (5.5)\\ &= \arcsin(\frac{|ax_A + by_A + cz_A|}{\sqrt{a^2 + b^2 + c^2}}) && (5.6) \end{aligned}$$

where disc_c denotes the normal Cartesian distance and point A has Cartesian coordinate (x_A, y_a, z_a).

So far, we have obtained the explicit form of dist(A, G). But the biggest problem is about to come.

5.1.3 Analysis of Step 3:

Based on the fact that we couldn't compare the dist(A, G) with $\frac{1}{n}$ from an arbitrary great circle G. the step 2 is theoretically impossible to finish. But if we consider the sum of all the distant from points to great circle G, denote it as S. theoretically, we could get a minimum state of distance among all the points. and later on compare the distance of optimal state with $\frac{1}{n}$. Then use it as our final result for step 3.

this is a really hard question to be honest, I have to make some compromise to at lease make the calculation become feasible.

Mathematically, we transfer our goal to achieve the following optimization problem:

$$\underset{G}{Min}\{\sum_{\forall P_i \in \mathbb{T}_n} \text{dist}(P_i, G)\}$$

Regarding to a variable great circle G lying on the variable plane P: ax+by+cz=0

Suppose this optimization problem could be solved, and denote the optimal great circle as G_o. Then, we could simply apply the distance formula 5.6 to calculate the distance from every point to the great circle respectively. if we denote these distances as:

$$D_i = dist(P_i, G_o) \qquad \forall P_i \in \mathbb{T}_n$$

then:

$$\#\{j : dist_{\mathbb{S}^2}(G, P_j) \leq \frac{1}{n}\} = \sum_{\forall P_i \in \mathbb{T}_n} \delta(D_i)$$

where $\delta(D_i) = \begin{cases} 1 & \text{for } D_i \leq \frac{1}{n} \\ 0 & \text{otherwise} \end{cases}$.

This is not the end of the story even after proceeding all the steps above. Based on our algorithm, we could merely calculate all the the constant $C_n > 0$ for a specific $\mathbb{T}_n$. But in

theory, we wish to find a universal $C > 0$ which is unrelated to the number n. Hence, our final result is only a reference on what the final C may look like. It is clear that $C \geq C_n \, \forall n$ since this C must satisfy the condition 2 for any n.

5.2 Code construction

In this subsection, a brief introduction to our code construction in Matlab will be shown.

5.2.1 point distribution generator

As we follow the analysis in subsection 5.1, we first need to generate the point distribution for $\mathbb{T}_2, \mathbb{T}_3, \mathbb{T}_4$ and possibly $\mathbb{T}_{10}$. As shown in Figure 11 and 12, we first load all the data into one big file called 'data.mat'. Then, use '$\mathbb{T}_4$ generator' to retrieve all 4 points in $\mathbb{T}_4$ from 'data.mat' and resize them to be a (3x4) vector array called 'X_{dist}'. follow the same idea, use '$\mathbb{T}_{10}$ generator' to retrieve and resize all the points in $\mathbb{T}_{10}$,we shall get a similar vector array with size (3x10) called 'Y_{dist}'. The data points we used in 'data.mat' are from our analysis above in subsection 5.1-1 and [7].

5.2.2 distance function

The second step in our code construction is to create a function for the geodesic distance calculation. based on the formula 5.6, we have the following code block in figure 13. In the code, it calculates the geodesic distance from one point X(X(1), X(2), X(3)) to a plane P where the great circle G lies. The plane P is a 1-dimensional row vector determined by 3 parameters [a, b, c], such that the plane equation is $P : ax + by + cz = 0$.

5.2.3 optimizer

The next step is to create our objective function and solve it with some optimizer. I originally write down my own gradient descent method, which follows the rudimental gradient descent. (calculate the gradient of variables in objective function and slowly move the function value from one starting points according to the gradient direction with the hope that it will eventually converge). But soon after I found this method is not well optimized for the system and cause a slow running time. Hence, in the figure 14, I simply adopt one of the built-in optimizer in Matlab toolbox to speed it up.

```
function X_dist = get_X_dist()
S = load("data.mat");
dist = zeros(3, 4);
for i = 1:4
    idx = sprintf('X_%d', i);
    dist(:, i) = S.(idx);
end
X_dist = dist;
end
```

Figure 11: $\mathbb{T}_4$ generator

In this solution function, it is necessary to first create the objective function based on our analysis in subsection 5.1-3. Then, we simply call the built-in optimizer 'fminunc' to solve the do the gradient descent. It is also worth noting that a starting point is necessary for this algorithm to proceed. As you can see this the code block, we set the starting point to [1, 1, 1] just for simplicity and with some sort of randomness.

