The local covering problem: producing and certifying local coverings

Jim van der Valk Bouman

A Thesis
in
The Department
of
Mathematics and Statistics

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Science (Mathematics) at
Concordia University
Montréal, Québec, Canada

July 2020

Concordia University
School of Graduate Studies

This is to certify that the thesis prepared

By: **Jim van der Valk Bouman**
Entitled: **The local covering problem: producing and certifying local coverings**

and submitted in partial fulfillment of the requirements for the degree of

Master of Science (Mathematics)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining commitee:

_____ Chair

_____ Examiner

Dr. H. Kisilevsky

_____ Examiner

Dr. M. Lipnowski

_____ Supervisor

Dr. G. Rosso

Approved by _____

Debbie Arless

July 13th 2020

_____

Prof. Pascale Sicotte

Abstract

The local covering problem: producing and certifying local coverings

Jim van der Valk Bouman

The covering problem is a classical problem that is concerned with find the smallest collection of structures such that a larger structure is covered. In this thesis we focus on a few examples of the local covering problem, which can be described by:

Given a connected set X on a manifold and a fixed $r > 0$, find the smallest set of points P such that $\bigcup_{p \in P} B_r(p)$ covers X.

We first explore the problem by picking a few easier settings and constructing an algorithm that can explicitly produce coverings of a certain set. We then discuss a result in algebraic topology based on the construction of the nerve of a covering, which allows us to certify that a given collection of opens provides a covering of the set. This inherently interesting result allows us to finally improve on the results of our algorithm. We do this by removing some points from its output configurations and applying gradient flow to move them around such that they repel each other. We then apply the homological criterion to verify that the points give a covering after gradient flow, in which case we have improved our results.

Contents

1 Introduction **1**

2 Producing explicit configurations **2**
2.1 Introduction 2
2.2 The 2-sphere 2
2.2.1 The algorithm 2
2.2.2 First results 3
2.2.3 Proving coverage 3
2.2.4 Two improvements and results of final algorithm 4
2.3 The unit disc in the Euclidean and the Hyperbolic plane 6

3 Homological criterion for verifying coverage **8**
3.1 Introduction 8
3.2 Main lemma and the nerve theorem 8
3.3 Computation through persistent homology 9

4 Gradient flow **11**
4.1 Introduction 11
4.2 Method 11
4.3 Results 13

5 Reference list **14**

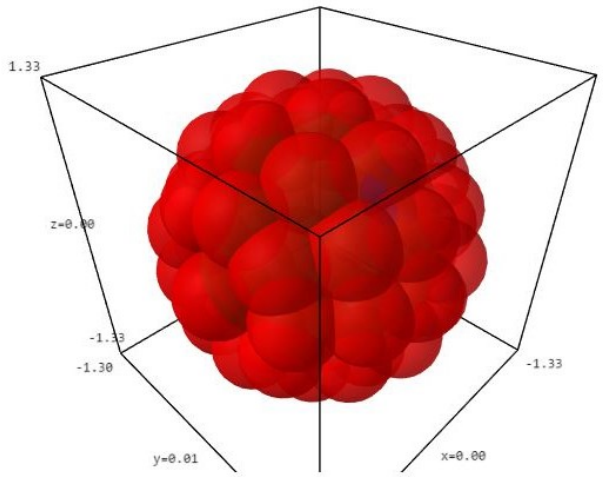
6 Appendix A **15**

1 Introduction

The covering problem is a classical type of problem that appears in not only geometry and applied mathematics, but also combinatorics and computer science. The problem is generally concerned with finding the smallest collection of a certain structure, so that the collection covers a larger structure. It has a dual problem called the packing problem, which is more well-studied and is concerned with packing structures together in the most efficient way without them overlapping.

The covering problem itself also has many different formulations and settings, and has inspired experimentation with methods based in varying fields of mathematics trying to yield optimal results [3] [4] [10]. In this thesis, we focus on a few intuitive examples that can be described by the following goal: **Given a connected set X on a manifold and a fixed $r > 0$, find the smallest set of points P such that $\bigcup_{p \in P} B_r(p)$ covers X.** Here we focus on the local covering problem; good algorithms found in the local setting have immediate implications on questions of global covering using expansion methods.

We first explore the problem by picking a few easier settings and constructing an algorithm that can explicitly produce coverings of a certain set. We want this algorithm to guarantee coverage, but the collections of points it outputs to be as small as possible. We then discuss a result in algebraic topology based on the construction of the *nerve* of a covering, which allows us to certify that a given collection of opens provides a covering of the set. This inherently interesting result allows us to finally improve on the results of our algorithm. We do this by removing some points from its output configurations and applying gradient flow to move them around such that they repel each other. We then apply the homological criterion to verify that the points give a covering after gradient flow, in which case we have improved our results.



2 Producing explicit configurations

2.1 Introduction

We started out by constructing an algorithm able to produce an explicit covering of some standard domains for any given covering radius r. We came up with the idea for this algorithm in the setting of the 2-sphere, one of the more intuitive examples. The resulting algorithm was then adapted to work in the case of a domain with a boundary (the unit disc in $\mathbb{R}^2$) and the case where the global vector space is not Euclidean (the unit disc in the hyperbolic plane $\mathbb{H}^2$). The ideas used in the algorithm could be applied to many different cases, and adapting it to work in for example a higher dimensional space would be very intuitive (conceptually at least; computational issues could arise in a high global dimension).

