A Mathematical Approach to Classifying Poi Patterns

a poi paper by

Ben Drexler aka "Drex"

Contact:
drex@drexfactor.com
http://www.drexfactor.com

Table of Contents

A. Introduction

The art of poi may be one of the most unique confluences of art, movement, math, and science in the modern era. At its root, the art is based in tribal dances practiced by the Maori of New Zealand for an unknown period before being exported to the Western world to integrate with a variety of martial arts, dance, and other prop manipulation styles. In its current form, spinning poi focuses on the creation of curves in space using a weighted end connected via a flexible tether to the performer's hand. Despite such simple apparatus, in the past decade poi has exploded in its vocabulary and number of practitioners.

As poi has acquired more practitioners and edged ever closer to a mainstream pursuit, it has also become the focal point of a variety of controversies among its practitioners. One such series of controversies is over the definition of even some of the most basic poi moves. As much of the movement vocabulary for poi has been created on an ad hoc basis or passed down via oral tradition rather than recorded or formalized in deliberate fashion, there exist strong regional differences in definitions, movement vocabularies, and interpretations of how movements interrelate to the point that such conflicts are all but inevitable.

We in the poi world are fortunate, however, in that much of the vocabulary of our art can be modeled using fairly trivial mathematics. Indeed, the vast majority of shapes and tricks we poi spinners produce can be modeled using trig functions that many of us learned in high school. This provides us with two opportunities: the first is that by understanding how the values in these equations generate the moves we are familiar with, we can easily enter alternate values to create patterns we have yet to experience. This expands our vocabulary as well as our understanding of the art. Second: we can find the boundaries of values necessary to produce given families of tricks. It is this second benefit that is most intriguing, for it offers us an opportunity to formalize the definitions of poi tricks in a way that is easily falsifiable and thus finally come to discuss poi within the context of a true shared vocabulary.

Given that this method of modeling poi movement through mathematics is not well known within the poi community at large, I have written this paper with the following two goals:

1. Educate the general public and flow community as to how to model poi moves utilizing the trig functions I was taught years ago.
2. Create the first formal classification scheme for poi tricks based in mathematical values rather than arbitrary metaphors or imprecise words

In order to restrict the length of what is already a painfully long paper, I am limiting the scope of what I cover only to patterns and tricks that are identifiable when performed with a single poi sans the body of the performer. As such, this paper does not include any information that covers the following topics: hybrids, inversions, timing and direction, etc (I've got to have some stuff to save for a follow-up if ever I get crazy enough to attempt a follow-up). With that in mind, this paper will consist of two sections: the first is meant to teach the reader the fundamentals of how trig functions can be used to model poi movement. It will cover trig equations that model a variety of different types of common poi moves. After covering all of these cases, I will use them to generalize an equation from which we can model any and all of the poi patterns covered in this paper (and hopefully pretty much every single poi pattern possible). I will use this equation in the second half of the paper to set down definitions for each category of poi movement modeled in the first half of the paper.

Let's start with the math.

B. Fundamentals

Poi consists of a small weight on the end of a tether connected to a person's hand. The usual practice is to perform with 2 poi, one for each hand, though explorations of 3 and 4 poi configurations have become more common in recent years. The materials used for this prop vary widely according to the performer, location, and resources at hand. However, fundamentally all tricks the performer may create with the poi will be based upon the idea of introducing accelerations in various directions through the tether to the head, leading to an abundance of possible patterns.

The dominant approach to poi performance for the better part of a decade has been to spin the poi in patterns that can be derived from vertical curves, presenting the audience with as full a profile of said curves as possible. Given that the most common approaches to these tricks are to induce the poi to travel in a circular or elliptical path around the hand/handle, we can use math designed to describe circles and gradually add layers of complexity as needed to make the equations capable of displaying more complex patterns. First, a few caveats:

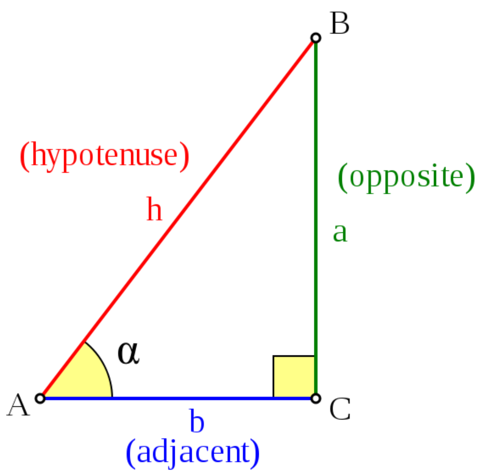
This type of math is excellent for describing the simple system of how two points in space relate to each other over time, but not how they interact with other objects introduced into the system. As such, we can use this method to describe how the poi moves in space, but how it moves in relation to the body is in many cases outside the bounds of this type of mathematical modeling. For this system, we will focus exclusively on the patterns the poi head and handle generate over time and not focus as much on how these patterns interact with the body nor how it presents obstacles that specific types of moves are designed to avoid. As such, any and all contact moves, wraps, inside moves, and throws are not covered in this breakdown.

This type of math is excellent for modeling complete poi patterns but is less helpful when modeling partial or composite patterns. While a section exists to describe some elementary versions of these patterns, the complexity of the modeling increases dramatically the more moves we try to understand in sequence.

Modeling poi in this fashion is agnostic of gravity--due to the forces of acceleration placed on the poi head, this frequently is not as important to understanding a pattern as the mathematically perfect version of said pattern is, but it is worth noting as a shortfall of this approach.

1. Background on Periodic Math

This section will work heavily in the fundamental math behind periodic functions. It is not necessary to read this section completely to understand everything that comes after it, especially if the reader already has an operating knowledge of Trigonometry. If the reader would like, they can skip this section to avoid the basic math and move on to how it applies to poi by clicking here.



Trigonometric functions provide the easiest means of modeling the behavior of cyclical functions over time. I am indebted to Adam Dipert for first teaching me how cycloid math could be simplified to provide a good model for poi movement and Will Ruddick for assisting me in creating computer simulations that taught me the depth to which these movements could be modeled using this mathematical approach.

To begin, we need an understanding of sine and cosine functions. To the left is a chart of a basic right triangle defined by angles ABC and sides abh. If we pick angle A as our home base, we can see it has two intersecting sides: b and h. Side h is referred to the hypotenuse as it is the only side that does not intersect with the right angle of the triangle, angle C. The remaining two sides are labeled according to their relationship with angle A: b is referred to as the adjacent side because it, along with the hypotenuse intersects at said angle. Side a is referred to as the opposite side as it is the side opposite angle A.

To define our first trigonometric function, we will establish a relationship between the lengths of sides a (the opposite) and h (the hypotenuse), that is if we divide a by h, we get a relationship referred to as the sine of angle A ($sinA = \frac{a}{h}$). Depending upon the measure of angle A, a and h will have set proportions ranging from 0:1 to 1:1. Sine is usually abbreviated as *sin* in these equations.

Our second trigonometric function is the cosine function, which has a different relationship to angle A than the sine function does. For cosine, we are dividing b (the adjacent side) by h (the hypotenuse), resulting in the equation $cosA = \frac{b}{h}$. As with the sine function, cosine will have set proportions depending on the size of angle A ranging from 1:1 to 0:1, but the proportion will always be different than sine A except in the special case when a = b, or the two sides are of equivalent length.

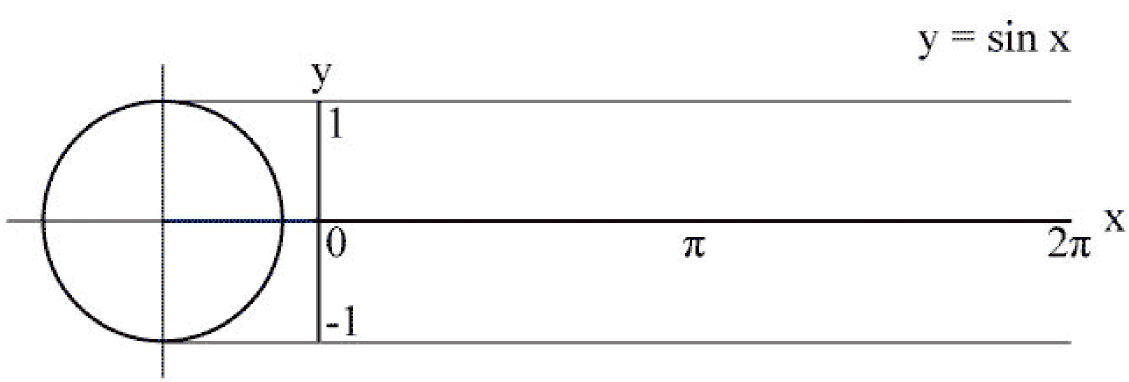
Now that we know the basics of these functions, let's learn a little bit about how we can graph the results of them out and change the resulting graph by adding particular values to different parts of the function.

To start, we will get results for a few values of $y = sinx$ (please note, I am using radians for values of x--to get the same values, ensure that your calculator is in RAD mode, not DEG. To find out why I'm doing this and what radians are, click here):

x	**y**
0	0
$\frac{\pi}{4}$	0.707*
$\frac{\pi}{2}$	1
$\frac{3\pi}{4}$	0.707*
π	0
$\frac{5\pi}{4}$	-0.707*
$\frac{3\pi}{2}$	-1
$\frac{7\pi}{4}$	-0.707*
2π	0

* For these values I have truncated down a much longer number to only three decimal places

As we can see, even though the value of x continues to rise, the value of y cycles between 1 and -1, passing through numbers in between in predictable intervals. The animated image below displays how this equation is graphed with all values of x between 0 and 2π:

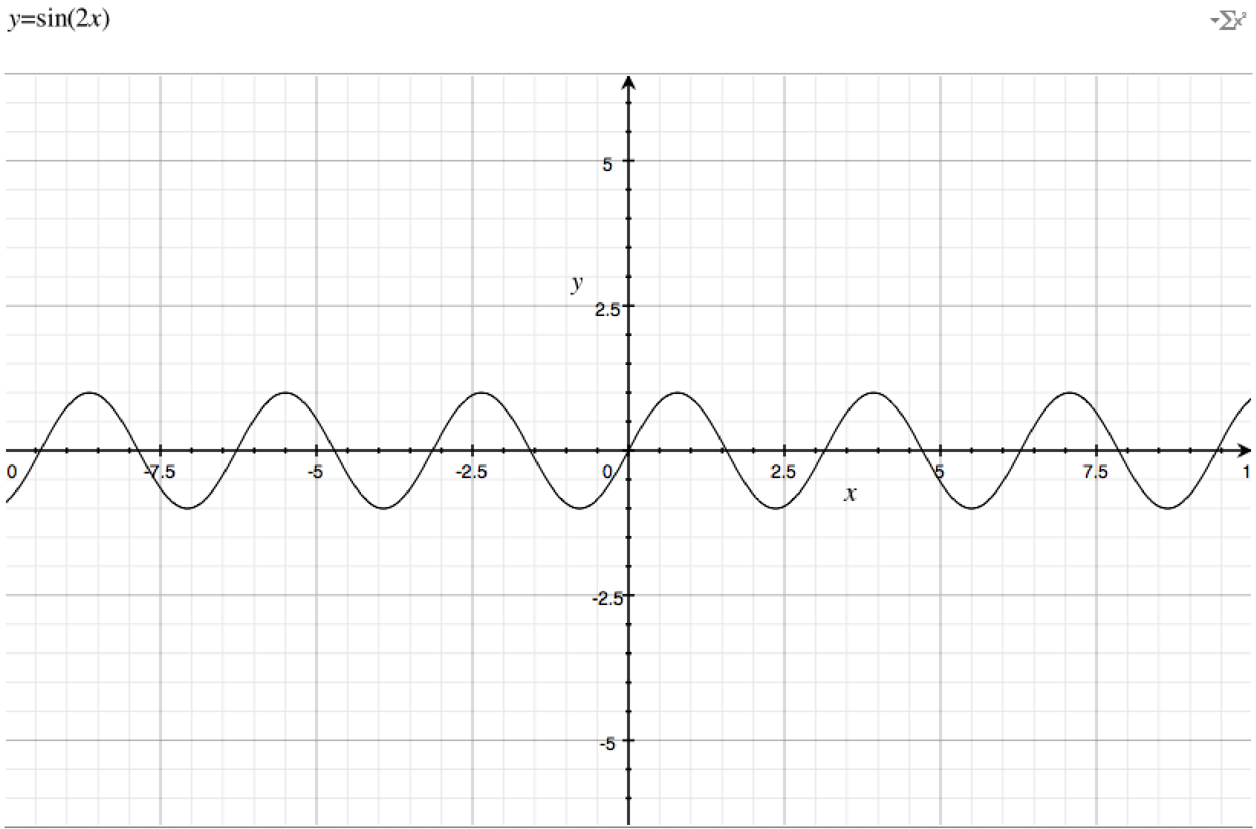


As you can see, graphing this equation produces a gentle, wave-like curve that is known as a sine wave. The sine wave goes no further than 1 unit positive or negative away from the x-axis and begins to repeat its results after 2π. We are going to introduce two terms here that will be used as placeholders to describe the exact same ideas: first, we are going to say that this wave has an amplitude of 1 (maximum distance from the x-axis) and second that it has a wavelength of 2π (distance between points where the wave starts to repeat itself). We are now going to see how we can change both the amplitude and the wavelength of $y = sin x$.

To start, we will change the wavelength of this equation. We will accomplish this by multiplying x before it's had a chance to pass through the sine function. Think about it this way: whenever we enter $\frac{\pi}{2}$ in, we want the answer we get at π --the point twice as far along the curve. We would write this equation out as $y = sin(2x)$, which changes the values from the table above to those below:

x	**y**
0	0
$\frac{\pi}{4}$	1
$\frac{\pi}{2}$	0
$\frac{3\pi}{4}$	-1
π	0
$\frac{5\pi}{4}$	1
$\frac{3\pi}{2}$	0
$\frac{7\pi}{4}$	-1
2π	0

Note how the value for y that we get now at $\frac{\pi}{2}$ is equivalent to our old value for π and now $\frac{3\pi}{4}$ is equivalent to our old value for $\frac{3\pi}{2}$ This, then produces the graph below.



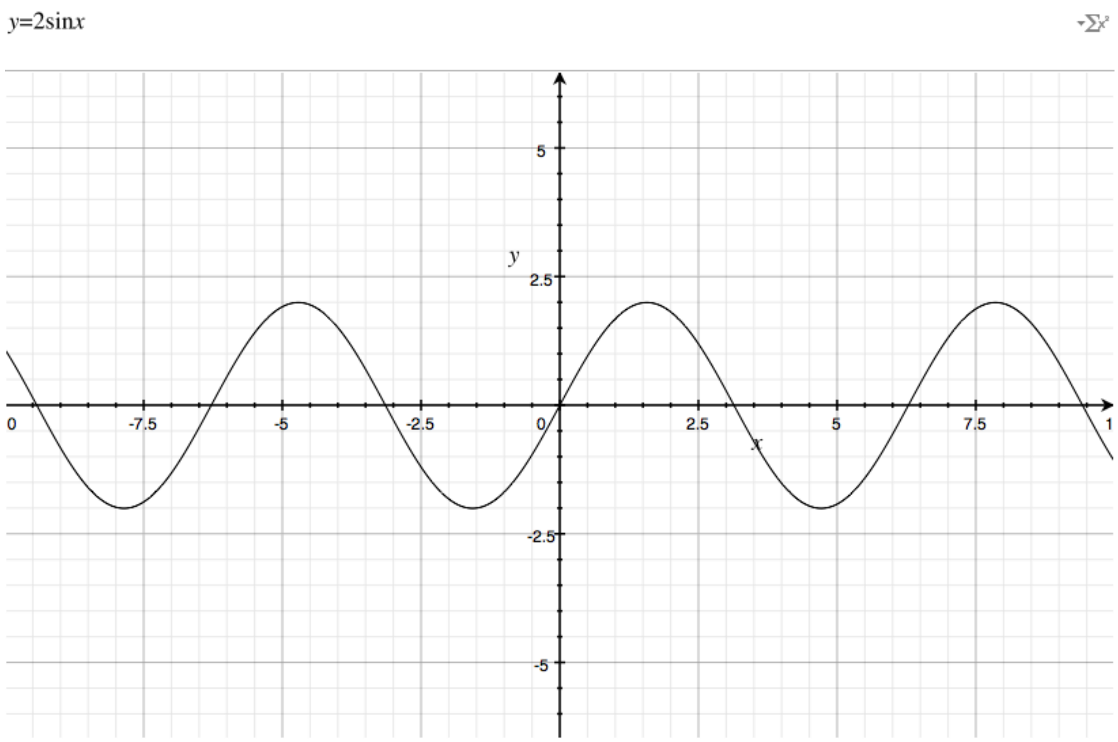
Sure enough--our graph still has amplitude 1, but now it begins repeating at π rather than 2π.

But what about the amplitude? Changing the value of x before it integrates into the sine function overall didn't have any effect the amplitude of the y values for the equation, so we will have to make this amendment after the sine function has already performed its function. Let's do this by checking our values for $y = 2 \cdot sinx$. Now our table comes out with these values:

x	**y**
0	0
$\frac{\pi}{4}$	1.414*
$\frac{\pi}{2}$	2
$\frac{3\pi}{4}$	1.414*
π	0
$\frac{5\pi}{4}$	-1.414*
$\frac{3\pi}{2}$	-2
$\frac{7\pi}{4}$	-1.414*
2π	0

* Again, I have truncated a longer number to only 3 decimal places for clarity's sake.

Now our y values go between -2 and 2 rather than -1 and 1 while the function goes back to repeating at 2π. As you can see, adding a 2 to this equation in these different spots produces radically different results! Here is the graph for this equation:



Success!

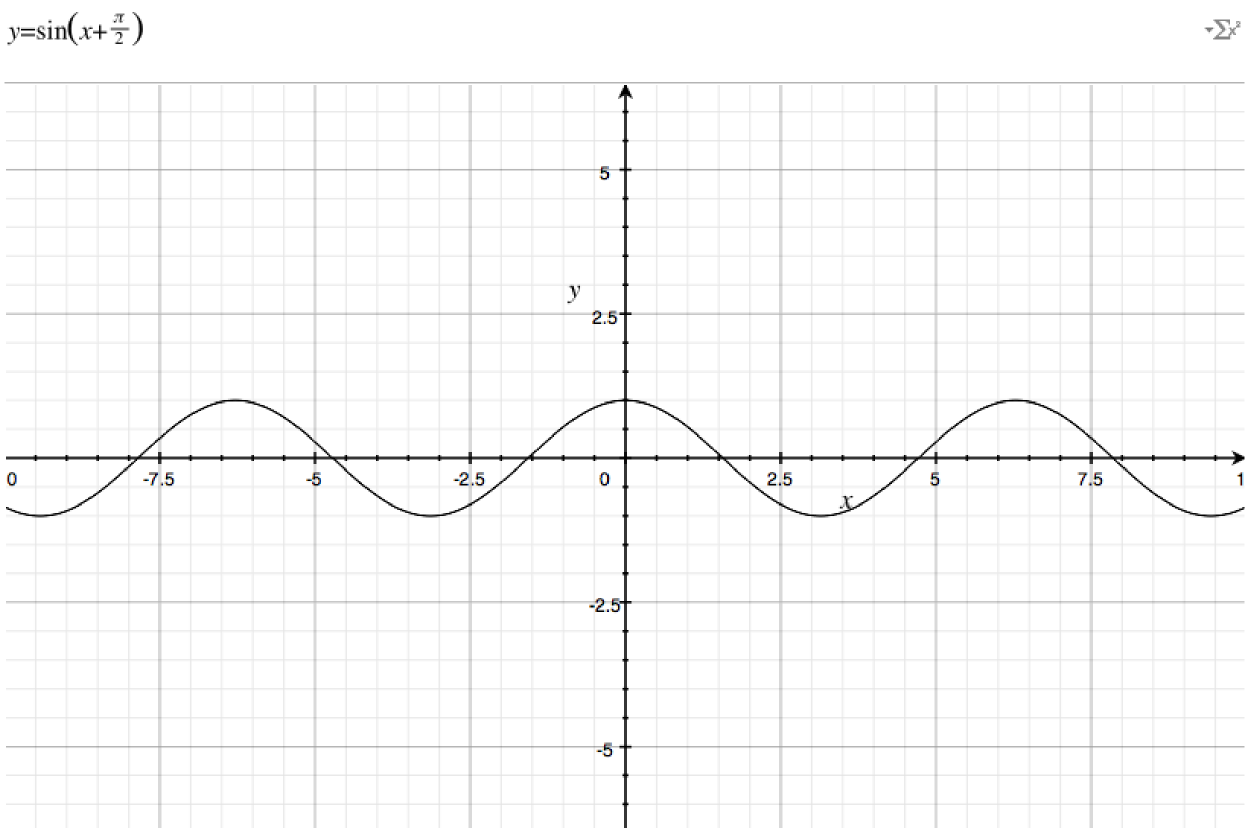
There is one final transformation we can perform on this equation that does not change the amplitude or the wavelength, but is nonetheless important. By now you may have noticed that the origin of this graph always occurs at 0 on the x axis. There is a relatively simple method we can use to bump this graph a little bit to either side slightly, starting it in a slightly different spot along the curve. To do this, we actually have to add or subtract values before the sine function, using the equation $y = sin(x + \frac{\pi}{2})$. Now our table of values comes out:

x	**y**
0	1
$\frac{\pi}{4}$	0.707*
$\frac{\pi}{2}$	0
$\frac{3\pi}{4}$	-0.707*
π	-1
$\frac{5\pi}{4}$	-0.707*
$\frac{3\pi}{2}$	0
	0.707*

$\frac{7\pi}{4}$	1
2π	

* Yup! Once again these values have been truncated!

Now our equation starts at 1 rather than 0. The resulting graph looks like this:



Now the graph begins on a peak at 0 rather than halfway between a peak and a valley. There's another fun side-effect of this transformation, as well. You may have noticed we have yet to graph anything with our cosine function--it turns out that $y = cos x$ produces the exact same result as $y = sin(x + \frac{\pi}{2})$. Please note: this will not be the case for most other values we could add to x in this equation--there are a whole range of different starting points available to us for this wave!

With these results in mind, then, we can think of a generalized equation for graphing wave functions that looks something like this:

$y = a \cdot sin(bx + c)$

Where a is the amplitude of the wave, b is the wavelength, and c is how far we are displacing the wave along the x axis. Knowing what these three variables do will be hugely helpful in navigating the rest of this paper!

Now let's apply this knowledge to graphing curves that meet back up upon themselves, such as circles. Because sine and cosine graphs of the same variable have peaks that are offset by a quarter of a wavelength, we can use the two functions together to graph a circle by using the equations $x = sint$ and $y = cost$ where t is equal to all values between 0 and 2π.

A Note on Radians

Why 2π? Instead of degrees, mathematicians using trigonometry more frequently use a method for measuring angles called radians. Because the circumference of a circle is equal to double the radius times pi ($2\pi r$), we can imagine setting the value of r aside to think of the distance around the circle as being 2π. We then can think of any fraction of that distance as being the measure of the angle we are hoping to achieve. For example, 90 degrees is the angle we achieve by dividing a circle into ¼, so we can also express the size of this angle by dividing 2π by 4, where we get a value of $\frac{\pi}{2}$. If it's easier to think in degrees than radians, simply take any value of t I present in the rest of the paper and perform the following operation on it to get the equivalent in degrees (d): $d = t\left(\frac{180}{\pi}\right)$. **For the rest of this paper we will assume t = all values between 0 and 2π unless otherwise noted.**

A Note on the Unit Circle

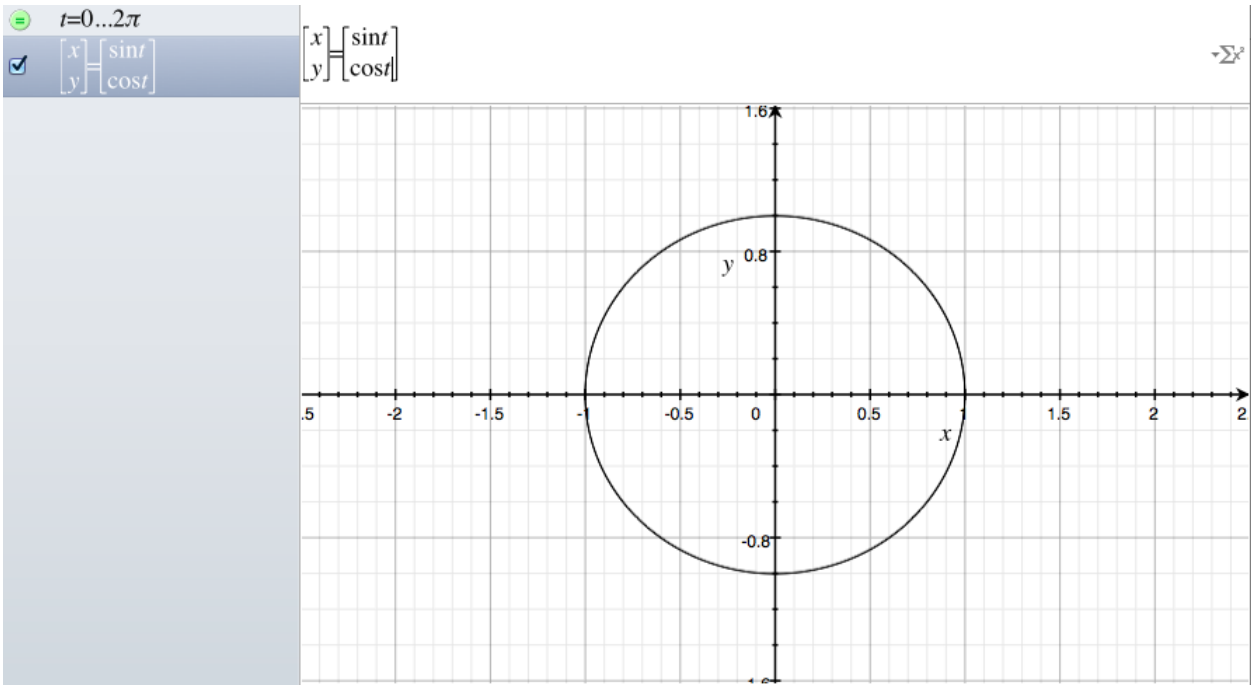
Using sine and cosine functions to graph a circle is a technique used in geometry called the unit circle--a circle with a radius of 1 unit. The traditional convention is to graph the unit circle and all derivatives of it by assigning the cosine function to the x axis and the sine function to the y axis, giving us $x = cost$ and $y = sint$. I'm not using the traditional approach to graphing the unit circle for two reasons: the first is that it sets the start point for any given pattern at the far right side of the x axis, leading to shapes like triquetras pointing to the right instead of up and I personally prefer the aesthetics of those patterns pointed up naturally. The second is much more subjective and it's that my dominant direction of spin is clockwise, whereas the traditional approach to graphing the unit circle leads to the pattern being built counter-clockwise.

If you're a traditionalist, you can just reverse the equations on the x and y axis to get results that will fit into the traditional convention for graphing curves of this sort. All of the conclusions reached in the course of this paper are valid whether this operation is performed or not.