The choose of the starting point could in theory affect our final results if there are a lot of local minimums in our objective function. But after the experiments of 15 different different starting points, we always get the same optimal solution. It doesn't confirm that we actually reach the global minimum, but at least we have a feeling that this optimal solution might be close enough to the global minimum.

Back to the function itself, it takes only one variable Pt_dist as input and output the optimal solution for plane P. Here, the Pt_dist represents the points distribution in $\mathbb{T}_n$ and the optimal plane P_o is represented by 3 optimal variables $[a_o, b_o, c_o]$ with the corresponding equation: $P_0 = a_o x + b_o y + c_o z = 0$.

5.2.4 Cadinality of the clustered points

At last, in figure 15 we still have to recover the cardinality of all the clustered points around the great circle. Hence, we create a simple 'for loop' to calculate the distance from every point in $\mathbb{T}_n$ to the optimal great circle G_o(or optimal plane P_o) and select those valid point according to $\delta(D_i)$ mentioned in subsection 5.1-3. Meanwhile, for the purpose of the future use, we also point out those valid points in the command window.

```
function Y_dist = get_Y_dist()
S = load("data.mat");
dist = zeros(3, 10);
for i = 1:10
    idx = sprintf('Y_%d', i);
    dist(:, i) = S.(idx);
end
Y_dist = dist;
end
```

Figure 12: $\mathbb{T}_{10}$ generator

```
function dist = dist_X_P(X, P)
abs_value = abs(P(1)*X(1)+P(2)*X(2)+P(3)*X(3));
sqrt_value = sqrt((P(1))^2+(P(2))^2+(P(3))^2);
dist = asin(abs_value / sqrt_value);
end
```

Figure 13: Distance function from point to G

```
function P_sol = get_sol(Pt_dist)
[pt_dim, n_pts] = size(Pt_dist);


%create the objective function:
    function obj_func = get_obj(P_var)
        % loop through the pts:
        sum_dist = 0;
        for i = 1:n_pts
            sum_dist = sum_dist + dist_X_P(Pt_dist(:, i).', P_var);
        end
        obj_func = sum_dist;
    end

% optimize the function:
    start_pt = [1, 1, 1];
    P_sol = fminunc(@get_obj, start_pt);

end
```

Figure 14: Solution function

```
function number_in_c2 = get_c2(P_sol, Pt_dist)
num_temp = 0;
[pt_dim, n_pts] = size(Pt_dist);
dist_idx = 1/n_pts;
for i = 1:n_pts
    geo_dist = dist_X_P(Pt_dist(:, i), P_sol);
    if geo_dist <= dist_idx
        num_temp = num_temp + 1;
        ans_row = Pt_dist(:, i).';
        fprintf('Found a valid pt in column %d:', i);
        disp(ans_row);
    end
end
number_in_c2 = num_temp;
end
```

Figure 15: Final answer of clustered point

5.3 Calculation results analysis

5.3.1 Results for $\mathbb{T}_2$

With no surprise, the optimal plane is $P_0 : x = 0$ with optimal variable [1, 0, 0] as shown in the figure 16. But one thing to notice is that our solution for $\mathbb{T}_2$ is only one of the infinite many solution. Theoretically, any great circle passes through the antipodal points could work. Since in any of these cases, the sum of geodesic distance of all the points in $\mathbb{T}_2$ to the great circle is 0 and thus make it an optimal solution. In such case, the number of clustered points around the great circle is 2. This is also the maximum number of clustered points we could get in such configuration with

$$Max\{\#\{j : dist_{\mathbb{S}^2}(G, P_j) \leq \frac{1}{2}\}\} = n = 2$$

A graphical result can be seen in the figure 17 as below. In this graph, great circle C represents our optimal great circle corresponds to plane $P_0 : x = 0$. Another possible solution C2 is also shown in the graph, which corresponds to another optimal plane $P_2 : y = 0$.

Click this link if you want to see the whole graph for $\mathbb{T}_2$ yourself and play with it.

https://www.geogebra.org/calculator/yry46acv

5.3.2 Results for $\mathbb{T}_3$

The solution for $\mathbb{T}_3$ is also not a surprise since all the 3 points are in the same plane. The calculation told us: the optimal plane in this case is $P_0 : y = 0$ with optimal variable

```
>> disp(P_sol_t2);
     1     0     0

>> disp(P_sol_t3);
     0     1     0

>> disp(c2_for_t2);
     2

>> disp(c2_for_t3);
     3
```

