

Let the Numbers Do the Walking: Generating Turtle Dances on the Plane from Integer Sequences

Adam Colestock
4th-8th Grade STEAM Teacher
Francis W. Parker School
Chicago, IL 60614, USA
acolestock@fwparker.org

There is no harm in repeating a good thing.
-Plato

Abstract

I tell a story of a dancing turtle and introduce a mathematical form based in turtle geometry and modular arithmetic. Sequences of integers generated from a discrete parametric function determine the turn angles for a single meandering point and produce symmetric and varied designs on the plane. I also analyze the mathematical properties of the form, highlight emergent features within the form set and show designed objects incorporating these patterns. Finally, I propose several variations of the initial algorithm that hold promise for future inquiry and mathematics-driven making.

Introduction

Let us begin with a story:

Maud is a turtle. She loves to go on walks. She is happiest when she is ambling along, letting her thoughts slowly drift as she picks up her left front foot, then her right back foot, then her right front foot and finally her back left foot, before beginning a gait cycle again. Her home is in the middle of a large meadow with ample room to roam. For a long time, Maud lived in fear that she would get lost on one of her walks and would not be able to find her way back. She would venture out but always walk within sight of her home.

Maud was methodical in her movements. She could walk and turn with the precision of a dancer. Each step she took was equal in measure and she could pivot in place by an exact angle. She enjoyed using her body to move with poise and accuracy and would choreograph sequences of moves and turns to create her own dances. She would practice them over and over until she could do them without thinking and her favorites would get their own names. She would whittle away hours dancing the *Widow's Waltz*, the *Penguin Promenade*, the *Falling Star* and others.

One day she set out from her home, having just invented a particularly intricate and delightful dance, the *Last Cookie*. She was so elated at having finally mastered the steps that she got swept away and performed it over and over and over again. She danced one *Last Cookie* after the next without pause or interruption. Finally, when she noticed that the light had started to change and the day was dwindling, she snapped out of her trance and with a growing dismay that felt like an eel wriggling in her stomach, realized that her home was nowhere to be seen.

Feeling lost and stranded, and unsure what to do, she tried to come up with a plan. But standing still, with her feet frozen to the ground, was her least favorite way to think. So she decided to do just a few more *Last Cookies* to help her cogitate a bit on a solution to her dilemma. She closed her eyes, to really

help focus her thoughts, and let her mind wander as her body effortlessly slipped back into motion. So caught up in the movement did she become, once again, that she forgot what problem she was trying to solve as she joyfully and gracefully stepped and turned her way through the meadow. Just as her body had once again succumbed to exhaustion, she finished off one *Last Cookie* and, opening her eyes, felt a flood of relief wash through her body as she found herself on the very doorstep that she had left earlier that day. Home!

This story illustrates the essence of the mathematics that I will explore in this paper. Maud was surprised to find that repeating her dance returned her to her exact starting point after some number of iterations. Would this happen for all possible dances or was she fortunate to have picked this one? Given a dance, can one predict how long it might take to complete the full journey? What would the dance look like from above if she were leaving tracks as she moved? The story of Maud and the questions that it suggests served as the basis for a recent episode of mathematical inquiry in which I developed an algorithm for creating symmetric 2D designs from a sequence of integers produced by a discrete parametric generating function.

Defining the Dance

Turtle geometry provides an intuitive way to describe the process of a point moving through space and tracing out a path. This simple way of mathematizing a line, first proposed by Seymour Papert [1] and further elaborated and explored by Abelson & DiSessa [2], provides a procedural description of drawing a line that has been implemented in the programming language LOGO and subsequent descendant languages such as Scratch. Turtle geometry provides a rich and accessible Mathland for people to learn geometry and engage in mathematical inquiry. There is also an active and vibrant community of art that has been produced using the framework of turtle geometry. A multitude of turtle art programming environments are available for interested artists [e.g. 3]. Mathematical art using turtle geometry has included work that mimics river patterns in nature [4], explores how certain infinite sequences can generate fractals [5], and provides techniques for producing 3D sculpture and analyzing its symmetries [6].

The moving point, or ‘turtle’, has 2 states that change as it moves. The turtle has a **HEADING**, a number between 0-360 degrees that indicates the bearing, similar to a compass reading. The turtle also has a **POSITION**, a 2 dimensional vector that determines where on the plane it is located. The **HEADING** changes by having the turtle **TURN** or rotate some number of degrees. The **POSITION** changes when the turtle is commanded to **MOVE** forward or backward by some number of steps.