2. Modeling Flowers and Other Simple Poi Patterns

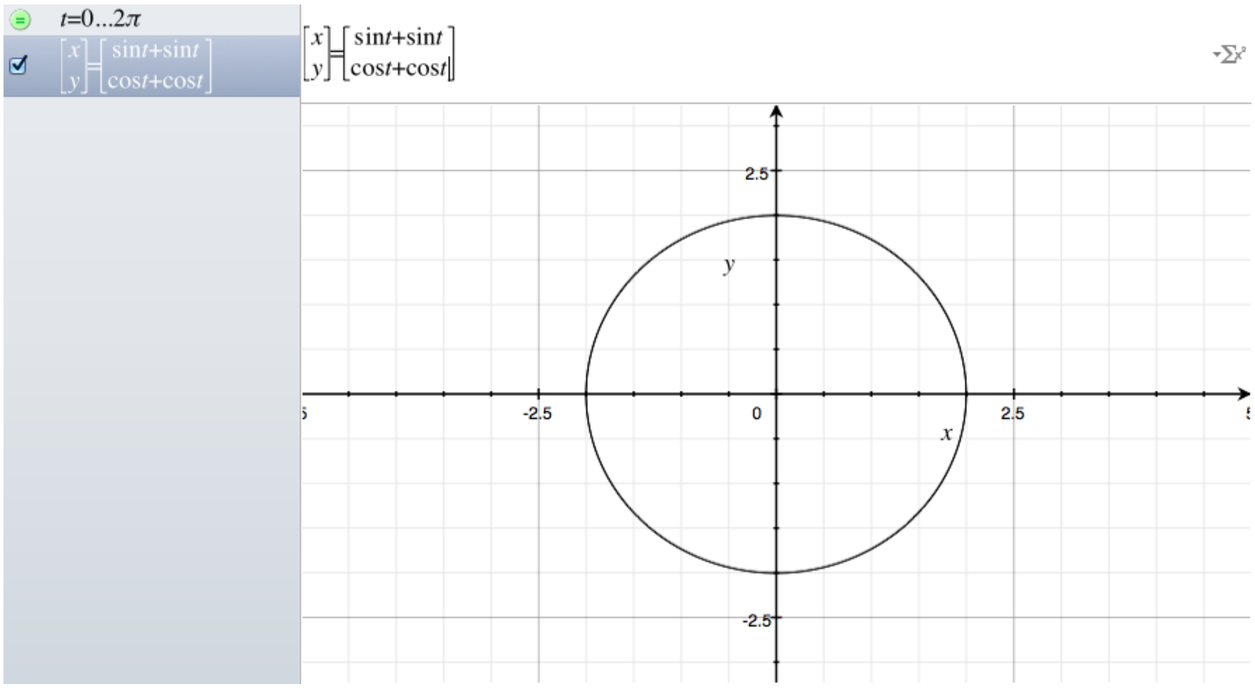
To be able to see how these functions relate to poi, it is helpful to be able to graph them to visualize the interactions of the different variables. This can be done with any graphing calculator or personal computer programs for graphing, including Grapher for Mac (found under

Applications => Utilities) or Microsoft Mathematics for PC (downloadable here). Most diagrams included from here on out will be screen captures of graphs produced in Grapher and will include the equation used to produce the graph as a visual aid. The first such graph can be found below:

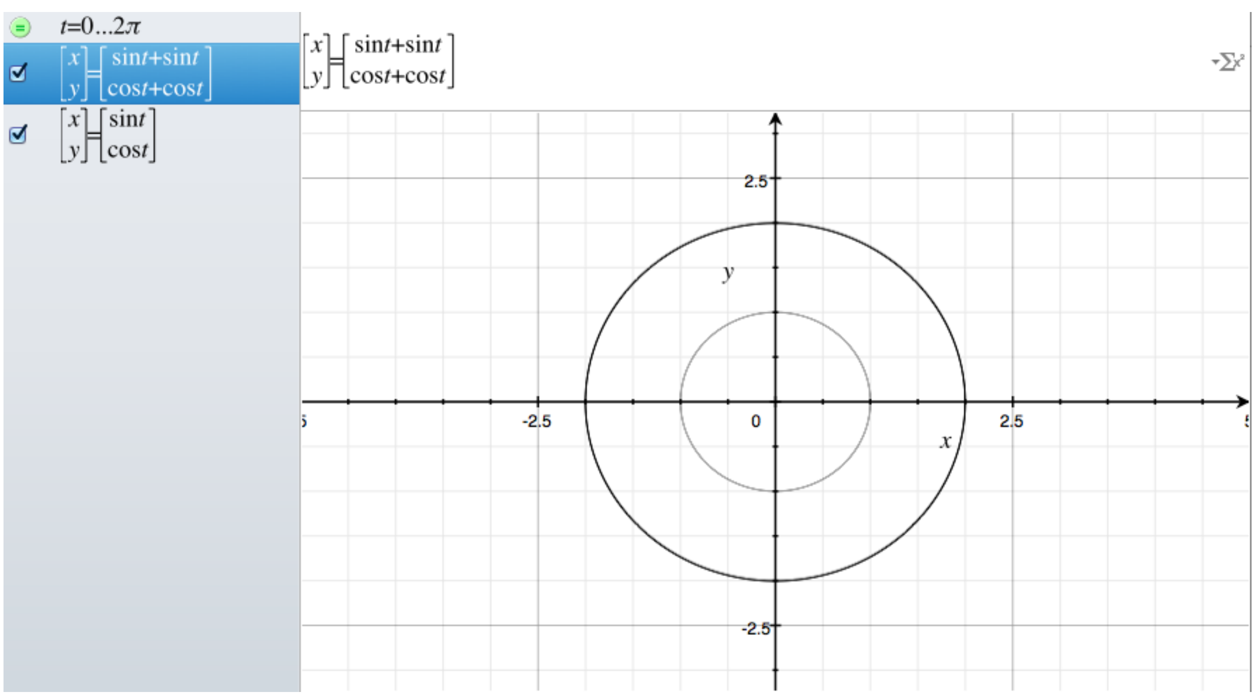


Above we have the graph of the equations we arrived at in the previous section: $x = sint$, $y = cost$ where t is equivalent to all values between 0 and 2π. First, ensure that if your calculator has one, that it is set to parametric mode. When we plug these equations into our graphing calculator, we should get the result shown above: a circle with a radius of 1. It is the first poi pattern we will model in this paper as it can be considered the pattern that is created by a single poi performing a static spin, or rotating around an unmoving hand. The hand and/or handle may be considered an infinitely small point at the very center of this graph. In real life, our hands do move a bit when performing static spin but the graph is meant to indicate an ideal state for this move--what we are hoping to achieve whether we are physically able to or not.

Now let's try a slightly more complex pattern, graphing $x = sint + sint$, $y = cost + cost$:



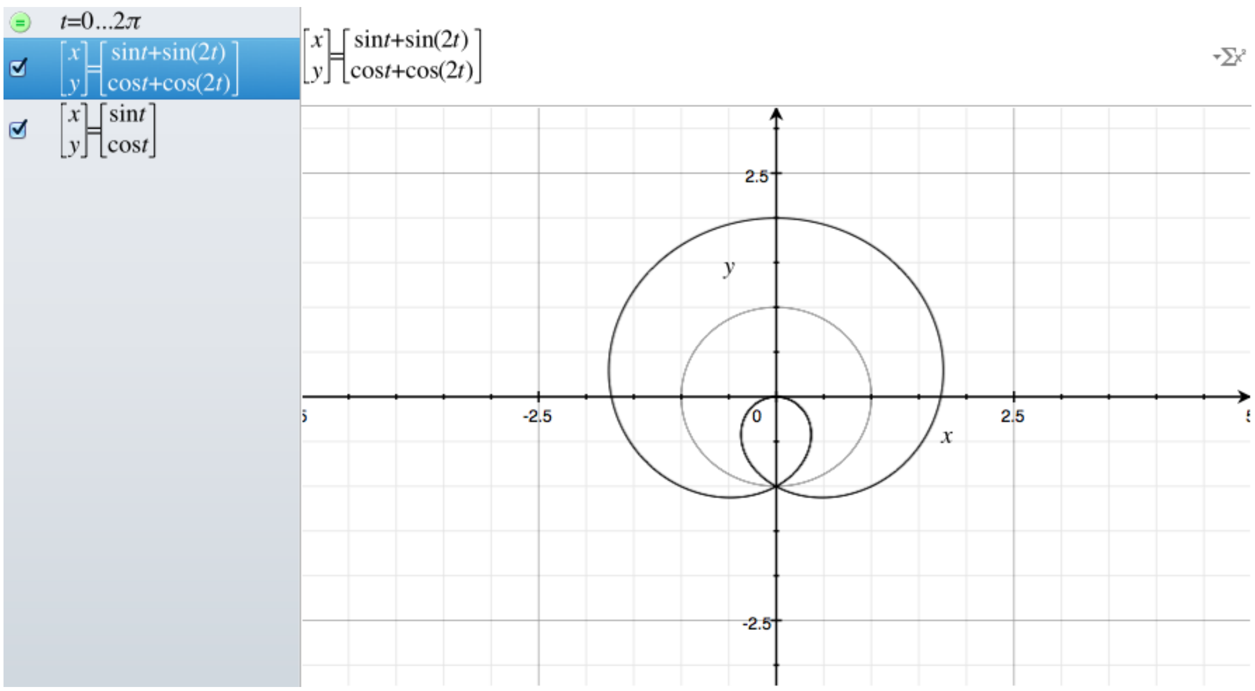
At first it may look as though this is the same graph, but looks can be deceiving. I have zoomed out of the graph to ensure it would completely fit on the screen. As a result, it may be easy to miss the fact that the radius of this circle is now 2 instead of 1, meaning that the circle is twice as far across as well as twice the circumference as the previous circle. So what does this have to do with poi patterns? Did the poi get twice as long? Not quite. Let me show you the same graph with one small addition:



Now you will note a smaller circle inside the bigger circle with the smaller circle being modeled with the same equation as the one we used in our first diagram. Here, the relationship between the two is easier to see--the bigger circle is intended to be the pattern of the poi as it traces an extension around a handpath that is one poi length in radius. In other words, we can think of the first pair of terms in the graph as describing the movement of the handpath and the second pair describing the movement of the poi as shown to the right:

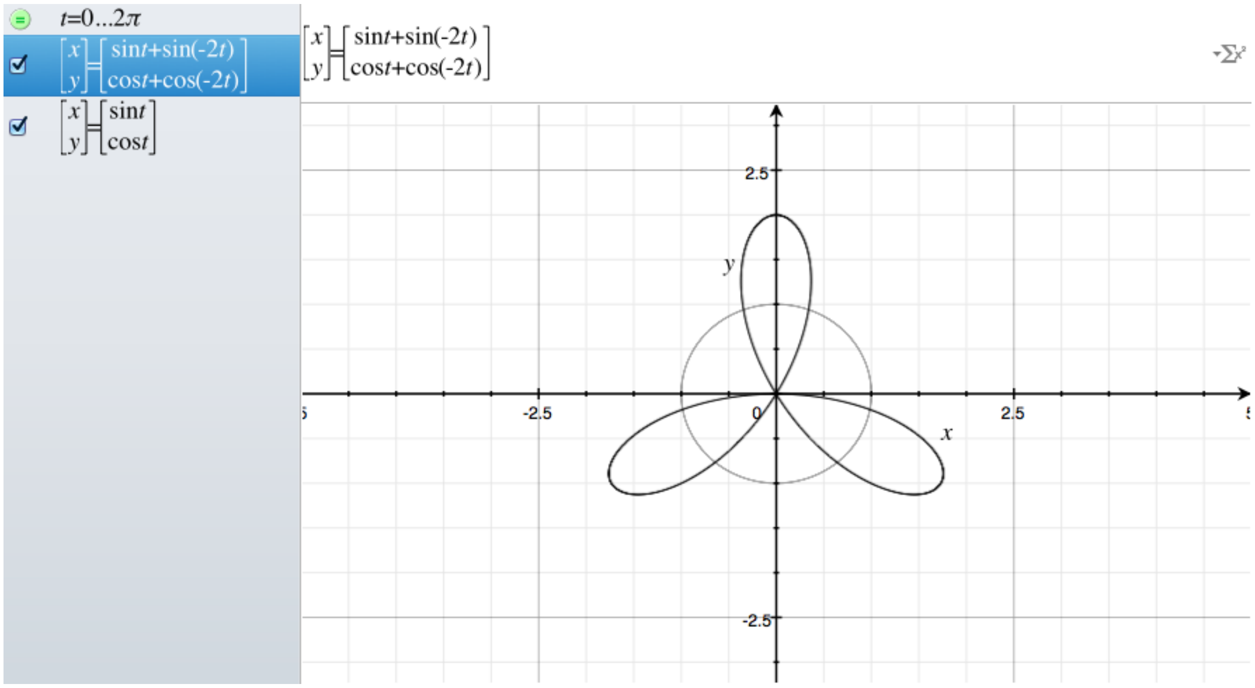
$$\begin{bmatrix}x\\y\end{bmatrix}=\begin{bmatrix}\overbrace{\boxed{\begin{matrix}\sin t\\ \cos t\end{matrix}}}^{\text{Movement of the hand}}+\overbrace{\boxed{\begin{matrix}\sin t\\ \cos t\end{matrix}}}^{\text{Movement of the poi}}\end{bmatrix}$$

With this relationship in mind, we can now model a more complex poi path--perhaps a flower using the equations $x = sint + sin(2t)$, $y = cost + cos(2t)$:

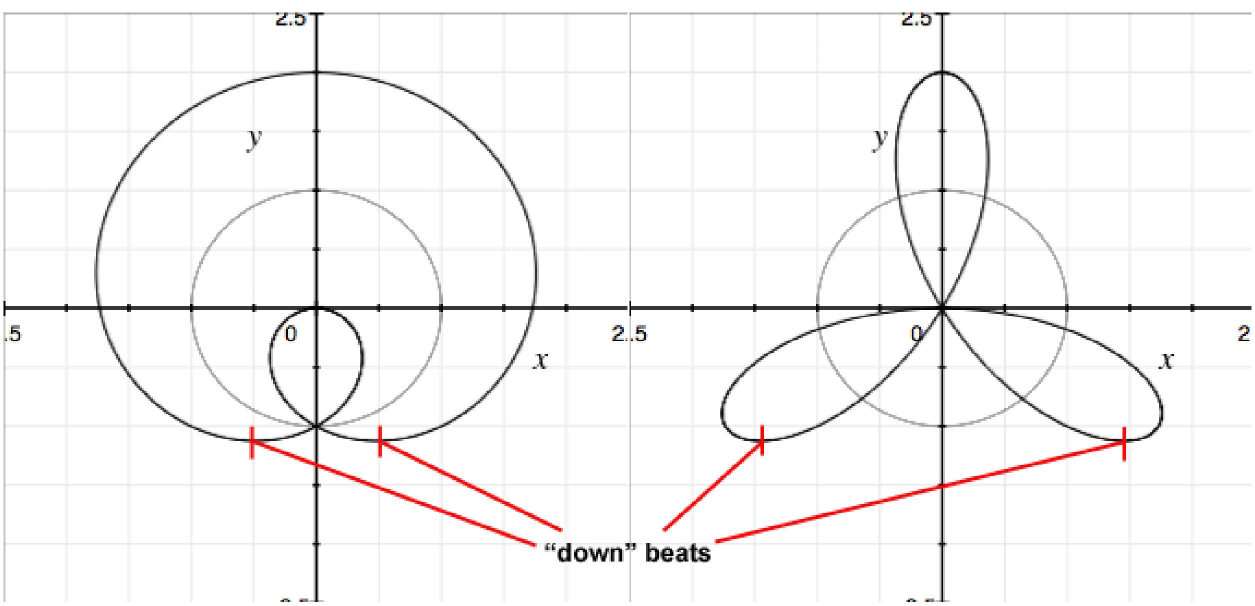


In this case, we can see we have graphed out a 1-petal inspin flower with a handpath that is still 1 poi length in radius--what changed? As you can see from the equation, the terms that govern the movement of the poi have been adjusted such that we are now multiplying t by 2 before we find the sine value of it. This means that the terms governing the movement of the poi are now completing 2 circles for every 1 circle the handpath terms are completing. But how can we be completing 2 circles while getting 1 petal? The method for graphing we are using is based upon Cartesian coordinates, wherein there is an “up” and a “down” (I know I am overgeneralizing this and I apologize). The poi in this pattern completes two “down” beats for every one “down” beat that the hand completes, thus giving rise to the pattern.

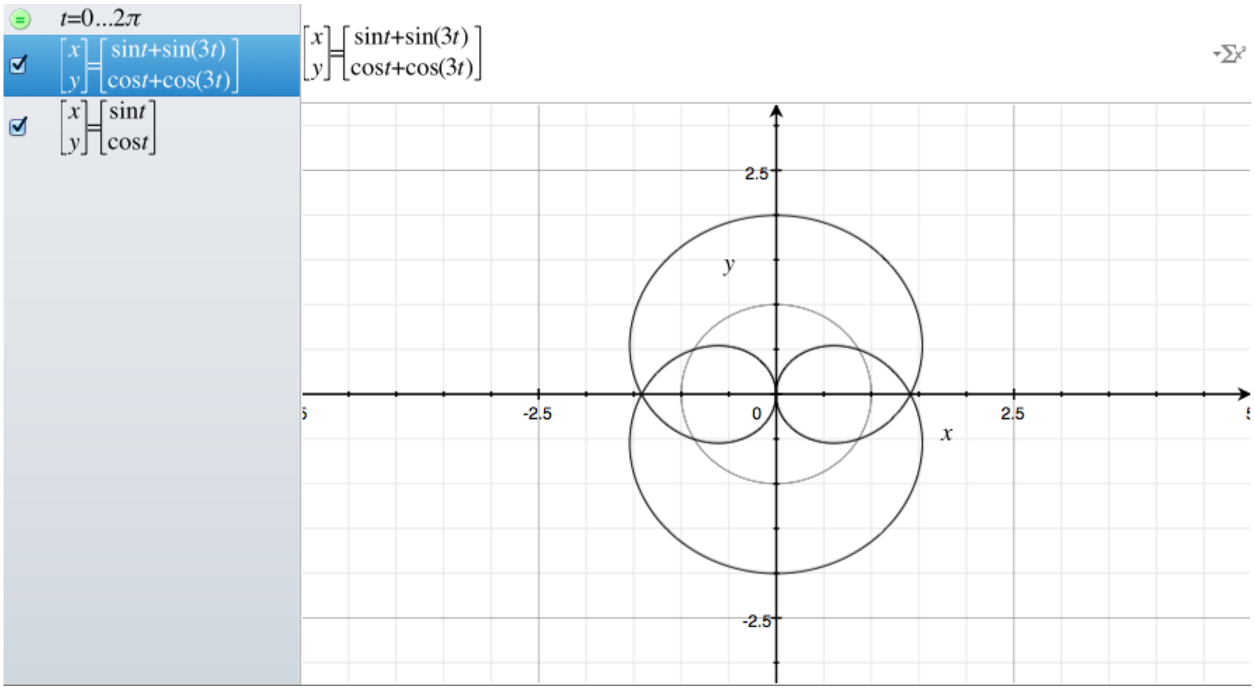
Let’s see what happens if the poi terms incorporate a -2 instead of a 2, resulting in the equations $x = sint + sin(-2t)$, $y = cost + cos(-2t)$:



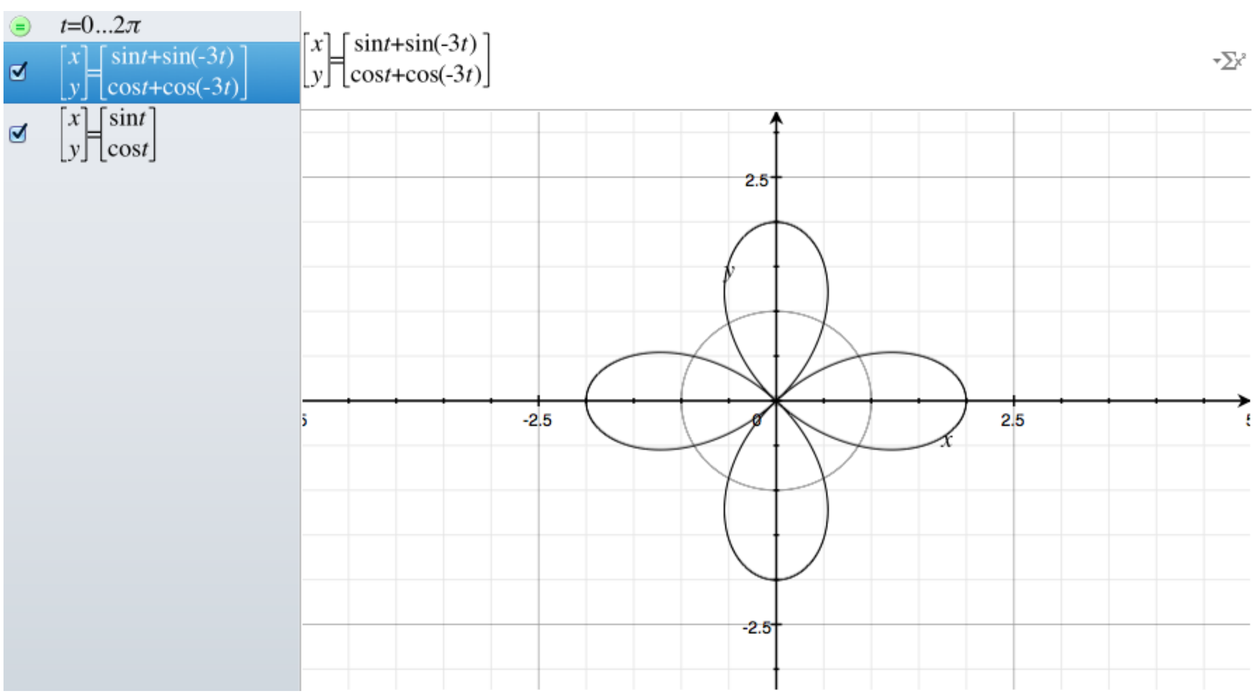
This results in a vastly different looking pattern, but still one wherein the poi has 2 “down” beats for every “down” beat the hand completes. You can see this clearly outlined below:



We would identify the pattern on the right as being a 3-petalled antispin flower, or colloquially a *triquetra*. A pattern should already be apparent from these two graphs: when t is multiplied by a positive number in the poi terms, the resulting graph will show an inspin pattern. When t is multiplied by a negative number in the poi terms, the resulting graph will show an antispin pattern. Let's verify this prediction by graphing $x = sint + sin(3t)$, $y = cost + cos(3t)$:

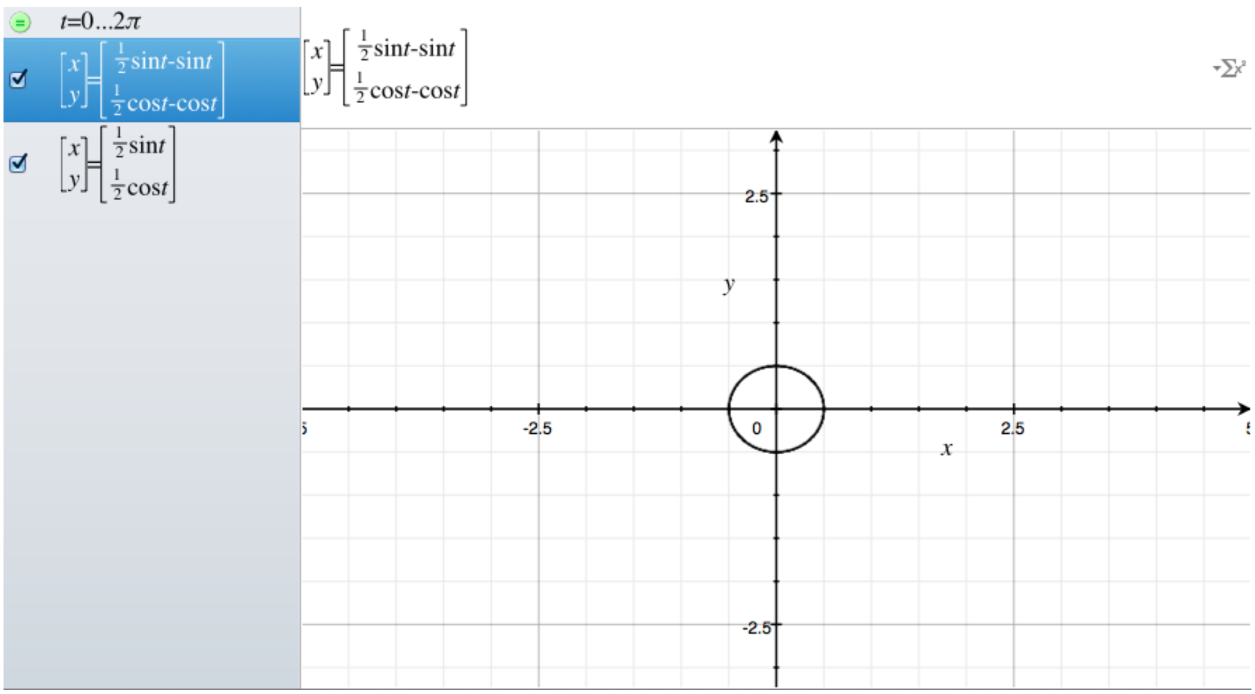


And $x = sint + sin(-3t)$, $y = cost + cos(-3t)$:

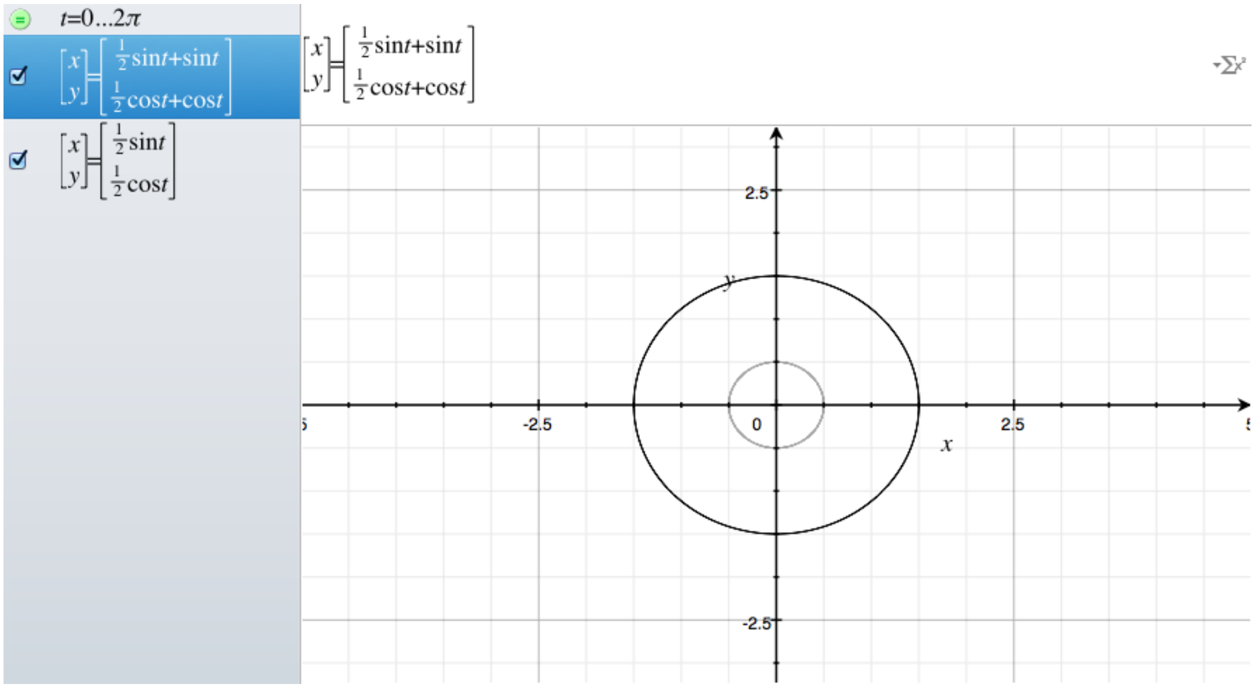


The prediction has been verified for these two cases! When we multiply t by 3 in the poi terms, the resulting graph is the pattern for a 2-petal inspin flower while multiplying t by -3 in the poi terms results in the pattern for a 4-petal antispin flower. You may have noticed that if we replace the number we multiply t by in the poi terms with the variable d ("down" beats), resulting in the following equation: $x = sint + sin(dt)$, $y = cost + cos(dt)$, we can then predict the number of petals p that will result with the following equation: $p = |1 - d|$. For d=-2, we can see that p=3 and for d=3 we can see that p=2. The equation works!