Figure 16: Results for $\mathbb{T}_2$ and $\mathbb{T}_3$

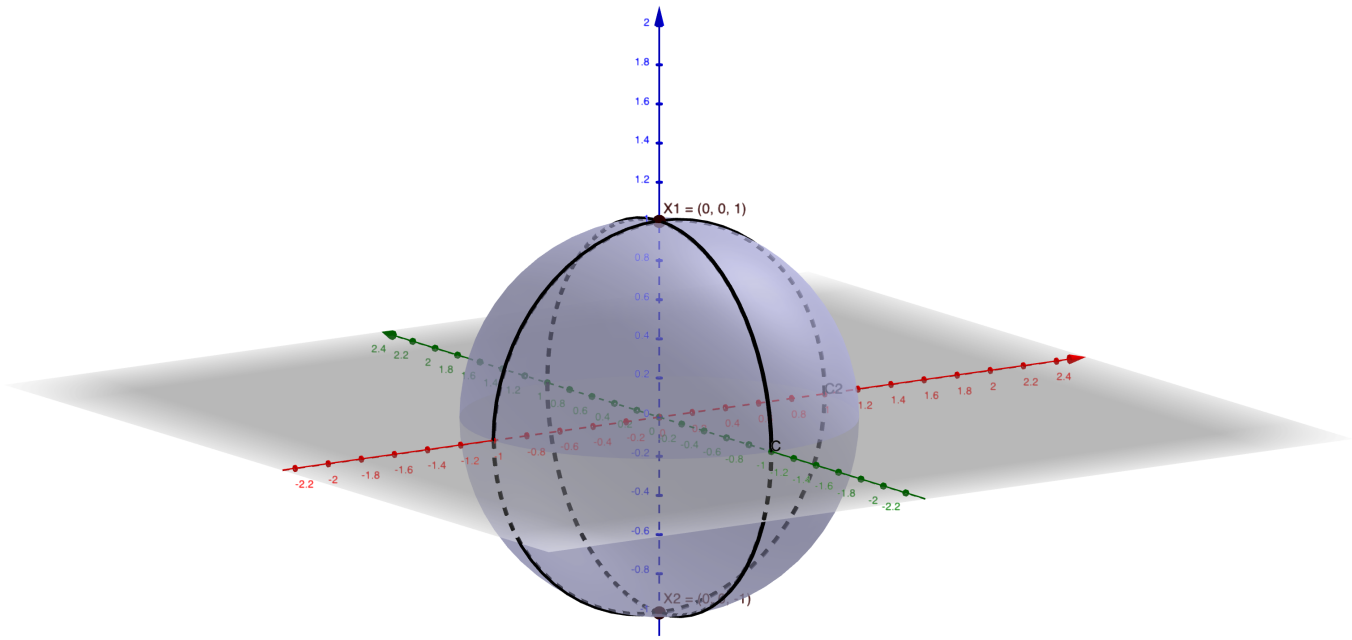

Figure 17: Graphical results for $\mathbb{T}_2$

[0, 1, 0]. this result also conform with our visualization since when 3 points lie in the great circle, then the sum of geodesic distance of all the points in $\mathbb{T}_3$ to the great circle is also 0 and reaches the optimal solution. Same as before in $\mathbb{T}_2$, this is the maximum number of clustered points we could possibly get. In this case:

$$Max\{\#\{j : dist_{\mathbb{S}^2}(G, P_j) \leq \frac{1}{3}\}\} = n = 3$$

But there is one difference between the results of $\mathbb{T}_2$ and the results of $\mathbb{T}_3$. it should not exist infinitely many optimal solutions for $\mathbb{T}_3$ whereas $\mathbb{T}_2$ has infinitely solutions. the calculation results also support our arguments. As we change the different starting points of the gradient descent algorithm, the optimal solution for $\mathbb{T}_2$ varies while the solution stays the same for $\mathbb{T}_3$ case as we change to the different starting points.

A graphical results could be found in the figure 18 with the optimal great circle C passing through all the points in $\mathbb{T}_3$.

The link for this graph is: https://www.geogebra.org/calculator/z3peebt5

5.3.3 Results for $\mathbb{T}_4$

The story starts to become complicated for the case of $\mathbb{T}_4$, there is simply no guessable answer here since not all 4 points in the distribution are in the same plane. The optimal answer we got from the command line could be found in figure 19. And the graphical results could be found in figure 20.

