Obtuse Billiards

George William Tokarsky*and Boyan Marinov
University of Alberta

September 2, 2021

Abstract

A 250 year problem, the simplicity, the intricacies, the marvels. Does every triangle have a periodic path?

Keywords: Triangle, Billiard, Periodic

1 Introduction

It has been known since Fagnano in 1775 that an acute triangle always has a periodic billiard path, namely the orthic triangle. In 1993, Holt [9] showed that every right triangle has a periodic path using simple arguments. It is currently unknown whether every obtuse triangle has a periodic path, and there don't appear to be any simple arguments. In between, there have been various partial results. In 1986, Masur [1] proved that every rational polygon and thus every obtuse triangle has a periodic path, and in 2006, Schwartz [13] showed that every obtuse triangle with obtuse angle at most 100 degrees has a periodic path.

The aim of this paper is to show that every obtuse triangle with all angles at least 11 degrees have a periodic path with the exception of a flare at the point (15,30) of radius less than .001 degrees. The flare (15,30) is the first interior flare we found that is not on the diagonal, the boundary or on the right triangle line $x + y = 90$. Further we will show that every obtuse triangle with obtuse angle at most 112.4 degrees has a periodic path. We stopped at 112.4 as 112.5 appears to be a natural barrier. To go further would need a new infinite family of triangles. From our research we don't think that there is one infinite family that will cover a small semicircle around the boundary point (0,67.5). We believe that it will need many infinite families possibly even infinite many infinite different families.

From the work of Hooper and Schwartz [12] certain obtuse triangles on the diagonal don't have any stable periodic paths. These are triangles of the Veech form $(90/2^n, 90/2^n)$ where n=2,3,4... A complete proof of the 180 degree theorem will thus require using unstable periodic paths. We did however extend the 100 degree Schwartz proof to 105 degrees just using 134 stable codes as in Appendix B. And in section 16 we needed to use more and more unstable periodic paths from 105 to 112.4 degrees and then more in to the 11 degree theorem.

1.1 What's new, a super baker's dozen.

1. Using code sequences instead of side sequences. "Coding the code." See Section 2.2
2. The Alphabet Algorithm for legal codes. See Section 2.3
3. The Unfolding Poolshot Tower Test. See Section 7
4. The XYZ Algorithm for stable and unstable codes. See Section 9.1 and 10.1
5. Using "thin lines" which are not stable. See Section 10.1
6. We introduce the interior flare (15,30). See Section 10.1
7. Classifying all codes into five types. See Section 10.2
8. The Prover for efficiently proving that squares lie inside stable regions. See Section 13

*tokarsky@ualberta.ca, bmarinov@ualberta.ca

9. The Triple Rule Algorithm for proving squares that combine two abutting stable regions and an unstable line in between. See Section 13.1
10. The Half Triple Rule Algorithm for proving squares that combine a stable region and an abutting unstable line. See Section 13.2
11. The Corner Rule Algorithm for proving squares that involve a stable region to form a corner from two unstable lines. See Section 13.3
12. The Straight-Curved Corner Rule Algorithm for proving squares that involve a stable region to form a rational corner from a unstable line and a curved side. See Section 13.4
13. The Two Curved Corner Rule Algorithm for proving squares that involve a stable region to form a rational corner from two curved sides. See Section 13.5
14. The Star Rule Algorithm for proving squares that involve an union of stable regions with corners that surround a rational point. See Section 13.6
15. A new infinite pattern using unstable lines. See Section 14
16. Using bounding polygons to limit the size of regions. See Section 15
17. Proving every obtuse angle triangle between 100 and 112.4 degrees has a periodic path. See Section 16
18. Proving every obtuse angle triangle with angles at least 11 degrees has a periodic path except for one flare at (15,30). See Section 16

2 Side, Code and Alphabet Sequences

Drawing inspiration from the game of billiards, consider a frictionless particle moving about the interior of a triangle. If the particle encounters a side, it bounces off according to the law of reflection, and if the particle hits a vertex, it is considered to end there. A **billiard trajectory** or **poolshot** is the path this particle traces out as it moves within the triangle. A poolshot is said to be **periodic** if the particle eventually returns to its starting point and repeats the same path over again. Observe that a periodic poolshot must hit all three sides as it must exit any corner.

2.1 Side Sequences

To begin classifying billiard trajectories, we first need some notation. See for example Vorobets [14]. Given a triangle ABC with angles x, y, and z, we label the side AB opposite z with 1, the side BC opposite x with 2, and the side AC opposite y with 3. For any periodic billiard trajectory in the triangle, we then define its **side sequence** to be the list of consecutive sides that are hit during one period. For example, for the periodic billiard trajectory shown in Figure 1, if we begin at side 2 and continue to side 1, it has the side sequence 213132313 of **length** 9. We define a **legal side sequence** to be a sequence consisting of 1,2,3 such that consecutive integers are distinct including the first and the last.

Observe that no two consecutive positive integers from 1,2,3 are the same including the first and the last which we formally call a **legal side sequence**. We call a finite side sequence **repeating** if the last integer in it is followed by the first integer and then the integers keep repeating.

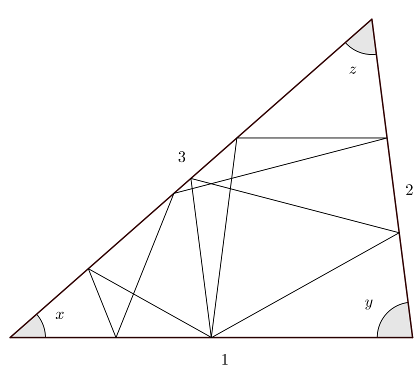


Figure 1: Periodic Billiard Trajectory 213132313 or equivalently 123132313

There are other equivalent side sequences for the trajectory in Figure 1 that we could consider. Given the periodic nature of the trajectory, the initial starting side and direction are both arbitrary, so we could instead start on side 3 and continue to side 1, giving the side sequence 313213132. These choices correspond to rotations and reversals of the original sequence, respectively. Likewise, we could continue around the triangle for as many periods as we wish, leading to the infinite family of side sequences 313213132313213132, 313213132313213132313213132, etc. All of these periodic sequences represent the same trajectory, so it is convenient to pick one among them as being canonical.

A periodic side sequence is in **standard form** if it has one period, and is lexicographically least among all its rotations and reversals. All side sequences can be converted to an equivalent one in standard form. For example, the standard form of the side sequence 213132313 is 123132313. In this way, every periodic billiard trajectory in triangle ABC can be uniquely identified by a side sequence in standard form.

It is said to be in **extra standard form** if we renumber the sides to make the side sequence minimal. For example the side sequence 3132 can be minimalized to 1213 by renumbering the sides. In this way any side sequence can be written in the form 12...23 or 12...13 by always starting with 12 and ending with 3.

Note that given a legal side sequence, there may not exist any triangle with a periodic billiard trajectory of that type. We call such side sequences **empty**. Observe also that any periodic path in a triangle must hit all three sides since it will eventually bounce out of any angle of the triangle and hence any periodic side sequence must include all three integers 1, 2 and 3.

2.2 Code Sequences

Long side sequences quickly become cumbersome to work with, so it is helpful to introduce a more compact notation. For each pair of adjacent integers in the side sequence (including the first and last), write out the angle between those corresponding sides. For example in the side sequence 123132313, this is $yzxxzzxxx$ where the first y is between the first 1 and 2 and right up to the last x coming between the last 3 and the first 1. Then, group the consecutive angles together, shuffling identical angles from the back to the front if necessary, and count the number of angles in each group. This gives $1y$ $1z$ $2x$ $2z$ $3x$. The sequence of integers 1 1 2 2 3 with spaces is called the **code sequence** consisting of the five individual positive integers each called a **code number**, and the sequence of angles y z x z x with spaces is called the **angle sequence**. The **length** of the code is the number of code numbers called an **odd code** or **even code** depending on its length and the **sum** of the code is the sum of the code numbers. It follows that the **sum** of the code equals the **length** of the corresponding side sequence.

Given a code and angle sequence, it is possible to recover the original side sequence if we start it with the first two integers 12... as in our example. Also observe that any code number by itself plus one yields the number of two successive alternate integers in the side sequence. For example the code number 2 corresponds to a side sequence of the form aba.

Note that specifying any two consecutive angles in the angle sequence will completely determine the rest of it as long as we know the code sequence. For example, for the sequence 1 1 2 2 3, consider the angles y z for the 1 and the 1. After bouncing once across angle y, the pool ball will bounce 1 time across angle z, and 1 being odd, will end at the opposite side from where it started. It will then bounce twice across angle x, and 2 being even, will end up at the same side as where it started, and then bounce back across the previous angle z. Continuing this pattern will produce the rest of the angles in the angle sequence and, crucially, wrap-around to match the last x with the first y we started with.

As with side sequences, a code sequence is in **standard form** if it is lexicographically least among its rotations and reversals. For example, the standard form of 1 1 3 2 2 is 1 1 2 2 3.

2.3 Notational conventions and Remarks

We will call a finite string of positive integers separated by spaces a **legal code sequence** if it is the **code sequence of a legal side sequence** and hence potentially represents a periodic path in a triangle.

Remark 1. We will only use the code sequence notation when dealing with a **repeating** side sequence corresponding to that code sequence.

Remark 2. Observe that if a side sequence starts 13 and ends in 2, then given its corresponding code sequence, we can recover the original side sequence by starting the sequence with 13... since successive integers are completely determined by the code numbers. If we choose to recover the side sequence by starting it with 12... then we will recover it with the 3's and 2's interchanged which means it will now end in a 3. This is not a problem as it just represents a relabeling of the sides of triangle ABC. Similarly we can start the side sequence 23 or any combination of 1,2 and 3 with the corresponding relabelling of the triangle. Caution: If the original side sequence is not in standard form and for example starts 13 and ends in 3, then we will recover some relabelling of the original side sequence from the code.

Remark 3. Given a code sequence representing a legal side sequence then we can always write in the form ab...bc or ab...ac where a,b and c are distinct.

We also make the convention that if there are three dots in front of (or following) a sequence of code numbers then this means there is at least one **code number** preceding (or following) that sequence in which case we will call it a **subcode**. For example ...2 4... is a subcode of the code sequence 2 2 4 4. Caution: A subcode need not be a legal code sequence by itself.

2.4 Alphabet Code Sequences

Given a code sequence, we can express it as as a set of evens and odds using the symbols E and O without spaces. For instance, the **alphabet code sequence** of 1 1 2 2 3 is OOEEO. A alphabet code sequence is said to be in **standard form** if it is alphabetically least among all its rotations and reversals. For example, the standard form of OOEEO is EEOOO.
From graph theory, it is now very easy to determine if a code sequence is legal or not as in the graph below.

The Alphabet Algorithm: An alphabet code sequence is legal if and only if it forms a closed **trail** (edges and vertices can be repeated and starting and ending at the same vertex) of E's and O's as in the directed graph.

Proof. The circles are vertices and each edge is labeled E or O. The only **cycles** (no repeated vertices except the first and the last) in the graph are those labeled EE, EOEO, OEOE, EOOEOO, OEOOEO, OOEOOE or OOO. These cycles all have corresponding legal side sequences of the form ab...c with a,b,c distinct. For example EE might correspond to the code sequence 2 4 which corresponds to the side sequence 121313. A closed trail must contain cycles and by successively removing the cycles, we can reduce the trail to the empty set. Now if we add the cycles back one by one, the trail would have a corresponding legal sequence. As an example suppose we add the cycle OOEOOE with code 1 1 2 1 1 2 and put it between EO (...1312...) to get EOOEOOEO (...131231321312...) which is still legal. On the other hand an open trail has no corresponding legal sequence since it can't be reduced to the empty set by removing all its cycles. We would end up with an open path (no repeated edges or vertices) and at most five edges. Say for example EOO which might correspond to the code sequence 2 1 1 but this doesn't correspond to any legal side sequence since it can't start 12 and end with 3 and have exactly length 4. Similarly for all cases. ☐

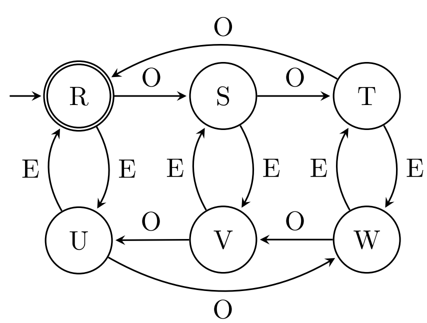
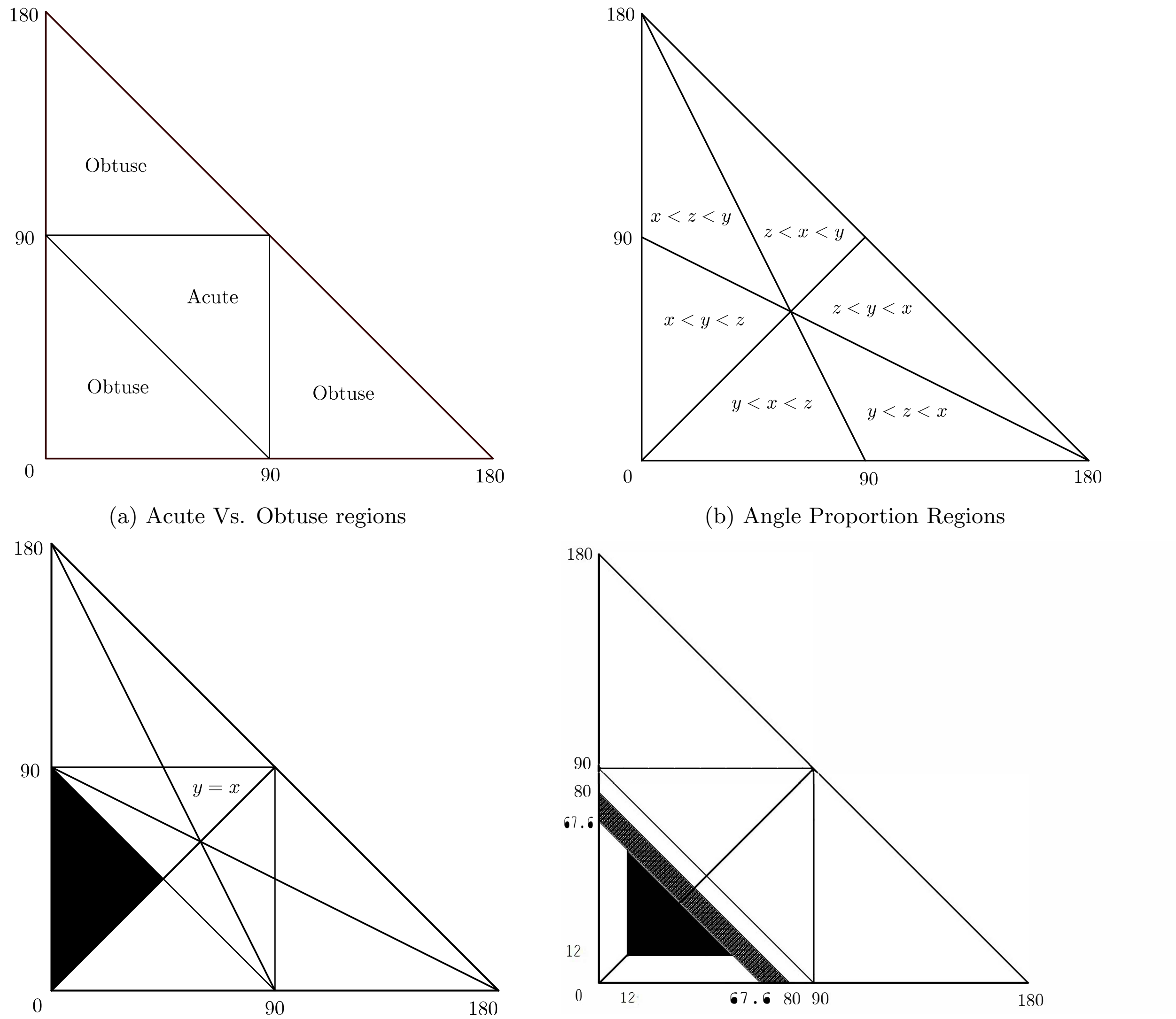


Figure 2: Graph for Alphabet Code Sequences

Observe from the graph that a code sequence always has an even number of even integers. It follows that the length and sum of a code sequence have the same parity.



(c) All Unique Obtuse Triangles

(d) The 12 and 112.4 Theorem

Figure 3: The Map of all Triangles

3 The Map of Triangles

Let $< A = x, < B = y$ and $< C = z$ using degrees and since $z = 180 - x - y$, the angles of the triangle are completely determined by the coordinates (x, y). We will then plot this point in an $X - Y$ coordinate system **only if that triangle has a periodic path**. Note this is independent of the size of the triangle. By symmetry to prove that all obtuse triangles have a periodic path amounts to plotting every point (x, y) with $x + y < 90$ and $x \leq y$ which means it is enough to fill in the shaded region of Figure 3c.

Our goal in this paper is to shade in the region bounded by $x = 0$, $y = 0$, $x + y = 80$, $x + y = 67.6$ and also by $x = 12$, $y = 12$, $x + y = 67.6$ as shown in Figure 3d. In conjunction with the Fagnano, Holt and Schwartz results we will prove that every triangle whose largest angle is at most 112.4 degrees or all angles are at least 11 degrees has a periodic path excepting one flare. We call this the 11 and 112.4 degree theorem.

4 The Unfolding Tower of Mirror Images of a Triangle

Let triangle ABC be oriented counterclockwise from A to B to C. A finite sequence of mirror images in the sides of triangle ABC will be called an unfolding tower or just a **tower** if successive mirror images are in different sides of the triangle. The **length** of a tower is the number of triangles in it. We will make the convention that side AB will form the **base** of the tower. It is a **parallel tower** if the last mirror image in side AC or side BC makes side AB parallel to the base and pointing in the same direction. AB is then the **top** of the tower.

If we take a successive subset of these mirror images then we will call it a **subtower** of the given tower which is a tower in its own right allowing its base to be any one of the three sides.

If we number the triangles consecutively the first or starting triangle will always be oriented counterclockwise and so will every odd numbered triangle while every even numbered triangle will be oriented clockwise. Successive "A" points are labelled A_0 to A_n, successive "B" points B_0 to B_m and successive "C" points C_0 to C_p. This means the orginal triangle is labelled $A_0B_0C_0$ and each vertex in the tower has an unique label.