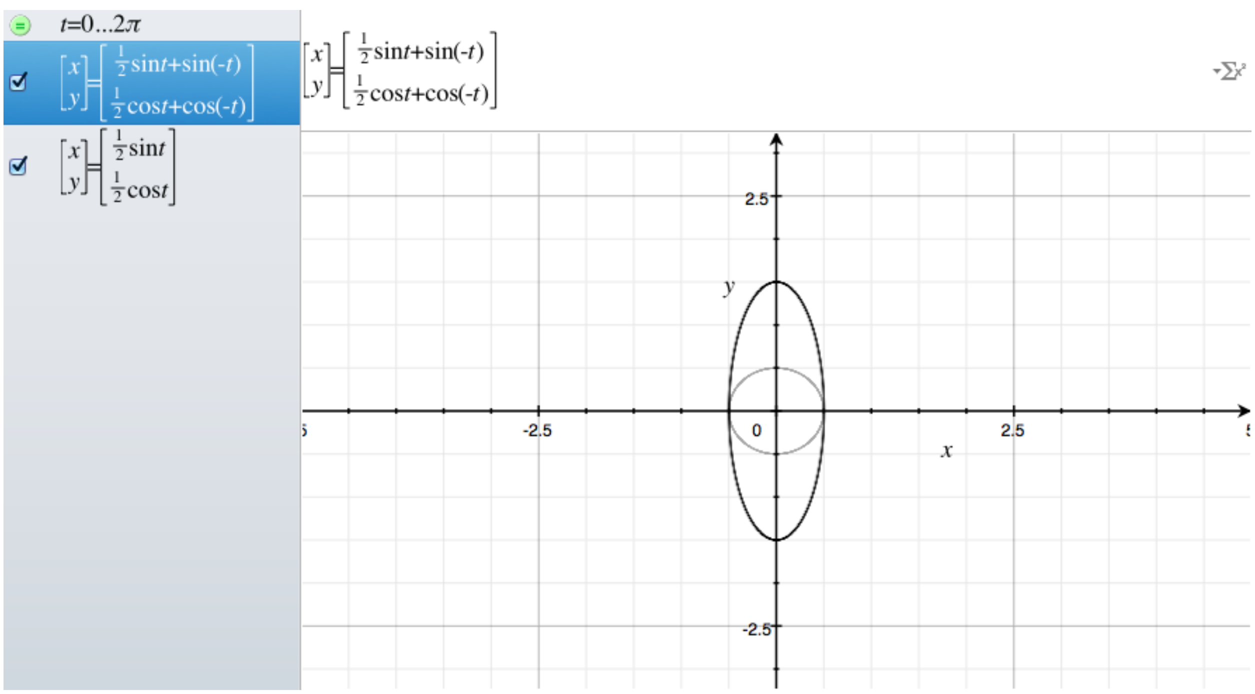
But what about if we want to model patterns that fit into the poi unit circle family--that is, poi patterns in which the handpath is half a poi length in radius rather than a full poi length in radius? We will model one such pattern, an isolation, with the following equation: $x = \frac{1}{2}sint - sint$, $y = \frac{1}{2}cost - cost$.



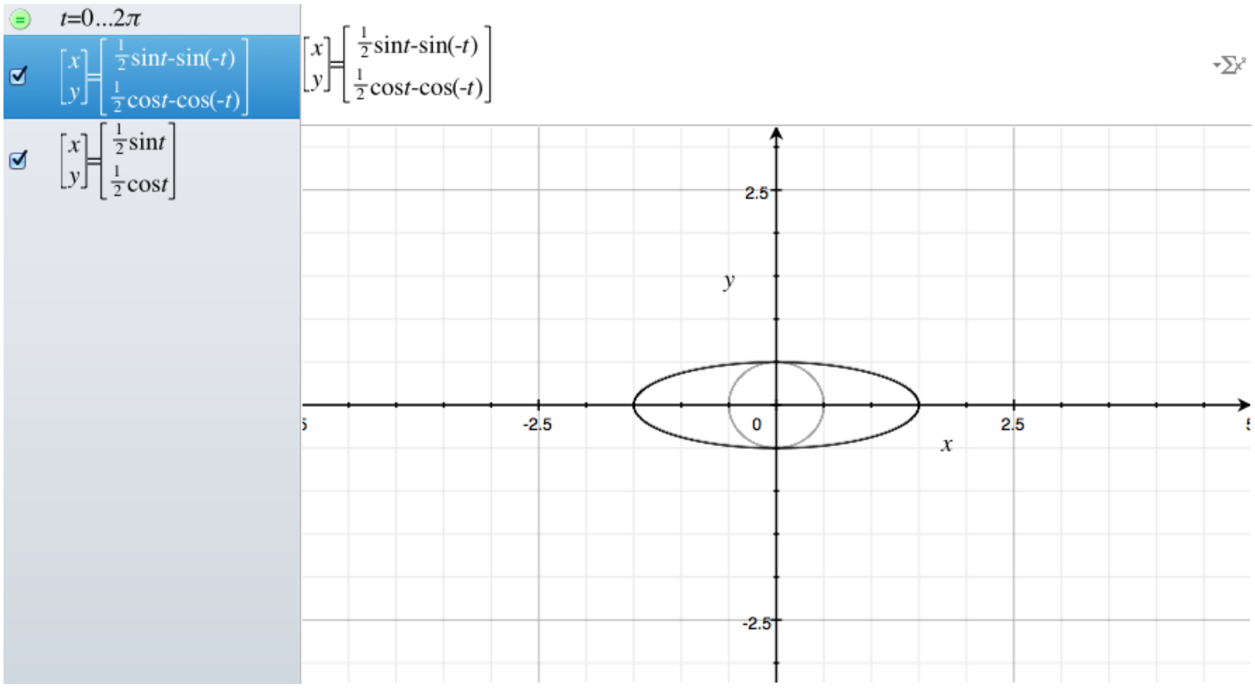
I've opted not to zoom into this graph to provide a proper sense of scale. For this equation, we do indeed find that now the radius of the circle produced is ½ the poi length. Even more interesting, the handpath and poi path graphs are now the same graph because the poi terms have been set to the opposite end of the circle by making both the poi terms negative terms instead of positive terms. We can switch this to the graph of a unit circle extension by switching the poi terms to positive, making the equations $x = \frac{1}{2}sint + sint$, $y = \frac{1}{2}cost + cost$:



Now we can see that the graph of poi terms describes a circle 1½ poi lengths in radius with a handpath in the center that has a ½ poi length radius. Now what about if we wanted to include *cateyes*? Since cateyes are antispin patterns, we will have to multiply t in the poi terms by a negative number: -1. The equation for a cateye, then, will be $x = \frac{1}{2}sint + sin(-t)$, $y = \frac{1}{2}cost + cos(-t)$. The graph for this pattern is below:



If you are curious, you can also model a horizontal cateye with the following equation:
$x = \frac{1}{2}sint - sin(-t)$, $y = \frac{1}{2}cost - cos(-t)$.



With these examples, we can then state with confidence that we can describe all flowers and unit circle patterns with the following equation:

$$x = (h \cdot sint) + p(sin(dt))$$
$$y = (h \cdot cost) + p(cos(dt))$$

Where h is the radius of the handpath, p is the phasing of the poi at the beginning of the pattern, and d is the number of downbeats of the poi relative to the handpath's downbeats. So given the examples we have above, we could plug the following values in for the variables:

Triquetra:	**Isolation:**	**2-petal Inspin:**	**Cateye:**	**Extension:**
$h = 1$	$h = ½$	$h = 1$	$h = ½$	$h = 1$
$p = 1$	$p = -1$	$p = 1$	$p = 1$	$p = 1$
$d = -2$	$d = 1$	$d = 3$	$d = -1$	$d = 1$

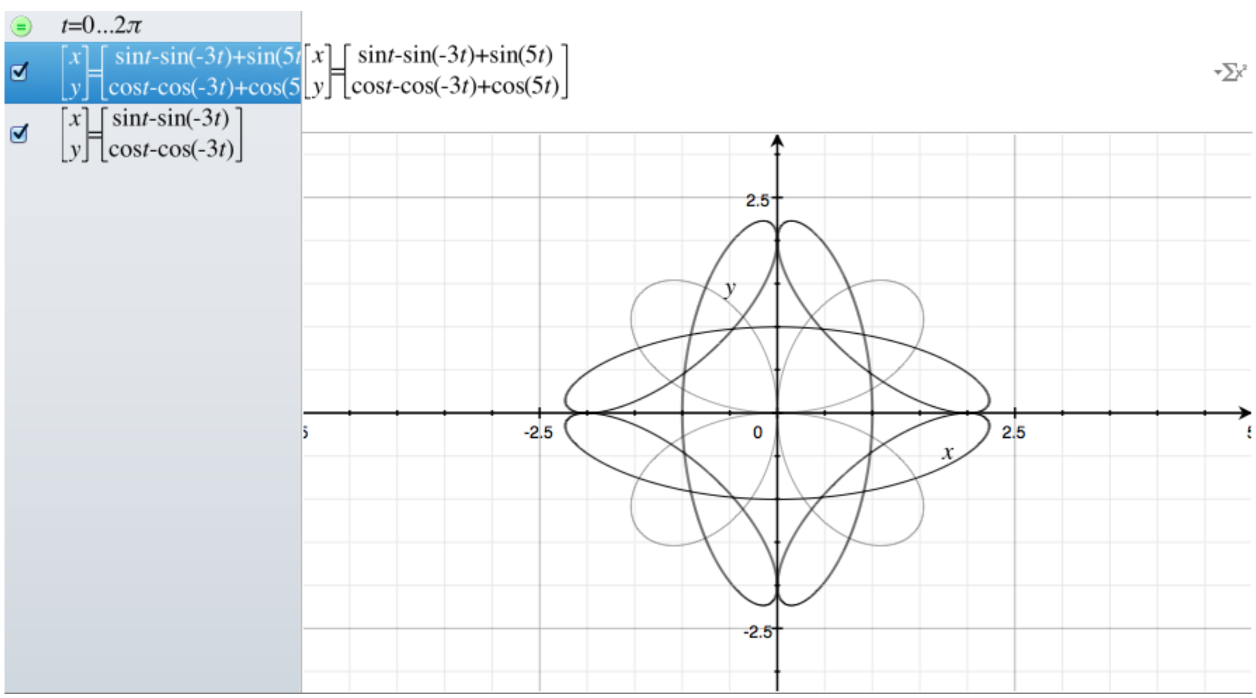
This in and of itself is enough of a model to describe most 2D spinning that poi performers engage in. But it is still limited in that it does not describe either third-order motions or any form of 3D spinning. We can expand our equation, however, to make these two types of movement possible to model. We will start with third-order motions because they can be modeled in a 2D environment as well.

3. Modeling Third-Order Motions

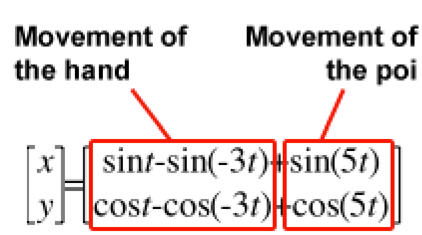
Third-order motions are an extension of flowers that were first named by Damien Boisbouvier[1] after analysis of a number of patterns including Zan's Diamond. Where flowers consist of two centers of rotation (the center of the handpath and the handle of the poi), third-order motions are generated by adding a third center of rotation. This is usually accomplished in two ways:

1. By treating the elbow as the center of the handle's rotation with the shoulder as the center of the elbow's rotation
2. Having the hand trace around the outside of an imaginary pattern with two centers of rotation, such that these first two centers are virtual rather than explicitly seen in the pattern.

Either way, we already possess the tools to model this type of movement by extrapolating what we already know about flowers. Since we now have three centers of rotation, we can add an additional term to our equation to accomplish this task. For example, we can graph Zan's Diamond with the following equation: $x = sint - sin(-3t) + sin(5t)$, $y = cost - cos(-3t) + cos(5t)$.

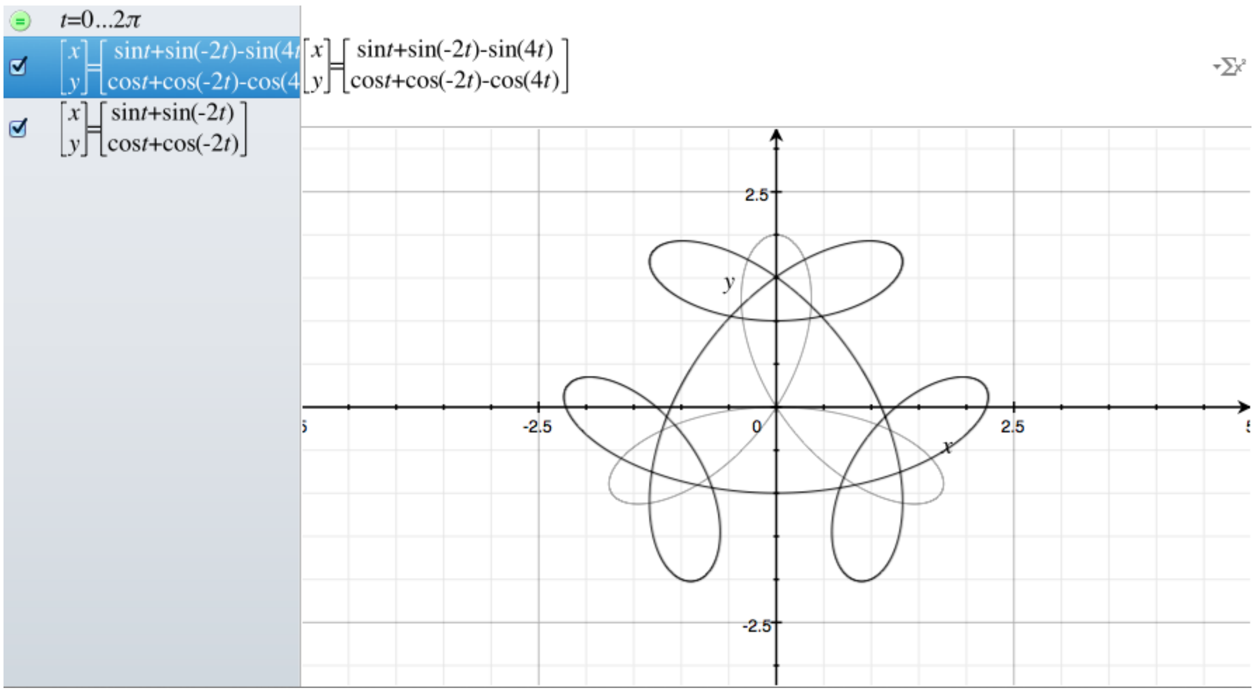


Again, I have also graphed the handpath so its relationship with the poi pattern is clear. In this case, the handpath follows the general pattern of a 4-petal antispin flower, with the poi adding



1 http://www.homeofpoi.com/community/ubbthreads.php/topics/92 ml#Post920391

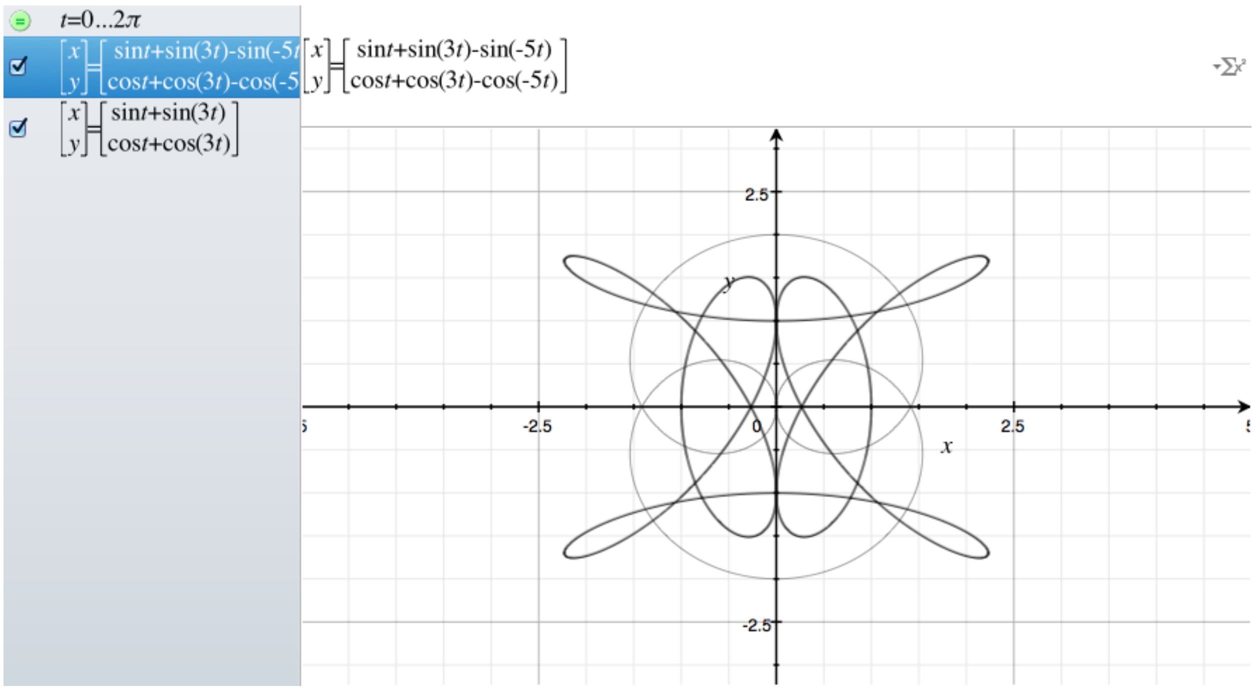
two petals for every one petal the handpath leaves. It is important to note that in this update to our equation, the terms governing the handpath and poi movement have changed slightly. While the last set of terms still governs the poi, now the first two sets of terms govern the hand instead of just the first one as outlined in the diagram on the side. We can see another example of this type of pattern with what has been dubbed a “fractal” flower and in graphing it we will see why the term applies. For this graph, we use the equation $x = sint + sin(-2t) - sin(4t)$, $y = cost + cos(-2t) - cos(4t)$.



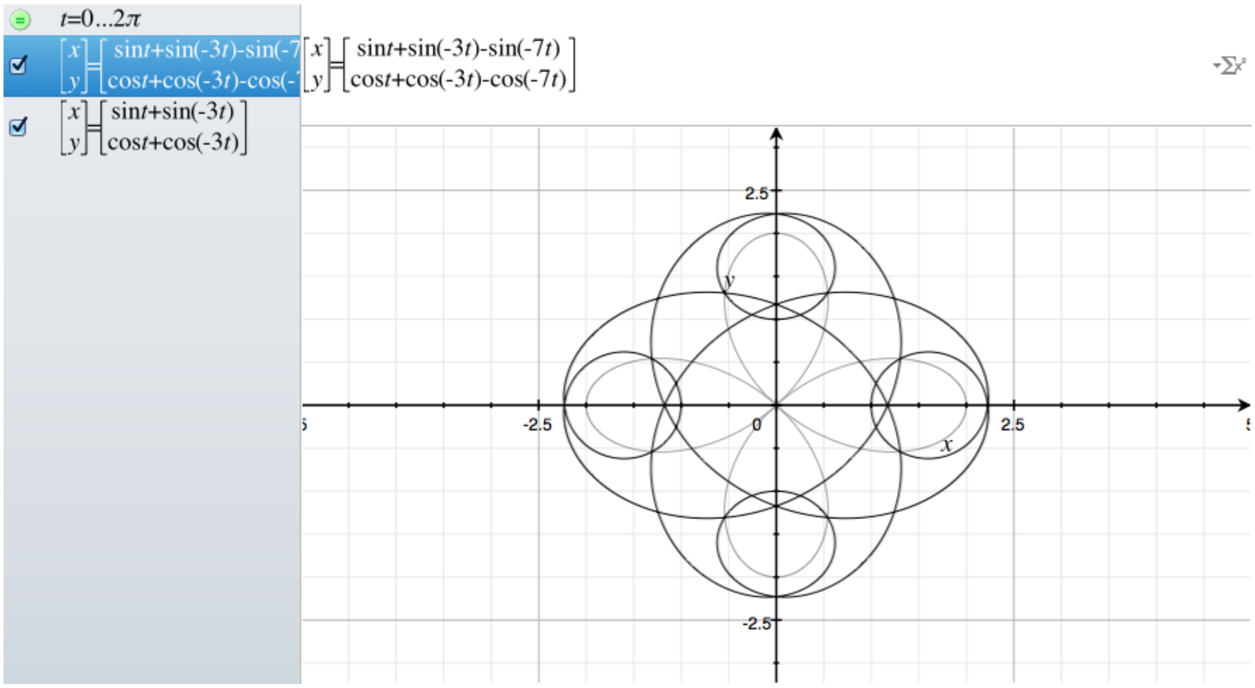
Here, we can see that the core handpath of the pattern is a Triquetra with a 6-petal pattern overlaid on top of it. If we imagine that the first set of terms multiplies its t variable by 1, then the relationship between the multipliers of the first two t’s comes out to 1:-2 or if we were to divide them, -½. Likewise, because the relationship between the t multipliers in the last term and the second is -2:4, we can divide them to arrive at a proportion of -½. In other words, the number of downbeats added between the first and second term is proportionally the same as that added between the second and third term. Because this pattern demonstrates a limited degree of recursion in its structure, we refer to it as a fractal flower after the concept of a mathematical set that displays self-recursion.

Both Zan’s Diamond as well as the fractal flower above also demonstrate a common phenomenon in performing third-order motions: adding two antispin petals for every petal encountered in the handpath. I will label this family of third-order motions “triquetra expansions” as they can also be thought of as adding one triquetra per petal. I will give formulas for both triquetra expansions as well as fractal flowers at the end of this section.

Thus far, all the patterns we have seen are examples of what would be called antispin-antispin third-order motions. That is, the handpath displays the patterns we saw above in antispin flowers and the direction of the poi is opposite that of the hand. This framework also predicts inspin-antispin, antispin-inspin, and inspin-inspin third-order motions. Though these latter three types are rarely performed, I will nonetheless provide modeled examples of each, starting with inspin-antispin: $x = sint + sin\,(3t) - sin\,(-5t)$, $y = cost + cos\,(3t) - cos\,(-5t)$.

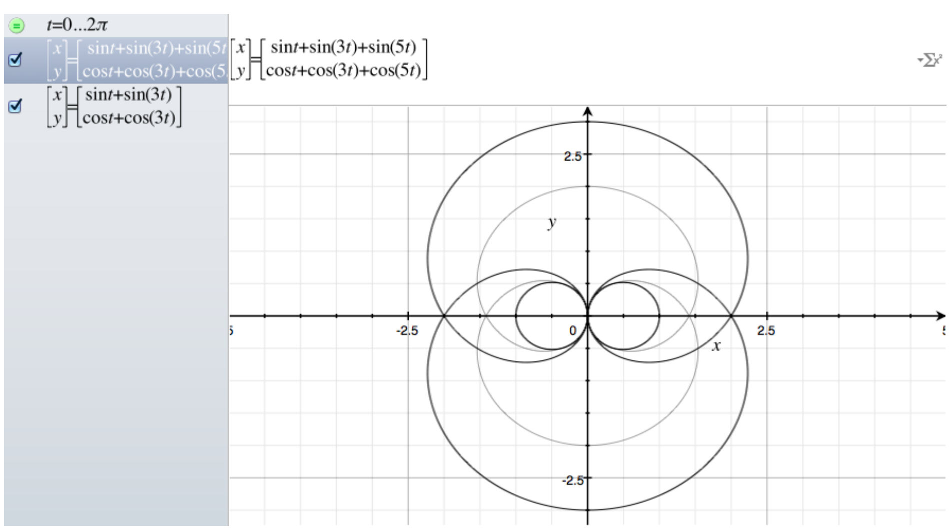


This pattern grafts a triquetra expansion on a 2-petal inspin handpath. For an antispin-inspin, we will use the equation $x = sint + sin\,(-3t) - sin\,(-7t)$, $y = cost + cos\,(-3t) - cos\,(-7t)$.



This pattern results in a core handpath pattern of a 3-petal antispin flower with a 1-petal inspin grafted onto each antispin petal. You may note that the third t multiplier is a negative number just like the second t multiplier--this doesn't mean that we've lost the antispin-inspin pattern we were shooting for, only that I had to do a little bit of mathematical wrangling to ensure proper petal placing on the poi pattern. Another approach to doing the same operation would be to add pi to the multiplier of last set of t terms, such that: $x = sint + sin(-3t) - sin(7t + \pi)$, $y = cost + cos(-3t) - cos(7t + \pi)$

Finally, inspin-inspin can be executed with the equation $x = sint + sin(3t) + sin(5t)$, $y = cost + cos(3t) + cos(5t)$:



Here, we've added one inspin petal for every inspin petal of a 2-petal inspin handpath. With several of these patterns now presented, we can assemble an equation with enough variable assignments to cover everything we can do with third-order motions. That equation is presented below:

$$x = (h \cdot sint) + p\,(sin\,(dt)) + q\,(sin\,(ft))$$
$$y = (h \cdot cost) + p\,(cos\,(dt)) + q\,(cos\,(ft))$$

Such that h is the radius of the first center of rotation (either the shoulder or imaginary), p is the phasing of the second center of rotation (handpath), d is the number of downbeats the handpath possesses, q is the phasing of the poi pattern, and f is the number of downbeats in the poi pattern. That's a lot of variables to keep track of! As mentioned above, here are some recipes both for triquetra expansions and fractal flowers:

For triquetra expansions of antispin-antispin third-order motions, $p = \pm 1$, $d \leq -1$, $q = -p$, $f = |d| + 2$

For fractal flowers in antispin-antispin, $p = \pm 1$, $d \leq -1$, $q = \pm 1$, $f = d^2$. Please note that values for p and q may need to be tried in different combinations until the desired aesthetic effect is reached for the resulting pattern.

We now have an equation that satisfies the requirements for all flowers (where $h = 0$ and $d = 1$) and all third-order motions. We still have yet to see both some of the most basic poi patterns and some of the most advanced patterns as well, including weaves and toroids. For these, we must add additional values for movement in the z-axis for movements that require depth.

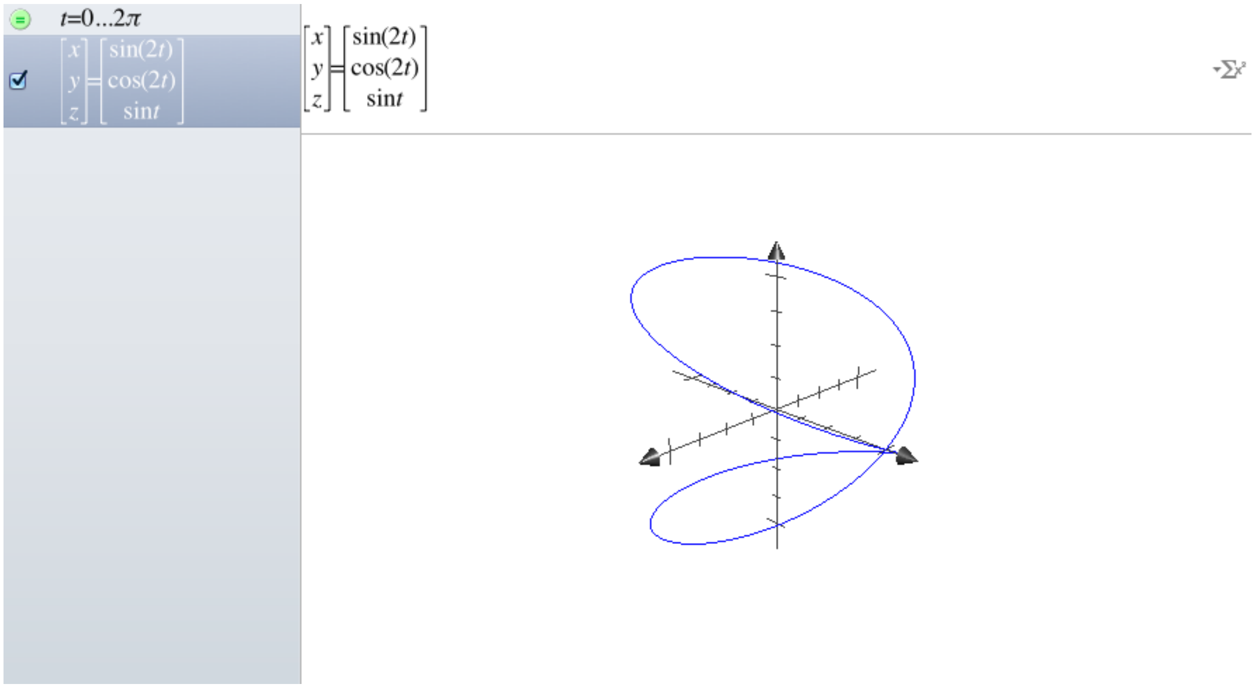
4. Modeling Simple Weaves

Most if not all 3D poi patterns that do not utilize plane breaks (more on those later) do utilize plane bends to attain their shapes. I am indebted to Alien Jon for the idea of christening this family of moves manifolds after the topological concept to which they all adhere.[2] This family of movement contains, but is not limited to weaves, thread the needles, inside moves (inversions, introversions, lovelaces, barrel rolls, etc), body tracers, and toroids. To begin to model these movements, we must add a z-dimension to our equations to begin to process them in terms of depth. As we examine these shapes, please note that as with our flowers, these patterns are being modeled agnostic of real-world coordinates such that x, y, and z axes are relative to the desired coordinates of the performer. x could just as easily be up-down as it could be side-side or even be measured along a diagonal axis.