2.2 The 2-sphere

2.2.1 The algorithm

The algorithm starts with the spherical cap $S_0 = \{x \geq 1 - \frac{r^2}{2}, x^2 + y^2 + z^2 = 1\}$ cut off by a sphere of radius r centered at $(1, 0, 0)$, and enlarges this spherical cap at every step by adding points with a fixed x-coordinate. The number of new points and their location are chosen in such a way that they cover a "slab" (spherical segment) which glues to the spherical cap covered thus far to create a larger spherical cap. We then choose a new x-coordinate for the next step, and continue in this way until we have covered the whole sphere.

Let us describe a step of this procedure in detail. Suppose we have so far covered the spherical cap $S_i = \{x \geq e_i, x^2 + y^2 + z^2 = 1\}$ cut off by the plane $\{x = e_i\}$. We let C_i be the circle $\{x = e_i\} \cap S$, i.e. the boundary of S_i. We now want to choose an x-coordinate x_i for the next layer of points. We require $x_i < e_i$, but we also need to make sure that the spheres of radius r centred at points with x-coordinate x_i intersect C_i, so that we can glue the spheres to S_i and get a new spherical cap. We therefore calculate the minimum x-value $x_{i,lower}$ such that $B_r\left(x_{i,lower}, \sqrt{1 - x_{i,lower}^2}, 0\right) \cap C_i \neq \emptyset$. $\left(\text{This turns out to be } x_{i,lower} := \frac{2e_i - e_i r^2 - r\sqrt{(e_i^2-1)(r^2-4)}}{2}\right)$.

We now have to make some choice for x_i with $x_{i,lower} < x_i < x_{i,upper} (:= e_i)$. There are different ways to do so and we will see in later discussion that this choice strongly impacts the size of the resulting covering, but for now we will just set $x_i = \frac{x_{i,lower} + x_{i,upper}}{2}$.

The next step is to determine the amount of points on $\{x = x_i\} \cap S$ we need (evenly spaced), such that the spheres of radius r centred at these points cover the slab of S between $\{x = x_i\} \cap S$ and C_i. Explicitly, we find the smallest natural number n so that the n spheres of radius r with centres:

$$\left(x_i, \sqrt{1 - x_i^2}\cos(k\frac{2\pi}{n}), \sqrt{1 - x_i^2}\sin(k\frac{2\pi}{n})\right)$$

for $k = 0, \ldots, n-1$ cover the aforementioned slab of S. To do this, note that by construction each of these spheres cuts off an equally sized piece of C_i. If we take the corresponding angle

of this piece on the circle C_i to be θ, then taking $n = \left\lceil \frac{2\pi}{\theta} \right\rceil$ will satisfy our condition, since the piece of the "new" circle $\{x = x_i\} \cap S$ cut off by one of the spheres will have a larger corresponding angle θ'. This last claim will be proven explicitly on the next page.

Finally, we note that our new collection of spheres will cover a slab larger than the one between $\{x = x_i\} \cap S$ and C_i. To calculate the size of the slab we intersect S with two adjacent spheres in our new collection (e.g. $B_r(x_i, \sqrt{1-x_i^2}, 0)$ and $B_r(x_i, \sqrt{1-x_i^2}\cos(\frac{2\pi}{n}), \sqrt{1-x_i^2}\sin(\frac{2\pi}{n}))$ and take e_{i+1} to be the smaller of the x-values of the two intersection points. The new spheres combined with S_i then cover a spherical cap $S_{i+1} = \{x \geq e_{i+1}, x^2 + y^2 + z^2 = 1\}$.

If the resulting collection of spheres covers S, i.e. if $\left|(-1,0,0) - (x_i, \sqrt{1-x_i^2}, 0)\right| \leq r$, we terminate the algorithm. If adding one more sphere centered at $(-1,0,0)$ completes the covering, i.e. if $\left|(-1,0,0) - (e_{i+1}, \sqrt{1-e_{i+1}^2}, 0)\right| \leq r$, we add this last sphere and terminate the algorithm there. Else, we repeat the same procedure with our new spherical cap S_{i+1}.

The pseudocode can be found in appendix A. Here P is the collection of points and α defines our choice of x_i at every step; as mentioned before, we will take different values of α later.

2.2.2 First results

The following table lists the amount of spheres in the coverings this algorithm produces for a few values of r. We can compare these amounts to the volume bound $\frac{4\pi}{r^2\pi} = \frac{4}{r^2}$, which is the surface area of the sphere divided by the surface area of one spherical cap corresponding to a sphere of radius r. Therefore the volume bound is a (strict) lower bound on the amount of spheres a covering can contain, and a good point of reference to compare the quality of our results. We see that our algorithm consistently produces coverings of around 1.8 times the volume bound.