Also note that a tower can overlap itself and that there may or may not be any poolshot associated with it as discussed in the next section. Further the A,B,C points are in fact ordered in an increasing order that follows the sequence of mirror images of the tower. So for example, $i < j$ if and only if C_i was formed before C_j. Caution: If the tower overlaps itself, some vertices could have multiple labels which is not a problem as they would belong to different triangles.

We will color the vertices of the tower according to the following rule. A_0 is a **blue point** and B_0 is a **black point**. If the first reflection is in side AC then C is a black point and if the first reflection is in side BC then C is a blue point. Inductively if vertex U has color blue *or* black and the next reflection is in side UV, then V has the opposite color.

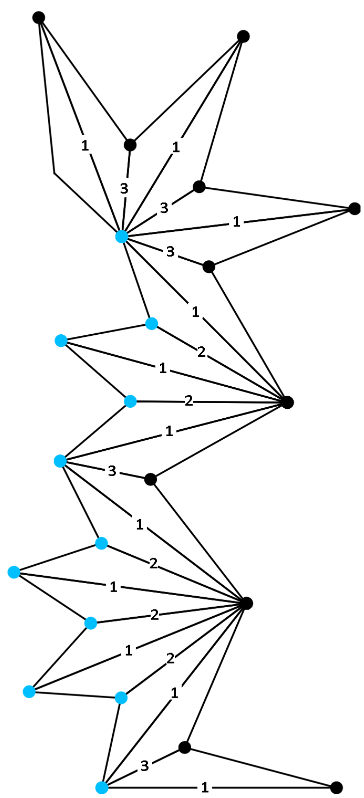


Figure 4: Unfolding Tower of Mirror Images

We can also use the side sequence notation to describe a tower where the first integer say 1 represents its base and successive integers represent the successive sides in which the mirror images are taken. For example the tower in figure 4 can be described by the **non-legal side sequence** 131212121312121313131

Remark: If a side sequence is symmetric about some integer say the integer "i" corresponding to side UV, then the corresponding subtower is symmetric about that same side UV. For example 312123**1**321213 is symmetric about the bolded 1. As a consequence, the line joining corresponding vertices of symmetric sides will be perpendicular to UV.

5 Unfolding Poolshot Towers

Given triangle ABC oriented counterclockwise from A to B to C and given a finite billiard trajectory or poolshot starting say at side AB and which doesn't hit a vertex, if we **straighten** out this poolshot upwards by successive reflections in the sides that the poolshot hits then we get a corresponding finite **tower** of mirror images in which the poolshot is now a straight line segment. See[2]which straightens the poolshot or [4] which unfolds the poolshot. All vertices occuring on one side of the straightened poolshot will be blue points while all vertices occuring on the other side will be black points.

It follows that the convex hull of the blue points and the convex hull of the black points are disjoint. This further means we cannot have a $blue - black - blue$ collinear situation where a black point is between two blue points. Similarly we cannot have a $black - blue - black$ collinear situation.

Since the poolshot starts at AB, if we straighten it out upwards with the base AB placed horizontal with A to the left of B and C above the base which we call **standard position**, then all blue points are on the $left\ side$ and all black points are on the $right\ side$ of the straightened poolshot and we get a **poolshot tower**. It is worth noting that a parallel poolshot tower must have an even number of triangles in it which is not necessarily the case for a parallel tower.

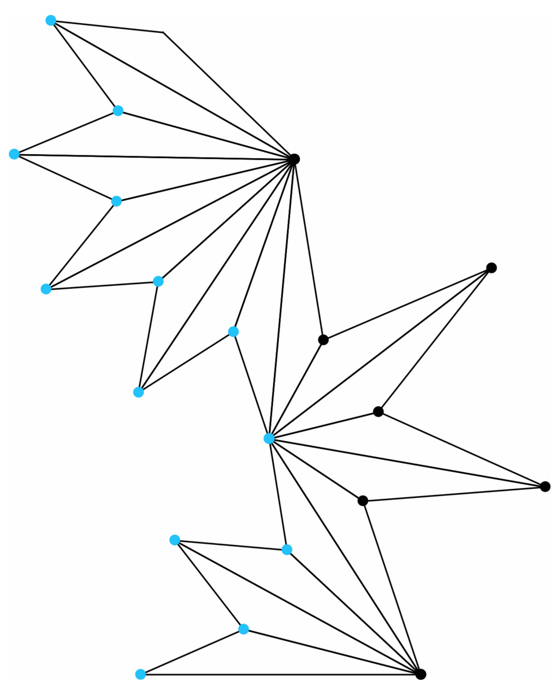


Figure 5: Unfolding Poolshot Tower

Convention: Any straightened poolshot can be viewed as forming the positive Y coordinate axis in an XY coordinate system by introducing a perpendicular X axis through the starting point of the poolshot on side AB. With this convention all blue points will be truly on the left side and all black points will be truly on the right side of the poolshot in this coordinate system.

6 Unfolding Periodic Poolshot Towers

If the poolshot forms a periodic path, then the corresponding poolshot tower is called a **periodic poolshot tower**. If it starts at side A_0B_0 and is periodic of even length, then it finishes at A_nB_m for some n and m where A_nB_m is **parallel** to A_0B_0 where A_0 and A_n are blue points and B_0 and B_m are black points and we have a parallel poolshot tower. If the periodic poolshot is of odd length and finishes at A_nB_m, then A_nB_m is **antiparallel** (in the sense that interior angles on the same side of the straightened poolshot are equal) to A_0B_0 with A_0 a blue point and A_n a black point and it follows that if we double the length of the poolshot and go around the periodic path twice then $A_{2n}B_{2m}$ will be parallel to A_0B_0 and both A_0 and A_{2n} will be on the same side of the straightened poolshot and both will be blue points and again we end with a parallel poolshot tower.

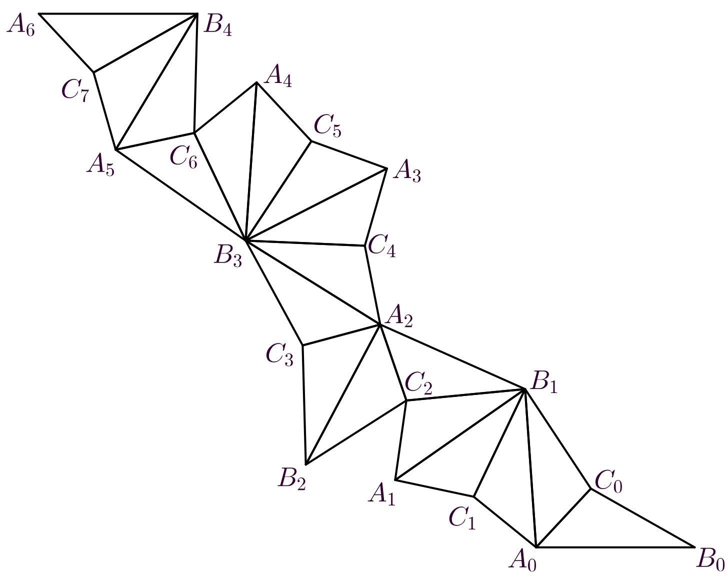


Figure 6: Unfolding Periodic Poolshot Tower

Remark: In a periodic poolshot tower, the side which is at the top of the tower is completely determined and is the same as the base.

7 The Unfolding Poolshot Tower Test

As previously noted given a poolshot tower the convex hulls of the blue and black points must be disjoint. Conversely if the convex hulls of the blue and black points respectively of a tower are disjoint then by a well known separation theorem there is a line separating the two sets and since A is a blue point and B a black point, that line must go through the base AB of the tower (and also through every segment joining a blue point to a black point) and hence there must be a straightened poolshot which produces the tower.

The Poolshot Tower Test: A tower is a poolshot tower if and only if the convex hulls of the blue and black points don't intersect.

Note: This test appears to be in many previous papers but none is quite as explicitly as above. Compare for example [14], [8] or [6]

8 The Unfolding Periodic Poolshot Tower Test

If we are given a periodic poolshot tower of **even length** then as stated previously the base A_0B_0 and the top A_nB_m of the tower are parallel line segments. The periodic poolshot that produces the tower will leave some point $P = P_0$ inside the base at an angle θ where $0 < \theta \leq 90$ and return to that point $P = P_q$ inside the top also at the angle θ but on the other side of the straightened poolshot. Since the base and top are parallel, this is the same acute angle between the line A_0A_n and the base A_0B_0 or between the line B_0B_m and the base A_0B_0. Indeed A_0A_n and B_0B_m are both parallel to the straightened poolshot P_0P_q. We will now simply say that the **poolshot vector** is any vector which has its head and tail on corresponding points of the base and the top.

Now observe that any line segment between a blue point and a black point must cross the line through P_0 and P_q. Vectorially if $v = (a, b)$ is a non-zero vector from any blue point to any black point and $w = (c, d)$ is a poolshot vector, then its determinant ad-bc is positive or equivalently $bc < ad$. In particular, any unit poolshot vector w=(-$cos\theta$, $sin\theta$) will work where θ is the clockwise angle between w and the horizontal.

Conversely if we are given a parallel tower of even length in which A_0B_0 is parallel to A_nB_m and where $bc < ad$ for every vector from a blue point to a black point then it must be a periodic poolshot tower. Since if U is a blue point farthest to the right (reminding the reader that we are orienting the coordinate system so that A_0A_n is vertical) and V is a black point farthest to the left and since $bc < ad$, there must be a band of non zero width between the two points between which there is a periodic poolshot which produces the given tower. Hence we get the

Periodic Poolshot Tower Test I: A parallel tower of even length with base A_0B_0 parallel to A_nB_m is a periodic poolshot tower if and only if $bc < ad$ for all vectors v where $v = (a, b)$ is a non-zero vector from any blue point to any black point and $w = (c, d)$ is a poolshot vector.

Now recall our discussion from section 2.2. As with repeating side sequences we can also use a code sequence to describe a repeating tower of mirror images of triangle ABC be it a periodic poolshot tower or not. For example the periodic poolshot tower in figure 6 can be described by the side sequence 131212313121212312 (the sequence of reflected sides) or the code sequence 2 3 1 3 5 1 1 2 (the sequence of groups of angles 2x 3y 1z 3x 5y 1z 1x 2y). By convention the first triangle in a tower is oriented counterclockwise. This means every odd numbered triangle is counterclockwise and every even numbered triangle is clockwise.

Note that corresponding to any subcode, there is a corresponding sequence of mirror images of triangle ABC which is a subtower.

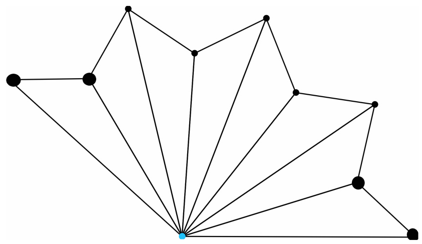
9 Fans

A **fan** is a tower of mirror images of a triangle in which all successive mirror images alternate between the same two sides which means that all triangles of the fan intersect at the same vertex which we will call its **center**. We can further classify the fans as **blue fans** if the center is black and all other vertices are blue points or **black fans** if its center is blue and all other vertices are black. The **central angle** of a blue or black fan is the maximal angle at its center produced by the blue or black vertices in the tower.

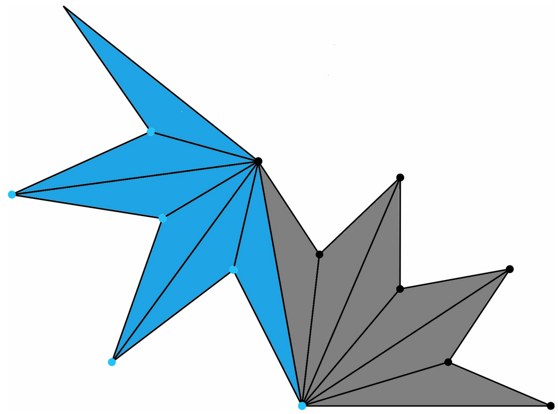
Any tower can be viewed as a succession of fans and in particular any poolshot tower can be viewed as a succession of alternating black and blue fans as cut off by the straightened poolshot. The blue points of a blue fan lie on one or two circular arcs whose endpoints are vertices of the fan. The endpoints of those arcs are called the **key points**. Each fan has either 2, 3 or 4 key points. In case if one of the arcs is a single point, there are 3 key points. Otherwise there are 2 or 4. Similar for the black points.

Fan Fact I: The center of a fan is a key point of the following and preceding fan of opposite color assuming it has a following or preceding fan.

Fan Fact II: Given a blue fan in a poolshot tower, then the key points of each blue arc are the points on the arc closest to the straightened poolshot. Similarly for black fans. This is a consequence of the fact that $sin\theta$ is a minimum at the endpoints of the interval [a,b] where $0 < a \leq b < 180$ or equivalently that $cos\theta$ is a minimum at the endpoints of the interval [a,b] where $-90 < a \leq b < 90$ and that in a poolshot tower the central angle of any fan is less than 180 degrees.



(a) Fan with its 4 Key Points Enlarged



(b) A Black and Blue Fan

Figure 7: Fan

Fan Fact III: Every blue(black) vertex lies on some blue(black) arc whose endpoints are key blue(black) points and whose center is black(blue).

Labelling of the Centers and Key points in a periodic poolshot tower

We will assume that the base is A_0B_0 and the first reflection is in side A_0C_0. Let the black point B_0 have the label $L_{(1,0)}$ and the blue point A_0 the label $L_{(2,0)}$. Now as the centers alternate between black and blue points the labels increase by one. Observe that all black centers have odd labels $L_{(2i-1,0)}$ and all blue centers have even labels $L_{(2i,0)}$ for $i \geq 1$. If the tower has 2m fans in it, then the last labels are $L_{(2m+2,0)}$ and $L_{(2m+1,0)}$ which belong to the last A and B vertices in the tower respectively.

For any fan and its center $L_{(k,0)}$, two of its key points are the previous center $L_{(k-1,0)}$ and the next center $L_{(k+1,0)}$. If there are more vertices in between, the next vertex is labelled $L_{(k,1)}$ and if it has another the last vertex is labelled $L_{(k,2)}$ so all key points are labelled. See Figure 8.

Note 1: The number of fans in a periodic poolshot tower in standard form equals the number of code numbers which is always even where we remind the reader that periodic paths of odd side sequence length are doubled in order to get a parallel tower.

Note 2: The first B and the last A vertices can be considered as centers of degenerate fans involving no triangles and are key points and are not counted in the number of fans.

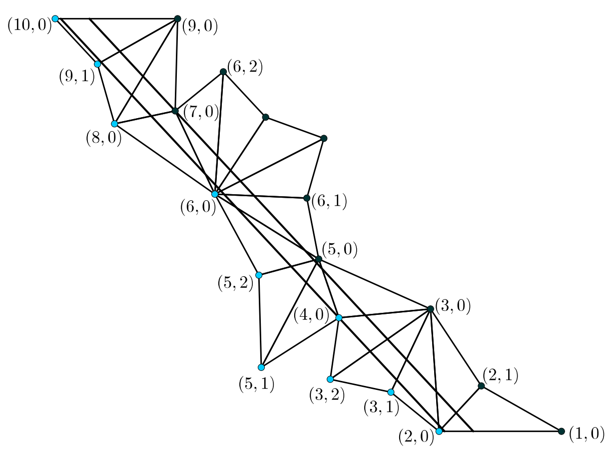


Figure 8: Labelled Periodic Poolshot Tower

Periodic Poolshot Tower Test II: A parallel tower of even length with base A_0B_0 parallel to A_nB_m is a periodic poolshot tower if and only if we apply Test I just to the key points **provided the central angle of any blue or black fan is less than 180 degrees**.

Proof. Choose our coordinate system so that A_0A_n is vertical. Let V be a black point with the smallest X coordinate in our system and let L_2 be a vertical line through V and let U be a blue point with the largest X coordinate and let L_1 be a vertical line through U. Then our tower is a periodic poolshot tower if and only if U lies to the strict left of V.

If the tower is a periodic poolshot tower then by the Test I $ad - bc > 0$ for all vectors from blue points to black points and hence this is certainly true for all vectors from key blue points to key black points.

On the other hand to show the other direction, it is enough to show that U and V are both key points since then the condition that $ad - bc > 0$ will guarantee that U lies to the strict left of V. Now since V is black, it must lie on some black arc whose center is blue. If that blue center lies to the left of or on L_2, then V must be a key black point otherwise since the central angle is less than 180 degrees one of the black endpoints of the black arc through V would lie to the left of V. If that blue center lies to the right of L_2, then since that center is a key blue point and the endpoints of the black arc through V are key black points, the condition $ad - bc > 0$ would force the central angle of that black arc to be greater than 180 degrees which is impossible. Hence this case can't arise and V is a key black point. Similarly U is a key blue point. □

Note 1: We can disregard the last blue and black points from this calculation since if $ad - bc > 0$ using a vector from the first blue point A_0 to some black point (not the last), then it follows that $ad - bc > 0$ using a vector from the last blue point A_n to that same black point since both vectors are on the same side of A_0A_n. Also we can disregard using a vector from A_n to B_m, since this works if and only if the vector from A_0 to B_0 works since these are the same vectors.

Note 2: If two blue points form a vector parallel to the poolshot vector then using either blue point and any fixed black point produces the same sign for $ad - bc$. This means we need only choose one blue point and disregard the others if they form a vector parallel to the poolshot vector.

Note 3: If two or more blue points determine a vector parallel to the poolshot vector and two or more black points determine a vector parallel to the poolshot vector, then we need only choose one blue point and one black point to find the sign of $ad - bc$.

Conclusion: In our computer calculations to show that a periodic path exists in a given triangle we use this second test taking into account the notes above.

Remark: Alternating succesive code numbers can be taken to represent alternating blue and black centers of alternating black and blue fans and we can call them **blue or black code numbers**. Observe that blue(black) code number gives one less than the number of black(blue) vertices in the corresponding fan and that the sum of the blue(black) codes plus one is the number of black(blue) vertices in the tower. Alternately each code number gives you the number of triangles in each fan and the sum of the code numbers gives you the number of triangles in the corresponding tower.

9.1 The XYZ Algorithm Rules

With successive code numbers in any code sequence, we can associate the X,Y or Z angles used in the central angles of the fans of the corresponding tower. Note that we use X and x, Y and y and Z and z represent the same angles. Notationally we like to use X,Y,Z when dealing with the alternating angles of the fans in a tower and x,y,z when dealing with the angles of an individual triangle.