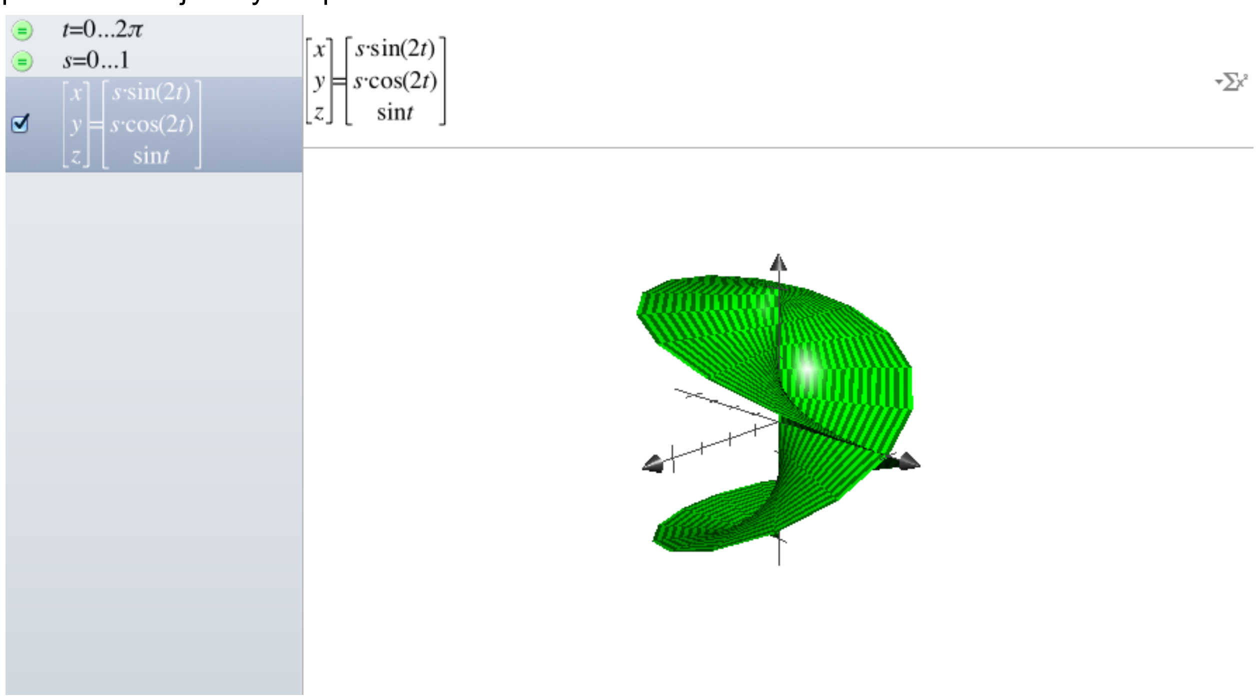
Let us begin with the simplest of all manifolds: the 2 beat weave. Functionally, modeling this

[2] http://en.wikipedia.org/wiki/Manifolds

move is identical to modeling one hand's performing the standard 2 beat weave, thread the needle, windmill, crosser, hip reel, and indeed any 3-dimensional 2 beat move with a single cross point. To model this, we use the equation $x = sin\,(2t)$, $y = cos\,(2t)$, $z = sint$.

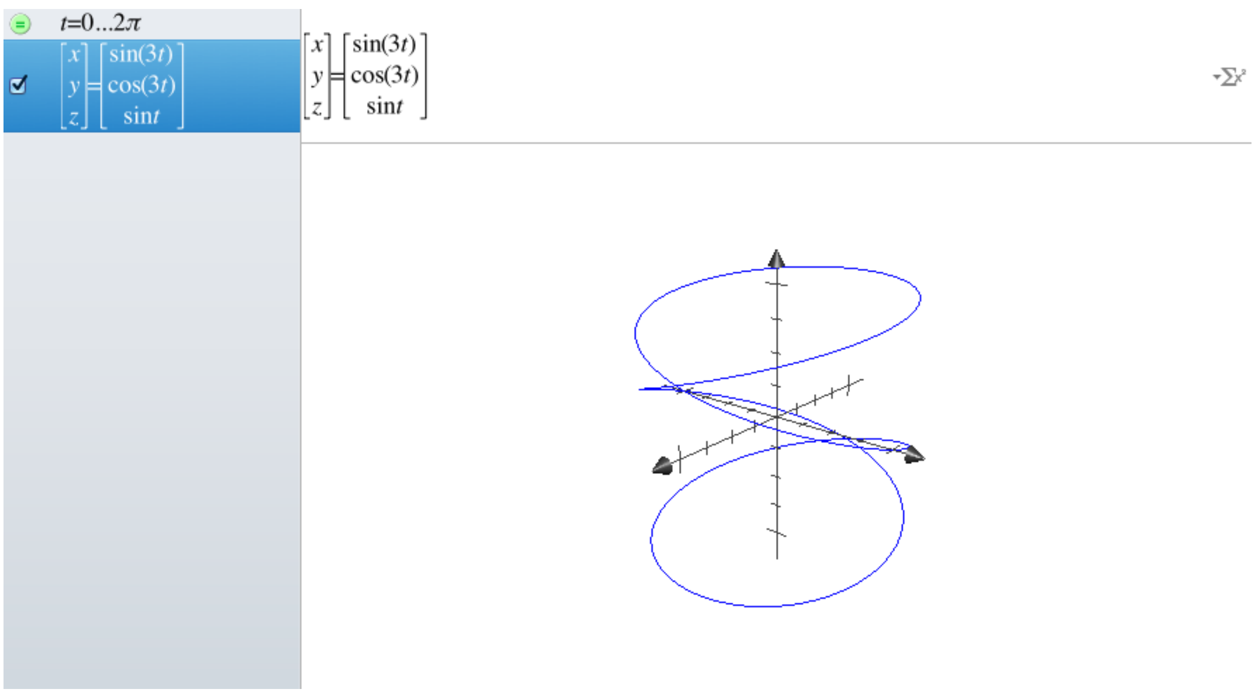


This results in a pattern in which the poi oscillates slightly different in two axes than it does in the third one. To better see this illustrated, I've tweaked the graph just slightly to see the shape produced not just by the poi head but also the tether in three dimensions.



Here we see the behavior of the poi is a plane that intersects and twists upon itself, producing what topologists refer to as a self-intersecting plane or real projective plane.[3] To tease out what is going on in this equation, let’s take the pieces we’re already familiar with: in $x = sin(2t)$, $y = cos(2t)$. We have an equation that theoretically should yield a circle with radius 1, we just happen to be drawing it twice. By adding $z = sint$ we are saying that for every time the poi returns to its starting point in the z axis, it has returned to its starting point twice in both the x and y axes, thus completing two “beats” for every one “beat” on the z axis. The hand path here can be understood as a straight line back and forth across the z axis.

But what about a more complex weave, like say a 3-beat weave? It is actually easier to achieve than you may think. All we have to do is change the 2 multiplier on the x and y axis to 3 instead, yielding this equation: $x = sin(3t)$, $y = cos(3t)$, $z = sint$.



Here we have the poi completing 3 “beats” in the x and y axis in the time it can return to its starting point on the z axis. Once again, the handpath has been simplified down to a straight line in very much the same way that we simplified the handpath of a static spin down to an infinitely small point. We can extrapolate this same pattern out to create 5 and 7 beat weaves easily. The equation we would use to describe these simple weaves, then, would look like this:

$x = sin(dt)$
$y = cos(dt)$

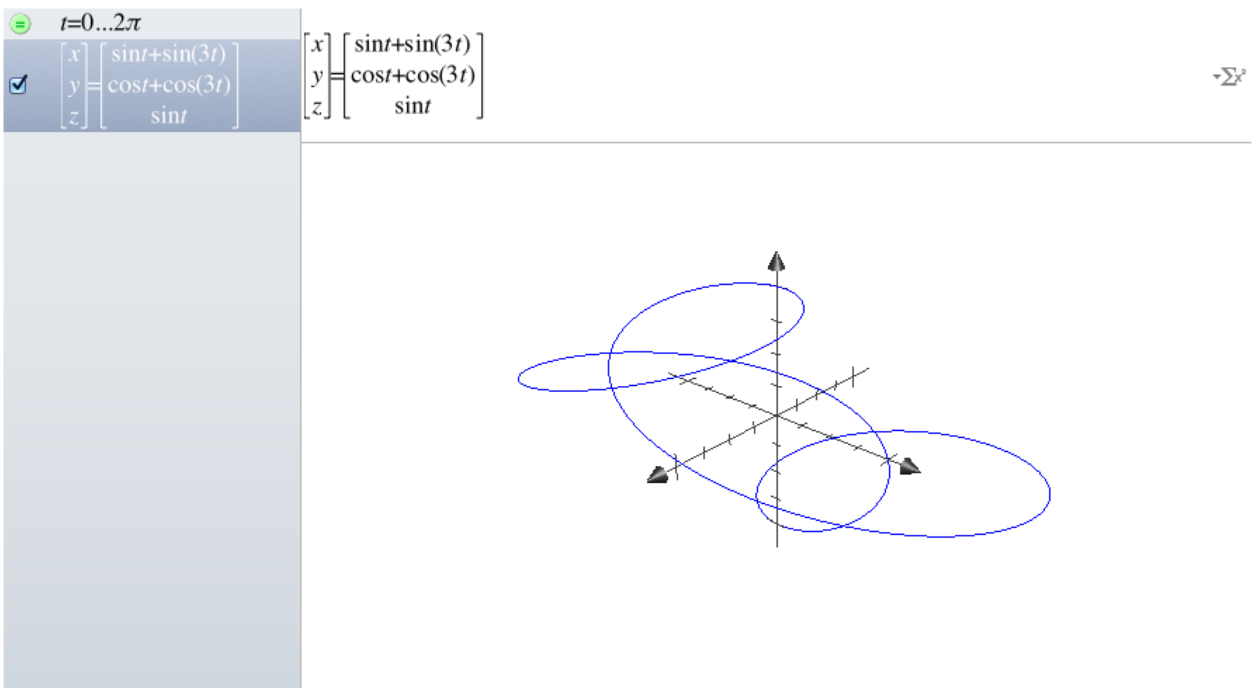
[3] http://en.wikipedia.org/wiki/Real_projective_plane

$z = sint$

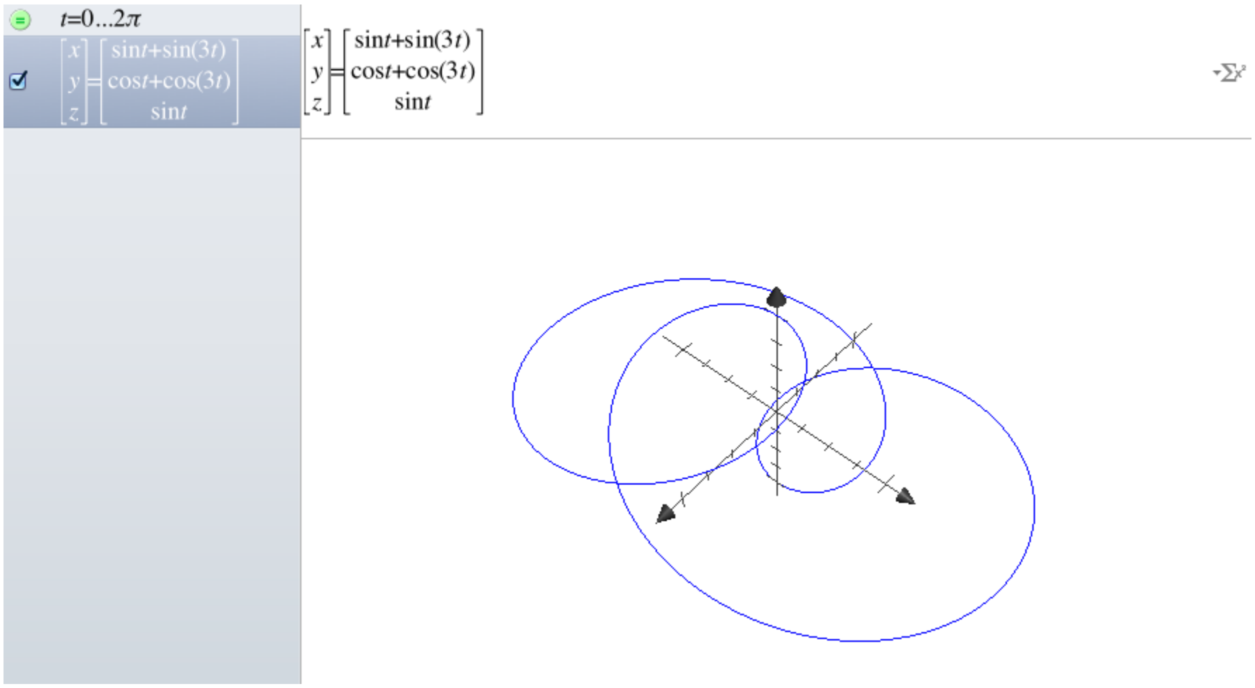
Where d is the number of beats in the weave we are trying to model. Now again bear in mind, this approach to modeling is totally agnostic of orientation to the body, so 5 beat weaves can come out looking identical to lovelaces and barrel rolls. A more comprehensive approach that includes the body will be needed to properly sort out all these subtypes of simple weave manifolds.

5. Modeling Body Tracers

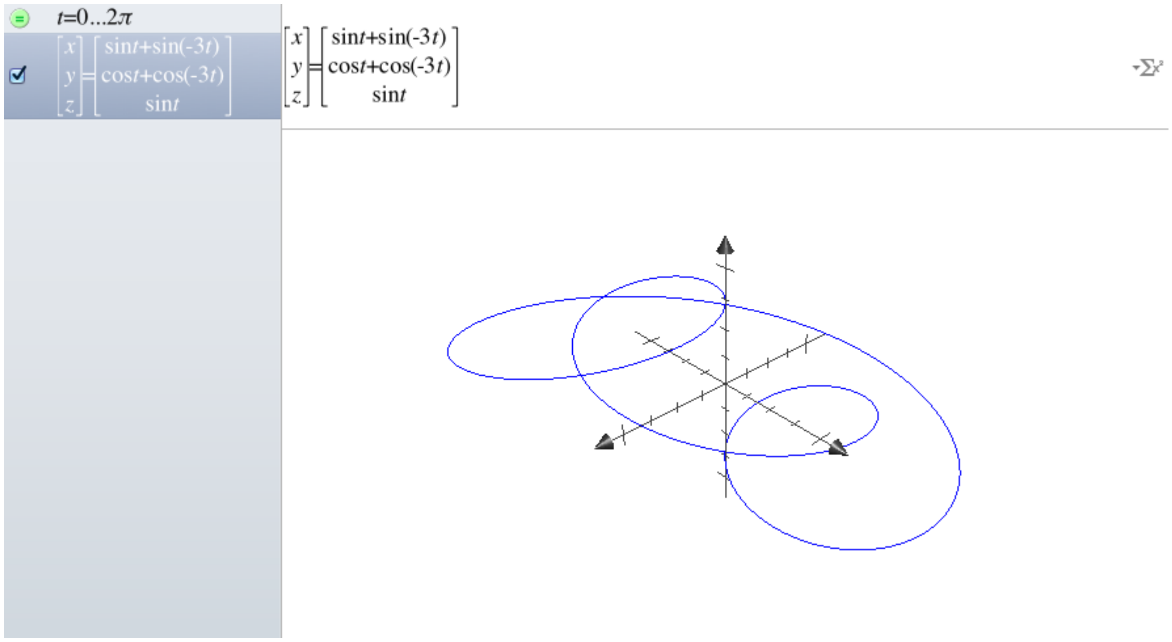
To make matters slightly more complex, however, we can model body tracers. These are flowers that do deviate along the z axis such that a petal or two can be placed behind the body. One simple example can be modeled using the equation $x = sint + sin(3t)$, $y = cost + cos(3t)$, $z = sint$:



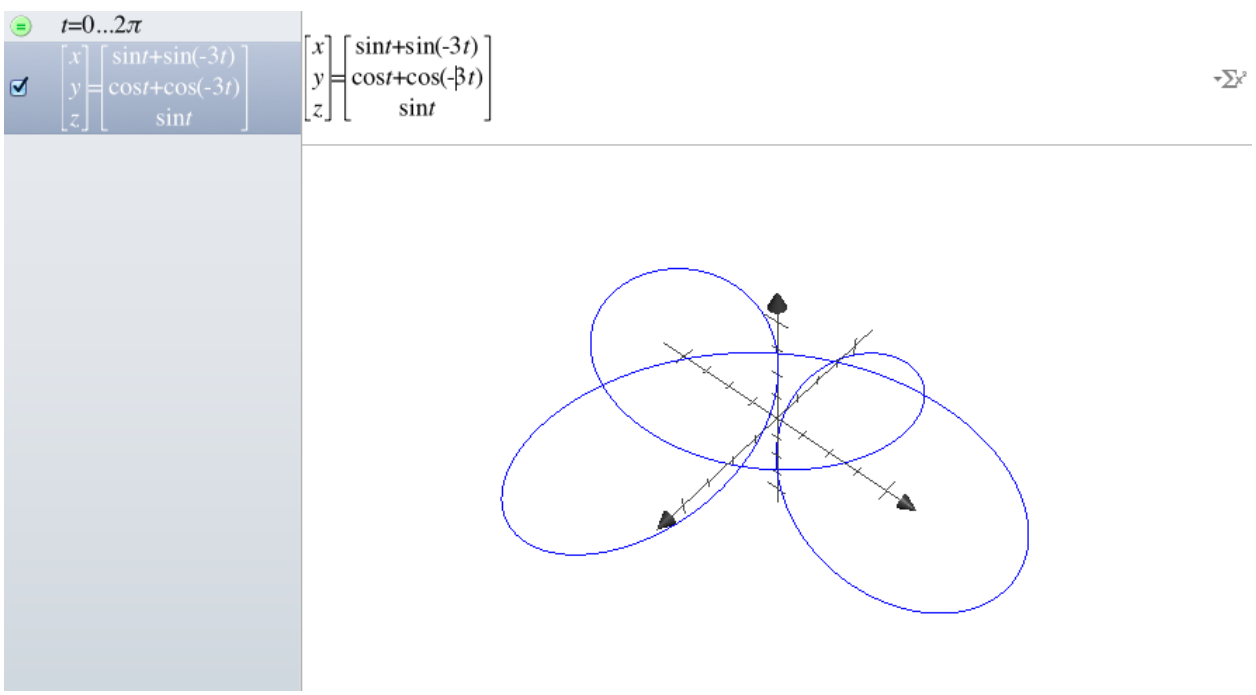
It may be difficult to tell from this angle, but this is actually a 2-petal inspin flower with one petal high on the top/down axis and one petal low down on this same axis. To aid with identification, I have added the following view from slightly above this one:



This will result in a 2-petal inspin flower in which one of those petals will be behind the performer and the other petal in front of the performer. It could also be a flower in which one petal is on their right side and the other is on their left side, or even perhaps one above their head and one below their shoulders. Either way, the performer is achieving not only a 2-petal inspin flower, but also varying the flower's depth in space to vary the placement of the petals around their body. We can achieve similar results by multiplying t in the poi terms by -3 instead of 3 to get a 4-petal antispin body tracer using the equation $x = sint + sin(-3t)$, $y = cost + cos(-3t)$, $z = sint$.



Again, the perspective is a little confusing. The top petal of the flower is the bit that is farthest to the left of the image while the bottom petal is the loop that crosses the arrow pointing to the lower right corner. Here it is from slightly above to clarify the shape slightly:



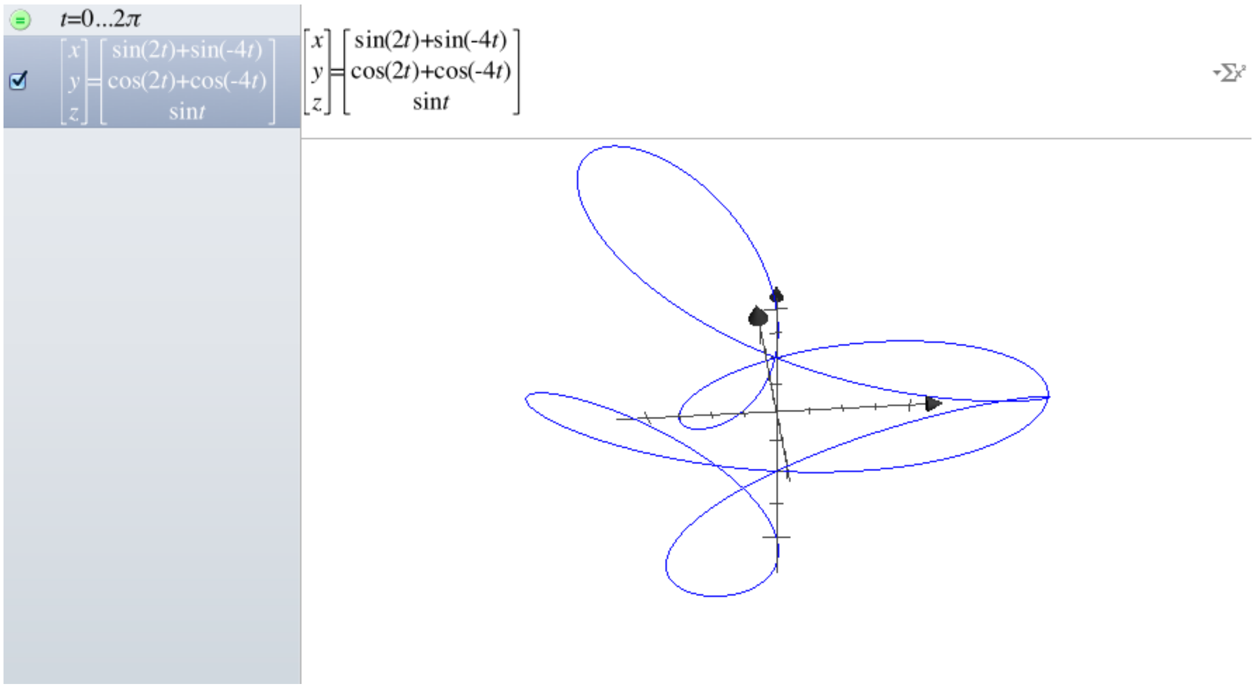
Just like with the 2-petal inspin flower, the top petal here can be placed on one side of the body and the bottom petal on the other side. Really, body tracers can be modeled with an equation almost identical to the one we used for flowers earlier in document only with the simple addition of a sine function on the z axis.

$$x = (h \cdot sint) + p(sin(dt))$$
$$y = (h \cdot cost) + p(cos(dt))$$
$$z = sint$$

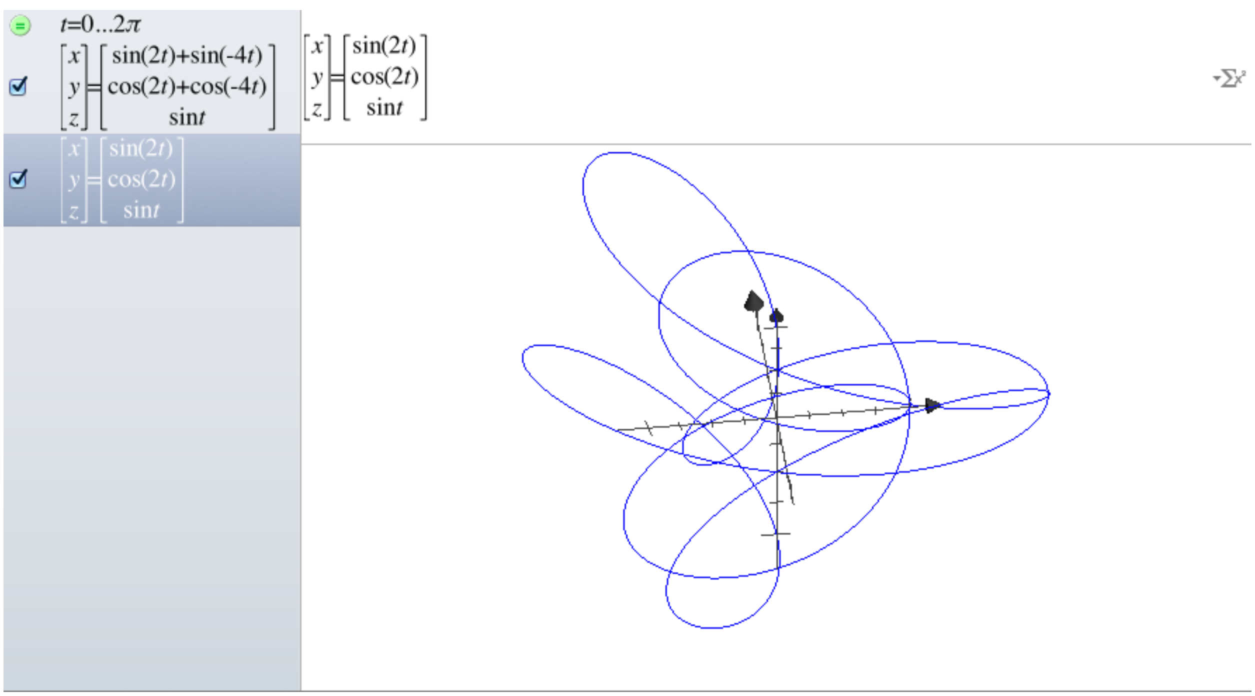
So far our exploration of manifolds has not required any math that is radically different from what we have already explored, but we have yet to explore more complex weaves or toroids. These will be the greatest challenges we encounter in our attempts to model poi patterns.

6. Modeling Complex Weaves

First: what is a complex weave? We've already discussed 2-, 3-, 5-, and 7-beat weaves and set aside inside moves for another time, what could be more complicated than these? Well, weaves have more or less become a *de facto* title assigned to the performance of moves that would be normally considered in the flower family, but wrap around the body in ways more complex than a body tracer. Body tracers fundamentally are built out of 2-beat weaves in that part of the pattern is on one side of the body and part is on the other side of the body. Complex weaves involve creating versions of patterns where the complete pattern exists on each side of the body. For example, one can perform a triquetra weave wherein rather than having part of the triquetra on each side of the body (as in a body tracer), we use one petal of the triquetra as a cross point to transition from a triquetra on one side of the body to one on the other side. This particular instance can be modeled using the following equation: $x = sin(2t) + sin(-4t)$, $y = cos(2t) + cos(-4t)$, $z = sint$. Here is that pattern already rotated to a point where it will be more intelligible:



Here is the same pattern with the handpath also graphed to provide a reference point as to how the hand and poi paths are interacting.



As you can see, the hand path is nearly identical to a 2-beat weave, so the resulting poi path is bent around it. I will not model additional complex weaves as I think it is easy enough to think of other patterns that conform to this framework. However, I will provide the equation with variables necessary to model these patterns for yourself. Bear in mind, when you hear people mention things like *isolation* or *cateye weaves*, this is how to model such a thing.

$$x = h(sin(dt)) + p(sin(ft))$$
$$y = h(cos(dt)) + p(cos(ft))$$
$$z = sint$$

Where h is the radius of the handpath in the x/y axes:

- d is the number of beats the hand gets for every beat along the depth or z axis,
- p is the phasing of the poi pattern,
- and f is the number of downbeats the poi has in relation to the number of beats in the handpath.

Bear in mind if you are trying to produce a flower where $f = -2$, you will now have to make $f = -2 \cdot d$ in order for the pattern to model properly. Here variable values for a few such patterns:

Isolated 3-beat weave:	**Cateye 3-beat weave:**	**2-petal inspin 2-beat:**
$h = ½$	$h = ½$	$h = 1$
$d = 3$	$d = 3$	$d = 2$
$p = -1$	$p = 1$	$p = 1$
$f = 3$	$f = -3$	$f = 3$

7. Modeling Toroids

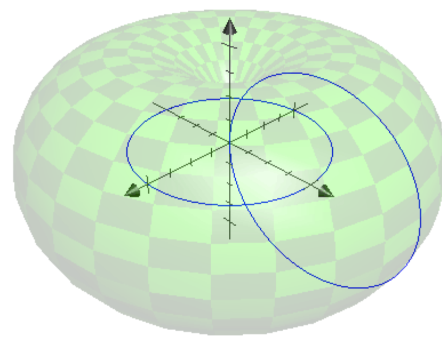
Toroids are the most complex type of movement we will address in this document. Modeling them requires not just tracking oscillations in all three axes, but simultaneously using the math of flowers coupled with the math for describing staff patterns. Why is this? Simply put, toroids are centered around the idea that rather than rotating the poi on an axis parallel to the axis of the handpath, we add the idea that it is now the poi plane that rotates on this axis parallel to that of the handpath. This mean that the poi plane now behaves much more like a staff than it does a poi and that the axis the poi rotates on is constantly shifting in relation to the hand. This results in patterns that are fully three-dimensional and consist of multiple plane bends. There are multiple frameworks for categorizing toroids, but for the purpose of this paper I am going to use the framework outlined by Ted Petrosky and Charlie Cushing[4]--known colloquially as the "East Coast" method[5] for categorizing toroids.