Radius	# of spheres in covering	Vol. bound
$\frac{1}{2}$	28	16
$\frac{1}{3}$	68	36
$\frac{1}{4}$	119	64
$\frac{1}{5}$	184	100
$\frac{1}{6}$	263	144

2.2.3 Proving coverage

The proof that this algorithm covers is based on the following claim:

Lemma 2.1. *Let $C = \{x = c, y^2 + z^2 = \sqrt{1-c^2}\}$ and $D = \{x = d, y^2 + z^2 = \sqrt{1-d^2}\}$ be two circles on the sphere of radius 1 with $-1 < d < c < 1$. Let S be a sphere of radius $0 < r < 1$ centered at a point on D, and suppose that this sphere intersects C at the points $a^{\pm}$ and the sphere D at the points $b^{\pm}$. Then the angle between b^+ and b^- (on D) is greater than the angle between a^+ and a^- (on C).*

Our algorithm satisfies these assumptions for every sphere at every step by construction. We then place the next sphere (again, centered at a point on D) in such a way that the a^+ of the

previous sphere aligns with the a^- of the new sphere. Using the claim, on D this new sphere will therefore leave no gap between the old b^+ and the new b^- either. Since we place the spheres exactly so that C is covered, the claim therefore implies that D is also covered. To see that the area in between C and D is covered, consider the parallelogram on the geometry of the sphere with vertices $a^\pm$ and $b^\pm$ and the fact that the smaller spheres are convex.

Proof. Without loss of generality, S is centered at the point $(d, 0, \sqrt{1-d^2})$.

$S : \{(x-d)^2 + y^2 + (z-\sqrt{1-d^2})^2 = r^2\}$.

We have for $(x, \pm y, z) = a^\pm = C \cap S$:

$$(c-d)^2 + (z-\sqrt{1-d^2})^2 - r^2 = z^2 - 1 + c^2 \implies z = \frac{1 - c - \frac{r^2}{2}}{\sqrt{1-d^2}} = \sqrt{1-c^2}\cos(\theta)$$

So the angle between a^+ and a^- is $2\arccos\left(\frac{1-cd-\frac{r^2}{2}}{\sqrt{1-c^2}\sqrt{1-d^2}}\right)$.

Similarly, the angle between b^+ and b^- is $2\arccos\left(\frac{1-d^2-\frac{r^2}{2}}{1-d^2}\right)$. Since arccos is a decreasing function, we want to show that $\frac{1-cd-\frac{r^2}{2}}{\sqrt{1-d^2}\sqrt{1-c^2}} > \frac{1-d^2-\frac{r^2}{2}}{1-d^2}$ under our assumptions ($0 < r < 1$ and $-1 < d < c < 1$). This inequality holds in general for $-1 < d < 1 - \frac{r^2}{2}$, which is always the case since the first e_i we consider is $e_0 = 1 - \frac{r^2}{2}$. □

2.2.4 Two improvements and results of final algorithm

Upon reviewing the first version of the algorithm for the 2-sphere we found two small possible improvements.

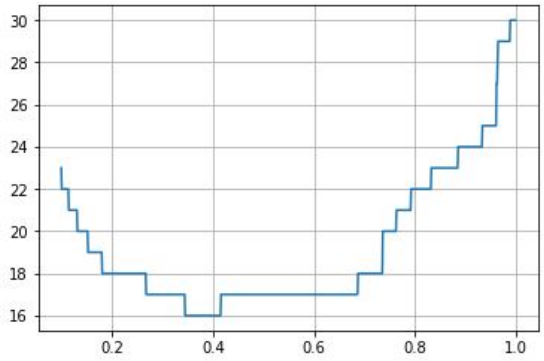
First, letting the amount of points at every step equal $n_i = \left\lceil \frac{2\pi}{\theta_i} \right\rceil$ (with θ_i as described before) means that the angle between $P_{i,0}$ and P_{i,n_i-1} will be smaller than the other pairwise angles; in other words, there is more overlap between the two spheres centered at these points than there is between the other spheres at the same step.

We can use this as follows: we first compute n_i as before and then lower x_i as far as we can while still requiring n_i regularly spaced spheres to cover the slab between $\{x = x_i\}$ and C_i. This is as easy as reverse engineering the procedure in the algorithm; after finding n_i, let $\Theta_i := \frac{2\pi}{n_i}$ and calculate the x-value x_i' such that the angle between the two points in $B_r(x_i', \sqrt{1-x_i'^2}, 0) \cap C_i$ on C_i is exactly Θ_i, then proceed with the algorithm as usual.

As a result we have the same amount of spheres added per step, but the total amount of spheres is lower since we move further down on the x-axis at every step. The result has been summarized in the following table, together with the improvement over the original algorithm and the volume bound.

Radius	# of spheres	Improvement	Vol. bound
$\frac{1}{2}$	27	1	16
$\frac{1}{3}$	66	2	36
$\frac{1}{4}$	115	4	64
$\frac{1}{5}$	174	10	100
$\frac{1}{6}$	252	11	144

The second improvement comes in the form of choosing α correctly. As mentioned before, at every step we take an x-value x_i for our next layer of points, which can be chosen anywhere between $x_{i,lower}$ and $x_{i,upper}$, which guarantee that the spheres centered at these points will intersect S_i, the spherical cap covered so far. We express this choice as the ratio $\alpha \in (0, 1)$, where we take $x_i = \alpha \cdot x_{i,upper} + (1 - \alpha)x_{i,lower}$ at every step.