Turtle geometry provides a system to describe and track Maud’s movement through the meadow. But how does she decide what sequence of turns to make to create her dance? For Maud’s dance, let’s imagine that she always takes one step forward and then turns by an amount determined by the following discrete parametric function with parameters A and B:

$$f(n) = n \bmod A - n \bmod B$$

As n goes from 1 to infinity (only taking on natural number values), this function produces a sequence of integers that can be used to determine the angle by which the turtle will turn after each step as it moves around the plane.

Figure 1 shows several examples of what happens when this algorithm is applied and the output of the generating function determines the turtle’s turn angle. Although all of the dances shown here are closed and retrace their paths, other dances (such as when $A=36$ and $B=20$) do not close but rather remain open and result in a translation of a base curve over and over again across the plane and off to infinity.

A few observations about closed paths emerge based on these examples:

- 1) Dances retrace their path after a finite number of steps
- 2) Some dances begin to retrace their path quickly while others take longer
- 3) Dances exhibit rotational symmetry but the orders of symmetry differ

These three observations suggest a set of mathematical questions that we might want to answer about these forms. Given specific values for parameters A and B , how can we determine:

- 1) Is the path open or closed?
- 2) If it is a closed path, how long will it take for the path to start to retrace itself?
- 3) What order of symmetry will a closed form exhibit?

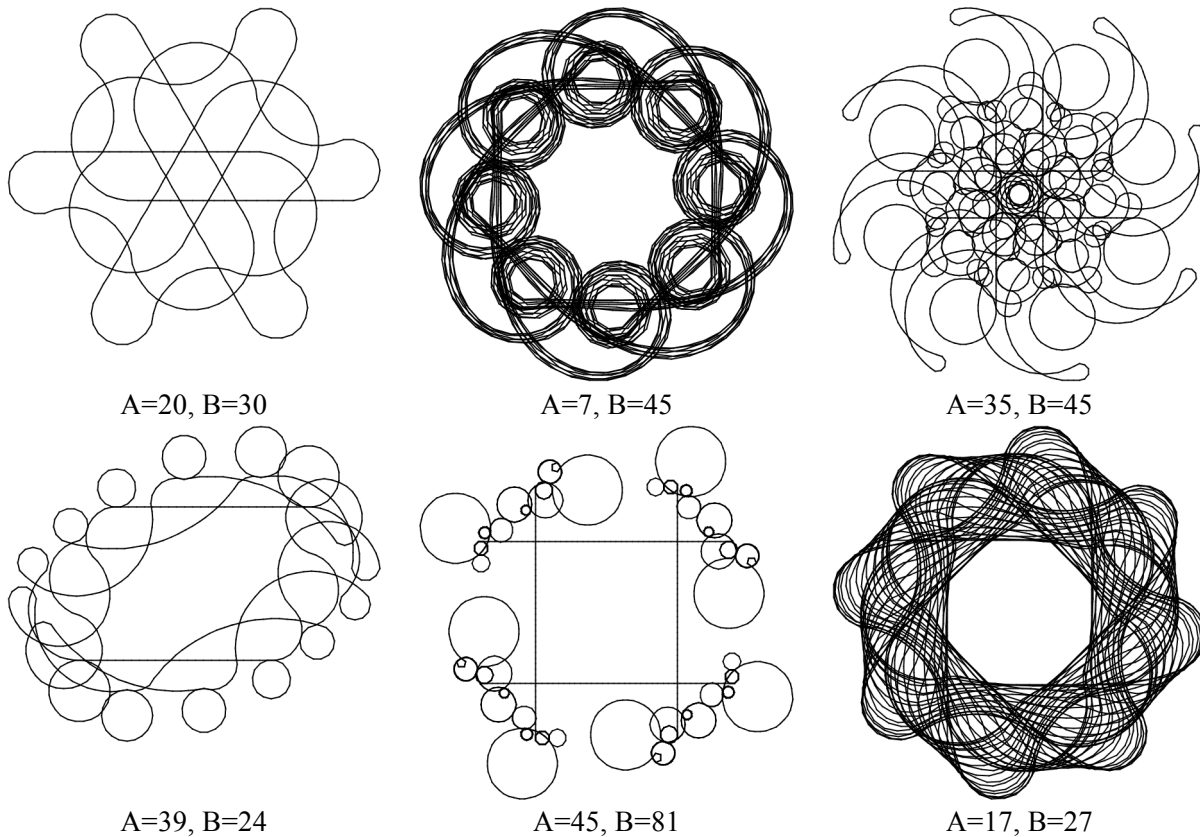


Figure 1: A sample of dances produced for different values of A and B

Understanding the Dance

We now turn to considering these three questions.