Rule 1. Let the first code number correspond to X and the second code number correspond to Y. (Note X and Y can be replaced by any of X,Y or Z.)

Rule 2. Now consider any two successive code numbers C_i and C_{i+1} that have angles say X and Y assigned to them. Then if C_{i+1} is even, then C_{i+2} must have the same angle as C_i (X in this example) whereas if C_{i+1} is odd then C_{i+2} must have the angle different from that of C_i or C_{i+1} (Z in this example).

Rule 3. We will write these angles successively above and below the corresponding code numbers starting with X say on top. We will call these the **top angles and the bottom angles** as in the example below.

Rule 4: Finally we multiply them by their corresponding code numbers and form their **sum** as in 1X+3Z+3Y=3Y+1X+3Z. If the sum on top equals the sum on botttom then the top of the tower is parallel to the base although not necessarily the same side as the base. Observe that the alternating central angles of the fans are 1X 3Y 3Z 1X 3Y 3Z in that order. We call this the **XYZ equation**.

Example
X Z Y
1 3 3 1 3 3
Y X Z

Given a periodic poolshot tower with corresponding code sequence n m ... then the successive acute angles as the poolshot crosses each side of a fan are called the **shooting angles.** If the first shooting angle from the base of the tower is θ where $0 < \theta < 90$ and the first fan it crosses involves the angle X and has central angle nX and if $n = 2k + 1$ then the successive shooting angles are
θ,
$\theta + x$,
$\theta + 2x$,
...
$\theta + kx$,
$180 - \theta - (k + 1)x$,
$180 - \theta - (k + 2)x$,
...
$180 - \theta - (2k + 1)x$.
If $n = 2k$, then the successive shooting angles are
θ,
$\theta + x$,
$\theta + 2x$,
...
$\theta + (k - 1)x$,
$\theta + kx//180 - \theta - kx$,
$180 - \theta - (k + 1)x$,
$180 - \theta - (k + 2)x$,
...
$180 - \theta - 2kx$
where the // indicates that either $\theta + kx$ or $180 - \theta - kx$ is the acute angle.

Note that the first shooting angle cannot be 90 since in any fan that contains the 90 degree shooting angle, the side with the 90 degree shooting angle is also right in the center of the fan. Also observe that all the angles are integer linear combinations of x, y, 180 and θ and that as the poolshot passes from one fan to the next all the shooting angles are completely determined. In particular, if θ can be expressed as an integer linear combination of x,y and 90 then so too can every shooting angle.

10 Classifying Codes

10.1 Stable and Unstable

A code sequence is **stable** if the XYZ equation is **identically** zero and **unstable** if the XYZ equation is **not identically** zero .

As in the previous example 1 3 3 1 3 3 is a stable code since 1X+3Z+3Y-(3Y+1X+3Z)=0X+0Y+0Z=0 Using a second example 1 2 1 2 is an unstable code since the XYZ equation 2X=2Y+2Z or X=Y+Z or X=90 or Y=90 or X+Y=90 is not identically zero.

Caution: There are stable and unstable codes for which there are no triangles (x, y) which have a periodic path corresponding to that code sequence.

Convention: Given a code sequence, we define the associated region to be the set (x, y) consisting of triangles which actually have a billiard periodic path realizing that sequence. We call the region **stable** if this region is open and **unstable** otherwise. The stable case corresponds to the situation where the XYZ equation is identically zero and the unstable case corresponds to the situation when it is not. In the unstable case, the region is an open line segment.

Remark: In the stable case there is a periodic path within the triangle whose straightened trajectory is a postive minimum distance from all vertices. This means we can always change the coordinates (x,y) by a small amount and have another different triangle with the same periodic path. The upshot of this is that a stable region corresponding to a stable code and covers a positive area.

We call a point (x, y) in the plane a **flare** if no stable region contains that point and needs an infinite family of regions to cover an open neighbourhood of it. Flares appear to be rare. We conjecture that the point (15,30) is an interior flare.

10.2 The Five Code Types

We will now classify the various codes into five types CS, CNS, OSO, ONS, OSNO according to their various properties. These include of being stable or not stable, of being even or odd length or of the billiard path hitting some side at ninety degrees (containing a 90 degree reflection) or not.

CS codes: These are stable, even codes which contain a 90 degree reflection. Compare with [14]

Properties:
1. The lowest such code is 1 1 1 1 2 1 1 1 1 2.
2. Their side sequence length is a multiple of 4 and code sequence length is even and the code sequence passes the stable test.
3. They are codes of the form E_1 C_1 C_2 ... C_k E_2 C_k ... C_2 C_1 or some cyclic permutation of the above where E_1 and E_2 are even code numbers. Note that this means there are two special parallel sides corresponding to E_1 and E_2 in the tower which are perpendicular to the poolshot and half the length of the tower apart. We will call these the two **special perpendiculars**. It follows that these are the only sides of triangles in the tower which are perpendicular to the poolshot.
4. The corresponding stable region covers a finite non zero area where the XYZ equation is identically zero and is an open set in the plane.
5. One can determine the **first** shooting angle of each fan by the following algorithm which we illustrate by the following example. Consider the lowest CS code and also see Figures 9 and 10.

```
X Z   Y   Z    X
1 1 1 1 2 1 1 1 1 2
  Y  X   X   Y   Z
```

Then the first shooting angle of each successive fan is assuming the first one is θ are

θ
$180 - \theta - 1X$
$\theta + 1X - 1Y$
$180 - \theta - 1X + 1Y - 1Z$
$\theta + 1X - 1Y + 1Z - 1X$
$180 - \theta - 1X + 1Y - 1Z + 1X - 2Y$
$\theta + 1X - 1Y + 1Z - 1X + 2Y - 1X$
$180 - \theta - 1X + 1Y - 1Z + 1X - 2Y + 1X - 1Z$

$\theta + 1X - 1Y + 1Z - 1X + 2Y - 1X + 1Z - 1Y$
$180 - \theta - 1X + 1Y - 1Z + 1X - 2Y + 1X - 1Z + 1Y - 1X$

Note that if we continued this pattern then the next shooting angle would be $\theta{+}1X{-}1Y{+}1Z{-}1X{+}2Y{-}1X{+}1Z{-}1Y{+}1X{-}2Z{=}\theta$ since $1X{-}1Y{+}1Z{-}1X{+}2Y{-}1X{+}1Z{-}1Y{+}1X{-}2Z =$ 0 since the code sequence is stable.

Remark: Further since there is a 90 angle associated with the first 2, the first shooting angle of that fan is $(180 - 2Y)/2$=90-Y and we can determine that θ=X+Y-90 and hence express all shooting angles in terms of X,Y and 90.

6. The first shooting angle of the first fan can be used to get the unit poolshot vector (c,d)=(-$cos\theta$, $sin\theta$) used in the periodic poolshot tower test using θ as above.

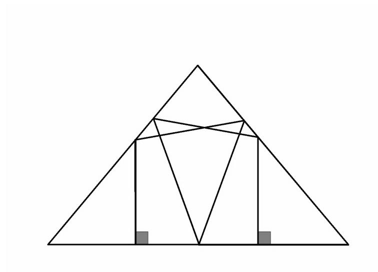


Figure 9: CS 1 1 1 1 2 1 1 1 1 2 Periodic Path

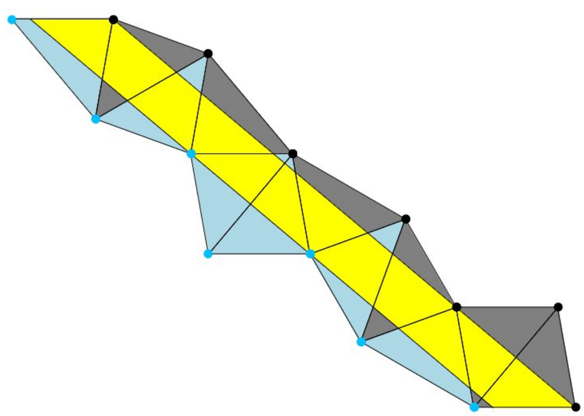


Figure 10: CS 1 1 1 1 2 1 1 1 1 2 Periodic Path with Fans and a yellow band of Poolshot Vectors

CNS codes: These are unstable, even codes which contain a 90 degree reflection.

Properties:
1. The lowest such code is 2 2.
2. Their side sequence length is a multiple of 2 and code sequence length is even and the code sequence doesn't pass the stable test.
3. As above they are of the form E_1 C_1 C_2 ... C_k E_2 C_k ... C_2 C_1 or some cyclic permutation of the above where E_1 and E_2 are even code numbers with the difference that they are not stable codes. The

corresponding tower would also contain two special perpendiculars.
4. The corresponding unstable line is a straight line segment where the XYZ equation is not identically zero. For the 2 2 CNS this becomes 2X-2Y=0 or Y=X.
5. One can determine the first shooting angle of each fan similar to the above which we illustrate by the following example. Consider the CNS code below and also see Figure 11.

Y Y
1 2 1 6
Z X

Then the first shooting angle of each successive fan is assuming the first one is θ are

θ
$180 - \theta - 1Y$
$\theta + 1Y - 2Z$
$180 - \theta - 1Y + 2Z - 1Y$

Note that if we continued this pattern then the next shooting angle would be $\theta + 1Y - 2Z + 1Y - 6X$=$\theta$ since $1Y - 2Z + 1Y - 6X$=0 or $Y = 90 + X$ is the equation associated with this code sequence.

Remark: Further since there is a 90 angle associated with the first 2, the first shooting angle of that fan is $(180 - 2Z)/2$=90-Z=X+Y-90 and we can determine that $\theta = 270 - X - 2Y = 90 - 3X$ and hence express all shooting angles in terms of X and 90.

6. The first shooting angle of the first fan can be used to get the unit poolshot vector (c,d)=(-$cos\theta$, $sin\theta$) used in the periodic poolshot tower test using θ as above.

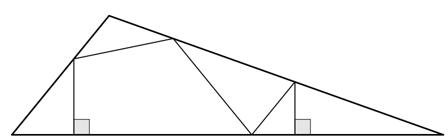


Figure 11: CNS 1 2 1 6 Periodic Path

Remark: There are faster ways to test a parallel tower for being a periodic poolshot tower if it is of the CS or CNS form and the XYZ equation is zero. This means the two special perpendiculars are parallel and perpendicular to the poolshot vector.

Periodic Poolshot Tower Test III for CS and CNS codes:
1. Let a parallel tower be of code sequence form E_1 C_1 C_2 ... C_k E_2 C_k ... C_2 C_1 or some cyclic permutation of the above where E_1 and E_2 are even code numbers and the XYZ equation is zero.
2. Let UV and WZ be the two special perpendiculars.
Then the tower is a periodic poolshot tower if and only if $ad - bc > 0$ where $v = (a, b)$ is a non-zero vector from any key blue point to any key black point which lie between or on the two special perpendiculars and $w = (c, d)$ is the poolshot vector **provided the angle of any blue or black fan is less than 180 degrees**.

OSO codes: These are stable code sequences whose sum is an odd length or some multiple of this and never contain a 90 degree reflection.

Properties:
1. The lowest such code is 1 1 1 which belongs to all acute triangles.
2. The side sequence length of an OSO with minimum period is odd and the code sequence length is also odd.

3. In order to create a parallel tower from an OSO, one has to double the code as in the example above the tower has to correspond to 1 1 1 1 1 1. This means OSO's as a parallel tower are of the form C_1 C_2 ... C_k C_1 C_2 ... C_k where C_1 + C_2 + ... + C_k is odd.
4. The corresponding stable region covers a finite non zero area where the XYZ equation is identically zero and is an open set in the plane.
5. One can determine the first shooting angle of each fan similar to the above which we illustrate by the following example. Consider the OSO code sequence below. Note we don't need to double its length in this calculation.

Z X X
1 1 2 2 5
Y Y

Then the first shooting angle of each successive fan is assuming the first one is θ are

θ
$180 - \theta - 1Z$
$\theta + 1Z - 1Y$
$180 - \theta - 1Z + 1Y - 2X$
$\theta + 1Z - 1Y + 2X - 2Y$

Note that if we continued this pattern then the next shooting angle would be
$180-\theta-1Z+1Y-2X+2Y-5X=\theta$ from which we can solve for θ to get $\theta=(180-1Z+1Y-2X+2Y-5X)/2 = 2Y - 3X$.

Remark: By induction, in an OSO legal code all shooting angles are integer linear combinations of X,Y and 180.

6. The first shooting angle of the first fan can be used to get the unit poolshot vector $(c, d) = (-cos\theta, \, sin\theta)$ used in the periodic poolshot tower test using θ as above.

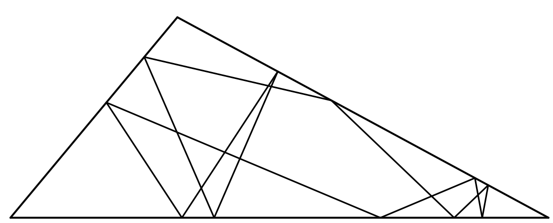


Figure 12: OSO 1 1 2 5 2 which is another different Periodic Path

ONS codes: These are the unstable even codes which don't contain a 90 degree reflection.

Properties:
1. The lowest such code is 1 1 1 1 3 3.
2. Their side sequence length is a multiple of 2 and code sequence length is even and the code sequence doesn't pass the stable test.
3. As above they are non stable codes of the form C_1 C_2 ... C_k which are not of the CNS form.
4. The corresponding unstable line is a straight line segment where the XYZ equation is not identically zero. For the 1 1 2 1 3 2 ONS this becomes 4X-2Y=0 or Y=2X.
5. One can determine the first shooting angle of each fan similar to the above which we illustrate by the following example. Consider the ONS code sequence below and also see Figure 13.

X X
2 2 4 4
Y Y

Then the first shooting angle of each successive fan is assuming the first one is θ are

θ
$180 - \theta - 2X$
$\theta + 2X - 2Y$
$180 - \theta - 2X + 2Y - 4X$

Note that if we continued this pattern then the next shooting angle would be $\theta + 2X - 2Y + 4X - 4Y$=$\theta$ and we **cannot** solve for θ since on the corresponding unstable line Y=X and then $2X - 2Y + 4X - 4Y$=0.

Remark: It can be shown that the shooting angles are not integer linear combinations of X,Y and 90. For example the ONS lowest code 1 1 1 1 3 3 has a periodic path in the (45,45,90) triangle and none of its shooting angles are 45 degrees.

6. Because of the above, the unit poolshot vector $w = (c, d)$=(-$cos\theta$, $sin\theta$) used in the periodic poolshot tower test is calculated another way using section 12 where θ is the angle between w and the horizontal.
7. There is a way to eliminate θ to form a bounding polygon which is discussed later in section 15 and which contains the straight line segment region corresponding to the ONS code.

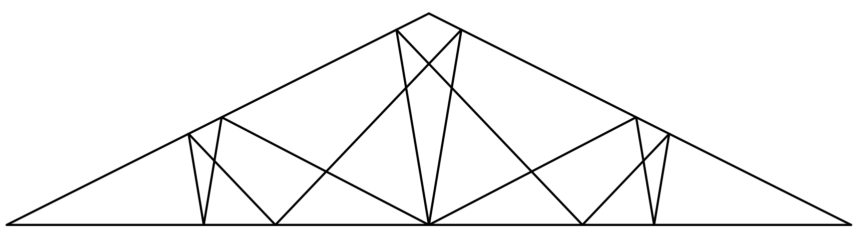


Figure 13: ONS 2 2 4 4 Periodic Path

OSNO codes: These are the stable, even codes which don't contain a 90 degree reflection.

Properties:
1. The lowest such code is 1 1 2 2 1 1 3 3.
2. Their side sequence length is a multiple of 2 and code sequence length is even and the code sequence passes the stable test.
3. They are codes of the form C_1 C_2 ... C_k which are not of the CS or even multiples of the OSO form.
4. The corresponding stable region covers a finite non zero area where the XYZ equation is identically zero and is an open set in the plane.
5. One can determine the first shooting angle of each fan similar to the above which we illustrate by the following example. Consider the OSNO code sequence

X Y Y Z
1 1 2 2 1 1 3 3
Z Z X Y

Then the first shooting angle of each successive fan is assuming the first one is θ are

θ
$180 - \theta - 1X$
$\theta + 1X - 1Z$
$180 - \theta - 1X + 1Z - 2Y$
$\theta + 1X - 1Z + 2Y - 2Z$
$180 - \theta - 1X + 1Z - 2Y + 2Z - 1Y$
$\theta + 1X - 1Z + 2Y - 2Z + 1Y - 1X$
$180 - \theta - 1X + 1Z - 2Y + 2Z - 1Y + 1X - 3Z$

Note that if we continued this pattern then the next shooting angle would be $\theta + 1X - 1Z + 2Y - 2Z + 1Y - 1X + 3Z - 3Y$=$\theta$ since $1X - 1Z + 2Y - 2Z + 1Y - 1X + 3Z - 3Y = 0$ since the code sequence is stable.

Remark: It can be shown that the shooting angles are not integer linear combinations of X,Y and 90. For example the OSNO code 1 1 2 2 1 1 2 2 1 1 2 2 has a periodic path in the (60,60,60) triangle and none of its shooting angles are 30 or 60 degrees.

6. Because of this, the unit poolshot vector $w = (c, d) = (-cos\theta,\ sin\theta)$ used in the periodic poolshot tower test is calculated using section 12 where θ is the angle between w and the horizontal.
7. There is a way to eliminate θ to form a bounding polygon which is discussed later in section 15 and which contains the open region corresponding to the OSNO code.

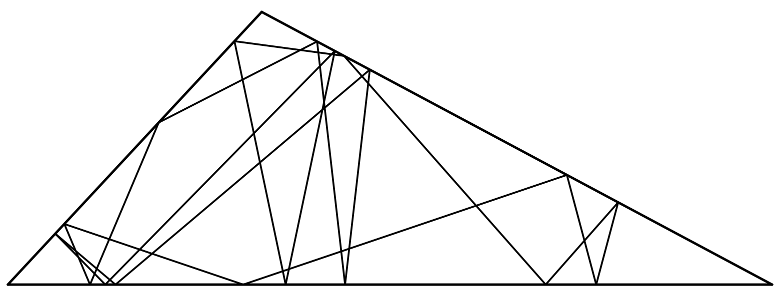


Figure 14: OSNO 1 1 2 2 3 1 3 5 which is another different Periodic Path

11 Previous results on periodic paths:

1. Acute triangles always have a periodic path of OSO code type 1 1 1 which has length 3 namely the orthic triangle whose vertices are the feet of the altitudes. See Figure 15.