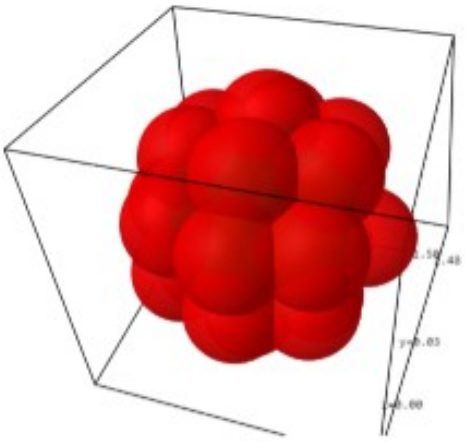


An example (for $r = \frac{1}{3}$) of a graph plotting α against the total number of spheres.

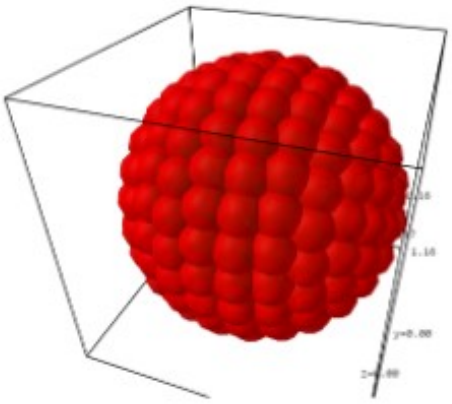
The way to think about this choice is that for larger values of α, we will need more steps in total since we advance less far along the x-axis, but we will need less spheres at every step since the new spheres lie closer to C_i and therefore less of them are needed to cover the slab between $\{x = x_i\} \cap S$ and C_i. Alternatively, smaller values of α will take less steps in total but more spheres per step (for $\alpha \to 0$, the amount of spheres needed per step goes to ∞ since each sphere will cover an extremely small part of C_i). For each value of r, the above graph is an example of possible values of α plotted against the total amount of spheres in the resulting covering. From these graphs we can read the optimal value of α; it is clear that different covering radii r require different values of α for optimal results, although the optimal value usually ends up in the interval $[0.2, 0.4]$.

Intuitively it looks like we could choose α differently at every step of the algorithm and get better results than for any fixed value of α throughout. After running some tests however, randomizing the value of α at every step and looking at the minimal covering out of $N > 10,000$ iterations, we found that this method struggles to perform as well as the previous best result for fixed α, and none of the randomized configurations was smaller than those found before. We were therefore satisfied with finding the best value of α for every r, and the following results are the best that this algorithm could produce on its own, around 1.6 times the volume bound.

Radius	# of spheres	Vol. bound
$\frac{1}{2}$	25	16
$\frac{1}{3}$	57	36
$\frac{1}{4}$	102	64
$\frac{1}{5}$	158	100
$\frac{1}{6}$	227	144



Result for $r = \frac{1}{2}$



Result for $r = \frac{1}{6}$

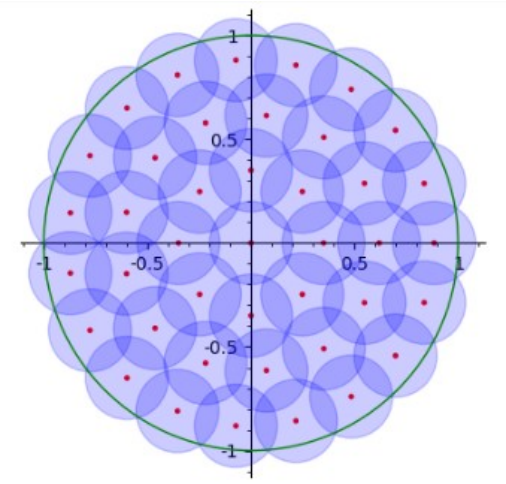
These coverings, while surprisingly small, are by no means optimal. In fact, R. H. Hardin, N. J. A. Sloane and W. D. Smith provide putatively optimal coverings [4] which outperform our results, and it is clear that our algorithm could never achieve these configurations due to its "slice-by-slice" construction, since the optimal coverings given by Sloane et. al. contain spheres that are spaced much more regularly. This is something we will try to overcome in the last chapter, after we study the necessary tools to improve on the results of our algorithm.