Open vs. closed paths. When is the path produced closed and when is it open? If we define the *period* of the generating function as the number of iterations before the sequence of integers begins to repeat, then for any integer values A and B , the period will be equal to the least common multiple of A and B (I will write this as $LCM(A, B)$). Each *move* of the turtle consists of a single step followed by a turn as determined by the generating function. Let us define a *cycle* of the dance as the sequence of moves produced by our generating function over a single period.

First, consider the case in which the accumulated total turn angle is divisible by 360 and thus the HEADING returns to 0 after a single cycle. Since the HEADING has returned to 0, subsequent cycles of moves will result in repeated identical translations of the base curve, resulting in an open path that is unbounded. Alternatively, if the HEADING doesn't return to 0 after a single cycle, then the result is some displacement in POSITION and a change in the HEADING as determined by the accumulated total turn angle at that point. Subsequent iterations of a cycle of the dance will eventually return the turtle to its original HEADING and POSITION when the accumulated total turn angle is divisible by 360. The guarantee that the HEADING will eventually return to 0 follows from the constraint that we are working in degrees and using integer values and avoiding irrational numbers. But when does this happen with the particular integer sequences produced from our generating function?

In order to determine this for our sequences, we need to know what the HEADING will be after $\text{LCM}(A,B)$ moves. The generating function consists of a positive component that depends on mod A and a negative component dependent on mod B. When calculating the total turn angle after $\text{LCM}(A,B)$ moves, it is helpful to consider these separately. For a single cycle of A turns, the term $n \bmod A$ will take on the values $1, 2, 3, \dots, A-1$ and therefore turn a total of $A(A-1)/2$ degrees in $A-1$ moves for an average turn of $A/2$ degrees. Similarly for every B turns, it will turn a total of $B(B-1)/2$ degrees in $B-1$ moves for an average turn of negative $B/2$ degrees. Therefore after $\text{LCM}(A,B)$ moves we have that the total turn angle is $\text{LCM}(A,B) * (A/2 - B/2)$.

Recall the form is only open when the HEADING after a single cycle is equal to 0. Thus the form will be open precisely when $(\text{LCM}(A,B) * (A/2 - B/2)) \bmod 360 = 0$ and otherwise it will be closed.

Orbit length. The *orbit length* or number of moves before the turtle starts to retrace its path can be calculated using the HEADING after $\text{LCM}(A,B)$ moves. The number of iterations of a single cycle required for the HEADING to return to 0 is equal to the LCM of the HEADING after a single cycle and 360. That is:

$$\text{Orbit Length} = \text{LCM}((\text{LCM}(A,B) * (A/2 - B/2)) \bmod 360, 360)$$