As usual, you could access this graph by clicking the link: https://www.geogebra.org/calculator/d9puknkh

In figure 19, we could see that the optimal plane for the point distribution is $P_o = [1.1491, 0.9496, 0.9013]$ with corresponding plane equation: $1.1491x + 0.9496y + 0.9013z = 0$. This is also the plane where the optimal great circle C lies in the figure 20. As we compare our optimal distance (The distance from each points to optimal great circle) to $\frac{1}{n} = \frac{1}{4}$,

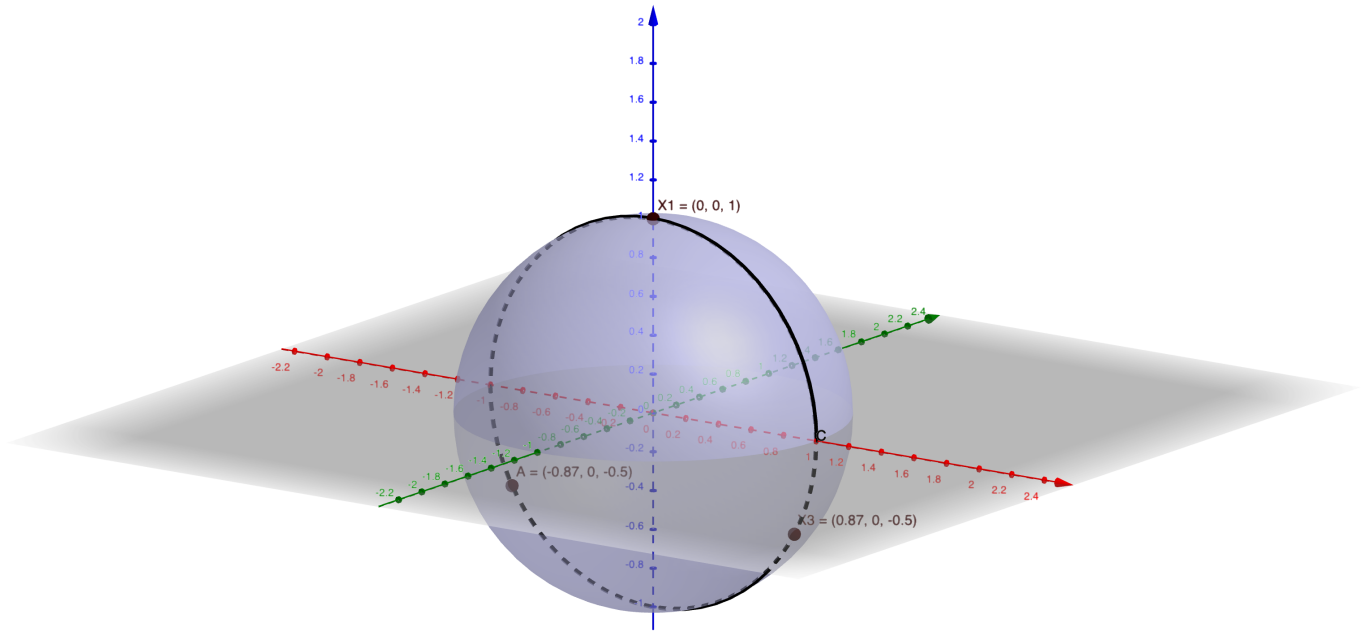

Figure 18: Graphical results for $\mathbb{T}_3$

```
>> P_sol_X = get_sol(X_dist);

Local minimum possible.

fminunc stopped because it cannot decrease the objective function
along the current search direction.

<stopping criteria details>
>> disp(P_sol_X);
    1.1491    0.9496    0.9013

>> c2_for_t4 = get_c2(P_sol_X, X_dist);
Found a valid pt in column 2:         0    0.8660   -0.5000

Found a valid pt in column 3:    0.7500   -0.4330   -0.5000

>> c2_for_t4;
>> display(c2_for_t4);

c2_for_t4 =

     2
```

Figure 19: command line output results for $\mathbb{T}_4$

only two points are valid to be categorized into clustered points. These two points are: $(0, \frac{\sqrt{3}}{2}, -\frac{1}{2}), (\frac{3}{4}, -\frac{\sqrt{3}}{4}, -\frac{1}{2})$ which has been highlighted in figure 20 using red colour.

Overall, in this case with n = 4:

$$Max\{\#\{j : dist_{\mathbb{S}^2}(G, P_j) \leq \frac{1}{4}\}\} = 2$$

But do notice that our calculation is based on the gradient descent algorithm. The existence of another optimal solution is possible by varying the starting points. As mentioned before, we have attempted many different starting points with the same settings. Although we always get the same optimal results in our experiments, it does not represent that we have found the global optimal solution. But this is at least a repetitive local optimal solution, it could help us to get some sense of the optimal configuration we are after. With some close observation of the figure 20, we could find that only one point (x_2) is on the optimal great circle C.