2.3 The unit disc in the Euclidean and the Hyperbolic plane

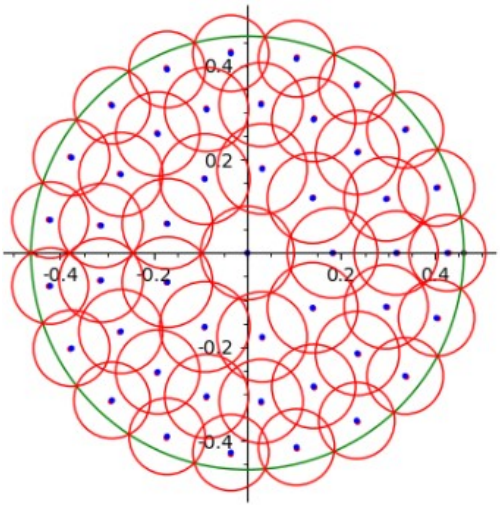
The algorithm constructed for the 2-sphere was fairly easy to adapt to the unit disc in $\mathbb{R}^2$. In this case, we started with the circle of radius r centred at the origin, and placed the centres of the next step of circles at a fixed radius r_0 from the origin, continuing the algorithm as usual. The optimizations made for the 2-sphere were also translated directly, and the resulting coverings performed about as well as the coverings for the 2-sphere did, at least below 2 times the volume bound.

Similarly, in the hyperbolic plane, after adjusting all the formulas to correspond to hyperbolic geometry the same algorithm worked perfectly fine, and in fact gave very similar results (comparing results to the volume bound in hyperbolic geometry). Computations were done in the hyperboloid model, although the results are presented in the Poincaré disc model for visualization purposes.

	$\mathbb{R}^2$		$\mathbb{H}^2$	
Radius	# of circles	Vol. bound	# of circles	Vol. bound
$\frac{1}{2}$	8	4	9	5
$\frac{1}{3}$	16	9	18	10
$\frac{1}{4}$	26	16	31	18
$\frac{1}{5}$	41	25	48	28



Result for $r = \frac{1}{5}$ in $\mathbb{R}^2$



Result for $r = \frac{1}{5}$ in $\mathbb{H}^2$

In the picture of the resulting configuration in $\mathbb{H}^2$, the circles lie in the Poincaré disc model and therefore have differing size depending on how far from the origin their center lies. Looking closely one can see a blue dot, which corresponds to the hyperbolic center, and a red dot, representing their "Euclidean" center corresponding to the circle in the Poincaré disc model. As mentioned before, the circles do all have the same size in hyperbolic geometry, something that is more intuitive but harder to visualize in the hyperboloid model where the computations were done.

3 Homological criterion for verifying coverage

3.1 Introduction

In the previous chapter we familiarized ourselves with the setting and possible approaches to the local covering problem, by constructing an explicit algorithm that creates coverings of certain domains. The algorithm is however heavily restricted by the fact that the output should guarantee coverage, and the actual minimal covering is unlikely to be achieved by such an explicit and direct method. One can imagine methods that would theoretically be able to produce a minimal covering (e.g. just randomly placing a fixed amount of points in the domain until we find a covering) but the problem lies in verifying (in an automated way) if a certain collection of opens actually covers. For this, we found an unexpected solution in the field of algebraic topology, inspired by a set of articles by Vin de Silva and Robert Ghrist [8][9]. Specifically we claim that for a d-dimensional manifold M the d-degree homology of a submanifold is non-zero if and only if it is equal to M. We then combine this fact with the so called nerve theorem, which states that a certain abstract simplicial complex constructed from a set of opens in a topological space has the same homology groups as the union of the opens. Therefore, checking if the homology of this complex is non-zero tells us if the opens cover the space.

3.2 Main lemma and the nerve theorem

Lemma 3.1. *Let M be a d-dimensional oriented closed manifold, $U \subseteq M$ an open submanifold and F a field. Then we have* $H_d(U, F) \simeq \begin{cases} 0 \text{ if } U \subsetneq M \\ F \text{ if } U = M \end{cases}$.

Proof. The case $U = M$ is immediate from the existence of a fundamental class, which corresponds to the generator of $H_d(M, F) \simeq F$.

Suppose that $U \neq M$, so that U is a non-compact submanifold. Since F is a field, Poincaré duality gives us $H_d(U, F) \simeq H^0_c(U, F)$, where we have to be careful to use cohomology with compact support in the statement of Poincaré duality since U is non-compact.

As a reminder, if we let $C^i_c(U, F)$ be the subgroup of $C^i(U, F)$ consisting of those cochains $\phi : C_i(U) \to F$ for which there exists a compact $K \subset U$ such that ϕ is zero on all chains in $U \setminus K$, then $H^0_c(U, F) = \ker(d_0 : C^0_c(U, F) \to C^1_c(U, F))$. However, for any 1-chain $[a, b]$ and $\phi \in C^0_c(U, F)$ we have $(d_0\phi)([a, b]) = \phi(b) - \phi(a)$. Therefore $\phi \in H^0_c(U, F)$ implies that ϕ is compactly supported and constant on points, and since U is non-compact the only such function is the zero function. We conclude that $H_d(U, F) \simeq H^0_c(U, F) \simeq 0$. □

Definition 3.1. *For a topological space X and a set of opens $\mathcal{U} = \{U_i\}_{i \in I}$ in X, the **nerve** of $\mathcal{U}$ is the set $\mathcal{N}(\mathcal{U})$ consisting of the subsets $J \subseteq I$ for which the intersection $\bigcap_{j \in J} U_j$ is non-empty. Since all subsets of a set $J \in \mathcal{N}(\mathcal{U})$ are also in $\mathcal{N}(\mathcal{U})$, the nerve is an abstract simplicial complex.*

Therefore we have a vertex for every open set U_i, and a face between vertices if the corresponding opens have non-empty intersection. The nerve theorem now gives us the desired

result and lets us compute the homology of a collection of opens only by its nerve.