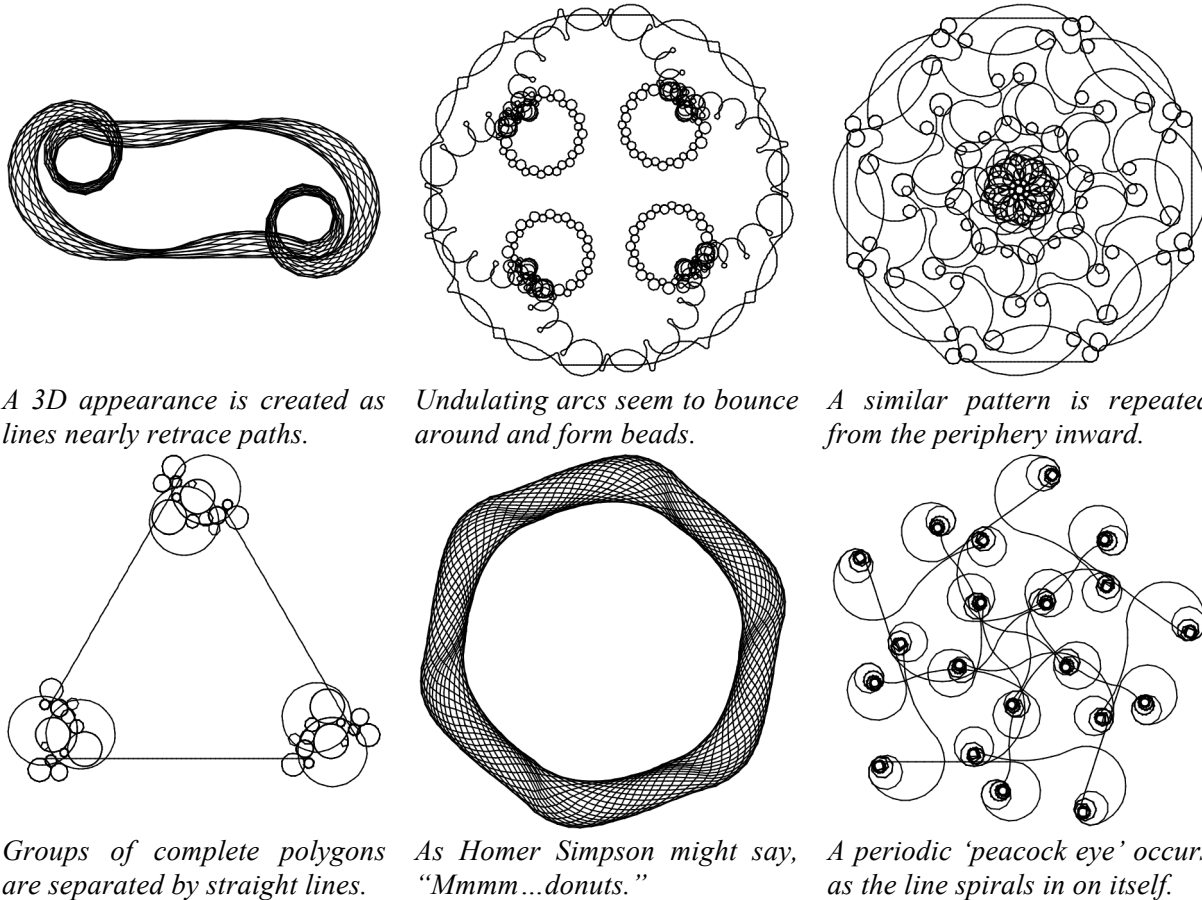
Symmetry order. The symmetry order of the resulting shape is determined by how many times a single cycle of $\text{LCM}(A,B)$ moves has to repeat itself before returning to the initial position with a HEADING of 0. Now that we have calculated the Orbit Length, we can use this value to calculate the symmetry:

$$\text{Symmetry Order} = \text{Orbit Length} / \text{LCM}(A,B)$$

I used these calculations for the orbit length in the Processing [7] programming language sketch that generated the images in this paper to determine when to terminate the algorithm.

Designing with the Dance

Exploring the form set. After delving into the mathematics of Maud dances and writing code in Scratch and Processing, I started to explore the diversity of designs that emerge from this simple algorithm. Slowly cycling through different values for A and B, I was engrossed in exploration, surprised to see some familiar forms and delighted to find new subtle effects that can be produced by the simple act of drawing a line and then repeating that line over and over until one returns to the start. I began to catalog some of the recurring features of the forms. I now give some examples to highlight a few recurring features and characteristics that I observed among the forms.



A 3D appearance is created as lines nearly retrace paths.

Undulating arcs seem to bounce around and form beads.

A similar pattern is repeated from the periphery inward.

Groups of complete polygons are separated by straight lines.

As Homer Simpson might say, "Mmmm...donuts."

A periodic 'peacock eye' occurs as the line spirals in on itself.

Figure 2: *A description of some features that emerge among the dances*

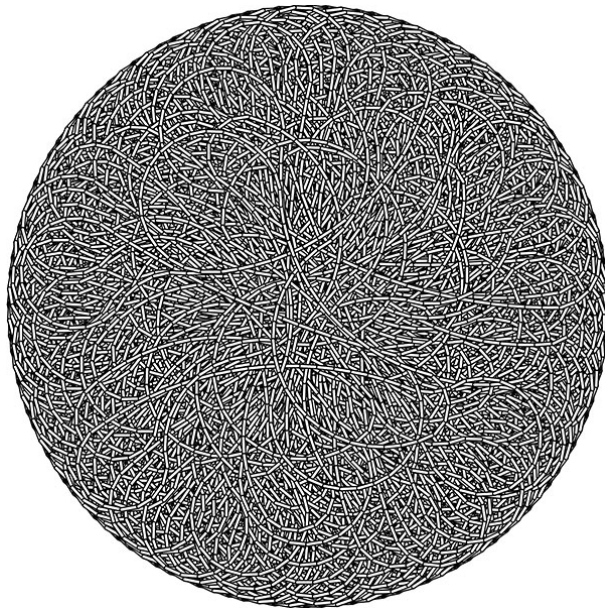
Presumably, the emergence of these forms is reflecting something about the nature of the relationship between the parameters A and B. With further exploration and analysis, I hope to be able to predict what features will appear given any values for A and B.