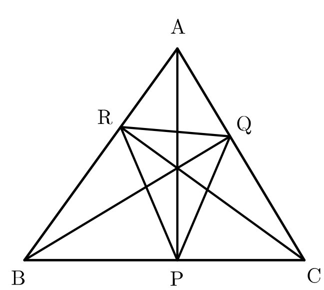


Figure 15: Orthic Triangle

2. Right triangles always have a periodic path of CNS code type 1 2 1 2. See Figure 16.

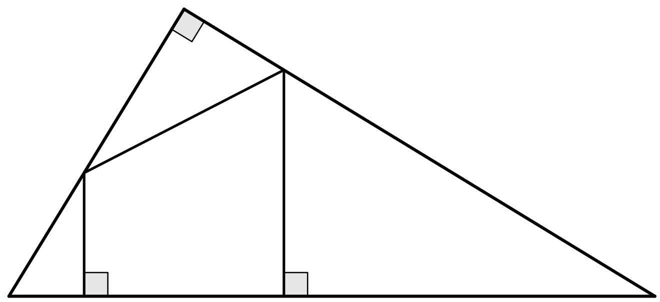


Figure 16: Right Triangle with CNS Periodic Path 1 2 1 2

3. Isosceles triangles always have a periodic path of CNS code type 2 2. See Figure 17.

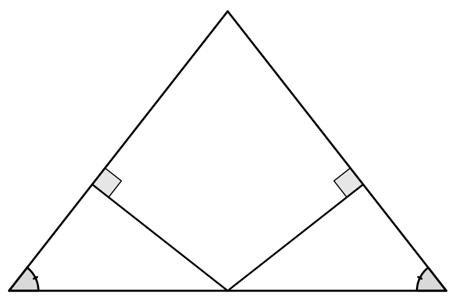
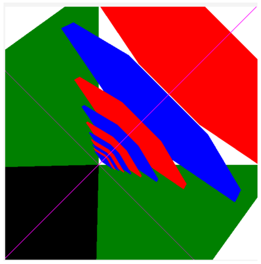


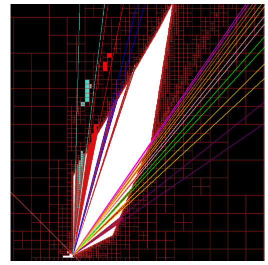
Figure 17: Isosceles Triangle with CNS Periodic Path 2 2

4. Rational triangles always have a periodic path of type CNS. See [1], [3], [4]. A triangle is **rational** if all angles can be expressed as integer multiples of x where x divides 90, say $< A = nx$, $< B = mx$ and $< C = px$. We can prove this from the following facts about rational triangles.
Fact I: If a poolshot leaves a side at 90 degrees and $90 = qx$ for some integer q, then it bounces off any side at some integer multiple of x. Conclusion: There are at most q angles involved as a poolshot bounces off the sides of the triangle.
Fact II: If a poolshot leaves a side at 90 degrees and $90 = qx$ for some integer x and hits a vertex say vertex B, then it must enter B along a ray making an angle tx where $0 < t < m$.

Conclusion: There are at most n+m+p-3 perpendicular poolshots which hit a vertex in a rational triangle. This means if a 90 degree poolshot leaves a side and doesn't hit a vertex, there is an open band around that poolshot of finite width $\delta > 0$ which leaves that side at 90 and never hits a vertex and in which δ is maximal. This is only possible if the band hits another side at 90 and becomes periodic since otherwise the band must hit some side say side AB infinitely often at some angle jx for a fixed integer j less than q and which is repeated infintely often. But each time this band hits this side, it hits on some open interval (a_i, b_i) of width δ no two of which can intersect without contradicticting the maximality of δ. But since AB is finite this is impossible.
5. Every obtuse triangle with obtuse angle at most 100 degrees has a periodic path. See [13].
6. Infinite families of periodic paths. See [8], [13], [14].
7. All conjectured flares on the diagonal are of the form $(90/2^n, 90/2^n)$ for $n \geq 2$ and are covered. See [12]. This is the work of W. Patrick Hooper and Richard Evan Schwartz who proved these points are unstable and conjectured that no neighbourhood of these points have a finite cover. As an example the flare at (22.5,22.5) is completely covered using an infinite red and blue pattern. See Figure 18a.
Note: From our work, we found and conjecture that a new point (15,30) is an interior flare and has no finite cover. See Figure 18b. By calculation, this flare has a neighbourhood of radius size less than .001 degrees.



(a) (22.5,22.5) diagonal flare cover



(b) (15,30) interior flare

Figure 18: Two flares

12 Calculating the coordinates of the vertices in a code tower

Given a code tower of even length, let us assume that the ordering of the angles in the code tower is such that the first top angle is $U_1 = X$ and that the last bottom angle is $U_{2k} = Z$ as shown.

$$\begin{array}{llll} U_1 & & U_3 & \\ C_1 & C_2 & C_3 \ldots & C_{2k} \\ & U_2 & & U_{2k} \end{array}$$

Then the base of the tower is AB with $< A = x$, $< B = z$ and $< C = y$ and if we let $AB = siny$, $BC = sinx$ and $AC = sinz = sin(x + y)$ since $z = 180 - x - y$ we then can recursively calculate the coordinates of each fan center $L_{(i,0)}$ as follows: Letting $a_i = U_iC_i$ be the center angle of the fan with corresponding label $L_{(i+1,0)}$ for $i \geq 1$, u_i be the length of the side between (x_i, y_i) and (x_{i+1}, y_{i+1}) where (x_i, y_i) are the coordinates of the center of the fan at $L_{(i,0)}$ then recursively let $x_1 = siny$, $y_1 = 0$, $x_2 = 0$, $y_2 = 0$ and

$$x_{2n} = x_{2n-1} - u_{2n-1}cos(a_{2n-2} - a_{2n-3} + a_{2n-4} \ldots - a_3 + a_2 - a_1)$$
$$y_{2n} = y_{2n-1} + u_{2n-1}sin(a_{2n-2} - a_{2n-3} + a_{2n-4} \ldots - a_3 + a_2 - a_1)$$
$$x_{2n+1} = x_{2n} + u_{2n}cos(a_{2n-1} - a_{2n-2} + a_{2n-3} \ldots + a_3 - a_2 + a_1)$$
$$y_{2n+1} = y_{2n} + u_{2n}sin(a_{2n-1} - a_{2n-2} + a_{2n-3} \ldots + a_3 - a_2 + a_1)$$

Observe that a poolshot vector $w = (c, d)$ goes from $L_{(1,0)}$ to $L_{(2k+1,0)}$ or equivalently from $L_{(2,0)}$ to $L_{(2k+2,0)}$. See an ONS example in Appendix A.

13 The Prover

The basic idea behind the prover is the Mean Value Theorem in two dimensions from calculus.

Given a code sequence and its corresponding code region R, we would like to verify that the code sequence represents a periodic poolshot at every point (x,y) in R. It turns out that it is enough to show that the following functions of two variables are positive on a small enough square to be completely contained in R. Then we can combine squares and even different code sequences to create a cover as in Figure 23.

Due to the massive dataset, we made no attempt to minimalize the cover. As we found codes and corresponding squares, we put them into the prover and proved that they worked. We simply took a poolshot in a triangle and used the Unfolding Periodic Poolshot Tower Test I, II or III. If it passed we used that code sequence and corresponding small enough squares and put it into the prover. If it worked and we needed it to fill a hole, we added it to the cover.

The Mean Value Theorem: Let $f(x, y)$ be a differentiable function of two variables, then $f(b_1, b_2) - f(a_1, a_2) = f_x(c_1, c_2)(b_1 - a_1) + f_y(c_1, c_2)(b_2 - a_2)$ for some (c_1, c_2) on the line between (a_1, a_2) and (b_1, b_2).

Fact I: If $f(x,y) = \sum_{i=1}^{k} \pm u_i cos(m_i x + n_i y)$, then $f_x = \sum_{i=1}^{k} \mp m_i u_i \sin(m_i x + n_i y)$ and f_y=$\sum_{i=1}^{k} \mp n_i u_i \sin(m_i x + n_i y)$ and hence $|f_x| \leq \sum_{i=1}^{k} |m_i u_i| = M$ and $|f_y| \leq \sum_{i=1}^{k} |n_i u_i| = N$. Similarly if $f(x,y) = \sum_{i=1}^{k} \pm u_i sin(m_i x + n_i y)$.

Fact II: If $f(b_1, b_2) > 0$ and $f(a_1, a_2) \leq 0$, then by the Mean Value Theorem $f(b_1, b_2) - f(a_1, a_2) = f_x(c_1, c_2)(b_1 - a_1) + f_y(c_1, c_2)(b_2 - a_2) \geq f(b_1, b_2)$ and since $f_x(c_1, c_2)(b_1 - a_1) + f_y(c_1, c_2)(b_2 - a_2) \leq M|b_1 - a_1| + N|b_2 - a_2|$, then $f(b_1, b_2) \leq M|b_1 - a_1| + N|b_2 - a_2|$.

Conclusion: If $f(b_1, b_2) > M|b_1 - a_1| + N|b_2 - a_2|$, then $f(a_1, a_2) > 0$.

Fact III: Let (b_1, b_2) be the center of a square of side $2r > 0$, then for any (a_1, a_2) in or on the boundary of the square we must have $(b_1 - a_1) \leq r$ and $(b_2 - a_2) \leq r$ and hence $M|b_1 - a_1| + N|b_2 - a_2| \leq (M + N)r$.

Conclusion I: If $f(b_1, b_2) > (M + N)r \geq M|b_1 - a_1| + N|b_2 - a_2|$ then $f(a_1, a_2) > 0$.

Conclusion II: If $0 < r < f(b_1, b_2)/(M + N)$, and $f(b_1, b_2) > 0$, then all points (a_1, a_2) in the square centered at (b_1, b_2) and of side 2r must also satisfy $f(a_1, a_2) > 0$.

The Gradient Algorithm: For every function $f_j(x,y) = \sum_{i=1}^{k} \pm u_i cos(m_i x + n_i y)$ or $f_j(x,y) = \sum_{i=1}^{k} \pm u_i sin(m_i x + n_i y)$ and which form the boundary equations of a stable code region

1. calculate $G_j = \sum_{i=1}^{k} |u_i|(|m_i| + |n_i|)$
2. then if a square centered at (b_1, b_2) satisfies $f_j(b_1, b_2) > 0$ and $f_j(b_1, b_2) - rG_j > 0$ for all f_j where 2r is the length of a side of the square, then every point in or on the boundary of the square satisfies $f_j > 0$ and that square lies completely within the given code region.

Note: If we are dealing with a CNS or ONS code and its corresponding linear region, then it is exactly the same algorithm as long as (b_1, b_2) is the center of the square intersecting the linear region.

13.1 The Triple Rule Algorithm

Suppose two stable code regions R_1 and R_2 intersect along a common boundary unstable code line from (x_1, y_1) to (x_2, y_2).

1. This can happen if R_1 is defined by the system of equations $f_i(x,y) > 0$ and $f(x,y) > 0$ and R_2 is defined by the system of equations $g_i(x,y) > 0$ and $g(x,y) > 0$.
2. Let the equations $f(x,y) = 0$ and $g(x,y) = 0$ have a common factor of the form $sin(ax+by)$ or $cos(ax+by)$.
3. Now observe that $sin(ax + by) = 0$ if and only if $ax + by = 180k$ and $cos(ax + by) = 0$ if and only if $ax + by = 90 + 180k$ for some integer k.

Lets illustrate with the case with $sin(ax + by)$ as the common factor where $f(x,y) = sin(ax + by)u(x,y)$ and $g(x,y) = sin(ax + by)v(x,y)$
4. Let us further suppose that R_1 lies between the parallel lines $ax + by = 180(k - 1)$ and $ax + by = 180k$ and that R_2 lies between the parallel lines $ax + by = 180k$ and $ax + by = 180(k + 1)$
5. Further suppose that $sin(ax + by) > 0$ between the first two parallels and $sin(ax + by) < 0$ between the second two parallels.
Now consider a square with sides parallel to the coordinate axis whose vertex coordinates are all rational numbers and which lies between the first and third parallels and which may or may not intersect the middle parallel. It is worth noting that if this square lies inside either code region then it cannot intersect any of the three parallel lines above since $sin(ax + by)$ is zero there. We can use the following to decide if every point in the square including its boundary has a periodic path.

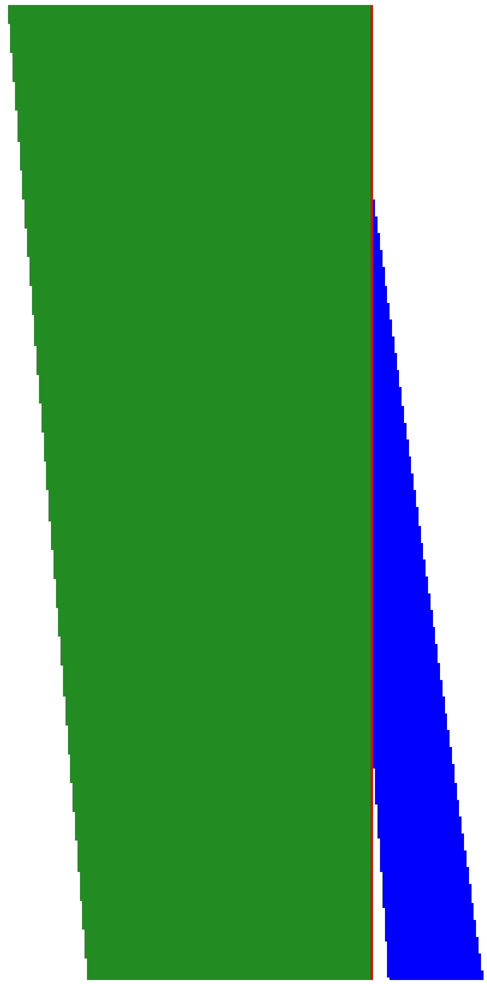


Figure 19: Two Stable Regions Sharing a Line Segment

Triple Rule Algorithm: Using interval arithmetic, suppose the following is true:

1. that each of the four corners (x, y) of the square satisfy $180(k+1) > ax + by > 180(k-1)$
2. that the center of the square (x_0, y_0) satisfies $f_i(x, y) > 0$, $g_i(x, y) > 0$, $u(x, y) > 0$ and $-v(x, y) > 0$ and each of these equations satisfies the Gradient algorithm with respect to the given square. It then follows that all points on the square and its boundary satisfy these inequalities. (Noting that we use $-v$ since we assume $sin(ax + by) < 0$ between the second pair of parallels)
3. that each point on the common boundary line segment $ax + by = 180k$ from (x_1, y_1) to (x_2, y_2) has a periodic path corresponding to a CNS or ONS code which includes this line segment and runs from (x_3, y_3) to (x_4, y_4) and that each of the four corners of the given square lies between the lines $x = x_3$ and $x = x_4$ or between the lines $y = y_3$ and $y = y_4$.

Then that square must lie within R_1 union R_2 union R_3 where R_3 is the unstable line corresponding to the CNS or ONS code from 3 and every point in that union has a periodic path.

Proof. If all points on the square satisfy $sin(ax+by) > 0$, it is within R_1. If all points satisfy $sin(ax+by) < 0$, it is within R_2. Otherwise it is within R_1 union R_2 union R_3. $\square$

13.2 The Half Triple Rule Algorithm

Suppose one stable code region R and has on its boundary an unstable code line T from (x_1, y_1) to (x_2, y_2).

1. This can happen if R is defined by the system of equations $f_i(x, y) > 0$ and $f(x, y) > 0$.
2. Let the equations $f(x, y) = 0$ have a factor of the form $sin(ax + by)$ or $cos(ax + by)$.
3. Now observe that $sin(ax + by) = 0$ if and only if $ax + by = 180k$ and $cos(ax + by) = 0$ if and only if $ax + by = 90 + 180k$ for some integer k.

Lets illustrate with the case where $f(x, y) = sin(ax + by)u(x, y)$.
4. Let us further suppose that R lies between the parallel lines $ax + by = 180(k - 1)$ and $ax + by = 180k$.
5. Further suppose that $sin(ax+by) > 0$ between the first two parallels $ax+by = 180(k-1)$ and $ax+by = 180k$ and $sin(ax + by) < 0$ between the second two parallels $ax + by = 180k$ and $ax + by = 180(k + 1)$.

Now consider a square S with sides parallel to the coordinate axis whose vertex coordinates are all rational numbers and which lies between the first and third parallel and which may or may not intersect the second parallel. It is worth noting that if this square lies inside the stable code region R then it cannot intersect any of the three parallel lines above since $sin(ax+by)$ is zero there. We can use the following to decide what points in the square including its boundary have a periodic path.

Half Triple Rule Algorithm: Using interval arithmetic, suppose the following is true for a given square:

1. that each of the four corners (x, y) of the square satisfy $180(k + 1) > ax + by > 180(k - 1)$
2. that the center of the square (x_0, y_0) is inside R, outside R or on its given straight boundary T and satisfies that $f_i(x, y) > 0$ and $u(x, y) > 0$ and that each of these equations satisfies the Gradient algorithm with respect to the given square. It then follows that all points on the square including its boundary satisfy these two inequalities.
3. that each point on the boundary line segment $ax + by = 180k$ from (x_1, y_1) to (x_2, y_2) has a periodic path corresponding to a CNS or ONS code which includes this line segment and runs from (x_3, y_3) to (x_4, y_4) and that each of the four corners of the given square lies between the lines $x = x_3$ and $x = x_4$ or between the lines $y = y_3$ and $y = y_4$.

Then that partial square must lie within R union T where T is the unstable line corresponding to the CNS or ONS code from 3 and every point in that union has a periodic path.

Proof. If all points on the partial square satisfy $sin(ax + by) > 0$, it is within R. If all points on the partial square satisfy $sin(ax + by) = 0$, it is on T. Thus all those points have a periodic path. □

13.3 The Corner Rule Algorithm

Suppose one stable code region R and has on its boundary two unstable code lines T_1, T_2 which intersect to form a corner at (x_1, y_1) and observe this point is rational and has a periodic path. One unstable T_1 includes from (x_1, y_1) to (x_2, y_2) and the other T_2 includes from (x_1, y_1) to (x_3, y_3). Note that R is an open set and T_1 and T_2 are open segments.