This framework distinguishes between three types of toroids based upon the movement of the plane relative to the movement of the hand. Like flowers, these toroids produce multiple petals[6],

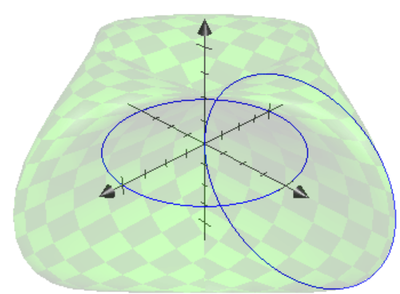
[4] https://www.facebook.com/groups/poitheory/permalink/266786586705417/
[5] http://youtu.be/IrS-NdpnEHo
[6] I realize the term petal isn't appropriately applied in this instance--I will be exploring this issue in depth later on in the paper.

but each of the different toroid types will produce differing numbers and locations of plane bends to produce the same number of petals. These three types are:

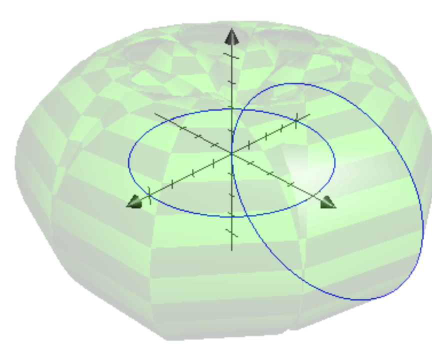
1. **Isobend**--in which the poi plane is locked in a synchronous orbit to the handpath, so that effectively the plane is always perpendicular to the handpath. This results in toroid patterns that look similar to the props of a propeller and tend to look like crosses and Xs when viewed head on. These are the toroids most similar to traditional flowers and indeed can be flattened down to traditional flowers by reducing the distance the poi travels along the z axis till the flower is flattened completely down to a standard flower. These are the toroid equivalent of an extension or isolation.



2. **Antibend**--in which the poi plane completes one or more turns in the direction opposite to the handpath's rotation in the time it takes to complete one handpath. This results in the poi plane switching between moments in which it is perpendicular to the handpath and moments when it is tangent to the handpath. If one conceptualizes a torus as having one side that faces the center and one side that faces the outside, antibend patterns tend to result in poi paths wherein the poi never travels on the inside of the torus. When viewed head-on, this approach can create polygonal patterns using straight lines that would otherwise be impossible in traditional flowers, such as triangles, squares, and pentagrams. These are the toroid equivalent of antispin in 2D spinning.



3. **Probend**--in which the poi plane completes more than one turn in the same direction as the handpath in the time it takes to complete one such handpath. Like antibend, this results in the



poi plane switching between being perpendicular and tangent to the handpath. Unlike antibend, probend patterns tend to result in poi paths where the poi never reaches the outside of the torus. This results in patterns that closely resemble inspin flowers in traditional 2D flower spinning, but the poi will switch between a plane in front of and behind the hand. These are the toroid equivalent of inspin in 2D spinning.

Given this framework, the astute reader may have noticed that many of the rules governing how the poi must move no longer apply to how a poi plane can move. Indeed, the most accurate model we have for understanding the behavior of the poi plane in a toroid is the behavior of a staff. With this in mind, we can easily understand the movement of the plane by updating our formula for deriving petals with poi movement to now work for staff. Where d is the number of staff/plane beats per handpath completion, to get the number of p petals, the equation will be $p = |2\,(1 - d)|$. Keep this in mind as it will become very important very quickly!

a. Modeling Isobend Toroids

We will start with the easiest type of toroid to model: the isobend. Because this type of toroid is closest to a perfect torus, we can apply the equations used to model a torus with a few minor tweaks to produce the result we want. The traditional equation for producing a torus is usually listed as such:[7]

$$x = (R + rcosu)sint$$
$$y = (R + rcosu)cost$$
$$z = r \cdot sinu$$

Where t is equal to all values between 0 and 2π, u is also equal to all values between 0 and 2π, R is the distance from the center of the torus to the center of its ring, and r is the radius of that ring itself. Now, this at first looks like a vastly different equation from what we are used to dealing with, but if we dive into it, we will find it is not so terribly different from what we are already used to using. First, if we get rid of the $(R + rcosu)$ terms, we can easily see that the resulting equation is very similar to what we already played with in our weaves section:

$$x = sint$$
$$y = cost$$
$$z = r \cdot sinu$$

So what is that extra term doing? If you notice the weaves equation we have separated out, you may notice something odd about it: when we were playing in 2D land and x and y were the only axes we needed to worry about, we had sin and cos working in tandem to give us circular motion. When we add a z axis oscillation for weaves, we are adding an odd sin term that has no

[7] http://en.wikipedia.org/wiki/Torus

cos to balance it out. This does not matter as much with weaves because the z value is just bouncing back and forth essentially along a straight line. But when we model a torus, we need to be creating circles both along the x and y axes, as well as circles between the x and z axes as well as the y and z axes. The term we removed to achieve the weave-like equation above is the missing cos term that balances out the sin function on the z axis to give us a circle--and it needs to be applied to both the x and y axis because we need to create a relationship both between them and also individual between each of those axes and the z axis.

To make the torus equation easier to understand, I will recontextualize it to look more similar to what we're used to dealing with so far:

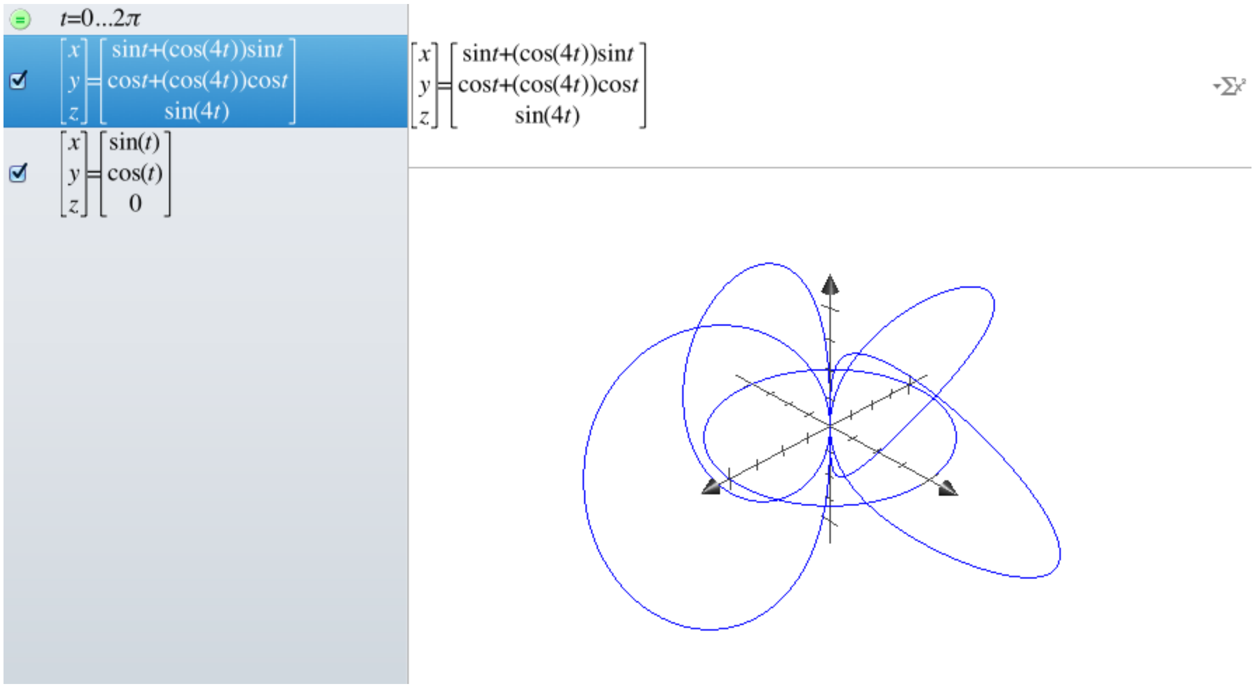
$$x = sint + (cosu)sint$$
$$y = cost + (cosu)cost$$
$$z = sinu$$

Movement of the hand **Movement of the poi**

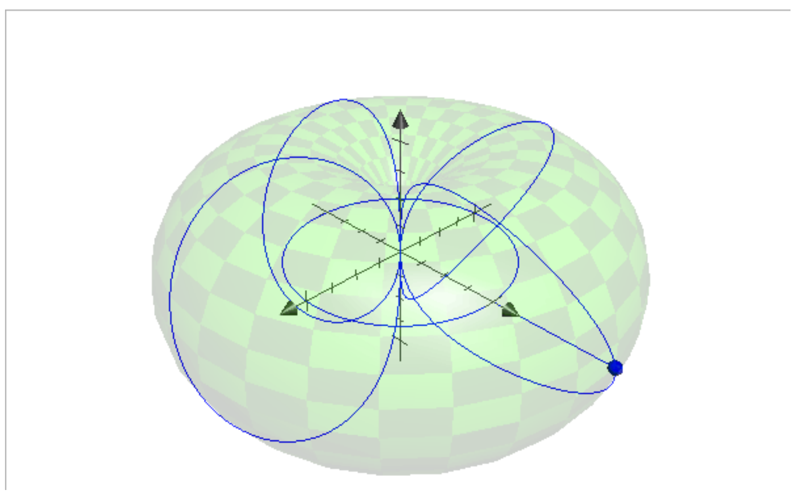
$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \sin t + (\cos u)\sin t \\ \cos t + (\cos u)\cos t \\ \sin u \end{bmatrix}$$

Where t is equal to all values between 0 and 2π and u is also equal to all values between 0 and 2π. Just like in our previous poi equations, the first set of terms on the x and y axes govern the movement of the hand. This equation produces the exact same result as the one I outlined above for a torus, but arranges the pieces in a way that is more in line with what we need to work with. Why are there two different variables that essentially do the same thing? Because to draw a torus we're drawing two different sets of circles. When we draw a toroid using poi, we are only tracking a path along the surface of the torus, so u will be replaced by t.

Given this, we can graph the path for an isobend with 4 petals using the following equation: $x = sint + (cos\,(4t))\,sint$, $y = cost + (cos\,(4t))\,sint$, $z = sin\,(4t)$.



Here, we have what would look like a 4-petal antispin flower as seen from the top axis, but each petal includes a single rotation in the z-axis. An example of this toroid being performed can be seen here where forward and back are treated as the z axis in this graph. The terms that govern the number of “petals” we get with an isobend toroid changes slightly as opposed to what we are used to seeing with 2D flowers. Now, the multiple of t that we find inside the cos functions on both the x and y axis in addition to the multiple of t on the sin function on the z axis are what controls the number of “petals” we wind up with. In this case, the multiple is equal to the number of “petals” we want to see in the resulting pattern. As noted above, petals is not the best term to describe the behavior of the poi relative to the handpath in toroids. We will explore a better term in the flower structure section of this paper under definitions.



An animation of how the poi tracks through this pattern can be found to the right.

b. Modeling Antibend Toroids

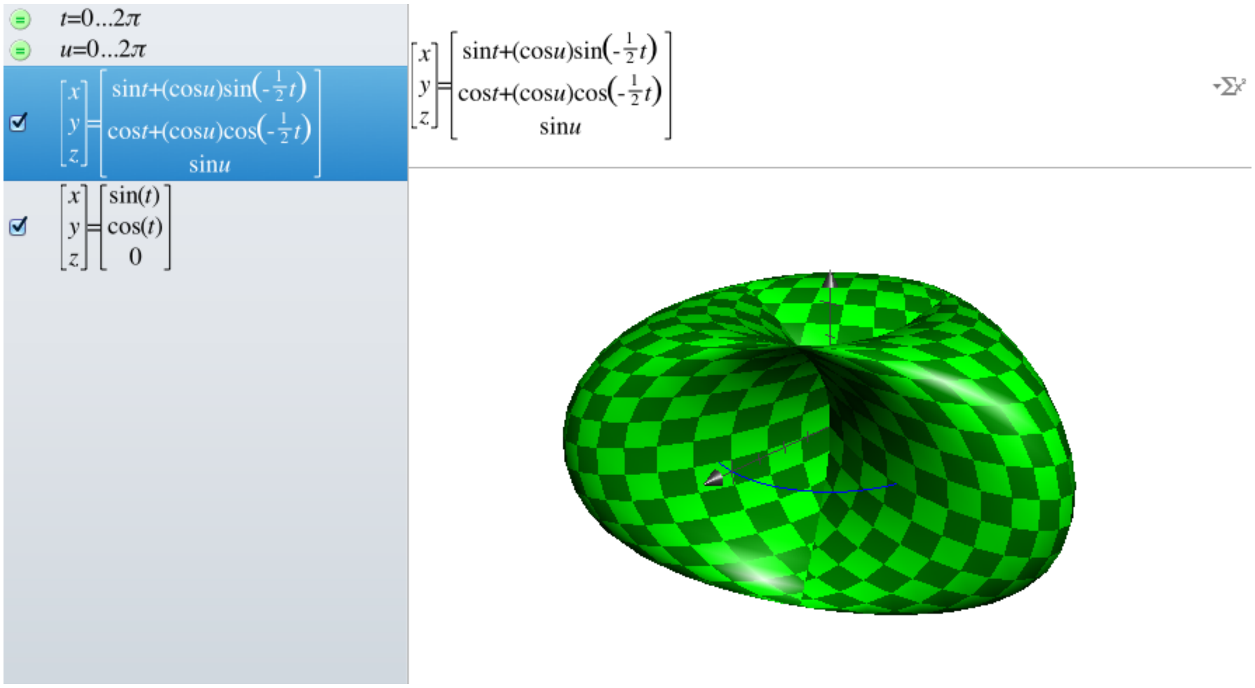
Antibend toroids are going to present a very different host of challenges to us. Where isobend toroids are planar locked to always be oriented perpendicularly to the handpath, antibends for the first time introduce the concept that the poi plane can turn in a different fashion, twisting up a torus into bizarre and wonderful self-intersecting shapes. This is one of the most popular types of toroids because it can be used to create shapes that leave a poi path that will appear similar to a polygon when seen from edge-on due to the poi plane functioning more like a straight line than a curve. This will also be the most diverse type of toroid we will explore as it presents multiple opportunities to create a variety of different shapes.

To start, we will need to update our equation open up additional possibilities for how the poi plane can rotate. Our equation now becomes:

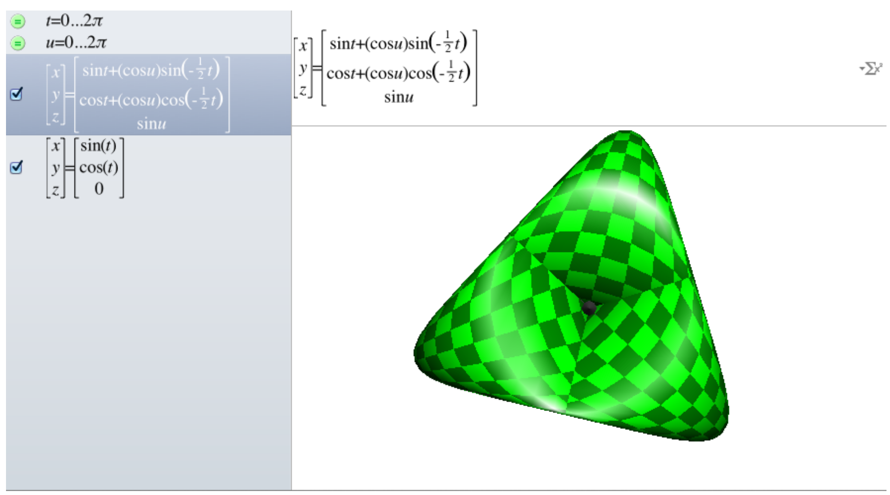
$$x = sint + (cos\,(ft))\,sin\,(dt)$$
$$y = cost + (cos\,(ft))\,sin\,(dt)$$
$$z = sin(ft)$$

Where f is the number of z-axis beats our pattern will have and d is the number of rotations the poi plane will have per handpath. This equation will also change what we know about the number of petals we are creating in a pattern. In isobend, because the poi plane is locked, the number of “petals” is essentially just determined by the number of times the poi is rotated as the plane goes around. For antibend, however, because the poi plane itself is rotating, we must for the first time consider what the shape generated by rotating the plane will be in addition to the number of beats the poi will have in the pattern.

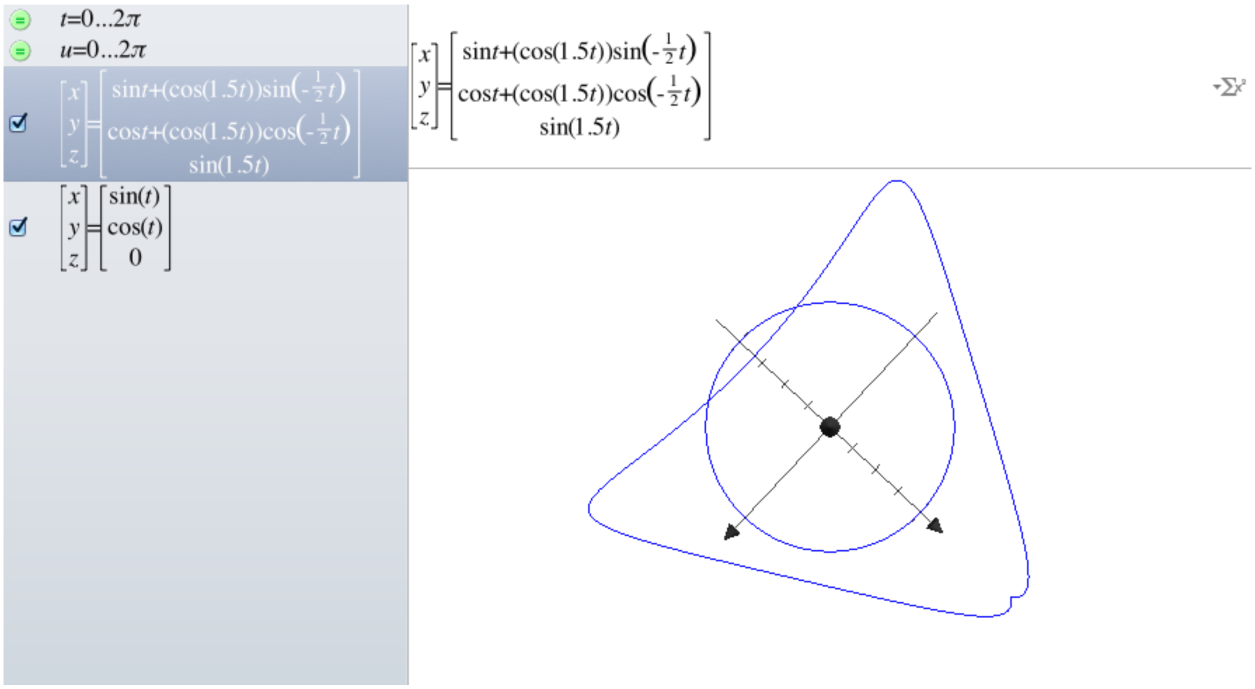
For example, if we elect to display a full torus with three points, the equation will be $x = sint + (cosu)\,sin\left(-\frac{1}{2}t\right)$, $y = cost + (cosu)\,sin\left(-\frac{1}{2}t\right)$, $z = sinu$ and it comes out looking like this:



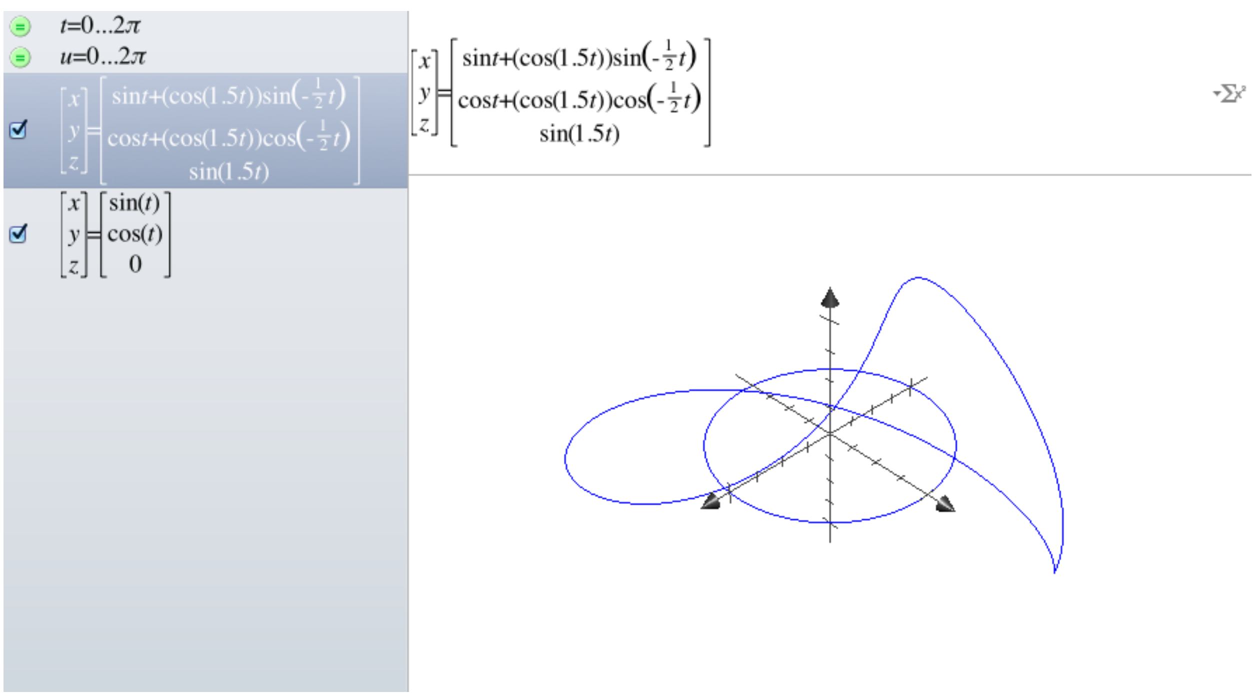
Which, when viewed from above comes out looking like a triangle with slightly rounded corners.



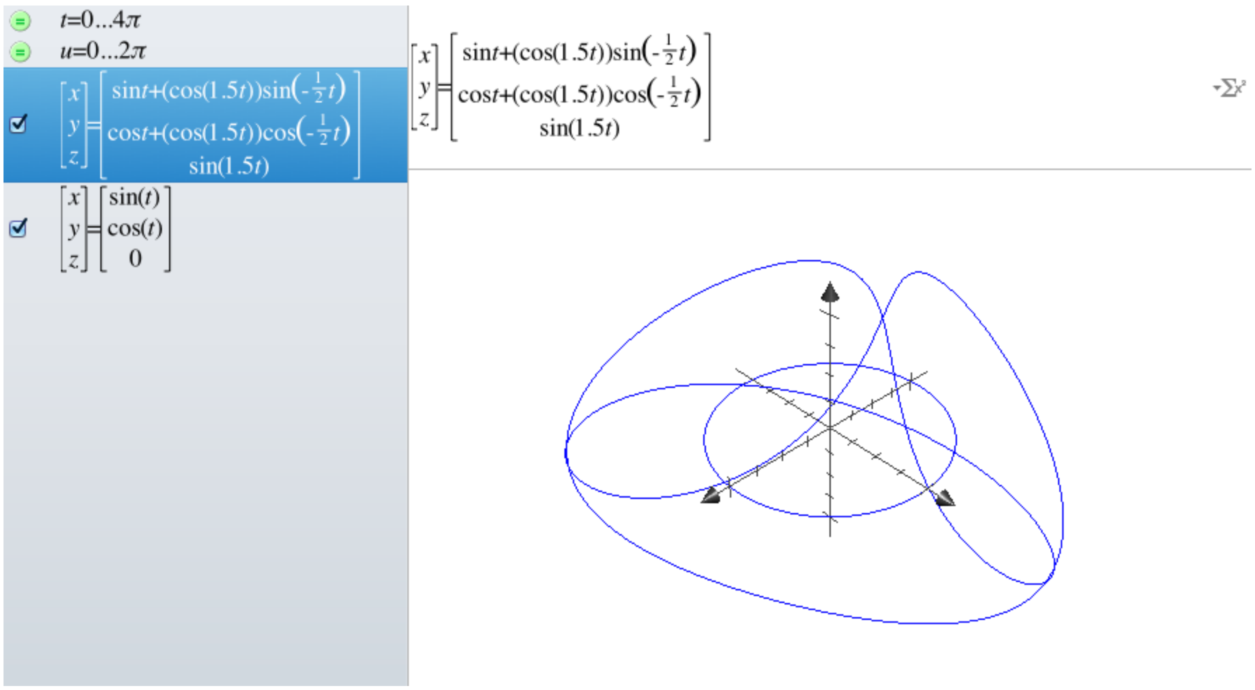
Now remember, this is just the movement of the poi plane. Unlike in this example, the poi never occupies all points in its plane simultaneously, so we must now assign a given number of beats to this pattern to see what the resulting poi path will become. One of the more popular ways to perform this pattern is to perform an half beat for each rotation of the plane, resulting in a poi path that likewise resembles a triangle when seen from above, resulting in the equation $x = sint + \left(cos\left(1\frac{1}{2}t\right)\right) sin\left(-\frac{1}{2}t\right)$, $y = cost + \left(cos\left(1\frac{1}{2}t\right)\right) sin\left(-\frac{1}{2}t\right)$, $z = sin\left(1\frac{1}{2}t\right)$.



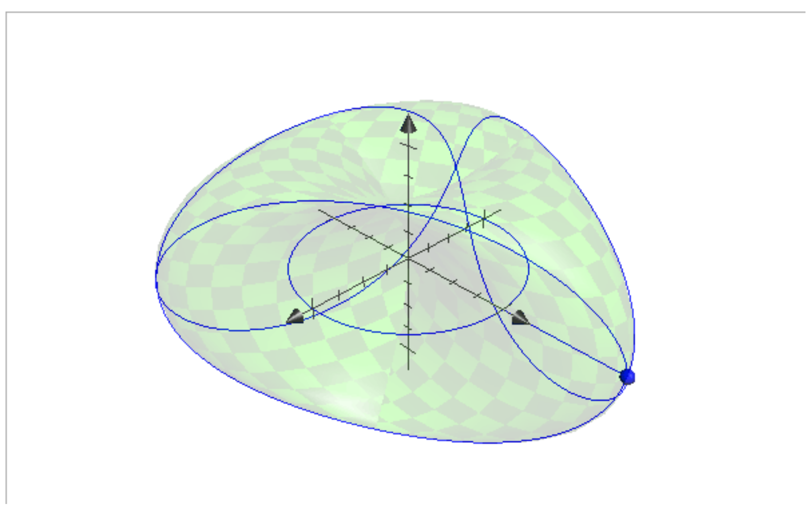
Now, something looks a little off, and a view from the side will help us see what it is. Because we are completing only 1.5 beats of the poi per handpath, the resulting poi path looks incomplete.



One way to fix this is to complete two repetitions of the pattern. Given that for each handpath, we are getting 1.5 poi beats, if we complete 2 handpaths we will get 3 poi beats, which because it is a whole number should produce a more complete looking pattern for us. We can easily perform this operation by setting t to equal all values between 0 and 4π, or completing two repetitions.