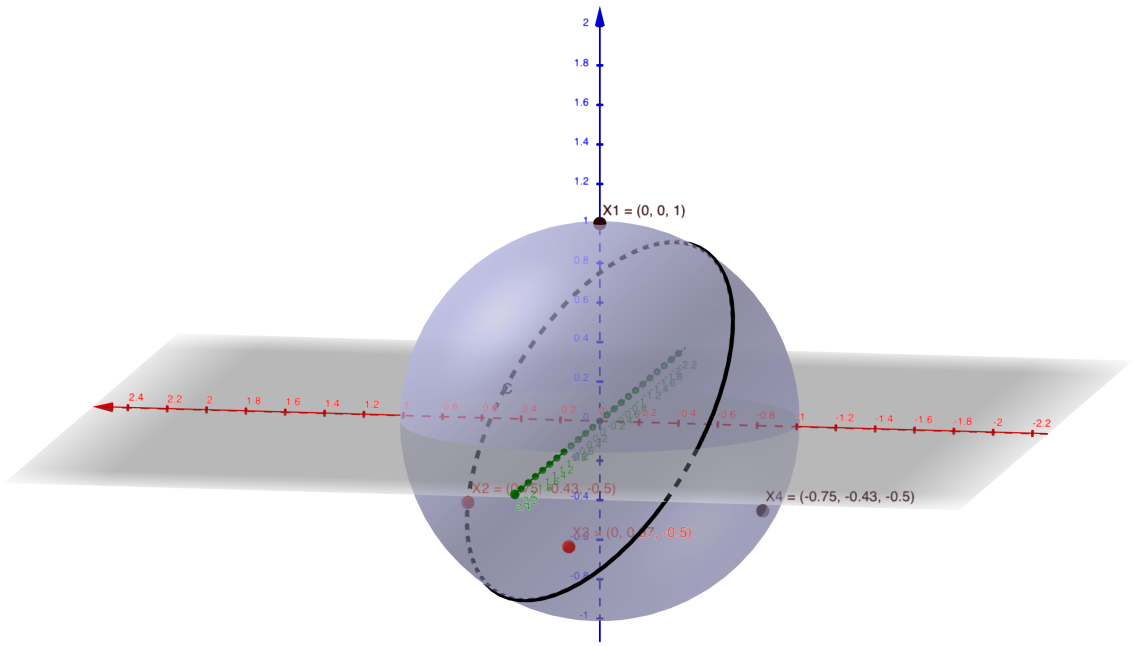

Figure 20: Graphical results for $\mathbb{T}_4$

5.3.4 Results for $\mathbb{T}_{10}$

As we mentioned in the previous section, the coordinates of all the points in $\mathbb{T}_{10}$ are extremely hard to get. However, thanks to the results from this paper [7], we are able to run our algorithm. And the result we get is pretty interesting as shown in the figure 21 and figure 22.

As usual, you could gain the access for this graph at : https://www.geogebra.org/calculator/n4yatawk

First of all, we have to emphasis this is not said to be the global optimal solution, rather than a repetitive local minimum which could give us a sense of intuition like above.

From figure 21, the results give us the optimal plane $P_o = [0.6830, 1.2316, 1.5448]$ with corresponding plane equation: $P_o : 0.6830x + 1.2316y + 1.5448z = 0$. The optimal great circle C is illustrated in the figure 22. Same as the case in $\mathbb{T}_4$, the optimal great circle C only passes through 1 point (x_3) in the distribution. one surprising result is that the final clustered points is still 2 (marked with red in figure 22) even though it has 10 points now. based on some observation on figure 22, it's not too hard to find:

1. $\mathbb{T}_{10}$ seems to have some sort of symmetry around the z-axis in this graph. and such symmetry scatters the points to be far enough within the bandwidth around one great circle.

2. As n increases, the bandwidth $\frac{2}{n}$ around the great circle decreases. It's harder for points to be in the bandwidth if they are distributed evenly.

Overall, in this case with n = 10:

$$Max\{\#\{j : dist_{\mathbb{S}^2}(G, P_j) \leq \frac{1}{10}\}\} = 2$$

```
>> P_sol_Y = get_sol(Y_dist);

Local minimum possible.

fminunc stopped because it cannot decrease the objective function
along the current search direction.

<stopping criteria details>
>> disp(P_sol_Y);
    0.6830    1.2316    1.5448

>> c2_for_t10 = get_c2(P_sol_Y, Y_dist);
Found a valid pt in column 3:    0.9146         0   -0.4044

Found a valid pt in column 7:   -0.1388   -0.7833    0.6060

>> disp(c2_for_t10);
     2
```