1. This can happen if R is defined by the system of equations $f_i(x, y) > 0$, $f(x, y) > 0$ and $g(x, y) > 0$.
2. Let each of the equations $f(x, y) = 0$ and $g(x, y) = 0$ form that corner and each have a factor of either the form $sin(ax + by)$ or the form $cos(ax + by)$.
3. Now observe that $sin(ax + by) = 0$ if and only if $ax + by = 180k$ and $cos(ax + by) = 0$ if and only if $ax + by = 90 + 180k$ for some integer k.

Lets illustrate with the case where $f(x, y) = sin(ax + by)u(x, y)$ and $g(x, y) = sin(cx + dy)v(x, y)$
4. Let us further as an example suppose that R lies between the parallel lines $ax + by = 180(k_1 - 1)$ and $ax + by = 180k_1$ and that R also lies between the parallel lines $cx + dy = 180k_2$ and $cx + dy = 180(k_2 + 1)$. Note R is in one of four possible areas.

5. Further suppose that $sin(ax+by) > 0$ is between the first two parallels and $sin(cx+dy) > 0$ is between the second two parallels. Note R is in one of the choices of four possible signed areas.
Now consider a square S with sides parallel to the coordinate axis whose vertex coordinates are all rational numbers and which lies between the first and third parallels of both families and which may or may not intersect the two middle parallels. It is worth noting that if this square S lies inside the code region R then it cannot intersect any of the six parallel lines above since $sin(ax+by)$ and $sin(cx+dy)$ is zero there. We can use the following to decide what points in the square including its boundary has a periodic path.

Corner Rule Algorithm: Using interval arithmetic, suppose the following is true:

1. that each of the four corners (x, y) of the square S satisfy $180(k_1+1) > ax+by > 180(k_1-1)$ and $180(k_2+1) > cx+dy > 180(k_2-1)$
2. that the corner (x_1, y_1) is inside or on the boundary of the square S.
3. that the center of the square (x_0, y_0) is inside R or outside R and satisfies that $f_i(x, y) > 0$, $\pm u(x, y) > 0$ and $\pm v(x, y) > 0$ (these are one of the four choices) and that each of these equations satisfies the Gradient algorithm with respect to the given square S and keeps its sign. It then follows that all points on the square including its boundary satisfy all these inequalities.
4. that each point on the boundary open line segments $ax+by = 180k$ from (x_1, y_1) to (x_2, y_2) and $cx+dy = 180k$ from (x_1, y_1) to (x_3, y_3) has a periodic path corresponding to a CNS or ONS code and intersects some side of the square S .

This means every point in the intersection of this square S and the corner R union T_1 union T_2 union (x_1, y_1) has a periodic path. Note this is a closed set.

13.4 The Straight-Curved Corner Rule Algorithm

Suppose one stable code region R and has on its boundary an unstable code line T_1 and a curved side of R which intersect to form a rational corner at (x_1, y_1). Let T_1 go from (x_1, y_1) to (x_2, y_2) and the curved side start at (x_1, y_1). Note that R is an open set and T_1 is an open segment.

1. This can happen if R is defined by the system of equations $f_i(x, y) > 0$, a straight line $f(x, y) > 0$ and a curved side $g(x, y) > 0$.
2. Let the equation $f(x, y) = 0$ have a factor of the form $sin(ax+by)$ or the form $cos(ax+by)$.
3. Let the equation $g(x, y) = 0$ be concave up or down at (x_1, y_1). Further suppose there is another unstable T_2 straight code line with equation $h(x, y) = 0$ that goes from (x_1, y_1) to (x_3, y_3) and lies inside R. The key is that these two lines form a corner K which is inside the closure of R.
4. Let $h(x, y) = 0$ have a factor of either the form $sin(ax+by)$ or the form $cos(ax+by)$.
5. Now observe that $sin(ax+by) = 0$ if and only if $ax+by = 180k$ and $cos(ax+by) = 0$ if and only if $ax+by = 90+180k$ for some integer k.

Lets illustrate with the case where $f(x, y) = sin(ax+by)u(x, y)$ and $h(x, y) = sin(cx+dy)v(x, y)$
6. Let us further suppose that R lies between the parallel lines $ax+by = 180(k_1-1)$ and $ax+by = 180k_1$ and that K also lies between the parallel lines $cx+dy = 180k_2$ and $cx+dy = 180(k_2+1)$. Note K is in one of four possible areas.
7. Further suppose that $sin(ax+by) > 0$ between the first two parallels and $sin(cx+dy) > 0$ between the second two parallels. Note K is in one of four possible signed areas.

Now consider a square S with sides parallel to the coordinate axis whose vertex coordinates are all rational numbers and which lies between the first and third parallels of both families and which may or may not intersect the two middle parallels. It is worth noting that if this square lies inside the code region K then it cannot intersect any of the six parallel lines above since $sin(ax+by)$ and $sin(cx+dy)$ is zero there. We can use the following to decide what points in the square including its boundary has a periodic path.

Straight-Curved Corner Rule Algorithm: Using interval arithmetic, suppose the following is true:

1. that each of the four corners (x, y) of the square S satisfy $180(k_1 + 1) > ax + by > 180(k_1 - 1)$ and $180(k_2 + 1) > cx + dy > 180(k_2 - 1)$
2. that the corner (x_1, y_1) of R is inside or on the boundary of the square S.
3. that the center of the square (x_0, y_0) is inside R or outside R and satisfies that $f_i(x, y) > 0$, $\pm u(x, y) > 0$ and $\pm v(x, y) > 0$ (these are one of the four choices) and that each of these equations satisfies the Gradient algorithm with respect to the given square S and keeps its sign. It then follows that all points on the square including its boundary satisfy all these inequalities.
4. that each point on the boundary open line segments $ax + by = 180k$ from (x_1, y_1) to (x_2, y_2) and $cx + dy = 180k$ from (x_1, y_1) to (x_3, y_3) has a periodic path corresponding to a CNS or ONS code and intersects some side of the square S .

This means every point in the intersection of this square S and the corner K union T_1 union T_2 union (x_1, y_1) has a periodic path. Note that it is a closed set.

13.5 The Two Curved Corner Rule Algorithm

Suppose one stable code region R and has on its boundary two curved sides which intersect to form a rational corner at (x_1, y_1). Note that R is an open set.

1. This can happen if R is defined by the system of equations $f_i(x, y) > 0$, a curved side $g_1(x, y) > 0$ and another curved side $g_2(x, y) > 0$ which form a corner at (x_1, y_1) which must be rational.
2. Let the two equations $g_1(x, y) = 0$ and $g_2(x, y) = 0$ be concave up or down at (x_1, y_1). Further suppose there are two unstables, one T_1 a straight code line with equation $h_1(x, y) = 0$ that goes from (x_1, y_1) to (x_2, y_2) and another one T_2 a straight code line with equation $h_2(x, y) = 0$ that goes from (x_1, y_1) to (x_3, y_3) that lie inside R. This forms a corner K which is inside the closure of R.
3. Let $h_1(x, y) = 0$ and $h_2x, y) = 0$ have factors of either the form $sin(ax+by)$ or the form $cos(ax+by)$.
4. Now observe that $sin(ax + by) = 0$ if and only if $ax + by = 180k$ and $cos(ax + by) = 0$ if and only if $ax + by = 90 + 180k$ for some integer k.

Lets illustrate with the case where $h_1(x, y) = sin(ax + by)u(x, y)$ and $h_2(x, y) = sin(cx + dy)v(x, y)$

5. Let us further suppose that R lies between the parallel lines $ax+by = 180(k_1-1)$ and $ax+by = 180k_1$ and that K also lies between the parallel lines $cx + dy = 180k_2$ and $cx + dy = 180(k_2 + 1)$. Note K is in one of four possible areas.
6. Further suppose that $sin(ax + by) > 0$ between the first two parallels and $sin(cx + dy) > 0$ between the second two parallels. Note K is in one of four possible signed areas.

Now consider a square S with sides parallel to the coordinate axis whose vertex coordinates are all rational numbers and which lies between the first and third parallels of both families and which may or may not intersect the two middle parallels. It is worth noting that if this square lies inside the code region K then it cannot intersect any of the six parallel lines above since $sin(ax + by)$ and $sin(cx + dy)$ is zero there. We can use the following to decide what points in the square including its boundary has a periodic path.

Two Curved Corner Rule Algorithm: Using interval arithmetic, suppose the following is true:

1. that each of the four corners (x, y) of the square S satisfy $180(k_1 + 1) > ax + by > 180(k_1 - 1)$ and $180(k_2 + 1) > cx + dy > 180(k_2 - 1)$
2. that the corner (x_1, y_1) is inside or on the boundary of the square S.
3. that the center of the square (x_0, y_0) is inside R or outside R and satisfies that $f_i(x, y) > 0$, $\pm u(x, y) > 0$ and $\pm v(x, y) > 0$ (these are one of the four choices) and that each of these equations satisfies the Gradient algorithm with respect to the given square S and keeps its sign. It then follows that all points on the square including its boundary satisfy all these inequalities.

4. that each point on the boundary open line segments $ax + by = 180k$ from (x_1, y_1) to (x_2, y_2) and $cx + dy = 180k$ from (x_1, y_1) to (x_3, y_3) has a periodic path corresponding to a CNS or ONS code and intersects some side of the square S .

This means every point in the intersection of this square S and the corner K union T_1 union T_2 union (x_1, y_1) has a periodic path. Note that it is a closed set.

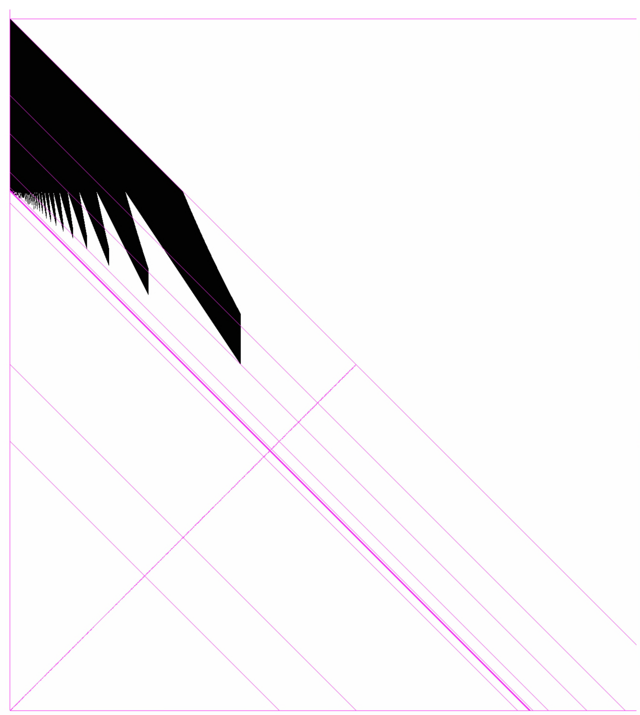
13.6 The Star Rule Algorithm

Suppose the union of n stable code regions has a rational point at (x_1, y_1) and fully surrounds that point with corners. This is expected if that rational point doesn't contain any stables.

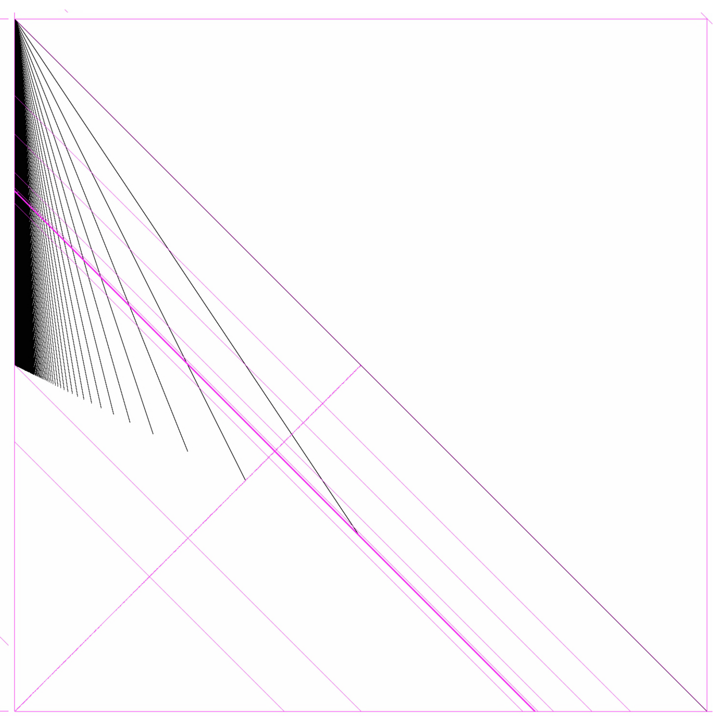
If each one of the stables and its associated unstables satisfies the corner, the straight-curved corner or the two curved corner algorithm, then they fill a square S.

14 The Two Infinite Patterns

There are two infinite patterns which converge to the line segment x=0, $67.5 < y < 90$ as follows. Infinite Pattern I is the same as the one that is used in [13]. Infinite Pattern II is new and uses unstable lines. These two patterns tessellate a cover and don't intersect. It is worth reminding that pattern I is an open region and pattern II is an open interval at the endpoints. Contrast this with [13] which uses overlapping patterns. Our proof for both I and II is new and uses a direct geometric proof.



(a) Infinite Pattern I

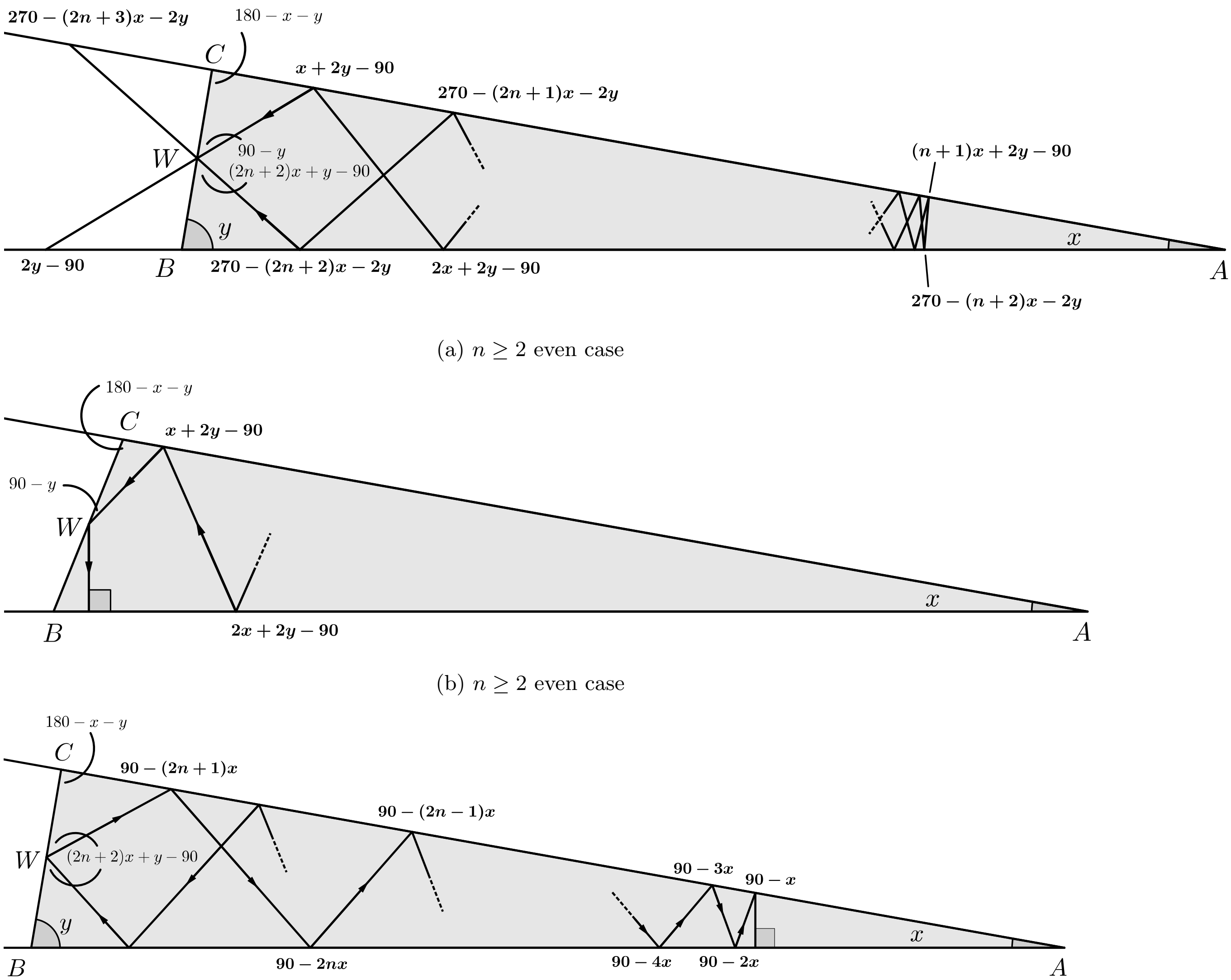


(b) Infinite Pattern II

Figure 20: The two infinite patterns

Infinite Pattern I: Given a triangle ABC with $< A = x$, $< B = y$ where $(n+1)x + 2y < 180 < (n+2)x + 2y$ and $0 < x < 90/(2n+2)$ for $n \geq 1$, then it contains a CS periodic path 1 1 $2n+1$ 1 2 1 $2n+1$ 1 1 $4n+2$. Note: Since $x < 22.5$ and $y + (n+2)x/2 > 90$ and $(n+2)x/2 < (n+2)22.5/(2n+2) < 22.5$ then $y > 67.5$. Also observe that since $n \geq 1$, then $2x + 2y \leq (n+1)x + 2y < 180$ which means that

$x + y < 90$. Finally observe that the successive regions determined by these conditions share the boundary line $(n+1)x + 2y = 180$ for $n \geq 2$.