Theorem 3.2 (Nerve theorem)**.** *Let X be a topological space and $\mathcal{U} = \{U_i\}_{i\in I}$ a collection of open sets in X such that for every $J \in \mathcal{N}(\mathcal{U})$, the (non-empty) intersection $\bigcap_{j\in J} U_j$ is contractible. Then $\mathcal{N}(\mathcal{U})$ is homotopy equivalent to $\bigcup_{i\in I} U_i$.*

An abstract proof can be found in Hatchers *Algebraic Topology* [5], but a more direct proof is given in an excellent paper by Kathryn Heal [6] where she gives the explicit homotopy equivalences and makes the theorem very intuitive.

It is easy to see why the condition of non-empty intersections being contractible is necessary. To give an example, let $X = S^1$ be the circle with two open arcs on S^1 that cover it. These opens would have to have non-empty non-contractible intersection, one contractible intersection on either end of the arcs. The nerve of these opens would then look like two connected vertices and would therefore be contractible, unlike their union S^1, so the nerve theorem would not hold. If we instead had three opens covering the circle, so that every intersection is contractible, we would get a nerve that reflects the homology of S^1 correctly.

3.3 Computation through persistent homology

It is common in for example topological data analysis to want to capture topological information about a set of points P in a metric space X [1] [8] [9]. In this context the term **Cech complex** is commonly used. By definition this is just the nerve of the collection of r-balls $\{B_r(p)\}_{p\in P}$ for some fixed $r > 0$, and the resulting simplicial complex is denoted $C_r(P)$ (so for $Q \subseteq P$ we have a simplex $Q \in C_r(P) \iff \bigcap_{q\in Q} B_r(q) \neq \emptyset$). By the nerve theorem, $\bigcup_{p\in P} B_r(p)$ is homotopy equivalent to $C_r(P)$ for any $r > 0$.

In our situation we would like to compute the Cech complex of a set of points and check if its top degree homology is non-zero, in which case we have a covering by the nerve theorem and lemma 3.1. Computing the Cech complex of a large number of points can however be challenging, since we have to check for every combination of balls whether they have non-empty intersection. We solve this by working instead with a similar but more manageable construction called the Rips complex (sometimes Vietoris-Rips complex).

Definition 3.2. *For a metric space X, a set of points P in X and $r > 0$, the **Rips complex** $R_r(P)$ of P is the abstract simplicial complex consisting of the subsets $Q \subseteq P$ for which $B_r(q) \cap B_r(q') \neq \emptyset$ for all $q, q' \in Q$.*

The Rips complex is a modification the Cech complex where one only considers pairwise intersection, and the computational advantages are clear since to compute it one only has to check if $d(p, p') \leq 2r$ for all pairs of points $p, p' \in P$. We now restrict ourselves to Euclidean space so that we can make the following chain of inclusions, relating the Rips complex to the Cech complex in a strong enough way to compute the homology of the latter using the former. Similar bounds are likely to exist in non-euclidean spaces, but for the remainder of the paper we will restrict ourselves to Euclidean space so that we can apply this theorem.

Theorem 3.3. *For a set of points P in $\mathbb{R}^d$ and any $r, r' > 0$ with $r' \leq r\sqrt{\frac{d+1}{2d}}$, we have the chain of inclusions $R_{r'}(P) \subseteq C_r(P) \subseteq R_r(P)$.*

Proof. The second inclusion is trivial since any simplex in $C_r(P)$ comes from a set of balls with non-empty intersection, whose pairwise intersection is then non-empty as well which means the simplex lies in $R_r(P)$. For the first inclusion it is enough to show that for any set of $n \leq d+1$ points with pairwise distance between them at most r', the balls of radius r centered at these points have non-empty intersection. The bound on r' was in fact chosen to satisfy this claim, which is shown in the article by V. de Silva and R. Ghrist [8]. The inclusion follows the following basic result known as Helly's theorem: if for any $m > d+1$ convex subsets $\{X_i\}_{i=1}^m$ in $\mathbb{R}^d$ the intersection of every $d+1$ of these sets is nonempty, then also $\bigcap_{i=1}^m X_i \neq \emptyset$. □

We now use this inclusion to state the desired criterion for verifying coverage.