Fine tuning with a rational scalar. As I began to explore dances produced by larger values of A and B, the integer sequences would get longer, resulting in more complex and intricate patterns. However, the turn angles also increase, resulting in tight spirals with dense pockets of overlapping lines that make it difficult to perceive this increasingly complex structure. One technique that I've begun to employ is to use a rational scalar to scale down the integers in a particular turn sequence so as to limit how much the turtle might turn after each step. This involves slightly modifying the generating function to:

$$f(n) = \left(\frac{p}{q}\right)(n \bmod A - n \bmod B)$$

For any particular value of A and B, this technique affords an abundance of variations of the form. For values of A and B that produce very small turns, the rational scalar can be used to increase the turns into a range that is more readily perceptible. And similarly, for values of A and B that produce very large turns, the rational scalar can be used to 'open them up' and create more gradually meandering paths.

Incorporating into designed objects. With such varied forms to pick from, I began to think about how I might use these 2D designs in the service of artistic expression. I found that different dances spoke to me in different ways, with different forms seeming better suited to certain applications. I also became interested in the different effects that might be realized in the selection of stroke style and color when trying to develop the basic form into more fully realized designs. Figure 3 shows two examples of objects I have created based on these forms so far.



Above: A design with an orbit length of 75,240 and 120-fold symmetry completely fills a circle with a serpentine path

Below: A steel side table with the dance for A=45, B=25 waterjet cut to form the top



Figure 3: Two examples of art based on the turtle dances

The Dual of the Dance

Thus far we have used the sequence of integers generated from our generating function to determine the turn angle for a turtle moving around the plane with each move having the same stepsize. What would happen if instead of using this integer sequence to determine the turn angles in Maud's dance, we kept the turn angle fixed and used the integer sequence to specify the stepsize of the turtle? Since we are in essence swapping the MOVES and TURNS, we might call this *the dual of our dance*. What might these dances look like? Will they have properties similar to the original dance?

To formalize this dance, we will introduce a new parameter, ANGLE, which describes the amount the turtle will turn between each move. Deriving integer sequences from the same generating function we introduced earlier, Figure 4 shows some dual dances produced from different values of A, B, and ANGLE.