(a) $n \geq 2$ even case

(b) $n \geq 2$ even case

(c) $n \geq 2$ even case

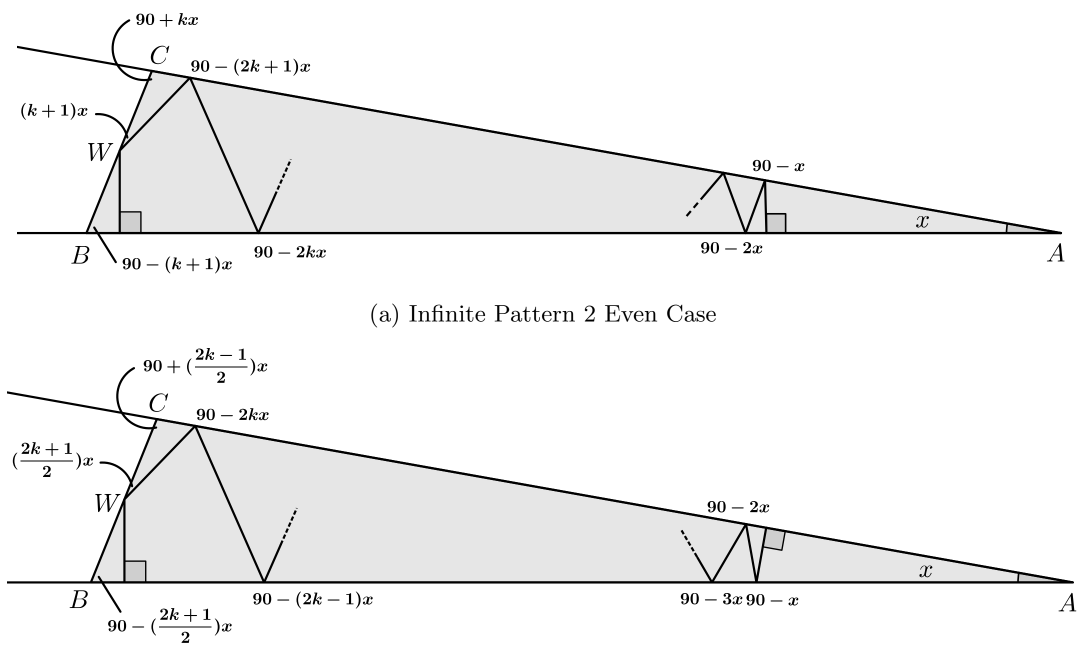
Figure 21: Infinite Pattern I

Proof. For the $n \geq 2$ even case, take a wedge of acute angle x and vertex A with arms l_1 and l_2 as shown on figure 21a and pick a point P on l_1 and shoot a poolball at an acute angle $0 < 270 - (n+2)x - 2y < 90$ as shown. Now since $270 - (n+2)x - 2y + x < 180$, it hits l_2 at an angle $0 < (n+1)x + 2y - 90 < 90$ and continues bouncing all on the wedge at angles $nx + 2y - 90 > (n-1)x + 2y - 90 > ... > x + 2y - 90 > 2y - 90 > 0$ as shown. If we consider the ray from P in the other direction, it bounces off the sides at the angles $270 - (n+3)x - 2y > 270 - (n+4)x - 2y > ... > 270 - (2n+3)x - 2y > 0$ where the last inequality holds since $(n+1)x + 2y < 180$ and $(n+2)x < (2n+2)x < 90$. Now let W be the intersection of the last two rays and draw a line through W hitting l_1 at B, l_2 at C and such that the angle at B as shown is y. Observe that B lies between the last two reflections on l_1 since $y > 2y - 90$ since $y > 45$ and $180 - y > 270 - (2n+2)x - 2y$ since $(2n+2)x + y > 90$ where this last inequality holds since $(2n+2)x + y > (n+2)x/2 + y > 90$. The last ray from l_2 hits BC at W at the angle $90 - y$ and bounces to hit AB at 90 whereas the last ray from l_1 hits W at $(2n+2)x + y - 90$ and since $(2n+2)x + y - 90 + 180 - x - y = 90 + (2n+1)x < 90 + (2n+2)x < 180$, it reflects off BC and hits l_2 at $90-(2n+1)x$. Observing that $90-(2n+1)x > x$ since $(2n+2)x < 90$, the ray then bounces off l_1 and l_2 until it hits at 90 producing a CS periodic path 1 1 $2n+1$ 1 2 1 $2n+1$ 1 1 $4n+2$.

The odd case is handled similarly interchanging l_1 and l_2.

$\square$

Infinite Pattern II: Given a triangle ABC with $< A = x$, $< B = y$ where $(n+1)x + 2y = 180$ and $0 < x < 90/n$ for $n \geq 1$, then it contains a CNS periodic path 1 2 1 $2n$.
Note: These are just the boundary lines (extended) between the regions of theorem 1.



(a) Infinite Pattern 2 Even Case

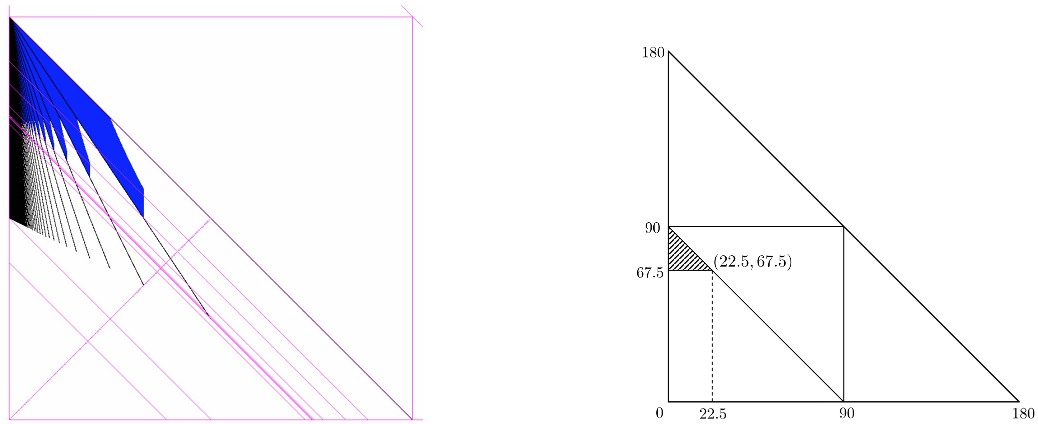
(b) Infinite Pattern 2 Odd Case

Figure 22: Infinite Pattern II

Proof. For the odd integer case $n = 2k+1$, $k \geq 0$, take a wedge of angle x and vertex A with arms l_1 and l_2 as shown in figure 22a and shoot a poolball at 90 degrees from l_1 which then bounces off the sides at angles $90-x > 90-2x > ... > 90-(2k+1)x > 0$ noting that $(2k+1)x < 90$. Now on the last ray leaving l_2 at angle $90-(2k+1)x$, choose any point W between l_1 and l_2 and draw a line through W hitting l_1 at B at an acute angle $y = 90-(k+1)x > 0$ (and so $(n+1)x+2y = 180$) and hitting l_2 at C. Observe that triangle ABC has a periodic path of type 1 2 1 $2n$.

For the even integer case n=2k, $k \geq 1$, again take a wedge of angle x and vertex A with arms l_1 and l_2 as shown in figure 22b and shoot a poolball at 90 degrees from l_2 which then bounces off the sides at angles $90-x > 90-2x > ... > 90-2kx > 0$ noting that $2kx < 90$. On the last ray leaving l_2 at angle $90-2kx$, choose any point W between l_1 and l_2 and draw a line through W hitting l_1 at B at an acute angle $y = 90-(2k+1)x/2 > 0$ (and so $(n+1)x+2y = 180$) and hitting l_2 at C. Observe that triangle ABC has a periodic path of type 1 2 1 $2n$. $\square$

IMPORTANT CONCLUSION: Given an obtuse triangle ABC, with $0 < x < 22.5$, $67.5 < y < 90$ and $x+y < 90$, then that triangle has a periodic path. This can also be seen in [13] using a different pair of infinite patterns.



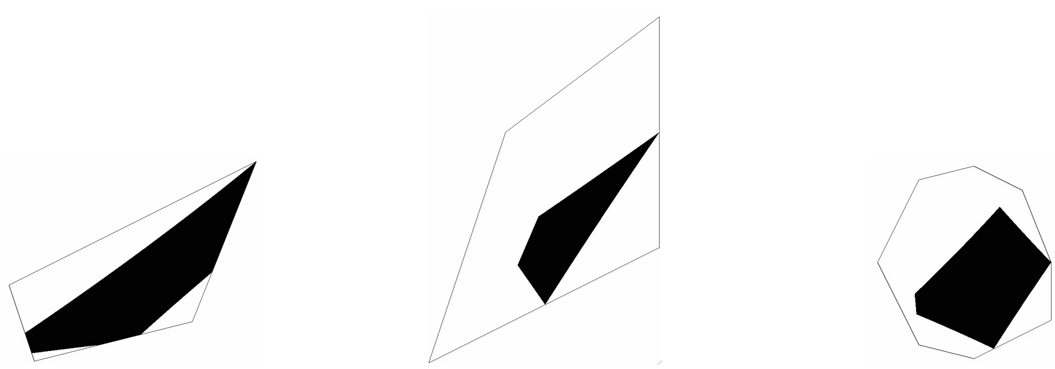
(a) Infinite Patterns (b) Region Covered

Figure 23: Infinite Pattern Cover

15 Bounding Polygons

We use bounding polygons which limit the size of code regions and code line segments. This was not used in [13].

A **bounding polygon** is a convex polygon in which a code region lies inside. Every code region has a bounding polygon for example the region bounded by $0 < x+y < 180$. It is useful in our calculations to find a bounding polygon with rational vertices which is as small as possible, the smallest being the convex hull of the region. There are up to six bounding polygons for each code, one corresponding to each permutation of the angles. For each code, we will assume we have a fixed order of the code angles X,Y and Z.
The corner bounding polygon: This is the polygon determined by the conditions that $0 < nX < 180$, $0 < mY < 180$ and $0 < pZ < 180$ where nX, mY, pZ are the code angles corresponding to the largest X,Y,Z code numbers in the code sequence.
The angle bounding polygon: Given a periodic side sequence leaving side AB of triangle ABC at an angle T where $0 < T \leq 90$, then we can calculate its successive angles as it reflects off each side. Since $z = 180 - x - y$, these reflecting angles will be linear combinations of x, y, 90 and T with integer coefficients. If T can be expressed in terms of x, y and 90 with integer coefficients, then so can all reflecting angles. This is the case for the OSO, CS and CNS periodic paths but not for the OSNO or ONS periodic paths. An example is given below.



(a) CS Code 1 3 6 3 1 7 3 1 8 1 3 7 (b) OSO Code 1 1 2 2 3 1 2 1 4 (c) OSNO Code 1 1 2 2 3 1 3 5

Figure 24: Angle Bounding Polygons

Now observe that in the OSO, CS and CNS cases each reflecting angle θ must satisfy $0 < \theta \leq 90$ and if we omit the 90 degree angles, then the set of (x, y) satisfying $0 < \theta < 90$ forms a bounding polygon which contains the region determined by the given periodic path. In the CNS case, it is a **bounding line segment**. On the other hand in the OSNO and ONS cases, we must have $0 < \theta < 90$ since there are no 90 degree angles. However since these reflecting angles are expressed in terms of x, y, 90 and T with integer coefficients, in order to form the bounding polygon we must eliminate T. This can be done as follows. For each reflecting angle of the form $mx + ny + p90 + T$, we can also say that $0 < 90 - mx - ny - p90 - T < 90$ and similarly for reflecting angles of the form $mx + ny + p90 - T$. We then get two sets of angles involving either T or $-T$ and if we add each equation with T to each equation with $-T$ and divide by 2, we end up with a set of linear cominations of x,y and 90 with rational coefficients which lie between 0 and 90 and hence produce a bounding polygon in these cases. In the ONS case it is a **bounding line segment**. These are the bounding polygons that we usually use in our calculations.

It is worth noting that the corner bounding polygon equations are included amongst the angle bounding polygon equations. This is a consequence of the fact that if a poolshot enters a corner A where $< A = x$ at an angle θ and bounces n times before it leaves then the angles involved are
θ,
$\theta + x$,
$\theta + 2x$,
...,
$180 - \theta - (n-2)x$,
$180 - \theta - (n-1)x$,
$180 - \theta - nx$
and then since $0 < \theta < 90$ and $0 < 180 - \theta - nx < 90$, we must have $0 < 180 - nx < 180$ or $0 < nx < 180$ which is one of the corner equations.

16 The Program and Proof

Because of the complexity and quantity of these equations, there is a dire need to automate the process of proving the codes work. Thus, we have written a program to crunch the numbers. In these calculations, each G_j is an integer and is exact. On the other hand r, b_1, and b_2 are in radians and in fact are rational multiples of $\pi/2$, so they too are exact. The f_j involve evaluating sines and cosines, so they are not exact. All these calculations are done by computer and have a certain degree of accuracy. We need to make sure that when we calculate that $f_j > 0$, that it is indeed true. To do this we use interval arithmetic which can show that f_j lies exactly within an interval $[u, v]$ with $u > 0$. The interval that we use for $\pi/2$ correct to 7 decimal places is (1.57079631,1.57079637). This precision can be increased as required.
We mainly use the arithmetical operations of

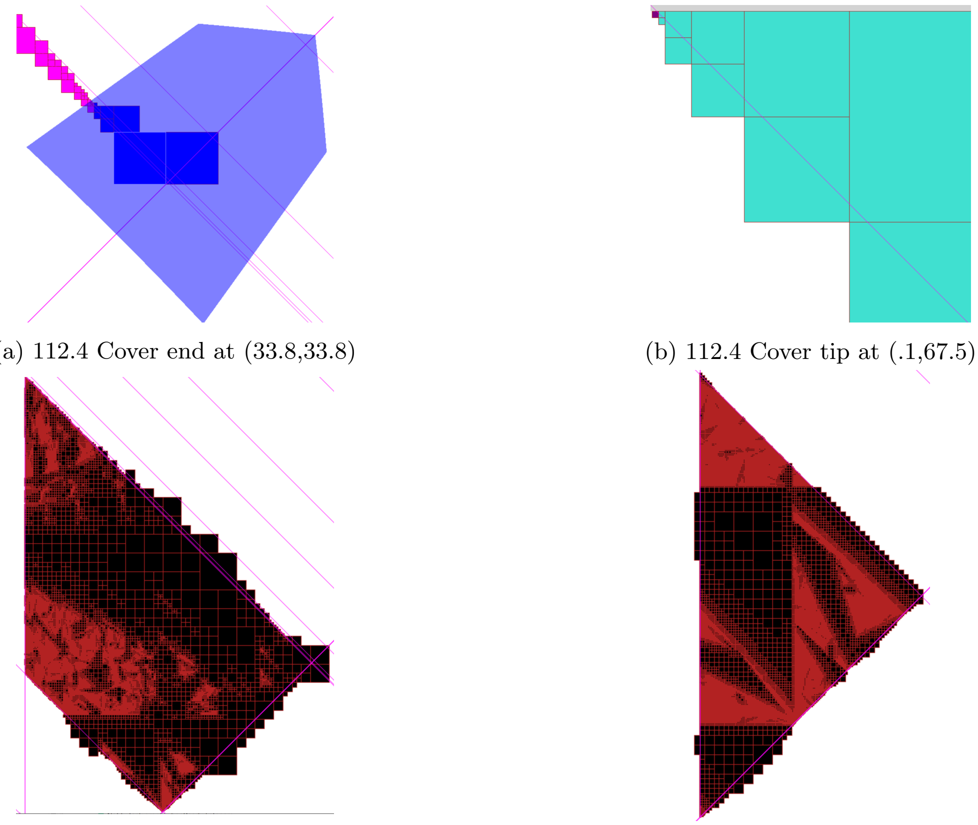
addition: $[x_1, x_2] + [y_1, y_2] = [x_1 + y_1, x_2 + y_2]$
subtraction: $[x_1, x_2] - [y_1, y_2] = [x_1 - y_2, x_2 - y_1]$
multiplication: $[x_1, x_2][y_1, y_2] = [min(x_1y_1, x_1y_2, x_2y_1, x_2y_2), max(x_1y_1, x_1y_2, x_2y_1, x_2y_2)]$

Note, we don't use any division operations.

In doing the calculations in the prover, we rely on the following two libraries for arbitrary precision arithmetic.
1. GMP: https://gmplib.org/
2. MPFR: http://www.mpfr.org/ [5]

Along with this paper, you should get our two standalone jars in the website called Billiards Everything. You will need to have a mac and have java8. In the first jar, you can find, test and prove the existance of periodic paths. This is the best way to get comfortable with this proof. You can play around with it or if you want to get more serious, you can join the Great Periodic Path Hunt.

In the second jar with the Covers jar, you can go square by square to see all the equations that produce a given code region and the lower bound of the calculations using interval arithmetic. This will show that



(a) 112.4 Cover end at (33.8,33.8)

(b) 112.4 Cover tip at (.1,67.5)

(c) Cover top of the 12 degree theorem

(d) Cover tip of the 12 degree theorem

Figure 25: The 12 and 112.4 Theorem

each square satisfies the prover. Because of how massive the proof is, there isn't a good way to present all of the calculations at once so we did it in smaller pieces.

A second way to trust this program is doing the calculations properly and to feel more comfortable about the proof, we recommend that you try some random triangles from previous papers. A good one to start with would be [11] and see that it gives the same shape. Others would be [8], [13] and [14].

To prove from 100 to 105 degrees there are 134 single codes that cover the region between $z = 75$ and $z = 80$ and which are not part of the two infinite patterns. See Appendix B. The endpoints of this region in clockwise order are (37.5, 37.5),(40, 40), (12.5, 67.5),(7.5, 67.5). We used 13,862 squares at 7 decimal accuracy starting from a single square and had to subdivide any square at most 20 times to be able to prove that these 134 code regions do cover this region. Our interval arithmetic calculations showed that the smallest lower bound on the prover at any equation used on a covering square was 6.65023 x 10^{-9}. The program shows the code type, a 2-tuple consisting of the code length and the side sequence length followed by the code sequence.

On this same program, you can find listed the 2439 single codes and 21 triples that are used to prove from 105 to 110, the 38,132 single codes and 310 triples to prove from 110 to 112 and the 118,809 single codes and 1,115 triples to prove from 112 to 112.3 and the 202,120 single codes and 2,915 triples to prove from 112.3 to 112.4.

In the 12 degree theorem from 22.4988 to the top, there are 7006 single codes and 94 triples, and from 17 to 22.4988 there are 40,474 single codes and 329 triples, and from 15 to 17 there are 47,350 single codes and 279 triples, and from 14 to 15 there are 39,487 single codes and 90 triples and from the tip to 14 there are 42,722 single codes and 306 triples. Make sure that you look at the tiny flare at (15,30) and load the 22.4988-33.8 cover.