Figure 21: command line output results for $\mathbb{T}_{10}$

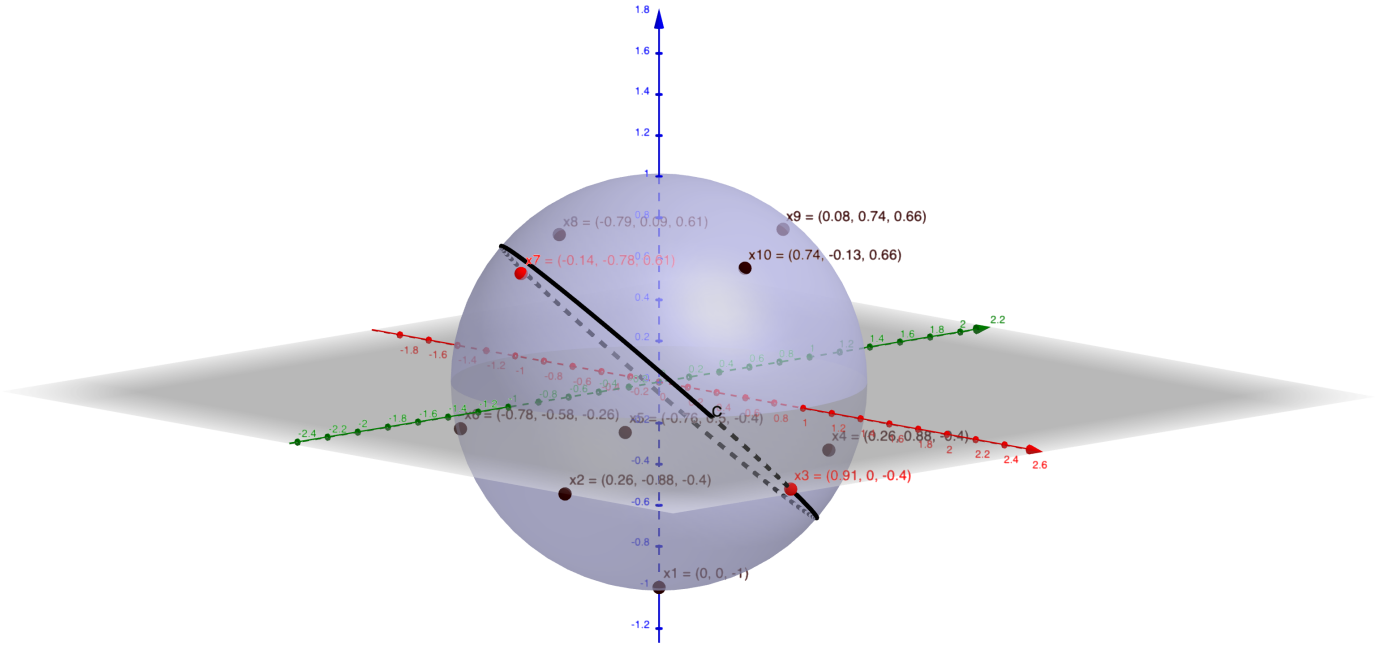

Figure 22: Graphical results for $\mathbb{T}_{10}$

5.3.5 Summery for the results

Based on the results we see so far in the last subsection, we still have nothing to say about the condition 2 since our goal is not to get C_n for some specific n. We want to find a special $C > 0$ which is independent of n. It is impossible to test all the $\mathbb{T}_n$, not to say that the solution of Tammes problem remains a mystery as well. Thus we could only hope that some abstract common symetries can be found between different $\mathbb{T}_n$, and use such symmetry to generate $C > 0$ that satisfies the condition 2.

One thing we could say based on the above results is that $C \geq C_n \, \forall n \in \mathbb{N}$, thus we get : $C \geq 3$.

One possible hypothesis is that: if $\mathbb{T}_n$ always contains some symmery $\forall n \in \mathbb{N}$ that makes points nearly evenly distributed in the sphere. As n increases, the bandwidth $\frac{2}{n}$ within any great circle G could only contain a finitely many number of points and thus to proof the C is converge as n $\to \infty$.

However, this is a huge hypothesis, there are many mysteries in this claim we cannot solve at this time.

Finally, here is a little summery table for all the final results:

n	$C_n = Max\{\#\{j : dist_{\mathbb{S}^2}(G, P_j) \leq \frac{1}{n}\}\}$
2	2
3	3
4	2
10	2

6 Condition 3: Equi-distribution/Evenly distributed

6.1: Equi-distribution

For any positive integer n, there is a collection of points $\{P_j^n\}_{j=1}^n$ such that For any $f \in C^\infty(\mathbb{S}^2)$,
$\lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^n f(P_j^n) = \frac{1}{4\pi} \int_{\mathbb{S}^2} f(P)dP$

This condition should have been the most difficult one to verify for me since it concerns the spherical harmonics and even coding theory. However, some paper have mentioned that Tammes' problem is actually one of the natural way to build equi-distributed point distribution.(see [5], 294) thus, the solution of any Tammes's problem naturally becomes a well-distributed point sets. This is an non obvious but no mystery result.