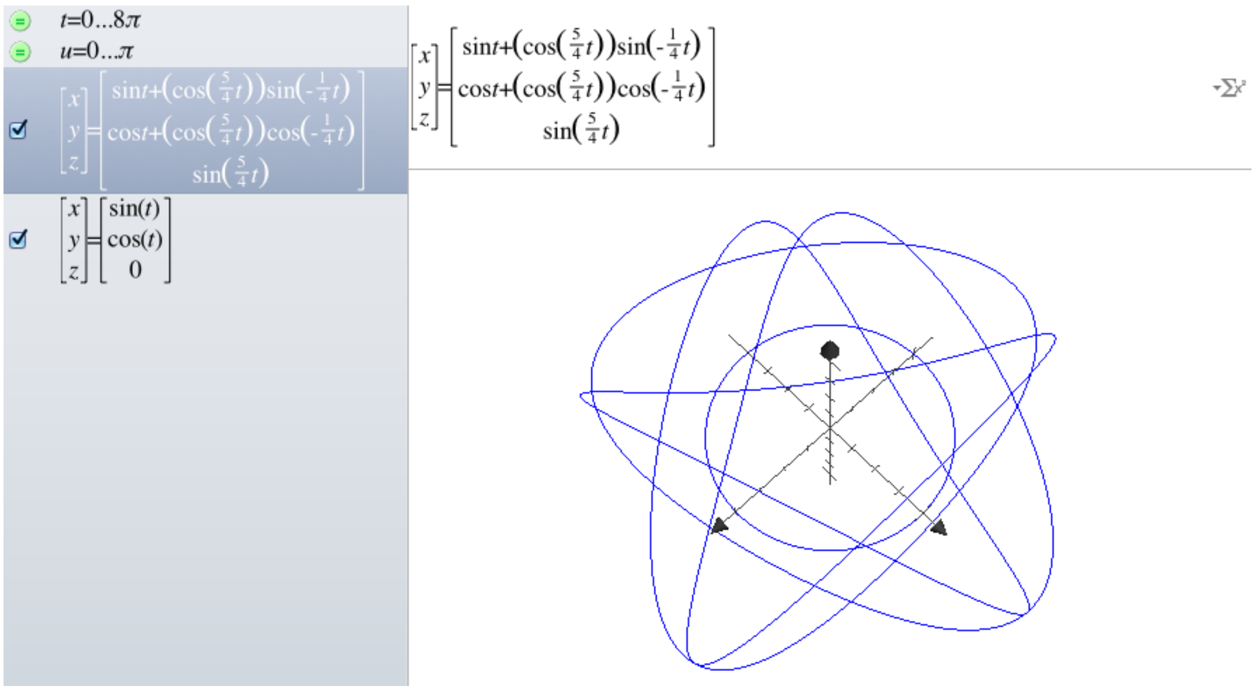
Now we have a completed pattern where the poi path intersects itself at each corner of the triangle. Viewed head-on, this pattern will appear to look very similar to a triangle, as it does in this long exposure photograph where it is performed with a triquetra.[8] Below is an animation of how the poi tracks through this pattern.



For another popular antibend toroid, we will model a pentagram. This toroid is much trickier! Pentagrams are generated via antispin when the poi completes $-1\frac{1}{2}$ beats for each handpath, but a toroid will be different because the poi's planes behave more like a staff than a poi. In this case, we are going to want the plane to complete $\frac{1}{4}$ of a rotation per handpath to set a base plane

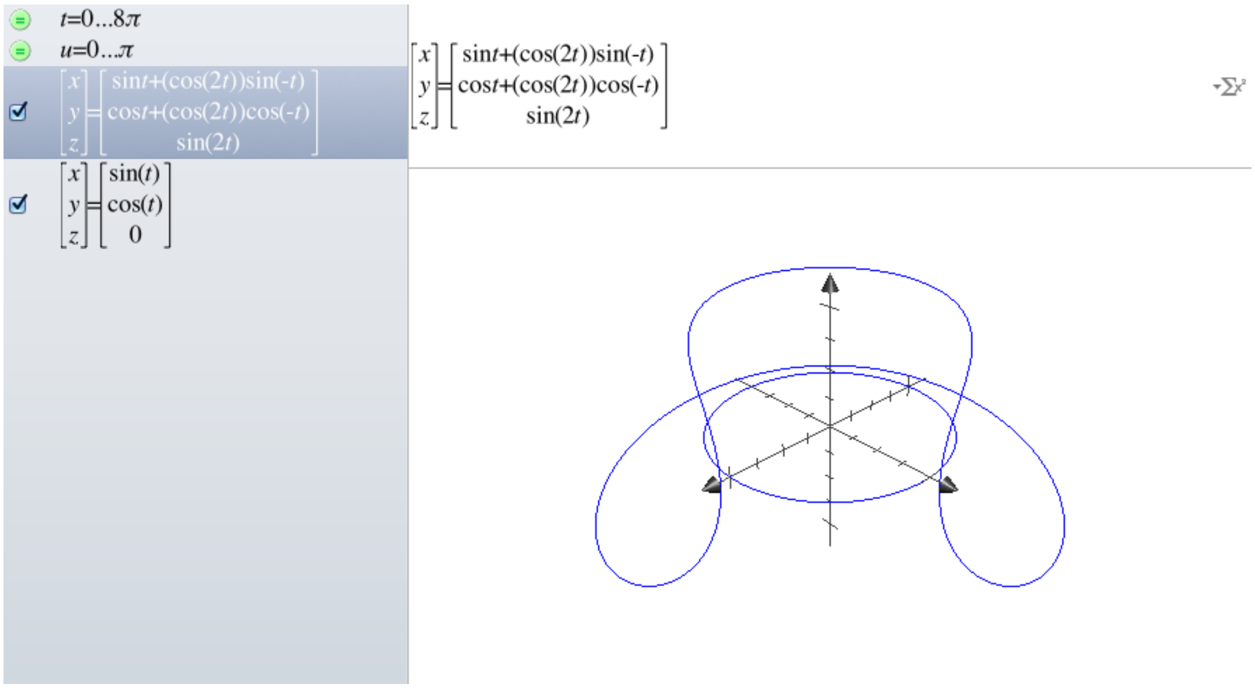
[8] http://www.pinterest.com/pin/373728469049273942/

shape that is a 5-pointed star polygon. The poi, then, will need to complete $1\frac{1}{4}$ beats per handpath to create the proper pattern. Thus, our equation is $x = sint + (cos\,(1\frac{1}{4}t))\,sin\,(-\frac{1}{4}t)$, $y = cost + (cos\,(1\frac{1}{4}t))\,sin\,(-\frac{1}{4}t)$, $z = sin\,(1\frac{1}{4}t)$. As we learned with the triangle, when the poi path has a number of beats that appears as a fraction, we have to perform multiple handpaths to catch up to it and generate a complete pattern. Because that number is $\frac{1}{4}$ in this case, we need four handpaths to complete the pattern, so t is equal to all values between 0 and 8π.

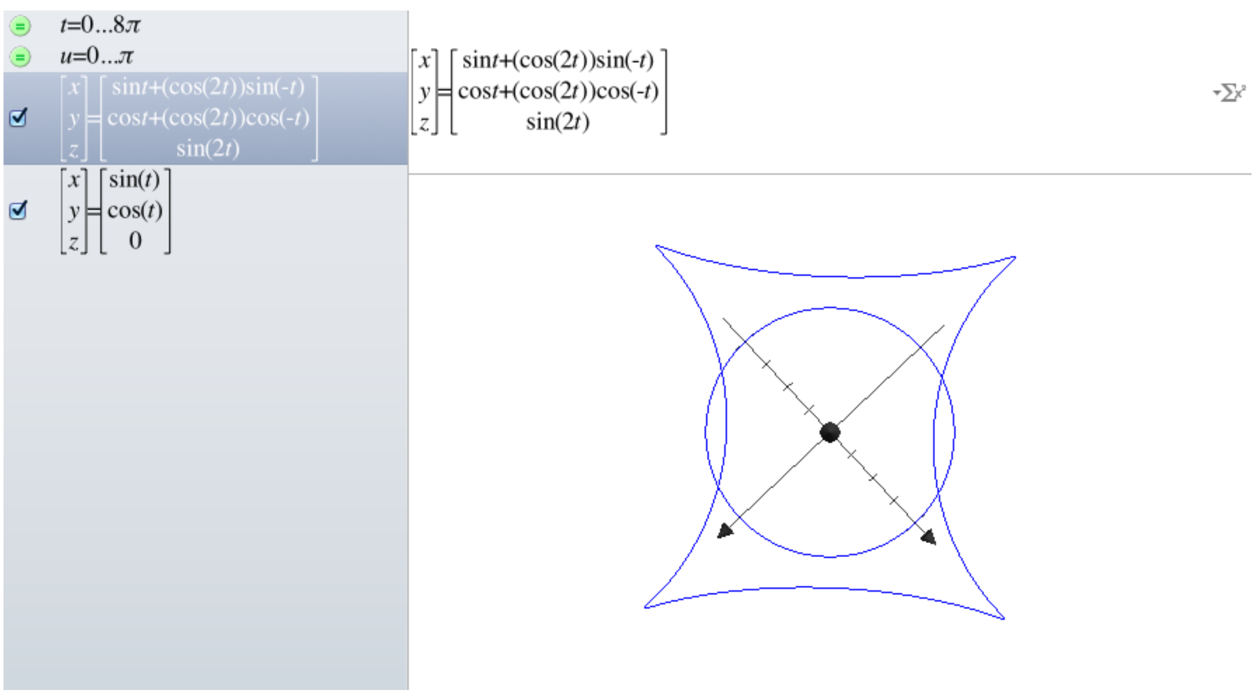


Again, we get an antibend toroid pattern where the poi traces a path along the outside edge of the torus, closely resembling a star polygon with straight edges and tight angles. An example of this toroid being performed with a long exposure photograph can be viewed here.

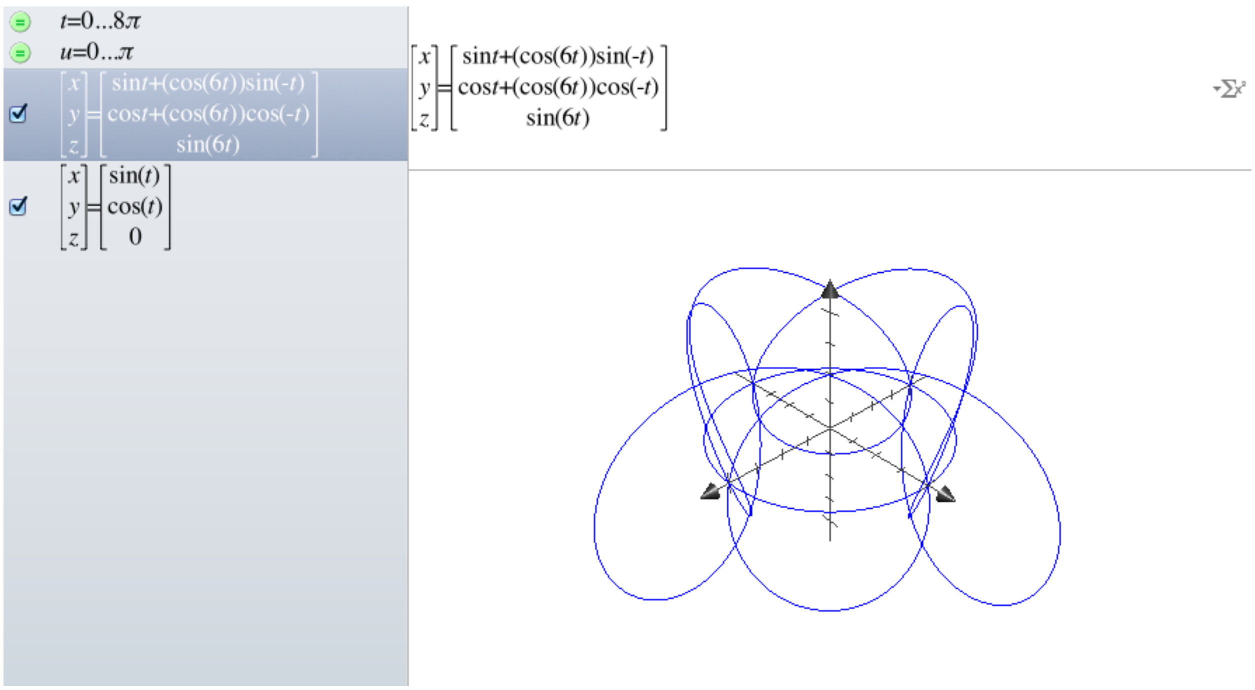
Let us model one final toroid with two variations to close out this section and see if we can't derive a few conclusions about toroid behavior from these examples. Our last example will be much simpler to model than either of our previous examples--this will be an antibend toroid with four corners rather than three or five. Compared to our last two examples, the equation for a four-pointed pattern is relatively simple. It will be $x = sint + (cos\,(2t))\,sin\,(-t)$, $y = cost + (cos\,(2t))\,sin\,(-t)$, $z = sin\,(2t)$ with t again returning to being equal to all values between 0 and 2π.



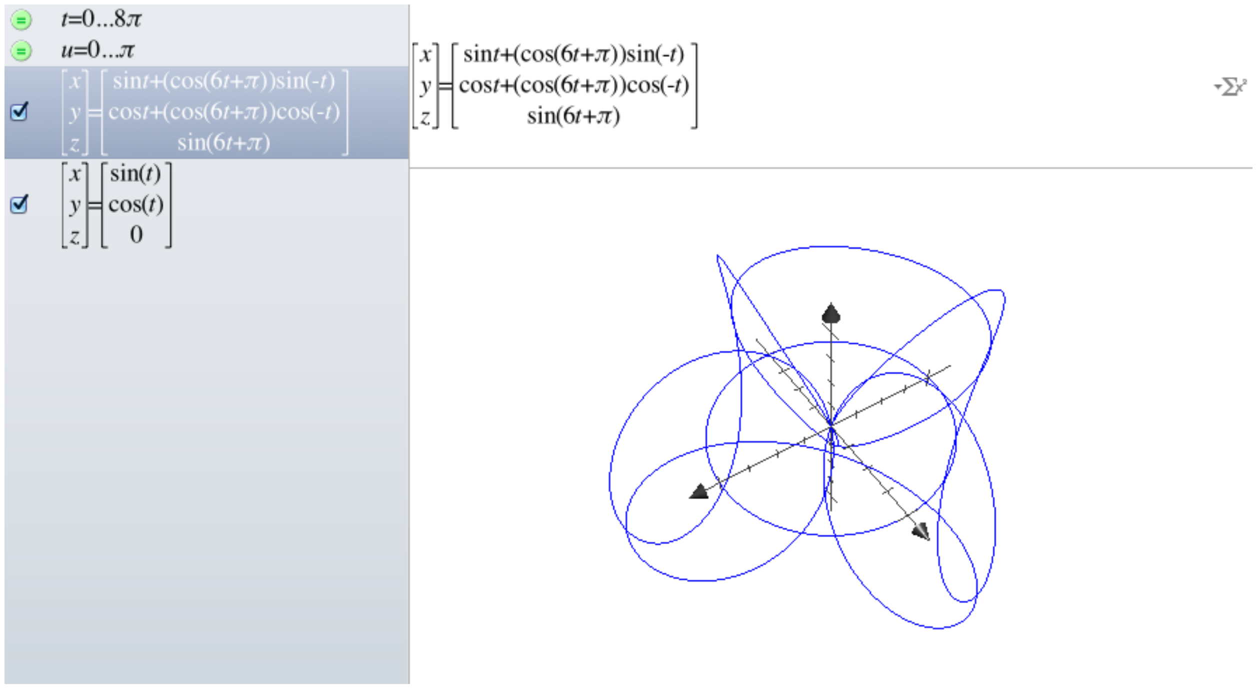
From above, this pattern looks not unlike a square or a diamond with slightly curved sides.



Because the angles at the corners of this pattern approach 90 degrees (or $\frac{\pi}{2}$ if we are working in radians), it is actually fairly difficult to accomplish this pattern with poi and maintain the angles cleanly, so it is a habit among poi spinners who perform this pattern to include what are called "grace" beats to stabilize the poi every quarter of the shape to prevent the pattern from looking a mess. Fortunately, we can model the grace beats and not lose the overall integrity of our equation. Because they require $1\frac{1}{2}$ beats per corner of the pattern, we just take the t multiplier on our two cos functions on the x and y axes as well as our sin function on the z axis and multiply it by $1\frac{1}{2}$, resulting in 6 being our new multiplier for this term. The resulting equation is $x = sint + (cos\,(6t))\,sin\,(-\,t)$, $y = cost + (cos\,(6t))\,sin\,(-\,t)$, $z = sin\,(6t)$ and the graph looks like this:



Those who have performed this pattern will instantly notice that something is wrong: the grace beat is being performed on the sides of the pattern rather than the corners. Fortunately, this is just a simple matter of changing the phasing of the pattern. We can accomplish this by now adding π to 6t to shift the pattern of the poi beats a half a circle. The resulting equation is $x = sint + (cos\,(6t + \pi))\,sin\,(-\,t)$, $y = cost + (cos\,(6t + \pi))\,sin\,(-\,t)$, $z = sin\,(6t + \pi)$.

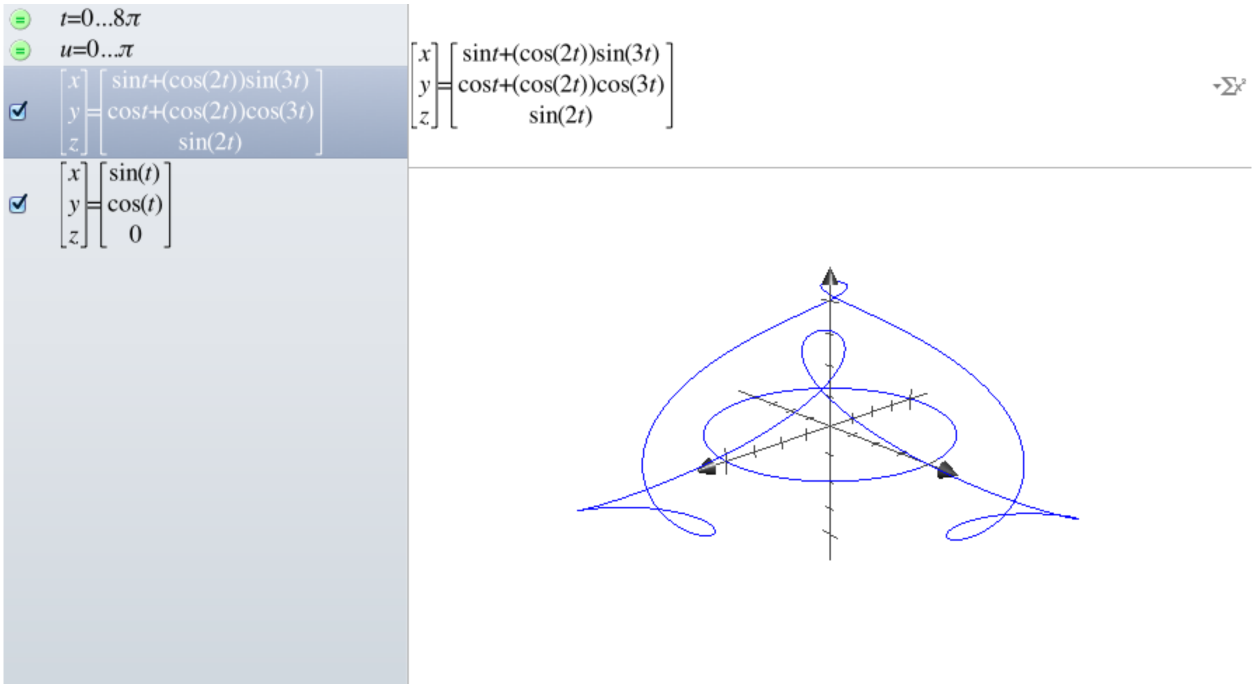


This path now looks like a cross between the 4-petal isobend we looked at in the last section and the 4-corner antibend because in essence it is. At each corner we have a grace beat from which we are able to jump to the next corner. This lesson on phasing the poi beats of an toroid will again become important in the section on probend toroids. You'll note, I'm accomplishing this in a different fashion than I did in our section on modeling isolations in 2D. The reason for this will become clear when I present an equation that includes variables to model each and every pattern presented in this document.

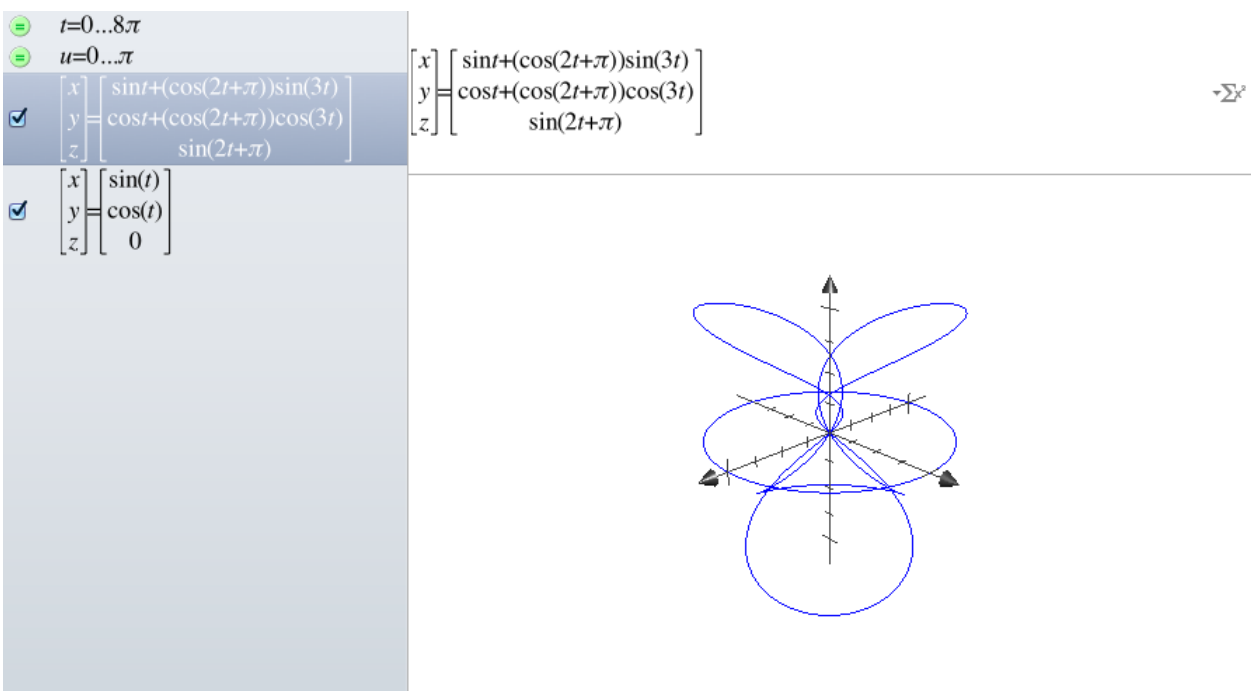
The reader will note that unlike the antibend triangle and pentagram, the antibend square required only one handpath to complete. This is just an interesting hiccup in the math--antibend patterns with an odd number of points will always require multiple handpaths of an even number to complete because each corner of the poi path reverses the direction of the poi. If you think of the rotation of the poi as having two sides (you could think of it as when the poi is rotating "up" vs "down" or in front of you vs behind you), the poi will return to the beginning of the handpath traveling in the opposite direction it started in. Thus, to return the poi to its origin, the number of handpaths needs to be doubled.

c. Modeling Probend Toroids

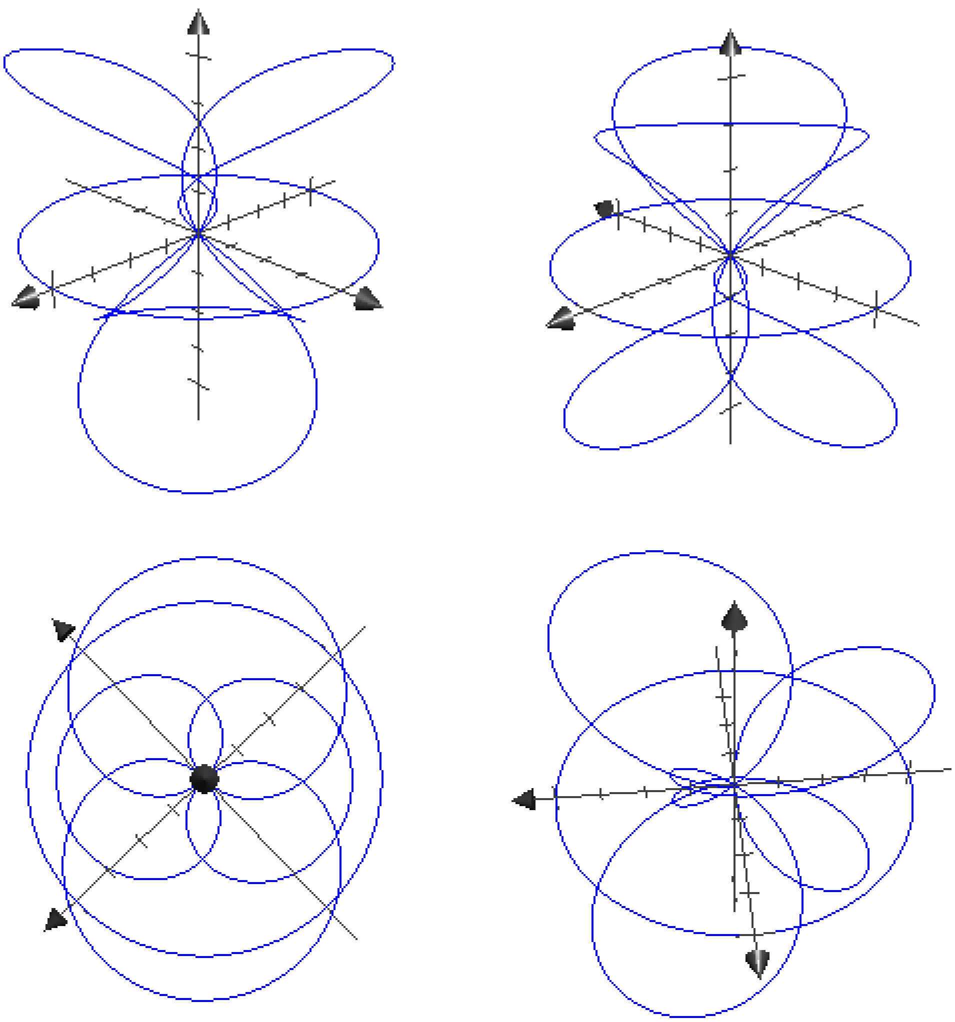
Like antibend toroids, probend toroids are also built on the notion that the poi plane rotates more freely in relation to the handpath. In this case, the plane will rotate in the same direction as the hand, but complete more than one rotation in the process. To give one such example, we will model a probend pattern that will have 4 "petals". To do this we will use the equation $x = sint + (cos\,(2t))\,sin\,(3t)$, $y = cost + (cos\,(2t))\,sin\,(3t)$, $z = sin\,(2t)$ to create this very odd-looking pattern:



It would be extremely difficult to perform this pattern given that a very rapid bend occurs at the top and bottom of the pattern. If, however, we use the same trick we learned for changing the phasing of an antibend with grace beats, we arrive at $x = sint + (cos\,(2t + \pi))\,sin\,(3t)$, $y = cost + (cos\,(2t + \pi))\,sin\,(3t)$, $z = sin\,(2t + \pi)$ and get a much different result:



I will present this shape from a few other perspectives as well:

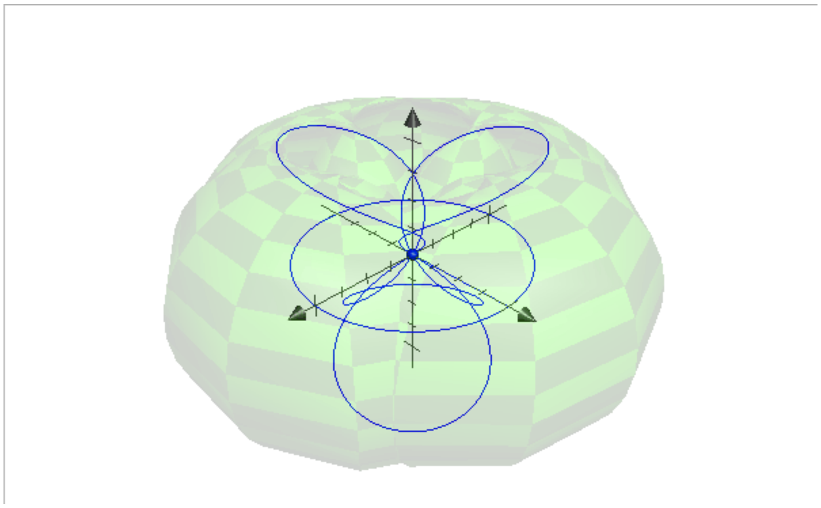


It may be difficult to make out from these perspectives, but this probend can be described as consisting of a series of 4 arcs between petals where the petals then switch the plane of the poi back and forth along the z axis. This is the least-explored and most difficult to understand type of toroid and results in the opposite pattern as the antibend: in a probend, the poi travels a path across only the inside surface of the torus. It is a difficult maneuver to perform cleanly, but there are a few video examples of this type of toroid being performed, one of which can be found [here].