Between the 11 and 12 degree theorem, see the covers jar from the tip at A11-12 to the top at A24-56.6.

17 Final Comments

From our data, we can see there is a large growth as we go from 100 to 112.4. See Figures 25 to 26. This is entirely due to the pile up of codes as we get closer and closer to the point (0,112.5). Over half of the codes lie in the sliver from 112.3 to 112.4 which amounts to less than one percent of the area of our cover.

From the work of Schwartz and Hooper [12] there are obtuse triangles of the Veech form which have only unstable codes. For neighbourhoods around those points they found a pile up codes using a double infinite family cover and conjectured that there is no finite cover.

The point (0,112.5) appears to be worse and we conjecture that it needs an infinite set of infinite families to surround.

Does every obtuse triangle have a periodic path? Is this problem intractable as many believe. We leave the reader with some clues and conjectures. We believe that the most enticing clue is the flares. There appears to be a connection with the Ward points [15] of the form $(180/2n, 180/n)$ for $n \geq 4$. Our calculations show that all Ward points for $4 \leq n \leq 36$ have no flares EXCEPT for possibly $n = 6, 9, 18, 27$ for which we could not find any stable codes. We thus have three conjectures.

1. There are an infinite number of flares amongst the Ward points.
2. There are no other flares in the interior.
3. Some can have infinite covers and some finite covers. We conjecture that $n = 6$ needs an infinite cover and $n = 9$ needs only a finite cover.

Have fun.

Acknowledgements

The authors would like to thank the NSERC USRA program, Arno Berger, Xi Chen, Gerald Cliff, Gerda de Vries, Terry Gannon, Thomas Hillen, James Lewis, Brendan Pass, Eric Primozic, Byron Schmuland, Zhongwei Shen, George Tokarsky, Suneeta Varadarajan and Eric Woolgar all of whom provided financial support to this project.

For you history buffs, we started this computing project in 2005 using laptops and all the theory and ideas in this paper came from Boyan Marinov and George Tokarsky. George wrote the paper and found and patiently inputted all the data. All of this data can be verified with laptops but to keep sane we eventually graduated to more powerful computers using the Cirrus Research Computing Cloud at the University of Alberta and using the cluster Cedar in the region Westgrid under Compute Canada. We thank William Campbell, Kamil Marcinkowski, Roman Paszewski, Igor Sinelnikov, Chris Want and Broderick Wood for maintaining these systems and answering our many questions.

We also would like to thank a succession of student coders Noor Al-Tamimi, Dylan Ashley, Raj Boora, Leah Brown, Wade Brown, Alexis Cerkiewicz, Bennett Csorba, Zheng En Than, Jacob Garber, Deeksha Gautam, Mingjun Hou, Colman Koivisto, Dan Moore, Kenneth Moore, Connor Peck, Cameron Ridderikhoff, Stephen Romansky, Shehraj Singh, Zijie Tan, Byron Tung, Shiyu Xiu, Alfred Ye and Chunyan Zhang for helping with the coding. We especially thank David Szepesvari for doing the finishing touches and fixing the previous broken code. We also thank Stefan Nychka for fixing the color codings. Finally we also thank Julie Wright for helping with the diagrams.

References

[1] Howard Masur, "Closed Trajectories for Quadratic Differentials with an Application to Billiards", *Duke Mathematical Journal* 53, no. 2 (1986): 307–314.

[2] Carlo Boldrighini, Michael Keane and Federico Marchetti, "Billiards in Polygons" *The Annals of Probability* 6, no. 4 (1978): 532–540.

[3] Michael D. Boshernitzan "Billiards and Rational Periodic Directions in Polygons" *The American Mathematical Monthly, Vol. 99, No. 6 (Jun. - Jul., 1992), pp. 522-529*

[4] M. Boshernitzan, G. Galperin, T. Kruger and S. Troubetzkoy, "Some Remarks on Periodic Billiard Orbits in Rational Polygons" *preprint SUNY Stony Brook* (1994)

[5] Laurent Fousse, Guillaume Hanrot, Vincent Lefèvre, Patrick Pélissier, Paul Zimmermann, "MPFR: A Multiple-Precision Binary Floating-Point Library With Correct Rounding", *ACM Transactions on Mathematical Software* 33, no. 2 (June 2007): Article 13.

[6] G. Galperin and D. Zvonkine, "Periodic Billiard Trajectories in Right Triangles and Right-Angled Tetrahedra" *Regular and Chaotic Dynamics* 8, no. 1 (2003): 29–44.

[7] Torbjörn Granlund et al., GNU Multiple Precision Arithmetic Library, https://gmplib.org.

[8] Lorenz Halbeisen and Norbert Hungerbuhler, "On periodic billiard trajectories in obtuse triangles" *SIAM review* 42, no. 4 (2000): 657–670.

[9] Fred Holt, "Periodic Reflecting Paths in Right Triangles", *Geometriae Dedicata* 46, no. 1 (April 1993): 73–90.

[10] W. Patrick Hooper, "Periodic billiard paths in right triangles are unstable" *Geometriae Dedicata* 125, no. 1 (March 2007): 39–46.

[11] W. Patrick Hooper, Richard Evan Schwartz, "Introduction to McBilliards" *Geometriae Dedicata* 125, no. 1 (Sept. 2004): 39–46.

[12] W. Patrick Hooper, Richard Evan Schwartz, "Billiards in Nearly Isosceles Triangles", Journal of Modern Dynamics 3, no. 2 (2009): 159–231.

[13] Richard Evan Schwartz, "Obtuse Triangular Billiards II: One Hundred Degrees Worth of Periodic Trajectories", *Experimental Mathematics* 18, no. 2 (2009): 137–171.

[14] Ya. B. Vorobets, G. A. Galperin, and A. M. Stepin, "Periodic billiard trajectories in polygons generating mechanisms" *Russ. Math. Surv.* 47, no. 5 (1992): 5–80.

[15] Clayton C. Ward, "Calculation of Fuchsian groups associated to billiards in a rational triangle" *Ergodic Theory Dynam. Systems* 18, no. 4 (1998): 1019–1042.

A Example of Calculating Coordinates of a Code Tower

Here we will calculate the coordinates of the vertices in the code tower of the code ONS below which runs along the diagonal in the map of all triangles. Refer to section 12 for the notation. By the Sine Law, sinx, siny and sinz are the lengths of the sides opposite the angles x, y and z. To distinquish lengths of sides from combinations of angles we use combinations of angles X, Y and Z. Finally we use subscripts to coordinates as in x_i and y_i.

X Z X
1 1 2 3 3 2
 Y Y Z
Then
$x_1 = siny$, $y_1 = 0$ which are the coordinates of $L_{(1,0)}$
x_2=0, $y_2 = 0$ which are the coordinates of $L_{(2,0)}$
$x_3 = sinzcosX$,
$y_3 = sinzsinX$ which are the coordinates of $L_{(3,0)}$
$x_4 = sinzcosX - sinxcos(Y - X)$,
$y_4 = sinzsinX + sinxsin(Y - X)$ which are the coordinates of $L_{(4,0)}$
$x_5 = sinzcosX - sinxcos(Y - X) + sinxcos(2Z + X - Y)$,

$y_5 = sinzsinX + sinxsin(Y - X) + sinxsin(2Z + X - Y)$ which are the coordinates of $L_{(5,0)}$
$x_6 = sinzcosX - sinxcos(Y - X) + sinxcos(2Z + X - Y) - sinzcos(3Y - 2Z + Y - X)$,
$y_6 = sinzsinX + sinxsin(Y - X) + sinxsin(2Z + X - Y) + sinzsin(3Y - 2Z + Y - X)$ which are the coordinates of $L_{(6,0)}$
$x_7 = sinzcosX - sinxcos(Y - X) + sinxcos(2Z + X - Y) - sinzcos(3Y - 2Z + Y - X) + sinycos(3X - 3Y + 2Z - Y + X)$,
$y_7 = sinzsinX + sinxsin(Y - X) + sinxsin(2Z + X - Y) + sinzsin(3Y - 2Z + Y - X) + sinysin(3X - 3Y + 2Z - Y + X)$ which are the coordinates of $L_{(7,0)}$
$x_8 = sinzcosX - sinxcos(Y - X) + sinxcos(2Z + X - Y) - sinzcos(3Y - 2Z + Y - X) + sinycos(3X - 3Y + 2Z - Y + X) - sinycos(2Z - 3X + 3Y - 2Z + Y - X)$,
$y_8 = sinzsinX + sinxsin(Y - X) + sinxsin(2Z + X - Y) + sinzsin(3Y - 2Z + Y - X) + sinysin(3X - 3Y + 2Z - Y + X) + sinysin(2Z - 3X + 3Y - 2Z + Y - X)$ which are the coordinates of $L_{(8,0)}$

We then use standard trig identities together with the conditions that $z = 180 - x - y$ and $y = x$ from the code pattern.

$sin(-A) = -sinA$
$cos(-A) = cosA$
$sin(180 - A) = sinA$
$cos(180 - A) = -cosA$
$sinA + sinB = 2sin(\frac{A+B}{2})cos(\frac{A-B}{2})$
$sinA - sinB = 2sin(\frac{A-B}{2})cos(\frac{A+B}{2})$
$cosA + cosB = 2cos(\frac{A+B}{2})cos(\frac{A-B}{2})$
$cosA - cosB = -2sin(\frac{A+B}{2})sin(\frac{A-B}{2})$
$2cosAcosB = cos(A + B) + cos(A - B)$
$2sinAcosB = sin(A + B) + sin(A - B) = sin(A + B) - sin(B - A)$
$2sinAsinB = cos(A - B) - cos(A + B)$

The poolshot vector is (c, d) where (c, d) is the vector from $L_{(2,0)}$ to $L_{(8,0)}$
$c = sin(y)cos(180) + sin(z)cos(x) + sin(x)cos(x - y + 180) + sin(x)cos(-x - 3y + 4(180)) + sin(z)cos(-x - 6y + 5(180)) + sin(y)cos(2x - 6y + 6(180))$
$d = sin(y)sin(180) + sin(z)sin(x) + sin(x)sin(x - y + 180) + sin(x)sin(-x - 3y + 4(180)) + sin(z)sin(-x - 6y + 5(180)) + sin(y)sin(2x - 6y + 6(180))$
CONVENTION: In using the last three trig identities to simplify to sums of sines and cosines, we multiply all coordinates by 2 so that all coefficients are integers.
The poolshot vector (c, d) then further becomes
$c = -2sin(y) - sin(3y) + sin(5y) - sin(2x - 7y) + sin(2x - 5y) - sin(2x - y) + sin(2x + y) + sin(2x + 3y) - sin(2x + 7y)$ a sum of sines.
$d = -cos(3y) + cos(5y) + cos(2x - 7y) - cos(2x - 5y) + cos(2x - y) - cos(2x + y) + cos(2x + 3y) - cos(2x + 7y)$ a sum of cosines.
To calculate the coordinates of the key points in the tower which are not centers of fans if any, we illustrate by the same example. Suppose we look at the coordinates of $L_{(4,1)}$. It is found by starting with the coordinates of $L_{(4,0)}$ and since the corresponding code is 2 adding one more reflection of the given triangle and proceeding as before.
$x_4 = sinzcosX - sinxcos(Y - X)$,
$y_4 = sinzsinX + sinxsin(Y - X)$
becomes
$x_{(4,1)} = sinzcosX - sinxcos(Y - X) + sinycos(Z - Y + X)$
$y_{(4,1)} = sinzsinX + sinxsin(Y - X) + sinysin(Z - Y + X)$
and then simplyfing to sums of sines or cosines.
To find the coordinates of $L_{(6,2)}$, we would start with the coordinates of $L_{(6,0)}$ and since the corresponding code is 3 adding two more reflections of the given triangle which corresponds to using the angle $2X - 3Y + 2Z - Y + X$ and we get
$x_{(6,2)} = x_6 + sinzcos(2X - 3Y + 2Z - Y + X) = x_6 + sinzcos(x - 6y)$

$y_{(6,2)} = y_6 + sinzsin(2X - 3Y + 2Z - Y + X) = x_6 + sinzsin(x - 6y)$
and then simplyfing to sums of sines or cosines.
It is now a simple matter to calculate a vector (a, b) from any key blue point to any key black point. As an example the blue-black vector from $L_{(6,0)}$ to $L_{(5,0)}$ is given by

$a = sinzcos(3Y - 2Z + Y - X) = sinzcos(6y + x) = sin(7y + 2x) - sin5x = sin9x - sin5x$ since y=x for this code

$b = -sinzsin(3Y - 2Z + Y - X) = -sinzsin(6y + x) = cos(7y + 2x) - cos5y = cos9x - cos5x$

B The Single Codes for the 105 theorem

1. OSO (3, 7) 1 3 3
2. OSO (5, 11) 1 1 2 2 5
3. OSO (5, 15) 1 1 4 2 7
4. OSO (5, 15) 1 3 2 6 3
5. OSO (5, 17) 1 1 4 2 9
6. OSO (5, 21) 1 1 6 2 11
7. OSO (5, 23) 1 1 6 2 13
8. OSO (7, 15) 1 1 3 1 2 1 6
9. OSO (7, 17) 1 1 3 1 2 1 8
10. OSO (7, 19) 1 1 2 2 6 2 5
11. OSO (7, 21) 1 1 2 2 8 2 5
12. OSO (7, 23) 1 1 4 2 6 2 7
13. OSO (7, 29) 1 1 6 2 8 2 9
14. OSO (7, 17) 1 2 1 2 1 3 7
15. OSNO (8, 18) 1 1 2 2 3 1 3 5
16. OSNO (8, 22) 1 1 4 2 3 1 3 7
17. OSO (9, 25) 1 1 2 2 6 2 4 2 5
18. CS (10, 20) 1 1 3 1 2 1 3 1 1 6
19. OSNO (10, 24) 1 1 3 1 2 1 5 1 1 8
20. CS (10, 28) 1 1 5 1 2 1 5 1 1 10
21. OSNO (10, 32) 1 1 5 1 2 1 7 1 1 12
22. CS (10, 36) 1 1 7 1 2 1 7 1 1 14
23. OSNO (10, 40) 1 1 7 1 2 1 9 1 1 16
24. CS (10, 44) 1 1 9 1 2 1 9 1 1 18
25. OSNO (10, 48) 1 1 9 1 2 1 11 1 1 20
26. OSNO (10, 26) 1 1 2 2 7 1 1 4 2 5
27. OSNO (10, 38) 1 1 4 2 11 1 1 6 2 9
28. OSO (11, 29) 1 1 2 1 1 7 2 4 1 1 8
29. OSO (11, 25) 1 1 2 1 1 6 1 1 2 2 7
30. OSNO (12, 38) 1 1 2 2 8 2 3 1 3 8 2 5
31. OSNO (14, 40) 1 1 3 1 2 1 7 2 4 1 1 7 2 7
32. OSNO (14, 44) 1 1 3 1 2 1 7 2 6 1 1 9 2 7
33. OSNO (14, 52) 1 1 3 1 2 1 11 2 6 1 1 11 2 9
34. OSNO (14, 36) 1 1 2 2 7 1 2 1 3 1 1 7 2 5
35. CS (14, 36) 1 2 1 5 3 1 3 2 4 2 3 1 3 5
36. CS (16, 44) 1 1 2 1 1 7 3 1 3 2 6 2 3 1 3 7
37. CS (16, 52) 1 1 4 1 1 9 3 1 3 2 8 2 3 1 3 9
38. OSNO (18, 42) 1 1 2 1 1 6 1 1 2 2 7 1 2 1 2 1 3 7
39. CS (18, 44) 1 1 2 2 5 1 2 1 5 2 2 1 1 5 2 4 2 5
40. CS (18, 52) 1 1 4 2 5 1 2 1 5 2 4 1 1 7 2 4 2 7
41. OSO (19, 57) 1 1 2 1 1 6 1 2 1 3 1 1 7 2 8 2 8 2 7
42. CS (20, 48) 1 1 3 1 2 1 6 1 2 1 3 1 1 8 1 1 4 1 1 8
43. CS (20, 56) 1 1 3 1 2 1 10 1 2 1 3 1 1 10 1 1 4 1 1 10