Theorem 3.4. *For P a set of points in a domain $X \subseteq \mathbb{R}^d$, let $\mathcal{U}_r = \bigcup_{p \in P} B_r(p)$. Then, if for some $r' \leq r\sqrt{\frac{d+1}{2d}}$ the homomorphism $\iota_* : H_d(R_{r'}(P)) \to H_d(R_r(P))$ induced by the inclusion $\iota : R_{r'}(P) \hookrightarrow R_r(P)$ is non-zero, $\mathcal{U}_r$ contains X.*

Proof. Suppose we have an element $[\alpha] \in H_d(R_{r'})$ such that $\iota_*([\alpha]) \neq 0$. Then since ι_* factors through $j_* : H_d(R_{r'}) \to H_d(C_r)$, we also have $j_*([\alpha]) \neq 0$. This implies that $H_d(C_r) \neq 0$, and by the nerve theorem, $H_d(\mathcal{U}_r) \neq 0$. Now apply lemma 3.1. □

As mentioned before, the Rips complexes in the theorem are much easier to compute than the corresponding Cech complex, and in fact even computing if a map on the homology of Rips complexes is non-zero is fairly common in topological data analysis and is called *persisent homology*. We used the software Perseus [7] for this computation in the next chapter, where we apply the theorem to improve on our results for the covering problem.

4 Gradient flow

4.1 Introduction

Now that we have a rigorous way of certifying that a collection of points gives a covering, we have many new options of generating good configurations. We decided to try a method based on *gradient flow*, inspired by a paper by Simanta Gautam and Dmitry Vaintob [2]. This method gives us a way of moving points on the 2-sphere around in such a way that they repel each other. We use this by removing points from the configurations generated by our algorithm and moving the remaining points around until we find a new covering (verified by theorem 3.4), therefore immediately improving on our results.

4.2 Method

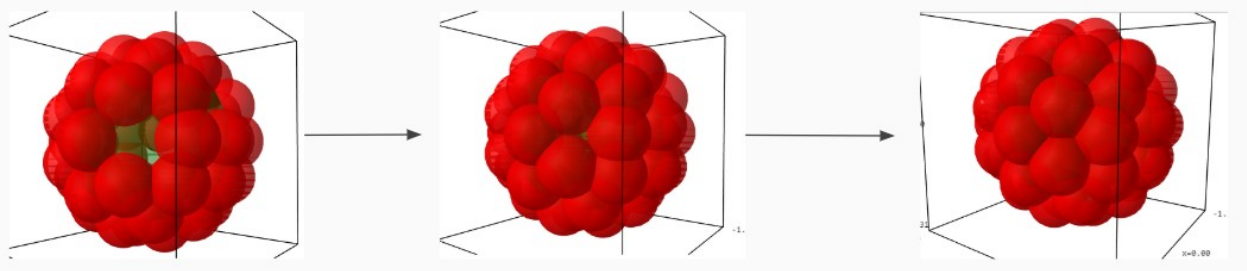
The method of gradient flow starts with a configuration of points on the 2-sphere and defines some global energy function on the set of points that takes a high value when points are close together. A simple example is the function

$$E((x_1, \ldots, x_n)) = \sum_{i<j}^{n} \frac{1}{|x_1 - x_n|}$$

which we took as the global energy function in our setting. We calculate the gradient on each vector, given by

$$\nabla E(x_i) := \nabla \sum_{j \neq i}^{n} \frac{1}{|x_i - x_j|} = \sum_{j \neq i}^{n} \frac{-(x_i - x_j)}{|x_i - x_j|^3}$$

(here the x_i are vectors) which gives a new vector that points towards a maximum of the energy function on x_i. We then move the points around by mapping, for ϵ small, $x_i \mapsto x_i - \epsilon \cdot \nabla E(x_i)$ and scale the result back to norm 1 so that it lies on the 2-sphere. This should move the points closer towards a local or global minimum of the energy function which corresponds to a well spaced configuration.



The gradient flow method: configuration for $r = 1/3$ after removing points, after 100 steps of gradient flow and after 200 steps.

One choice to consider is the value of ϵ. In the paper [2] a value of $\epsilon = 0.001$ is suggested, and the tradeoff of taking a higher value is clear: a higher value of ϵ will move the configuration

towards a minimum of the energy function faster, but taking ϵ too large can result in the points "skipping over" their optimal position, and can cause configurations that cover to be broken after more iterations. This is demonstrated in the following table. Here, we show the total energy of a configuration (taken from our algorithm for $r = \frac{1}{3}$ after removing two fixed points) after a number of steps for different values of ϵ. Here, E is the total energy minus 1200 and a star indicates that the corresponding configuration covers the 2-sphere, something we can verify using Perseus and theorem 3.4. We can see directly that a lower total energy is reached faster for higher values of ϵ, but taking it too high results in undesirable behaviour like coverings getting broken up after more iterations. These differences were even more apparant for smaller values of r; already for $r = \frac{1}{5}$, taking a value of $\epsilon > 0.01$ resulted in the overall energy going up instead of down. For our final results we have applied the gradient flow method for appropiate choices of ϵ per case.

# of iterations	150	175	200	250
$E, \epsilon = 0.05$			88.0260	87.9156*
$E, \epsilon = 0.1$		87.9748	87.9097*	*
$E, \epsilon = 0.2$	87.9807	87.9056*	87.8576*	
$E, \epsilon = 0.4$	87.9404	87.8743	87.8379*	
$E, \epsilon = 0.8$	87.9200*	87.8607	87.8295*	
$E, \epsilon = 1$	87.9160*	87.8582	87.8280*	

Another choice worth mentioning lies in the removal of points from the collection generated by our algorithm. It is clear that removing different points will result in different global energy values, and while in theory the gradient flow method should eventually converge to the same global minimum, picking the points that we remove intelligently will improve the speed of our method. After some testing, we concluded that removing points that are in some sense symmetrically located on the sphere gives the best results, as one would expect. In the following table, the gradient flow method was applied to a configuration with two random points removed, as well as the same configuration with two opposite points removed (namely the points with the highest and lowest z-coordinates). The configuration used was generated by our algorithm for $r = \frac{1}{3}$, and again we denote the total energy minus 1200 by E.