More study regarding this subject is required.

Part III

Evenly distributed point distribution construction

7 Introduction to group-theoretic point distribution construction

In this last part of the report, we will use the point distribution construction method explained in [14] and discuss its possibility of satisfying those 3 conditions in part 2 of this report. In their paper [14], they pose a new method to build a point distribution with the optimal uniformity in an efficient way.

Worth-noting that their 'uniformity' has a high resemblance to the meaning of 'evenly-distributed' in condition 3.

They wish to make δf as small as possible, where

$$\delta f = \frac{1}{n}\sum_{j=1}^{n} f(x_j) - \int f$$

(From here onwards, we shall use $\mathbb{D}_n$ to denote the n-points distribution generated by group-theoretic construction method in [14].)

However, in condition 3, we want to verify whether such δf is asymptotically 0 as n $\rightarrow \infty$. A detailed discussion will reserved for later sections.

Their idea of construction originates from the 3-D rotational group $SO(3)$. Instead of constructing each point individually, they form a countable subgroup $\Gamma \leq SO(3)$ and apply each rotational elements in the subgroup to one of the points on the sphere. Thus creating the whole distribution systematically. One benefit of such construction is rotation invariant. No matter what starting point you choose on the sphere, the resulting distribution will be identical with respected to sphere rotation.

As for the construction of such particular subgroup Γ, it should satisfy the condition mentioned in Theorem 1.5 in [14].

8 $\mathbb{D}_n$ and Condition 1

To verify condition 1, we need to show that for any positive integer n, there is a corresponding $\mathbb{D}_n$ such that for any $j, k = 1, ..., n$ and $j \neq k$,

$\text{dist}(x_j, x_k) \geq \frac{c}{\sqrt{n}}$

for constance c and $x_j, x_k \in \mathbb{D}_n$.

To tackle this problem, we need to consider the following 2 steps:

1. we wish to find such distribution $\mathbb{D}_n$ for any n. But so far, we only know how to construct a subgroup Γ that satisfies the condition in Theorem 1.5 in [14]. Thus, the order of such group would determine the size of n. Reversely, it is still unknown to us that if we could find such $\mathbb{D}_n$ given a particular size n.

2. if we only consider a known n with its distribution $\mathbb{D}_n$. By the remark (b) in (S150, [14]), we have:

$$dist(\gamma_i, \gamma_j) \geq \frac{1}{\sqrt{n}} \quad \text{for} \quad i \neq j$$

Here, the distance between γ_i and γ_j is the distance of two points in $\mathbb{D}_n$ which transformed from a same fixed point on the sphere by γ_i and γ_j respectively. Hence, such distance is the distance of any two distinct points in the distribution. Clearly, if we simply choose c = 1, the distance condition will be satisfied for such a particular n.

In conlusion, The points in $\mathbb{D}_n$ are well separated for many n. but whether the condition 1 can be fully satisfied depends on whether we could find a 1-1 correspondence between n and $\mathbb{D}_n$ as stated in step 1 above.

9 $\mathbb{D}_n$ and Condition 2

The condition 2 is always the hardest one to verify among these 3 conditions. I had a few ideas to potentially verify this condition.

1. I tried to followed the path of section two and calculate the optimal circle for a particular n and $\mathbb{D}_n$. But the size of the distribution is simply too big for manual calculation. Maybe it is due to my lack of understanding of the core idea of theorem 1.5 in [14] and thus finding a proper small size distribution becomes harder.

2. If we change our point of view. Instead of focusing on calculate every individual points in the distribution, we choose to observe the global property of the point generator—The rotational group Γ. By the property of group, every element in Γ should have an inverse. Thus if γ_i takes $x \in \mathbb{S}^2$ to a point a and γ_i^{-1} takes $x \in \mathbb{S}^2$ to a point b, then a and b should be symmetrical regarding to one of the great circle across x. In such case, we could thus choose a reference great circle (or plane) across x. By only considering the distance from one point to this reference great circle. We always get the identical distance from two inverse pairs (eg. $\gamma_i x$ and $\gamma_i^{-1} x$) separated by the reference great circle. from this process, we could simplify the problem by cutting the other half of the sphere and only consider half of the points on the sphere. I tried to use the geometrical inequality of the distance to proceed and failed. Some other works must be done in order to pursue this idea.