44. CS (20, 64) 1 1 5 1 2 1 8 1 2 1 5 1 1 12 1 1 6 1 1 12
45. CS (20, 88) 1 1 9 1 2 1 8 1 2 1 9 1 1 18 1 1 10 1 1 18
46. OSNO (20, 74) 1 1 2 1 1 7 1 1 13 2 7 1 2 1 7 1 1 13 2 9
47. CS (20, 52) 1 1 2 1 1 7 2 2 1 1 5 2 6 2 5 1 1 2 2 7
48. CS (20, 60) 1 1 2 1 1 7 2 4 1 1 7 2 6 2 7 1 1 4 2 7
49. CS (20, 68) 1 1 4 1 1 9 2 4 1 1 7 2 8 2 7 1 1 4 2 9
50. CS (20, 92) 1 1 6 1 1 13 2 6 1 1 11 2 10 2 11 1 1 6 2 13
51. CS (20, 60) 1 1 2 2 9 1 1 4 2 8 2 4 1 1 9 2 2 1 1 6
52. CS (20, 92) 1 1 6 2 13 1 1 8 2 10 2 8 1 1 13 2 6 1 1 12
53. CS (20, 60) 1 1 4 2 6 2 4 1 1 7 2 4 1 1 8 1 1 4 2 7
54. OSO (21, 49) 1 1 1 1 2 1 7 2 5 1 1 2 1 1 7 2 5 1 2 1 4
55. OSNO (22, 54) 1 1 1 1 3 7 1 1 3 1 2 1 5 2 6 2 2 1 1 5 2 5
56. OSNO (22, 82) 1 1 2 1 1 7 1 1 12 1 1 4 1 1 11 2 8 1 1 13 2 9
57. OSNO (22, 58) 1 1 2 2 7 1 2 1 3 2 7 1 1 3 1 2 1 5 2 6 2 5
58. OSNO (22, 64) 1 1 2 2 7 1 2 1 3 2 6 2 4 1 1 7 2 4 2 6 2 5
59. OSNO (22, 60) 1 1 4 2 7 1 2 1 4 1 2 1 5 1 1 9 2 3 1 2 1 8
60. OSNO (22, 74) 1 1 2 2 8 2 4 2 6 2 7 1 1 4 2 6 2 6 2 6 2 5
61. OSO (23, 55) 1 1 2 1 1 5 1 1 8 1 2 1 2 1 2 1 8 1 1 4 1 1 8
62. CS (24, 64) 1 1 1 1 2 1 9 3 1 2 1 3 9 1 2 1 1 1 1 5 2 8 2 5
63. OSNO (24, 60) 1 1 3 1 2 1 5 2 3 1 3 6 2 3 1 3 7 1 1 3 1 2 1 6
64. OSNO (24, 88) 1 1 3 1 2 1 11 2 9 1 2 1 5 1 1 12 1 1 6 1 1 13 2 9
65. OSNO (24, 68) 1 1 3 1 2 1 8 1 1 4 1 1 9 2 4 1 1 9 2 3 1 2 1 8
66. CS (24, 64) 1 1 2 1 1 5 1 1 10 1 1 6 1 1 10 1 1 5 1 1 2 1 1 8
67. CS (24, 76) 1 1 4 2 6 2 3 1 3 8 3 1 3 2 6 2 4 1 1 7 2 4 2 7
68. CS (26, 64) 1 1 1 1 2 1 7 2 4 2 7 1 2 1 1 1 1 5 2 5 1 2 1 5 2 5
69. CS (26, 104) 1 1 5 1 2 1 9 2 12 2 9 1 2 1 5 1 1 13 2 7 1 2 1 7 2 13
70. OSNO (26, 72) 1 1 2 1 1 5 1 1 8 1 2 1 3 2 9 1 1 4 2 9 1 1 4 1 1 8
71. CS (26, 64) 1 1 2 1 1 6 1 1 2 1 1 7 3 1 3 2 5 1 2 1 5 2 3 1 3 7
72. CS (26, 124) 1 1 6 2 12 2 7 1 2 1 7 2 12 2 6 1 1 11 2 8 2 12 2 8 2 11
73. CS (28, 80) 1 1 3 1 2 1 7 2 4 1 1 8 1 1 4 2 7 1 2 1 3 1 1 7 2 6 2 7
74. CS (28, 112) 1 1 3 1 2 1 11 2 8 1 1 14 1 1 8 2 11 1 2 1 3 1 1 9 2 12 2 9
75. OSNO (28, 118) 1 1 3 1 2 1 11 2 8 2 10 2 11 1 1 6 2 12 2 8 2 11 1 1 4 1 1 10
76. CS (28, 76) 1 1 3 1 2 1 10 1 1 4 1 1 10 1 2 1 3 1 1 8 1 2 1 6 1 2 1 8
77. CS (28, 80) 1 1 2 1 1 5 1 1 9 2 4 2 9 1 1 5 1 1 2 1 1 9 3 1 2 1 3 9
78. OSNO (28, 74) 1 1 2 1 1 5 2 6 2 5 1 1 2 2 7 1 2 1 3 2 6 2 5 1 1 2 2 7
79. OSNO (28, 68) 1 1 2 2 6 2 5 1 1 2 2 7 1 1 3 1 2 1 6 1 1 3 1 2 1 5 2 5
80. CS (28, 88) 1 1 4 2 10 2 4 1 1 7 2 9 1 1 5 1 2 1 4 1 2 1 5 1 1 9 2 7
81. CS (28, 92) 1 1 4 2 6 2 6 2 6 2 4 1 1 7 2 6 2 4 1 1 8 1 1 4 2 6 2 7
82. OSNO (30, 76) 1 1 2 2 5 1 2 1 4 1 2 1 5 3 1 2 1 3 6 3 1 3 2 4 2 4 2 4 2 5
83. OSO (31, 87) 1 1 3 1 2 1 6 1 2 1 3 1 1 7 2 6 2 4 1 1 7 2 5 1 2 1 5 2 6 2 7
84. OSNO (32, 84) 1 1 2 1 1 5 1 1 8 1 1 2 1 1 5 2 6 2 7 1 1 4 2 7 1 1 2 1 1 5 2 7
85. CS (32, 76) 1 1 2 1 1 5 1 1 8 1 2 1 2 1 2 1 8 1 1 5 1 1 2 1 1 8 1 1 4 1 1 8
86. CS (32, 76) 1 1 2 1 1 6 1 2 1 3 1 1 8 1 1 4 1 1 8 1 1 4 1 1 8 1 1 3 1 2 1 6
87. CS (32, 80) 1 1 2 2 5 1 2 1 5 3 1 2 1 3 5 1 2 1 5 2 2 1 1 5 2 4 2 4 2 4 2 5
88. OSNO (34, 80) 1 1 1 1 3 7 1 1 3 1 2 1 5 2 5 1 2 1 3 2 7 1 1 3 1 2 1 5 2 5 1 2 1 4
89. CS (34, 96) 1 1 3 1 2 1 5 2 6 2 4 2 6 2 5 1 2 1 3 1 1 7 2 4 2 5 1 2 1 5 2 4 2 7
90. CS (36, 84) 1 1 2 1 1 5 1 1 8 1 1 2 1 1 6 1 1 2 1 1 8 1 1 5 1 1 2 1 1 8 1 1 4 1 1 8
91. CS (36, 108) 1 1 2 1 1 7 1 1 12 1 1 4 1 1 10 1 1 4 1 1 12 1 1 7 1 1 2 1 1 10 1 1 4 1 1 10
92. CS (36, 132) 1 1 4 1 1 9 1 1 14 1 1 6 1 1 12 1 1 6 1 1 14 1 1 9 1 1 4 1 1 12 1 1 6 1 1 12
93. CS (36, 84) 1 1 2 1 1 6 1 1 2 1 1 6 1 1 2 1 1 7 3 1 3 2 5 1 2 1 4 1 2 1 5 2 3 1 3 7
94. CS (36, 84) 1 1 2 1 1 6 1 1 2 2 7 1 2 1 2 1 3 7 1 1 2 1 1 7 3 1 2 1 2 1 7 2 2 1 1 6
95. OSNO (36, 102) 1 1 2 1 1 6 1 2 1 5 2 7 1 1 3 1 2 1 6 1 2 1 5 2 6 2 6 2 7 1 1 4 2 6 2 7
96. CS (36, 112) 1 1 2 2 9 1 1 4 1 1 9 3 1 3 2 8 2 4 2 8 2 3 1 3 9 1 1 4 1 1 9 2 2 1 1 6
97. CS (38, 100) 1 1 2 1 1 6 1 1 2 1 1 7 2 7 1 1 3 1 2 1 7 2 5 1 2 1 5 2 7 1 2 1 3 1 1 7 2 7

98. OSNO (40, 96) 1 1 1 1 3 7 1 1 3 1 2 1 5 2 5 1 1 2 2 7 1 1 3 1 2 1 5 2 6 2 2 1 1 5 2 5 1 2 1 4
99. OSNO (40, 128) 1 1 3 1 2 1 10 1 1 5 1 2 1 7 2 13 1 1 7 1 2 1 6 1 2 1 7 2 13 1 1 7 1 2 1 6 1 2 1 8
100. OSNO (40, 120) 1 1 3 1 2 1 8 1 1 4 2 8 2 5 1 2 1 5 1 1 9 2 5 1 2 1 5 2 8 2 4 1 1 9 2 3 1 2 1 8
101. CS (40, 148) 1 1 2 1 1 7 1 1 13 2 9 1 2 1 5 1 1 13 2 8 2 13 1 1 5 1 2 1 9 2 13 1 1 7 1 1 2 1 1 10
102. CS (40, 120) 1 1 2 1 1 7 2 4 1 1 7 2 6 2 6 2 6 2 7 1 1 4 2 7 1 1 2 1 1 7 2 4 1 1 8 1 1 4 2 7
103. CS (40, 100) 1 1 2 1 1 6 1 1 3 1 2 1 5 2 6 2 5 1 2 1 3 1 1 6 1 1 2 1 1 7 2 4 1 1 8 1 1 4 2 7
104. OSNO (42, 116) 1 1 2 1 1 5 1 1 9 2 4 2 9 1 1 4 2 9 1 1 4 1 1 8 1 2 1 3 1 1 8 1 1 2 1 1 6 1 1 2 2 9
105. OSNO (44, 118) 1 1 2 1 1 7 2 4 1 1 8 1 1 2 1 1 7 2 4 1 1 8 1 2 1 3 2 9 1 1 5 1 2 1 4 1 2 1 7 2 4 1 1 8
106. CS (48, 140) 1 1 1 1 3 9 1 1 2 1 1 5 1 1 9 2 4 2 9 1 1 4 1 1 9 2 4 2 9 1 1 5 1 1 2 1 1 9 3 1 1 1 1 5 2 8 2 5
107. CS (48, 136) 1 1 2 1 1 6 1 1 2 1 1 7 2 2 1 1 5 2 6 2 4 2 7 1 1 4 2 6 2 4 2 6 2 4 1 1 7 2 4 2 6 2 5 1 1 2 2 7
108. CS (50, 156) 1 1 1 1 3 9 1 1 4 2 8 2 5 1 2 1 5 2 8 2 5 1 2 1 5 2 8 2 4 1 1 9 3 1 1 1 1 5 2 8 2 5 1 2 1 5 2 8 2 5
109. OSNO (50, 156) 1 1 2 1 1 7 2 6 2 6 2 8 2 4 1 1 9 2 3 1 2 1 8 1 1 3 1 2 1 8 1 1 4 2 8 2 6 2 6 2 7 1 2 1 3 1 1 7 2 7
110. CS (56, 132) 1 1 1 1 2 1 7 2 4 1 1 8 1 1 4 2 7 1 2 1 1 1 1 4 1 2 1 5 2 7 1 1 3 1 2 1 6 1 1 2 1 1 6 1 2 1 3 1 1 7 2 5 1 2 1 4
111. CS (56, 160) 1 1 2 1 1 5 1 1 9 2 4 2 9 1 1 4 1 1 9 2 4 2 9 1 1 5 1 1 2 1 1 9 3 1 2 1 3 9 1 1 2 1 1 6 1 1 2 1 1 9 3 1 2 1 3 9
112. OSNO (56, 216) 1 1 4 1 1 9 2 11 1 1 5 1 2 1 8 1 2 1 5 1 1 12 1 1 6 1 1 13 2 6 2 13 1 1 6 2 13 1 1 6 1 1 12 1 1 4 1 1 9 2 12 2 6 1 1 12
113. CS (56, 216) 1 1 2 1 1 8 1 1 2 1 1 9 2 12 2 9 1 2 1 5 1 1 13 2 7 1 2 1 7 2 13 1 1 6 1 1 13 2 7 1 2 1 7 2 13 1 1 5 1 2 1 9 2 12 2 9
114. OSNO (56, 142) 1 1 2 1 1 6 1 2 1 3 1 1 7 2 5 1 2 1 4 1 2 1 6 1 1 2 1 1 7 2 5 1 1 2 2 8 2 5 1 1 2 2 9 1 1 5 1 2 1 4 1 2 1 5 2 7
115. OSNO (56, 170) 1 1 2 1 1 6 1 2 1 5 2 8 2 4 1 1 8 1 2 1 3 1 1 9 2 2 1 1 5 1 1 9 2 4 1 1 9 2 4 2 8 2 6 2 6 2 7 1 2 1 3 1 1 7 2 7
116. CS (56, 172) 1 1 2 2 9 1 1 4 1 1 9 2 2 1 1 6 1 1 2 2 9 1 1 4 1 1 9 3 1 3 2 8 2 4 2 8 2 4 2 8 2 3 1 3 9 1 1 4 1 1 9 2 2 1 1 6
117. OSNO (58, 158) 1 1 2 1 1 6 1 2 1 5 2 7 1 2 1 3 1 1 7 2 7 1 1 2 1 1 6 1 2 1 5 2 8 2 4 1 1 9 2 2 1 1 5 1 1 9 2 4 1 1 8 1 2 1 3 1 1 8
118. OSNO (60, 162) 1 1 2 1 1 5 1 1 9 2 4 2 9 1 1 4 2 9 1 1 4 1 1 9 2 3 1 2 1 8 1 2 1 2 1 3 9 1 1 2 1 1 6 1 1 2 1 1 8 1 1 2 1 1 6 1 1 2 2 9
119. CS (60, 144) 1 1 2 1 1 5 1 1 8 1 2 1 2 1 2 1 8 1 1 4 1 1 8 1 2 1 2 1 2 1 8 1 1 5 1 1 2 1 1 8 1 1 4 1 1 8 1 2 1 2 1 2 1 8 1 1 4 1 1 8
120. CS (60, 180) 1 1 2 1 1 7 2 4 1 1 7 2 6 2 6 2 6 2 6 2 6 2 7 1 1 4 2 7 1 1 2 1 1 7 2 4 1 1 8 1 1 4 2 7 1 1 2 1 1 7 2 4 1 1 8 1 1 4 2 7
121. OSNO (60, 186) 1 1 2 1 1 7 2 6 2 6 2 8 2 4 1 1 8 1 2 1 3 1 1 8 1 2 1 3 2 9 1 1 4 2 8 2 6 2 6 2 7 1 1 2 1 1 7 2 6 2 7 1 2 1 3 1 1 7 2 7
122. CS (64, 156) 1 1 2 1 1 6 1 1 2 1 1 8 1 1 3 1 2 1 7 2 6 2 7 1 2 1 3 1 1 8 1 1 2 1 1 6 1 1 2 1 1 9 2 2 1 1 6 1 1 2 1 1 8 1 1 2 1 1 6 1 1 2 2 9
123. CS (66, 200) 1 1 1 1 3 9 1 1 2 1 1 5 1 1 9 2 4 2 9 1 1 4 1 1 9 2 4 2 9 1 1 4 1 1 9 2 4 2 9 1 1 5 1 1 2 1 1 9 3 1 1 1 1 5 2 8 2 5 1 2 1 5 2 8 2 5
124. CS (66, 208) 1 1 1 1 3 9 1 1 4 2 8 2 5 1 2 1 5 2 8 2 5 1 2 1 5 2 8 2 5 1 2 1 5 2 8 2 4 1 1 9 3 1 1 1 1 5 2 8 2 5 1 2 1 5 2 8 2 5 1 2 1 5 2 8 2 5
125. OSNO (66, 230) 1 1 2 1 1 5 1 1 9 2 4 1 1 9 2 4 2 8 2 6 2 6 2 8 2 4 1 1 9 2 4 2 8 2 6 2 6 2 8 2 4 2 9 1 1 4 1 1 9 2 4 1 1 9 2 4 2 9 1 1 5 1 1 2 2 9
126. CS (68, 188) 1 1 2 1 1 6 1 1 2 1 1 8 1 2 1 3 2 9 1 1 4 2 8 2 5 1 2 1 6 1 1 2 1 1 8 1 1 4 2 8 2 4 1 1 8 1 1 2 1 1 6 1 2 1 5 2 8 2 4 1 1 9 2 3 1 2 1 8
127. CS (72, 216) 1 1 2 1 1 7 2 4 1 1 8 1 1 2 1 1 7 2 4 2 8 2 4 2 8 2 4 2 7 1 1 2 1 1 8 1 1 4 2 7 1 1 2 1 1 7 2 5 1 2 1 5 2 8 3 1 3 2 8 2 3 1 3 8 2 5 1 2 1 5 2 7
128. OSNO (74, 220) 1 1 2 1 1 6 1 2 1 5 2 8 2 4 2 9 1 1 4 2 8 2 6 2 6 2 7 1 1 2 1 1 7 2 7 1 1 3 1 2 1 7 2 6 2 7 1 1 3 1 2 1 7 2 5 1 2 1 5 1 1 9 2 4 1 1 8 1 2 1 3 1 1 8
129. CS (80, 324) 1 1 2 1 1 7 2 13 1 1 8 2 10 2 9 1 2 1 5 1 1 12 1 1 6 2 13 1 1 6 1 1 12 1 1 6 2 12 2 8 2 12 2 6 1 1 12 1 1 6 1 1 13 2 6 1 1 12 1 1 5 1 2 1 9 2 10 2 8 1 1 13 2 7 1 1 2 1 1 10

130. CS (84, 260) 1 1 1 1 3 9 1 1 2 1 1 5 1 1 9 2 4 2 9 1 1 4 1 1 9 2 4 2 9 1 1 4 1 1 9 2 4 2 9 1 1 4 1 1 9 2 4 2 9 1 1 5 1 1 2 1 1 9 3 1 1 1 1 5 2 8 2 5 1 2 1 5 2 8 2 5 1 2 1 5 2 8 2 5
131. CS (92, 252) 1 1 1 1 3 9 1 1 5 1 1 2 1 1 8 1 1 4 1 1 8 1 2 1 2 1 2 1 8 1 1 4 1 1 8 1 1 2 1 1 5 1 1 9 3 1 1 1 1 5 2 6 2 7 1 2 1 2 1 2 1 8 1 1 4 2 8 2 4 2 8 2 4 2 8 2 4 1 1 8 1 2 1 2 1 2 1 7 2 6 2 5
132. CS (100, 300) 1 1 2 1 1 7 2 4 1 1 7 2 6 2 6 2 6 2 6 2 6 2 6 2 6 2 6 2 6 2 7 1 1 4 2 7 1 1 2 1 1 7 2 4 1 1 8 1 1 4 2 7 1 1 2 1 1 7 2 4 1 1 8 1 1 4 2 7 1 1 2 1 1 7 2 4 1 1 8 1 1 4 2 7 1 1 2 1 1 7 2 4 1 1 8 1 1 4 2 7
133. CS (102, 320) 1 1 1 1 3 9 1 1 2 1 1 5 1 1 9 2 4 2 9 1 1 4 1 1 9 2 4 2 9 1 1 4 1 1 9 2 4 2 9 1 1 4 1 1 9 2 4 2 9 1 1 4 1 1 9 2 4 2 9 1 1 5 1 1 2 1 1 9 3 1 1 1 1 5 2 8 2 5 1 2 1 5 2 8 2 5 1 2 1 5 2 8 2 5 1 2 1 5 2 8 2 5
134. OSNO (116, 326) 1 1 2 1 1 6 1 2 1 5 2 7 1 2 1 4 1 2 1 6 1 2 1 5 2 7 1 1 2 1 1 8 1 1 4 2 8 2 5 1 2 1 5 2 8 2 5 1 2 1 5 2 9 1 1 5 1 2 1 5 2 8 2 5 1 2 1 5 2 8 2 4 1 1 8 1 1 2 1 1 7 2 5 1 2 1 6 1 2 1 4 1 2 1 7 2 5 1 2 1 5 1 1 9 2 4 1 1 8 1 2 1 3 1 1 8