Opposite points						
k	0	10	20	30	40	50
E	93.1577	90.0420	89.8718	89.7812	89.7044	89.6240
Random points						
k	0	10	20	30	40	50
E	93.6620	91.7094	91.6555	91.5863	91.4669	91.2019

Therefore, for our final results we made sure to remove points that were symmetrically placed on the configuration we were trying to improve on. Usually this meant removing the points with the maximum and minimum coordinates on each of the three coordinate axes.

4.3 Results

For our final results, we started out with the optimal configuration our algorithm could provide, removed a reasonable number of points and applied the gradient flow method for up to 7000 steps. Afterwards, we used Perseus [7] and theorem 3.4 to verify that the resulting configuration provided a covering. If it did, we tried to remove more points, and if it didn't, we went back and tried to remove less points from the original covering. In this way we found optimal coverings for this method and in fact the size of the final configurations were under 1.5 times the volume bound, which is a very satisfying result.

Radius	Volume bound	# of spheres, algorithm	# of spheres after gradient flow
1/2	16	25	21
1/3	36	57	52
1/4	64	102	90
1/5	100	158	148

5 Reference list

[1] G. Carlsson, "Topology and Data" in *Bulletin of the American mathematical society*, vol. 46, no.2, pp. 255-308, 2009.

[2] S. Gautam and D. Vaintrob, "A Novel Approach to the Spherical Codes Problem", Massachusetts Institute of Technology, 2012.

[3] J. Hamkins and K. Zeger, "Asymptotically dense spherical codes" in *IEEE Transactions on information theory*, vol. 34, no. 6, pp. 1774-1798, 1997.

[4] R. H. Hardin, N. J. A. Sloane and W. D. Smith, "Spherical codes", book in preperation. http://neilsloane.com/coverings/index.html, 1994.

[5] A. Hatcher, "Algebraic Topology". *Cambridge University Press*, 2001.

[6] K. Heal, "Variations on the Nerve Theorem". kathrynheal-com.files.wordpress.com/2017/03/final.pdf, 2017.

[7] V. Nanda. Perseus, the Persistent Homology Software. http://www.sas.upenn.edu/ vnanda/perseus, accessed April 2020.

[8] V. de Silva and R. Ghrist, "Coverage in sensor networks via persistent homology," in *Algebraic & Geometric Topology*, 7, pp. 339-358, 2007.

[9] V. de Silva and R. Ghrist, "Coordinate-free coverage in sensor networks with controlled boundaries via homology" in *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1205–1222, 2006.

[10] T. Tarnai and Z. Gaspar, "Covering a sphere by equal circles, and the rigidity of its graph" in *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 110, no. 71, pp. 71-89, 1991.

6 Appendix A

The following is the pseudocode of our original algorithm, as described in section 2.2.1. The adjustments to the algorithm described in 2.2.4 are easy to implement by adjusting the value of α to be optimal and taking x_i' as described at every step instead of x_i after calculating n_i.

Initialization
$P = \{(1, 0, 0)\}$
$e_0 = 1 - \frac{r^2}{2}$
$cover = 0$
$\alpha = \frac{1}{2}$
$i = 0$
while *cover=0* **do**
  $x_{i,upper} = e_i$
  $x_{i,lower} = \frac{2e_i - e_i r^2 - r\sqrt{(e_i^2-1)(r^2-4)}}{2}$
  $x_i = \alpha \cdot x_{i,upper} + (1-\alpha) \cdot x_{i,lower}$
  if $\left|(-1,0,0) - (x_i, \sqrt{1-x_i^2}, 0)\right| \leq r$ **then**
    $cover = 1$
  Let θ_i be the angle between the two points in $B_r(x_i, \sqrt{1-x_i^2}, 0) \cap C_i$ on C_i
  $n_i = \left\lceil \frac{2\pi}{\theta_i} \right\rceil$
  $P_{i,k} = \left(x_i, \sqrt{1-x_i^2}\cos(k\frac{2\pi}{n_i}), \sqrt{1-x_i^2}\sin(k\frac{2\pi}{n_i})\right)$ for $k = 0, \ldots, n_i - 1$
  $P = P \cup \{P_{i,k}\}_{k=0}^{n_i-1}$
  Let e_{i+1} be the smaller x-value of the two points in $B_r(P_{i,0}) \cap B_r(P_{i,1}) \cap S$
  if $cover = 0$ *and* $\left|(-1,0,0) - (e_{i+1}, \sqrt{1-e_{i+1}^2}, 0)\right| \leq r$ **then**
    $P = P \cup \{(-1, 0, 0)\}$
    $cover = 1$
  i=i+1