10 $\mathbb{D}_n$ and Condition 3

The condition 3 is also the main focus of their paper in [14]. Just as we discussed before in the introduction section. We wish to make $\delta f \to 0$ as $n \to \infty$. This is not a hard question anymore given we have many results from the paper [14]. It shows that:

$$\delta(\gamma_1, \cdots, \gamma_{2n}) \leq \frac{log(n)}{\sqrt{n}}$$

(S150, [14])

Here, the $\delta(\gamma_1, \cdots, \gamma_{2n})$ is defined as the operator discrepancy of the sequence from γ_1 to γ_{2n}:

$$\delta(\gamma_1, \cdots, \gamma_{2n}) = \sup\{||\delta f||_2 : ||f||_2 = 1\}$$

Since we have the upper bound of $\delta(\gamma_1, \cdots, \gamma_{2n})$ and by definition, we have the upper bound of $||\delta f||_2$.

Next, if we calculate:

$$\begin{aligned}
0 \leq \lim_{n\to\infty} \delta(\gamma_1, \cdots, \gamma_{2n}) &\leq \lim_{n\to\infty} \frac{log(n)}{\sqrt{n}} \\
&= \lim_{n\to\infty} \frac{2}{\sqrt{n}} \quad \text{(by L'hospital's rule)} \\
&= 0 \\
\Rightarrow \lim_{n\to\infty} \delta(\gamma_1, \cdots, \gamma_{2n}) &= 0 \\
\Rightarrow \lim_{n\to\infty} ||\delta f||_2 &= 0 \\
\Rightarrow \lim_{n\to\infty} \delta f &= 0 \\
\Rightarrow \delta f \to 0 \quad \text{as} \quad n \to \infty & \qquad \square
\end{aligned}$$

We end up with exactly what we want, hence the condition 3 is fully satisfied for $\mathbb{D}_n$

References

[1] L. Hars, *Numerical solutions for midsize tammes problems: N = 61...100*, Dec. 2020. (visited on 06/24/2022).

[2] L. Hars, *Numerical solutions of the tammes problem for up to 60 points*, Nov. 2020. [Online]. Available: https://www.hars.us/Papers/Numerical_Tammes.pdf.

[3] E. Antoniano, "The tammes problem projective stiefel manifolds view project the tammes problem el problema de tammes," 2019. DOI: 10.46571/JCI.2019.1.8. (visited on 06/24/2022).

[4] J. Berry, M. Dannenberg, J. Liang, and Y. Zeng, *Relaxed disk packings*, 2015. (visited on 06/24/2022).

[5] J. S. Brauchart and P. J. Grabner, "Distributing many points on spheres: Minimal energy and designs," *Journal of Complexity*, vol. 31, pp. 293–326, Jun. 2015. DOI: 10.1016/j.jco.2015.02.003. (visited on 06/30/2022).

[6] O. R. Musin and A. S. Tarasov, *The tammes problem for ¡i¿n¡/i¿ = 14*, Jul. 2015. DOI: 10.1080/10586458.2015.1022842. (visited on 06/24/2022).

[7] T. SUGIMOTO and M. TANEMURA, *Exact value of tammes problem for n = 10*, Sep. 2015. [Online]. Available: https://arxiv.org/abs/1509.01768.

[8] C. Hai-chau and W. Lih-Chung, *A simple proof of thue's theorem on circle packing*, Sep. 2010.

[9] A. Tomaso and W. Denis, *The pursuit of perfect packing second edition*, 2008. (visited on 06/24/2022).

[10] S. E.B. and K. A.B.J., "Distributing many points on a sphere," vol. 9, p. 5, 1997. [Online]. Available: https://perswww.kuleuven.be/~u0017946/publications/Papers97/art97a-Saff-Kuijlaars-MI/Saff-Kuijlaars-MathIntel97.pdf (visited on 07/19/2022).

[11] P. Classical and A Horváth, *On dirichlet-voronoi cell*, 1995. (visited on 06/24/2022).

[12] E. Mooers, *Tammes's problem*, 1994. [Online]. Available: https://pdodds.w3.uvm.edu/teaching/courses/2009-08UVM-300/docs/others/1994/mooers1994a.pdf.

[13] F. Chung and S. Sternberg, *Mathematics and the buckyball*, 1993. (visited on 06/24/2022).

[14] A Lubotzky, R Phillips, and P Sarnak, *Hecke operators and distributing points on the sphere i*, 1986. (visited on 07/19/2022).

[15] H. Xiaolong, *Uniformly bounded spherical harmonics and quantum ergodicity*, CSUN. [Online]. Available: https://www.csun.edu/~xiaolong/Han_BddSH.pdf.