Now that we have viewed examples of all three types of toroids, we can set down a series of variables and equations that are capable of describing any toroid pattern as well as set down certain rules governing the relationships between the values of variables in said equations.

$$x = sint + (cos\,(ct + d))\,sin\,(ft)$$
$$y = cost + (cos\,(ct + d))\,cos\,(ft)$$

$z = sin(ct + d)$



Where c is the number of poi beats in the overall pattern, d is the phasing offset of the poi path within the pattern and f is the number of beats the poi plane will complete per handpath. To start making sense of how these values interrelate, let us start with f because it will likely be the most difficult variable to comprehend. This is because f now functions like a staff rotating rather than a poi rotating and a staff has two ends instead of one. Now we derive the number of petals p in the equation with $p = 2 \cdot |1 - f|$. Or, if we'd like to think of antispin flowers as having negative petals and inspin flowers positive petals, the equation will be $p = 2(1 - f)$. Which will be more useful to us because we can now solve the equation for f in order to find the value of f when we know how many petals we are hoping to produce in our pattern. For example, if we wish to have a -3 petal pattern (triangle), we can use $f = \frac{p}{2} + 1$ to find $f = -\frac{1}{2}$, exactly the number we used for our antibend triangle!

If instead we want to find the value of c in a given equation, first we can go back to always thinking of p as being a positive number, but now the answer will depend on the type of toroid we wish to create. If we are building an isobend and we know the number of petals p we are attempting to generate, $c = p$. Easy as pie. If we are building either a probend or an antibend toroid without grace beats, $c = \frac{p}{2}$. Finally, to create an antibend toroid with grace beats, we use $c = \frac{3p}{2}$.

8. A Generalized Equation for Modeling Poi Patterns in this Paper

Given all the different equations we have so far encountered in our study of poi patterns, we can now set down all the variables and equations we would need to model any one of them. This equation is as follows:

$$x = (a \cdot sint) + b(sin(mt)) + c(sin(nt))(cosp + s)$$
$$y = (a \cdot sint) + b(cos(mt)) + c(cos(nt))(cosp + s)$$
$$z = d(sin(ot + s))$$

This equation may look complex, but it involves no variable assignments we haven't worked with already. One question you may be asking is why I have skipped certain series of letters--the short answer is that my graphing program has preset values for e, i, j, and r such that I cannot override them in the program. To ensure consistency with my diagrams, I have opted to write out the equation to avoid using any such letters.

Let's dive into each variable so we can understand exactly what it means. For a, this is the radius of the core handpath of a third-order motion (almost always 1) or it is 0 if the pattern we intend to produce is not a third-order motion. For b, this is the radius of the handpath for most flowers and the second center of rotation for a third-order motion or it is 0 if we intend to produce a simple weave. For c, this is the phasing of the poi path and will be either 1 or -1 depending on whether we are producing a fractal flower or isolation in 2D spinning. For d, this is the number that tells us if our pattern will utilize a sin function on its z axis (in the case of weaves and most toroids, this value will be 1, for all other patterns 0). For variable m, this is the variable that determines how many downbeats the handpath pattern has in a third-order motion or how many beats the handpath has in a complex weave. For n, we have the number of poi beats per handpath in 2D spinning as well as the number of beats the poi plane completes per handpath in toroids. For o, we determine the number of z-axis beats a pattern has (in 2D spinning, this number will be 0). In toroids, p determines how many beats the poi path will have in the x and y axes--in this case $p = o$, but for all 2D spinning and weaves, $p = 0$ (and yes, math geeks, this variable is arguably redundant given that we could effectively replace c with it to accomplish much the same thing, but I am trying to keep as much consistency as possible between the equations in this document). Finally, for s we have the phasing of the poi path in toroid patterns (if the pattern is not a toroid, this number is always 0).

Wow! That's a lot. To help make sense of it, I have included a table below with sample values for these variables to create many of the patterns we've already worked with.

Flowers:

Triquetra	2-petal Inspin	Isolation
$a = 0$	$a = 0$	$a = 0$
$b = 1$	$b = 1$	$b = \frac{1}{2}$
$c = 1$	$c = 1$	$c = -1$
$d = 0$	$d = 0$	$d = 0$
$m = 1$	$m = 1$	$m = 1$
$n = -2$	$n = 3$	$n = 1$
$o = 0$	$o = 0$	$o = 0$
$p = 0$	$p = 0$	$p = 0$
$s = 0$	$s = 0$	$s = 0$

Third-order Motions:

Zan's Diamond	Triquetra Fractal Flower	Inspin-inspin 2-petal
$a = 1$	$a = 1$	$a = 1$

$b = -1$	$b = 1$	$b = 1$
$c = 1$	$c = -1$	$c = 1$
$d = 0$	$d = 0$	$d = 0$
$m = -3$	$m = -2$	$m = 3$
$n = 5$	$n = 4$	$n = 5$
$o = 0$	$o = 0$	$o = 0$
$p = 0$	$p = 0$	$p = 0$
$s = 0$	$s = 0$	$s = 0$

Weaves:

2-beat Weave	3-beat Weave	Triquetra Weave
$a = 0$	$a = 0$	$a = 0$
$b = 0$	$b = 0$	$b = 1$
$c = 1$	$c = 1$	$c = 1$
$d = 1$	$d = 1$	$d = 1$
$m = 1$	$m = 1$	$m = 2$
$n = 2$	$n = 3$	$n = -4$
$o = 1$	$o = 1$	$o = 1$
$p = 0$	$p = 0$	$p = 0$
$s = 0$	$s = 0$	$s = 0$

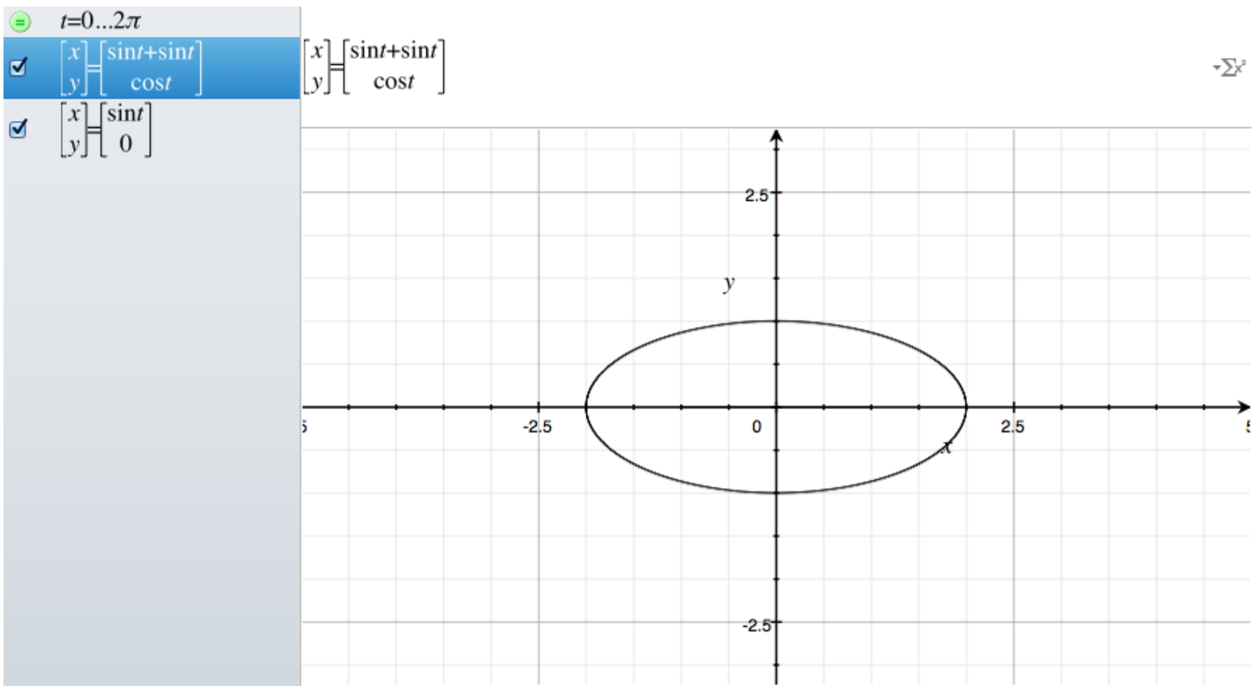
Toroids:

4-petal Isobend	3-petal Antibend	4-petal Probend
$a = 0$	$a = 0$	$a = 0$
$b = 1$	$b = 1$	$b = 1$
$c = 1$	$c = 1$	$c = 1$
$d = 1$	$d = 1$	$d = 1$
$m = 1$	$m = 1$	$m = 1$
$n = 1$	$n = -\frac{1}{4}$	$n = 3$
$o = 4$	$o = 1\frac{1}{2}$	$o = 2$
$p = o$	$p = o$	$p = o$
$s = 0$	$s = 0$	$s = \pi$

Given these values, you should be able to go back through any pattern already mentioned in this paper and be able to produce the graph associated with it by inferring the values of each variable in it. That does not mean that we are done, however. There are still two types of patterns we have not discussed, though neither requires significant revision of the equations we've laid out. Those two types of patterns are linear extensions and composite patterns.

9. Linear Extensions

Given much of the work we have just completed, linear extensions will come as incredibly simple patterns. They haven't been mentioned yet because I rarely have need of modeling them as generally speaking they are used as stepping stones to more complex patterns that have already been described in this paper. We can model one easily with the following equation: x=sint+sint, y=cost. That's it. It results in the following graph:



The handpath here is hidden by the x axis, but literally is a straight horizontal line that reaches for -1 to 1. You'll note, this move was accomplished by using different values for our previous variable b depending up whether it's on the x axis or the y axis. Again, I think these patterns are elementary enough they don't need to be included in our universal equation, but if the reader prefers to have a complete model, they can do so using the following equation:

$$x = (a \cdot sint) + b_0 (sin (mt)) + c (sin (nt)) (cosp + s)$$
$$y = (a \cdot sint) + b_1 (cos (mt)) + c (cos (nt)) (cosp + s)$$
$$z = d (sin (ot + s))$$

All other variables remain the same, but now b_0 is used to denote the radius of the handpath along the x axis and b_1 is used to denote the radius of the handpath along the y axis. For the graph above, $b_0 = 1$, $b_1 = 0$.

10. Composite Patterns

We have now discussed a suite of different poi patterns, but have yet to talk about what happens when they are combined together in different ways. After all, nobody wants to just spin the same pattern over and over again forever! In addition, there are examples both obvious as well as subtle of patterns we commonly take for granted that are the product of changing a pattern as we spin that can be mistaken for patterns in and of themselves. To distinguish between these types, I'm utilizing the terms elementary pattern and composite pattern.

First, a **pattern** is any poi pattern in which the poi and hand complete a movement over time t in which they depart from and return to the positions both occupied at $t = 0$.

An **elementary pattern** is any pattern for which every variable in the universal equation outlined above has been assigned a value and the path of the poi and hand has returned to the position and direction of the origin of the pattern. Most elementary patterns can be recorded between t values of at least 0 and 2π, though some patterns may require higher t values to complete.

A **composite pattern** is any pattern that is created via the transition of one incomplete elementary pattern to one or more additional elementary patterns. In other words, though poi and hand depart from and eventually return to their positions at $t = 0$, they may also at any point switch to a different set of variables for the universal equation before returning to the positions they occupied at $t = 0$. (I am indebted to Alien Jon for introducing me to the term and its applications).

Given these requirements, every single pattern we have examined so far in this paper qualifies as an elementary pattern. We have yet to work with composite patterns, however, as they add an additional degree of complexity. We will investigate three such types of composite patterns: CAPs, stalls, and antibend toroids with grace beats.

a. CAPs

CAP stands for *Continuous Assembly Pattern*, a type of movement originally identified by Damien Boisbouvier based upon a move seen in an old Yuta performance video.[9] He has done considerable work on the topic of fleshing out this concept[10] but fundamentally defines it as a single hand and poi performing continuously elements of two or more elementary patterns in series. This definition adheres closely to what we've already laid out here, but Damien's approach differs slightly in that he models his patterns using polar instead of Cartesian coordinates, meaning that he can model a CAP with a single equation while for us it is more easy to assemble a CAP out of two or more equations.

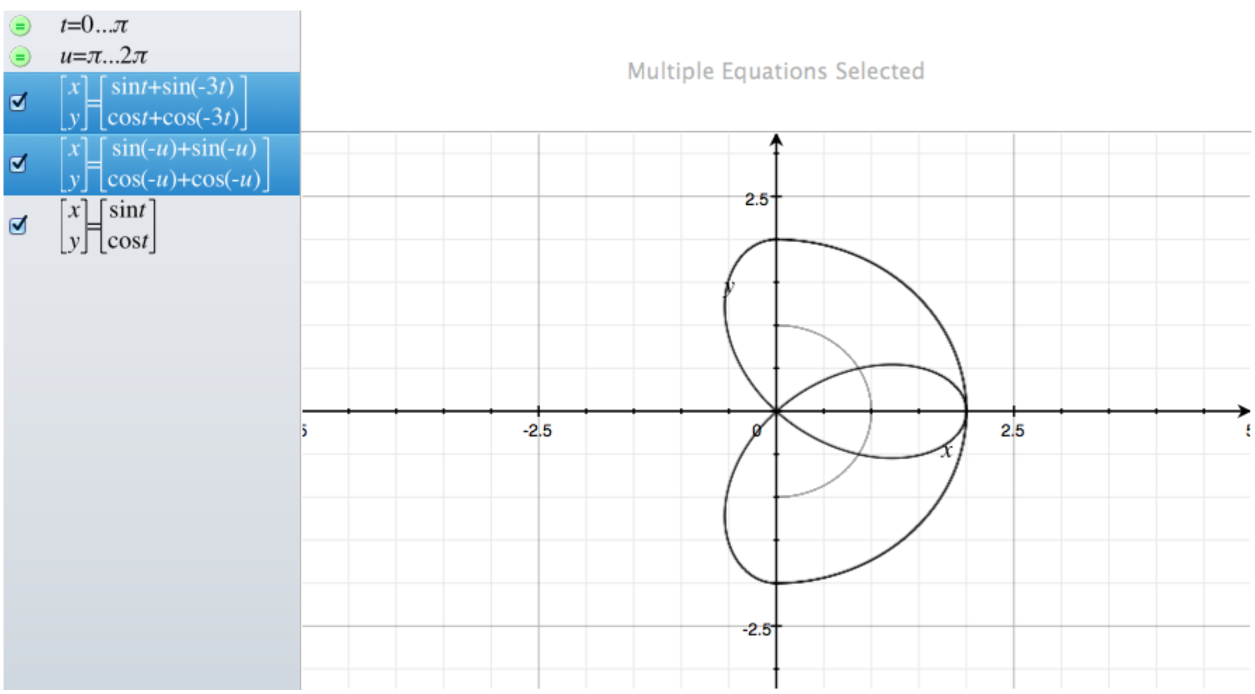
The most commonly performed type of CAP is that performed in the original Yuta video, frequently referred to as the C-CAP. In our system, it can be thought of as half of an extension combined with half of a 4-petal antispin flower. But how do we model this? Well, it will take two

[9] http://www.homeofpoi.com/community/ubbthreads.php/topics/836555/Yuta_moves_analysis.html
[10] http://www.homeofpoi.com/community/ubbthreads.php/topics/891193/What_are_CAP_s.html

copies of our universal equation to do so, but we would set out the following variable assignments:

Side 1	Side 2
$a = 0$	$a = 0$
$b = 1$	$b = 1$
$c = 1$	$c = 1$
$d = 0$	$d = 0$
$m = 1$	$m = -1$
$n = -3$	$n = -1$
$o = 0$	$o = 0$
$p = 0$	$p = 0$
$s = 0$	$s = 0$

Where side 1 is assigned to all values of t between 0 and π and side 2 is assigned to all values of t between π and 2π. In other words, we are breaking two different elementary patterns in half and sticking them together. For simplicity's sake in graphing these equations, I have set the variable t to only equal those values between 0 and π and assigned a second variable, u, to be assigned to the values between π and 2π.



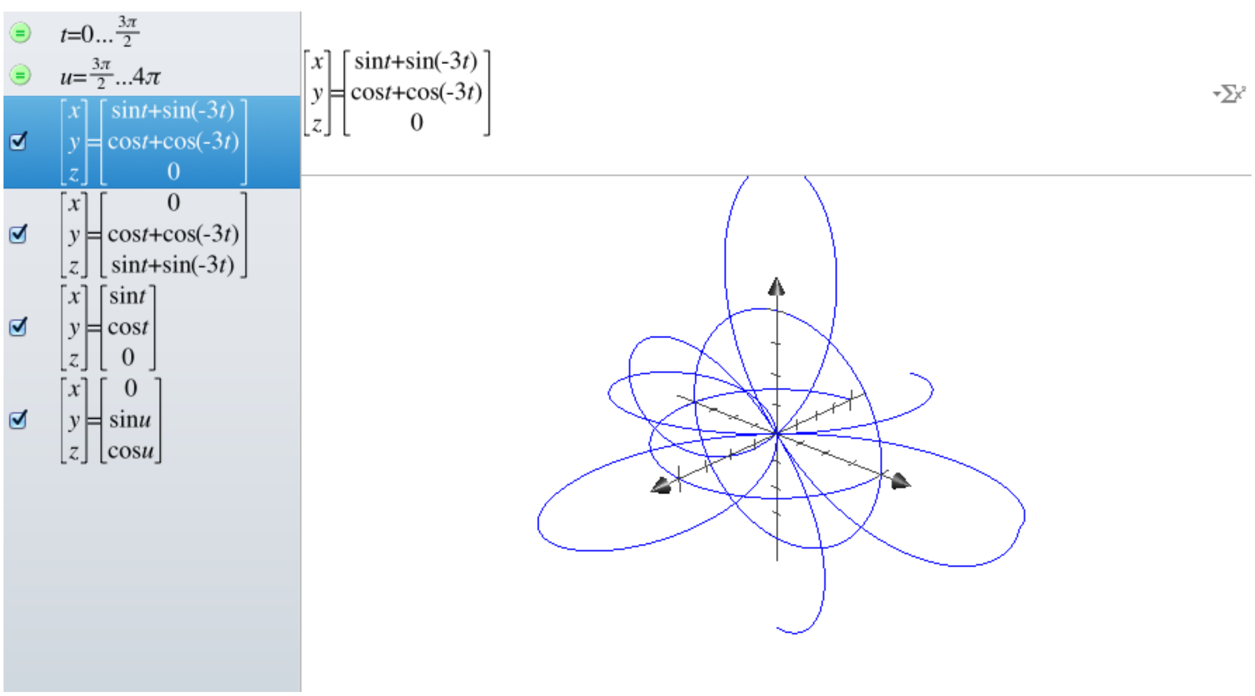
To those of us who have been spinning the past few years, this is a very familiar pattern indeed. We can also see easily how it is created by taking parts of two different equations and combining them together. One obvious question might be to ask why m and n have both been assigned values of -1 in the second equation. The reason for this is that it means that this part of the equation will track backwards. Just as we know that when m is a positive number and n is a negative number, we get antispin (because a negative n forces the poi to rotate in the opposite direction as a positive number m for the hand), assigning a negative value to m for the second half of the equation means that the hand is turning in the opposite direction as the first side, thus retracing the same path. You will note that between the two equations, the poi does not actually change direction--only the hand does. The poi, however, does change the number of downbeats it completes in relation to the number of downbeats the hand performs.

b. Stalls

If one is looking for a solution that is just "good enough," stalls are relatively easy to write equations for. They can simply be taken as a fraction of a flower that completes only to the tip of a given flower petal. This will not precisely model the behavior of stalls, but it will provide opportunities to easily model things like plane breaks. One such example can be obtained by compositing together the following two equations:

Side 1: $x = sint + sin(-3t)$, $y = cost + cos(-3t)$, $z = 0$
Side 2: $x = 0$, $y = cost + cos(-3t)$, $z = sint + sin(-3t)$

To get this result:

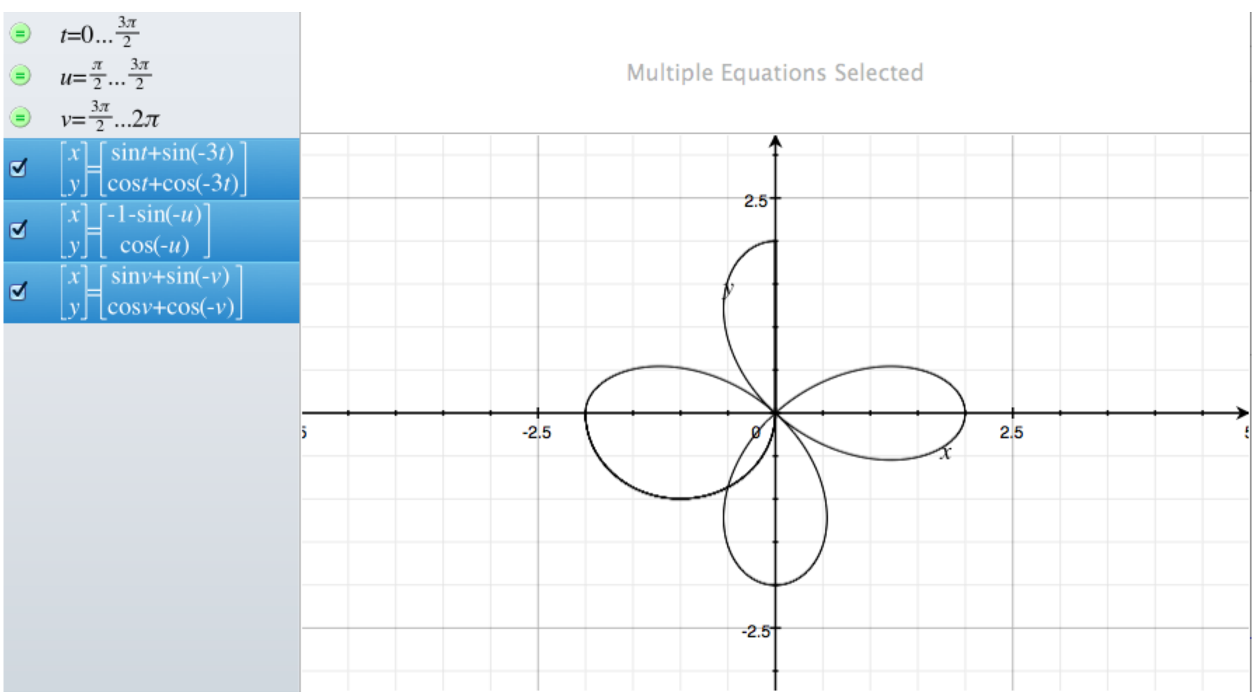


This can be treated as a plane shift between two atomic plane orientations--perhaps wall vs horizontal, for instance. While this is a workable solution, it does not model what occurs in a stall very accurately. To that end, stalls have been one of the most difficult things I have ever attempted to model with trigonometric equations--I still don't believe I have found a good solution to doing so. A few years ago, Charlie Cushing and I engaged in a long study on how stalls could be effectively modeled using these equations and came up with two primary approaches--both are difficult to write out, comprehend, and predict the properties of. One approach was to add an additional cyclical function to the hand's t term in both the x and y axes, effectively making the hand move along its path faster in some places than in others. This changes the shape of the flower petals just slightly to bring them to a point rather than rounded edges. The other approach is to effectively freeze the hand on the penultimate flower petal till the poi can come around to the axis along which it will be stalling.

It is this latter style that slow motion studies of videos reveals creates the cleanest stalls and which will be demonstrated here. In this particular case, a stall consists of three parts: ¾ of a 4-petal antispin flower so that the tip of the third petal is reached, then half a static spin with the hand immobile before finishing the handpath with the poi completing ¼ of a linear isolation to take it straight out the axis upon which it is stalling. The Byzantine equations that describe this series of actions are:

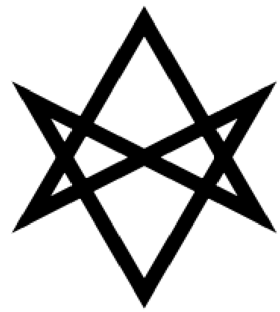
Side 1: $x = sint + sin(-3t)$, $y = cost + cos(-3t)$
Side 2: $x = -1 - sin(-u)$, $y = cos(-u)$
Side 3: $x = sinv + sin(-v)$, $y = cosv + cos(-v)$

Where t is equal to all values between 0 and $\frac{3\pi}{2}$, u is equal to all values between $\frac{\pi}{2}$ and $\frac{3\pi}{2}$, and v is equal to all values between $\frac{3\pi}{2}$ and 2π.



While this composite pattern may at first appear quite bizarre, the most noticeable quality of it when performed in real life is the cleanliness of the stall along the y axis. To put it mildly, modeling stalls in this fashion, while relatively accurate, can be very difficult and might be more work than is worth it. Nonetheless, if the reader is up for undertaking this challenge, the method outlined here will produce the desired result.

c. Composite Toroids



One of the benefits of spinning toroids, especially in antibend, is that they produce straight lines rather than the curves we are accustomed to seeing when spinning poi. Because of this, we are able to generate patterns with poi that would otherwise be impossible. One such example is the *unicursal hexagram*, a star polygon that, should we attempt to perform it as a standard poi flower, would result in alternating between regions of antispin and extension such that the intention of the pattern would be completely lost. This is due to the fact that unlike most other star polygons, the unicursal hexagram is an irregular polygon[11], meaning that each of its sides is not equally long, resulting in angles of different sizes that alternate regularly throughout the figure. Because the figure can be drawn with straight lines, however, it can be produced as a toroid, but not as one that meets the requirements of an elementary pattern.

Producing a unicursal hexagram requires six separate equations that all interlink with each other. I won't lie, modeling this results in a less than ideal result due to the fact that each segment has a significantly different curve from the previous one. It is likely possible to write a single equation that could graph the entire shape, but said equation would be highly complex and take quite a long time to puzzle out. When we physically spin this shape, we are automatically making constant adjustments to make each segment match up to the next. In lieu of this, I offer up the following six equations that together graph out a unicursal hexagram in toroid form. Please note: in producing this shape I am doing so with the radius of the handpath dilated down to 0 as this is the most common arrangement for performing this particular shape.

Segment 1:
$x = \left(cos\left(\frac{9}{6}t\right)\right) sin\left(-\frac{1}{2}t + \pi\right)$
$y = \left(cos\left(\frac{9}{6}t\right)\right) cos\left(-\frac{1}{2}t + \pi\right)$
$z = sin\left(\frac{9}{6}t\right)$

Segment 2:
$x = (cos(-w))\, sin\left(0w + \frac{2\pi}{3}\right)$

[11] http://en.wikipedia.org/wiki/Irregular_polygon

$y = (cos\,(-w))\,cos\left(0w + \frac{2\pi}{3}\right)$
$z = sin\,(-w)$

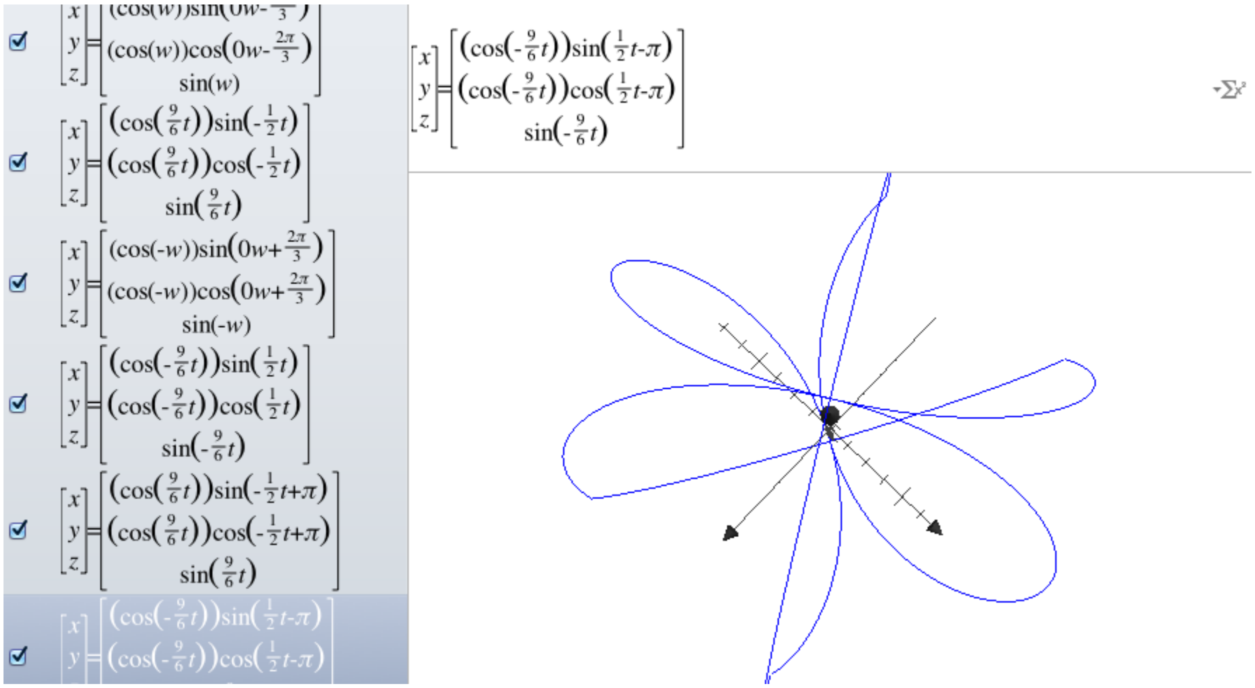
Segment 3:
$x = \left(cos\left(\frac{9}{6}t\right)\right) sin\left(-\frac{1}{2}t\right)$
$y = \left(cos\left(\frac{9}{6}t\right)\right) cos\left(-\frac{1}{2}t\right)$
$z = sin\left(\frac{9}{6}t\right)$

Segment 4:
$x = \left(cos\left(-\frac{9}{6}t\right)\right) sin\left(\frac{1}{2}t\right)$
$y = \left(cos\left(-\frac{9}{6}t\right)\right) cos\left(\frac{1}{2}t\right)$
$z = sin\left(-\frac{9}{6}t\right)$

Segment 5:
$x = (cos\,(w))\,sin\left(0w - \frac{2\pi}{3}\right)$
$y = (cos\,(w))\,cos\left(0w - \frac{2\pi}{3}\right)$
$z = sin\,(w)$

Segment 6:
$x = \left(cos\left(-\frac{9}{6}t\right)\right) sin\left(\frac{1}{2}t - \pi\right)$
$y = \left(cos\left(-\frac{9}{6}t\right)\right) cos\left(\frac{1}{2}t - \pi\right)$
$z = sin\left(-\frac{9}{6}t\right)$

Where t is equal to all values between 0 and $\frac{2\pi}{3}$ and w is equal to all values between 0 and π . This produces the following graph:



It is neither as pretty nor as clean as we may prefer, but it does contain all our essential segments and they do all interconnect. I apologize for the piecemeal assembly of this particular pattern, but it took a lot of trial and error to get all the segments to connect properly. Consider this an open invitation to the reader to find the equation that would give us a cleaner version of this pattern. A photograph of this pattern being performed can be viewed here.

C. Defining Elementary Poi Patterns

Now that we've been presented with several different types of poi tricks that can all be modeled, we're going to take this capability a step further by utilizing the ranges of variables we have found for each particular classification of tricks and using them as a diagnostic by which we can determine whether a particular poi pattern fits within this classification. Each variable we have examined can be expressed in lay terms such that any trick classified by the ranges of its variables can also be described in language the common person can comprehend.

For example, if we take the variable values for a triquetra as outlined in the universal equation section above:

For:

$$x = (a \cdot sint) + b\,(sin\,(mt)) + c\,(sin\,(nt))\,(cosp + s)$$
$$y = (a \cdot sint) + b\,(cos\,(mt)) + c\,(cos\,(nt))\,(cosp + s)$$
$$z = d\,(sin\,(ot + s))$$

$a = 0$
$b = 1$
$c = 1$
$d = 0$
$m = 1$
$n = -2$
$o = 0$
$p = 0$
$s = 0$

We can then translate this series of variables into lay language by stating it is a pattern created by having the hand travel around a circular pattern 1 poi length in radius as the poi completes 2 beats rotating in the opposite direction. Simple. For variables that are not currently in use, there is no reason we need to express them in this context, they will only lead to clutter. With this approach in mind, I humbly submit the following variable ranges and accompanying vernacular to classify the poi tricks we have examined so far in this paper.

1. Flowers

In this case, I am using flowers as an umbrella term to describe any and all of the 2D patterns that we examined above, including subsets of inspin and antispin flowers as well as the poi unit circle as a subfamily.

Variable Range:

$a = 0$
$0 < |b| < 4$ (depending on the performer's poi and their armspan)
$|c| = 1$
$d = 0$
$m = 1$
$|n| \geq 1$
$o = 0$
$p = 0$
$s = 0$

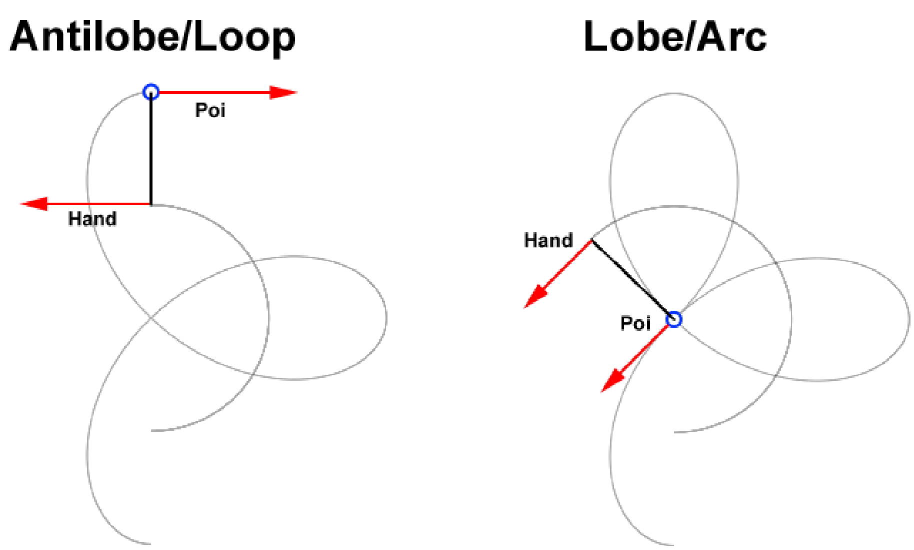
Vernacular Description:

Flowers are the family of moves in which the poi and hand both rotate in the same two axes, the hand creating a circle that has a radius greater than zero and ultimately bounded by the armspan of the performer while the poi completes one or more rotations in either the same or opposite direction as the hand.

a. Petals vs Lobes/Antilobes

A major point of contention in discussing flowers has been the use of the term "petal," for which no formal definition has yet been set down. It has been used both to connotate the enclosed loops that are produced when the poi completes more than one rotation in relation to the hand in either direction as well as regions of a flower in which the poi and hand are moving in opposite directions. Not surprisingly, this leads to a great deal of confusion as to how many petals there are in smaller antispin patterns like cateyes.

I am going to avoid that debate entirely by shifting my vocabulary. Rather than trying to reframe the concept of a petal to be easily diagnosable and court controversy over the resulting definition, I'm going to embrace a vocabulary developed by Alien Jon and explored in greater detail by myself and Charlie Cushing to shift the conversation over the structure of flowers to focus on the movement of the poi vs the movement of the hand.



In this framework, at the tips of the structures we traditionally regard as "petals," the movement vectors of the poi and the hand are parallel, but pointed in opposite directions. These regions were originally referred to as loops by Alien Jon, but lately we have been favoring the vocabulary Charlie developed[12], referring to these regions as "antilobes." At the opposite extreme are regions in which the vectors of the poi and hand are parallel, but are pointed in the same direction. These regions (originally dubbed arcs by Alien Jon) are called lobes in Charlie's vocabulary. It is important to note that these states do not exist in binary--any flower will exhibit gradients between the two extremes wherein the movement vectors of the hand and poi twist around in different directions before settling to parallel in either direction, thus producing a lobe or an antilobe. To put it simply:

Lobes are regions of a flower where the hand and poi are moving in parallel as well as the same direction.

Antilobes are regions of a flower where the hand and poi are moving in parallel but opposite

[12] https://www.facebook.com/groups/113059425470308/permalink/456352241141023/?stream_ref=2

directions.

With this classification in mind, all flower patterns display equivalent numbers of lobes and antilobes with two exceptions: isolations and extensions. In the former case, the poi and hand are eternally locked in a state where their movement vectors are parallel and moving in opposite directions (essentially, and isolation is an endless lobe) and in the latter case, the poi and hand are locked in a state where their vectors are parallel and moving in the same direction (thus, an extension is an endless antilobe).

Framing flower movements in this fashion is helpful because given the fact that all flowers feature these regions, they can be thought of as transition points between different flowers. As long as two flowers share the position of a lobe or antilobe, one can easily shift between them.

b. Antispin and Inspin

With the structure of our flowers now defined, it is time to define the difference between antispin and inspin. Fortunately, this is a trivial matter both mathematically and in the vernacular given that antispin and inspin flowers have long been defined in terms of whether the poi spins in the same or opposite direction as the hand. Below are the variable values that make up these definitions.

Antispin:
$a = 0$
$0 < |b| < 4$ (depending on the performer's poi and their armspan)
$|c| = 1$
$d = 0$
$m = 1$
$n \leq -1$
$o = 0$
$p = 0$
$s = 0$

Inspin:
$a = 0$
$0 < |b| < 4$ (depending on the performer's poi and their armspan)
$|c| = 1$
$d = 0$
$m = 1$
$n \geq 1$
$o = 0$
$p = 0$
$s = 0$

One logical question based upon these equations is why $|n| \geq 1$. The short answer is because gravity exists. When $|n|$ has values of less than 1, we encounter regions of a pattern in which the hand would have to have the capacity to push the poi upwards against the pull of gravity or straight sideways, a feat that is impossible for poi because the tether is not a rigid structure. When using a rigid prop like a club or a staff, however, n can be equal to any value at all.

Vernacular Description:

Antispin flowers are a subset of flowers (and thus subject to the restrictions laid out in the previous section) wherein the poi rotates in the opposite direction of the hand and completes one or more beats per handpath. Inspin flowers are a subset of flowers wherein the poi rotates in the same direction as the hand and completes one or more beats per handpath.

c. Unit Circle Patterns

Unit circle patterns are a specialized subset of flowers that incorporate both inspin and antispin patterns but set the handpath at ½ a poi length in radius. Technically we could include absolutely any pattern with said handpath in this family, but for the purposes of this paper I will adhere to the restrictions outlined by Alien Jon when he originally defined this family as only the patterns possible in this framework that have a single poi downbeat.[13] With that in mind, here is the mathematical definition:

$a = 0$
$b = \frac{1}{2}$
$|c| = 1$
$d = 0$
$m = 1$
$|n| = 1$
$o = 0$
$p = 0$
$s = 0$

Thus, the acceptable values for both c and n are ± 1, which covers all possible phasings of unit circle patterns, from extensions, to isolations, to both vertical and horizontal cateyes.

Vernacular Description:

Unit circle patterns are a subset of flowers wherein the radius of the handpath is equivalent to half the length of the poi and the poi completes a single beat for each handpath.

Believe it or not, this simple description actually covers every possible unit circle pattern.

2. Third-order Motions

[13] http://techpoi.tribe.net/thread/7d1509cc-1f2b-4448-b2a5-f1ef100ae18e

As seen previously in this paper, third-order motions require an additional center of rotation to be generated, so they will also require an expanded set of variables to be fully realized. Given this, we can realize all possible third-order motion patterns with the following variable values:

$a > 0$
$|b| > 0$
$a + |b| \leq 4$ (again, depending upon the armspan of the performer)
$|c| = 1$
$d = 0$
$|m| \geq 1$
$|n| \geq 1$
$o = 0$
$p = 0$
$s = 0$

Deciphering this equation, we find that now a and b together must equal the total armspan of the performer rather than b being solely responsible for carrying this value. Because b can be phased as a positive or negative number, we take the absolute value of it in formulating this equation. Similarly, now that the handpath can have multiple downbeats relative to the "virtual" handpath at the core of this movement, it is assigned the same possible range of values as the poi.

Vernacular Description:

Third-order motions are the family of poi patterns wherein both the poi and the hand rotate along the same pair of axes, but in such a way as to simulate 3 different sources of rotation. While the hand follows a path analogous to a flower pattern[14], the poi completes 1 or more beat for each completion of this flower pattern in either the same or opposite direction to the rotation of the hand.

a. Fractal Flowers

Fortunately, we already outlined many of the necessary concepts for defining this subset of third-order motions and can reach back to our work there to set the relationships necessary for our variables.

$a > 0$
$|b| > 0$
$a + |b| \leq 4$ (again, depending upon the armspan of the performer)
$c = -b$
$d = 0$
$|m| \geq 1$

[14] I realize it's annoying to use a recursive definition like this, but I'm trying to save space.

$n = m^2\left(-\frac{-m}{m}\right)$
$o = 0$
$p = 0$
$s = 0$

In this special case of third-order motions, we want to establish that the phasing of the poi pattern is opposite that of the handpath in addition to setting the poi beats to be hand beats squared, but in the opposite direction as the hand's movement. To achieve this, we assign to it a positive or negative value opposite that of m, the hand's direction of movement and number of beats.

Vernacular Description:

Fractal flowers are the subset of third-order motions wherein the number of poi beats is equivalent to the number of hand beats times itself and phased opposite to the handpath.

b. Triquetra Expansions

Again, we explored this idea in depth before, so assigning our variables should be relatively easy. Like fractal flowers, we are deriving our value of n (poi beats) from m (hand beats) but now we are doing so in a linear rather than exponential fashion.

$a > 0$
$|b| > 0$
$a + |b| \leq 4$ (again, depending upon the armspan of the performer)
$c = -b$
$d = 0$
$m \leq -1$
$n = |m - 2|$
$o = 0$
$p = 0$
$s = 0$

In this case, our work is simplified because we know both that triquetra expansions have to be antispin-antispin (and thus, that m must be a negative value) and that for each beat of m we are adding 2 poi beats n (and that since the poi is rotating opposite the hand, that this value must be positive).

Vernacular Description:

Triquetra expansions are a subset of third-order motions wherein the hand follows the handpath of an antispin flower while the poi completes 2 beats for each beat of the handpath while phased opposite the handpath.

3. Manifolds

Manifolds are a word usually not applied to poi except in specialized circles. Here, I am going to use the term in an umbrella fashion to apply literally to every single possible 3D pattern that can be produced with poi. Thus, it will apply to both simple and complex weaves, body tracers, and toroids. Think of flowers as covering all 2D poi movement and manifolds as covering all 3D poi movement. With this much ground to cover, the possible ranges of the variables will actually be pretty boring at this stage.

For:

$x = (a \cdot sint) + b\,(sin\,(mt)) + c\,(sin\,(nt))\,(cosp + s)$

$y = (a \cdot sint) + b\,(cos\,(mt)) + c\,(cos\,(nt))\,(cosp + s)$

$z = d\,(sin\,(ot + s))$

$a \geq 0$

$|b| \geq 0$

$a + |b| \leq 4$ (depending on the armspan of the performer)

$c = \pm 1$

$d > 0$

$m \geq 1$

$|n| \geq 1$

$o > 0$

$p \geq 0$

$0 \leq s \leq 2\pi$

This means that for the first time, we can have a range of values available for every variable. One will note, however, that there is no pattern we have explored in the course of this paper that *does* include a value for each variable. Breaking this down, we see once again that a and b together are limited by the performer's armspan, that the phasing of the poi can have two different values, and that for the first time we can start playing around with the poi spinning in all three dimensions thanks to possible values of d, o, and s. Given the multitude of possibilities that can emerge from all these different possible variable values, I am going to keep the description of this umbrella term as simple as possible.

Vernacular Description:

Manifolds are a family of poi moves in which the movement of the poi and hands together result in a poi path that travels through all three dimensions.

That's it. That covers everything we're about to look at.

a. Weaves

Our first subfamily of manifolds will also be the easiest to define with its variable ranges. I'm going to include both simple and complex weaves in the variable ranges we will define. This likely is one of the reasons that weaves are among the first tricks that poi spinners learn.

$a \geq 0$
$|b| \geq 0$
$a + |b| \leq 4$ (depending on the armspan of the performer)
$c = \pm 1$
$d > 0$
$m \geq 1$
$n \geq 2o$
$o > 0$
$p = 0$
$s = 0$

In this case, we are shutting down p and s values because we're not dealing with a toroid. We're also ensuring that that we have at least 2 poi beats for every oscillation of the hand across the z axis given that I've never heard of a ½ beat weave.

Vernacular Description:

Weaves are a subfamily of manifolds wherein the poi completes multiple rotations in the x and y axes, bending between parallel planes by the hand dragging the poi back and forth across the z axis in the time it takes the hand to return to its origin.

Please note: as we discussed in the section modeling weaves, this covers everything from 3-beat weaves to corkscrews, thread-the-needles, windmills, barrel rolls, and countless other moves to distinguish between which we would need to define the location of the poi relative to the body and include the timing and direction between the two poi. This is outside the scope of this paper but will hopefully be included in a follow-up.

b. Body Tracers

Body tracers are thankfully even more simple to set the variables for than weaves were. We can actually copy most of our variable values from flowers to achieve this result with only minor tweaks to our z axis terms.

$a = 0$
$0 < |b| < 4$ (depending on the performer's poi and their armspan)
$|c| = 1$
$d = 1$
$m = 1$
$|n| \geq 1$
$o = 1$
$p = 0$
$s = 0$

Here, the only variable values that have changed at all from our flower values are the values for d and o. These two variables in effect give us a single oscillation along the z axis. Now, one may look at these variable values and state that they fall within the range of values we've defined for weaves--and you would be absolutely right. It has been noted that 3-beat weaves can also be conceptualized as 2-lobe inspin flowers performed as body tracers with the radius of the handpath reduced to near 0. With this in mind, our vernacular definition can lean heavily on our definition of weaves.

Vernacular Description:

Body tracers are a specialized subset of weaves wherein the handpath completes a rotation along any two pairs of axes while the poi completes 1 or more beats bent between planes parallel to one pair of the handpath's axes.

c. Toroids

Our final set of elementary patterns, toroids will require the most diverse us of variable values we have yet seen, including for the first time the use of both p and s. In theory we could also turn toroids into weaves by giving a value to a, but again, it is outside the bounds of what this paper hopes to achieve.

$a = 0$
$0 \leq b \leq 4$ (depending on the performer's poi and their armspan)
$c = 1$
$d = 1$
$m = 1$
$|n| \geq 1$
$o \geq 0$
$p = o$
$0 \leq s < 2\pi$

Now there is only one variable in our model for which there is no value at all. Amazingly, the majority of our variables have scarcely from the body tracer model with two significant shifts: now p has been locked to the value of o, meaning it is now p and o that determine the number of lobes in the resulting pattern. Also, we have for the first time established a value for s. Given that s is the variable that determines the overall phasing of the poi path within the pattern, it's silly if it is to be used to allow it to be either 0 or 2π given that either value would in effect cancel out the effects of tweaking the phasing of the pattern.

You'll note I am also using the term lobe to apply to structures that appear within toroids, despite the fact that in the most rigid sense they don't fit the definition of lobes that I included in the section on the structure of flowers. The problem is compounded by the fact that with toroids there are two different possible points of reference from which structures analogous to lobes and antilobes can be derived. For instance, are we discussing the lobes produced by the rotation of the poi plane or by the path of the poi across said rotating plane? The answer can change depending on the type of toroid we are describing given that if we follow the movement of the plane, an isobend seems analogous to an extension, producing an endless lobe. To complicate matters, the poi can complete multiple beats within an isobend, creating patterns that when viewed from head-on can strongly resemble both lobes and antilobes. For both antibends and probends, however, it is fairly easy to produce patterns in which the plane rotates in a way that produces multiple lobes and antilobes and then assign each lobe its own poi beat to arrive at patterns that are able to assign these definitions to both the shape created by the rotating poi plane as well as the resulting poi plane.

So how to talk about lobes and antilobes in toroids in an internally consistent way? Honestly, there really isn't one. The best we can hope to do is define the type of toroid and then treat its lobes/antilobes as unique situations.

For isobend toroids, given that they are analogous to extensions, we can reasonably say that they lack antilobe structures but define their lobes as the regions in which the poi crosses the plane of the handpath traveling in the positive direction of the z axis.

We can, however use the same definitions of lobes and antilobes interchangeably between antibend and probend toroids. For these two types:

Toroid Lobes are regions in which the plane of the poi aligns tangent to the handpath.

Toroid Antilobes are regions in which the plane of the poi aligns perpendicular to the handpath.

Utilizing these definitions provide us a credible way to talk about the structure of toroids so that analogous structures between toroids and flowers are referred to with the same vocabulary. This means that just like with flowers, antibend and probend toroids display an equivalent number of lobes to antilobes while isobend toroids display only lobes--just like extensions in flowers.

Vernacular Description:

Toroids are a family of manifolds wherein the hand completes a rotation in any two axes while the poi completes multiple rotations in all three pairs of axes by rotating the plane of the poi's rotation on an axis perpendicular to the plane of the the handpath.

With that in mind, we can define the three subsets of toroids as such:

i. Isobend Toroids

Fortunately, isobends require only minor tweaks to our previous toroid equation and in the end simplify the equation slightly.

$a = 0$
$0 \leq b \leq 4$ (depending on the performer's poi and their armspan)
$c = 1$
$d = 1$

$m = 1$
$n = 1$
$o \geq 1$
$p = o$
$s = 0$

That's it! The only variable we really need to worry about here is o, which simply defines how many antilobes are produced in the pattern.

Vernacular description:

Isobend toroids are a subset of toroids wherein the plane of the poi completes a single rotation per handpath and the poi may complete one or more beats in the course of a single handpath.

ii. Antibend Toroids

For antibend toroids, we will have to set our poi plane rotating in the opposite direction as the handpath.

$a = 0$
$0 \leq b \leq 4$ (depending on the performer's poi and their armspan)
$c = 1$
$d = 1$
$m = 1$
$n < 0$
$o = |1 - n|$
$p = o$
$s = 0$

By setting n to be a number less than zero, we are setting it to rotate in the opposite direction as the handpath, whereas we can derive the value of o (the number of x, y, and z oscillations for the poi) from two equations we outlined earlier, where we can derive the number of lobes u from n with $u = 2(1 - n)$ and the number of x, y, and z oscillations from u with the equation $o = \left|\frac{2(1-n)}{2}\right|$, which simplifies to $o = |1 - n|$.

Vernacular Description:

Antibend toroids are a subset of toroids wherein the poi plane rotates in the opposite direction as the handpath and the poi completes as many beats as the number of lobes that result from the poi plane's rotation.

iii. Probend Toroids

Our final toroid type, probend toroids will only require minor tweaks to our antibend equations to switch the direction of the poi plane rotation and the phasing of the resulting poi pattern.

$a = 0$
$0 \le b \le 4$ (depending on the performer's poi and their armspan)
$c = 1$
$d = 1$
$m = 1$
$n > 1$
$o = |1 - n|$
$p = o$
$s = \pi$

That's it! The value for n has now become a positive number greater than 1 to ensure we are not modeling an isobend, but aside from that no changes are necessary to our value for o. We've changed the phasing of the pattern by setting s to phase the pattern halfway off (π is ½ of 2π, the number of radians in a complete circle).

Vernacular Description:

Probend toroids are a subset of toroids wherein the poi plane completes multiple rotations in the same direction and amount of time as a single handpath. The poi completes a number of beats equivalent to the number of lobes that result from the poi plane's rotation.

D. Conclusions

Here you are at the end--congratulations, you made it! You now know most of what I do about poi math. When I first started working on poi from a mathematical perspective, I had no idea both how far down that particular rabbit hole I was going to wind up exploring, nor the diversity of movement I was going to be able to describe with it. While we've covered a lot of ground in this paper, there are clearly many different possible variable values and combinations that have not been addressed in the course of this paper that may yield interesting results. We have barely scratched the surface on the hand moving in all three dimensions and modeling composite patterns is still in its infancy. There are scores of opportunities to expand and flesh out this model to describe a host of additional movements.

One great uncrossed frontier is the modeling of weaves using trig as it requires tracking the phasing of the poi in multiple axes and how the poi moves around obstacles to create its final path. The litany of weave-based inside moves and how they vary depending on the phasing of the poi within the pattern is another area ripe for exploration and modeling.

For now, I hope the reader has learned something new or at the very least received some moments of entertainment from reading this paper. It wound up being a vastly larger undertaking than I'd originally expected, but one I have not been able to put down nor truly focus on anything else while it lay incomplete. It is the longest written work I've produced in my life, so I hope it winds up being useful to someone other than myself at some point.

If you have any questions, comments, or corrections, please feel free to email me at drex@drexfactor.com. I welcome feedback both positive and negative as well as the opportunity to correct any mistakes I might have made in the multiple insomnia-fueled nights that led to the completion of this document.

Cheers.

E. Thanks and Acknowledgements

The one and only reason I have any shred of a reason to call myself a mathematician, hobby or otherwise, and by extension this paper exists due to a short but patient lesson from the one and only Adam Dipert. Those ten minutes spent learning the basics of modeling poi with trig functions literally changed my life from that of an artist who believed his entire life he was bad at math to an individual who now explores advanced math concepts for fun. It is often said we have little comprehension of the footprint we leave in this world--Adam, you may be a little guy, but the footprint you're leaving is that of a giant. Thank you, thank you, a million times thank you!

It's likely I would have forgotten most of what I'd learned from Adam had I not had the opportunity to practice my knowledge of the math through writing a program to model poi with Will Ruddick during my stay with him in Kenya. Teaching poi on the streets of Mombasa while we wrote that program at night is entirely too much awesome for any single human being to get to claim they got to participate in in one lifetime. Will made it possible for me to test my understanding of the math in visual form that radically changed the way I conceptualized poi. In addition, he's an amazing human that is doing fantastic work raising people in Kenya out of poverty through the use of local currencies. If you are reading this, you should visit http://koru.or.ke/bangla and learn how Will is changing lives while I spin balls on strings.

As I explored the math more deeply over the course of many years, I could ask for no better partner in crime than Charlie Cushing. We learned and explored together over phone calls, emails, video chats, exchanging Python programs over nips of scotch (I guarantee I am making our exchanges sound vastly classier than they truly were). Every curious mind should be so lucky as to find a partner who matches or exceeds their facilities, curiosity, and drive to push forward. Thank you for the most engaging moments of my spinning life, old friend :)

I would be remiss if I did not credit the original master of poi theory as well, Alien Jon Everett, for his inspiration, wonderful suggestions, and pioneering work expanding the vocabulary of the poi world. The language and concepts he pioneered are integral to the DNA of this paper. My understanding of poi owes you a debt I shall never be able to repay, my friend.

I would also like to thank Damien Boisbouvier for his work on CAPs and third-order motions. In many ways I believe we've been working on parallel tracks for years and again, the concepts Damien has pioneered are also deeply ingrained in the DNA of this paper.

Thanks are also in order to Jonathan Alvarez, whose post to Poi Chat on Facebook[15] asking if the definitions of poi moves were available anywhere on the web was the catalyst for the writing of this paper. I hope I have made some progress in addressing the need Jonathan identified.

Also, I would like to thank all the people outside the spinning community who've contributed directly and indirectly to making this paper happen. My girlfriend Debbi for putting up with my obsessive attempts to complete this paper in less than a week (no, I'm not joking), my customers and clients for patiently waiting while I completed this work and could think of absolutely nothing else, and the many friends who patiently listened to my description of my breakthrough modeling toroids the night I finally made the necessary connections after three interminable days running into dead ends.

I'd like to thank you, the reader for picking this paper up. I can't imagine the list of people who are curious enough about this topic to read through nearly 70 pages of it is terribly long, so I salute you for your patience and curiosity.

Finally, I'd like to thank those folks who took the time out to edit this paper--combing through my endless run-on sentences and equations, ensuring that I'd made no grammatical errors. Thanks to Jennifer Longo (Jexime) for taming my long-winded prose and Pierre Baudin for looking over my math. I am greatly indebted to the both of you. Another shout-out is due to Adam once again for looking over this paper and offering a number of helpful suggestions for making many of the mathematical ideas more clear and concise. Seriously, you rock, man :)

[15] https://www.facebook.com/groups/189779111046713/permalink/732862366738382/?stream_ref=21