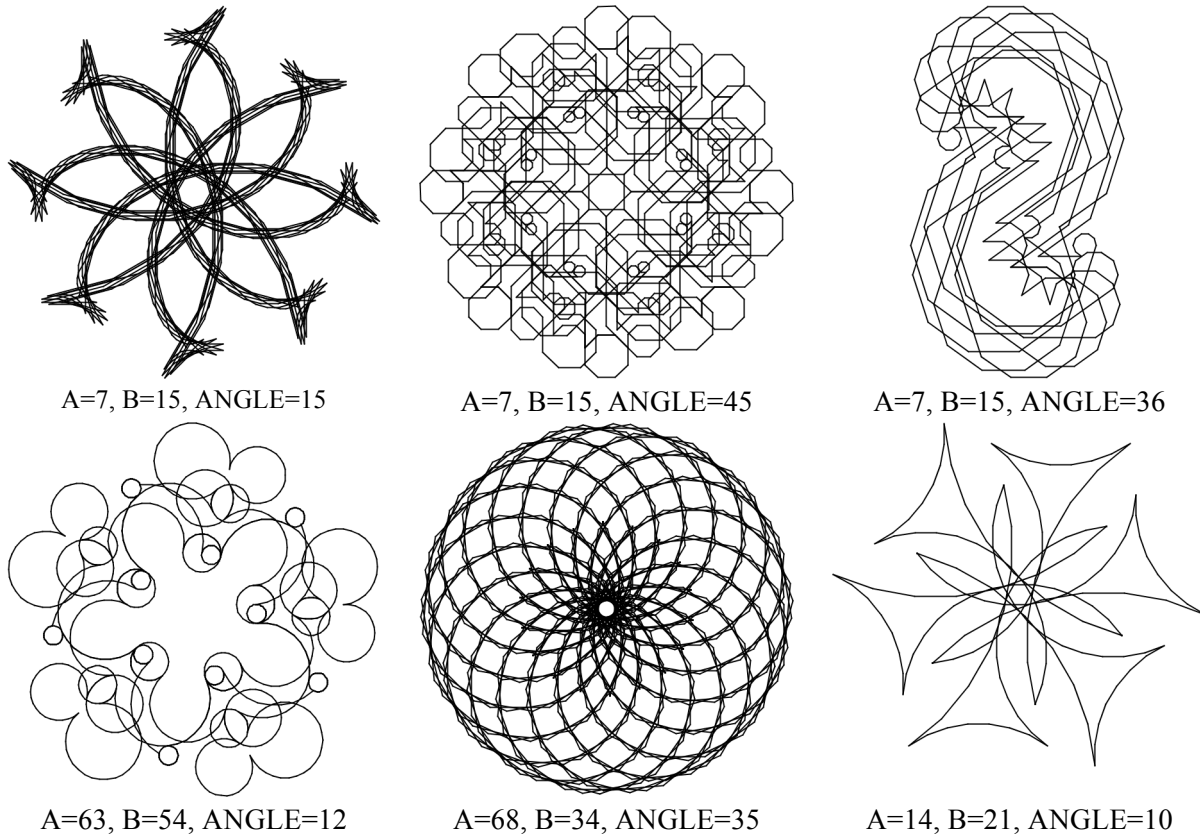


Figure 4: A sample of dual dances produced by different values of A , B , and $ANGLE$

This set of forms seems to have something in common with the original dances. Once again, the shape will be closed if the $HEADING$ after a single cycle of $LCM(A,B)$ moves is not a multiple of 360. The $HEADING$ after $LCM(A, B)$ moves is easier to calculate because the turn angle is constant, so it is just $(LCM(A,B) * ANGLE) \bmod 360$. Therefore the number of cycles before the $HEADING$ returns to 0 is $LCM((LCM(A,B) * ANGLE) \bmod 360, 360)$. The Orbit Length and Symmetry Order of these shapes can then be calculated in a similar fashion to the original dance.

Generating Dances from Other Integer Sequences

Thus far we have considered two variations of a turtle dance based on the integer sequence produced by our generating function. I selected this discrete function not at random but rather because it has certain properties such as being periodic and tapping into the relationships among natural numbers that made it a good candidate for providing an interesting variety of forms. However, the algorithm for Maud's meandering can be generalized to any integer sequence. I now turn to a few of the particular variations that I have explored.

But what if I love 7-fold symmetry? Readers with a bloodhound's nose for the arbitrary may have noted that the choice of dividing the circle into 360 degrees is a somewhat arbitrary one and limits the kinds of symmetry that can be realized since the order of symmetry of the form is limited to factors of 360. In order to achieve any symmetry desirable, we can divide the 360-degree circle into any number of unit angles of equal size that we want.

If we let a unit angle equal $2\pi / X$ radians, then we can create any number of divisions of the circle into X equal angles. Doing so would allow us to produce dances with any order of symmetry that we desire. The 7-fold symmetry seen in Figure 5 is generated from $A=10$ and $B=9$ and a unit angle set to $2\pi / 7$ radians.

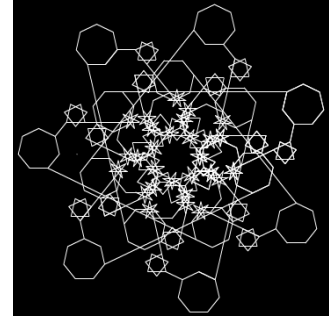


Figure 5

Discretizing continuous functions to produce sequences. The turtle dances described here use a discrete parametric equation to generate a periodic sequence of integers to control the turn sequence of the turtle as it moves. Another alternative approach would be to take a continuous parametric equation and strategically sample from it at regular intervals in order to create a turn sequence that will also be periodic and thus, eventually result in a closed form.

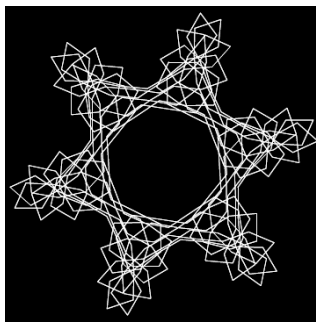


Figure 6

The design in Figure 6 is based on the parametric function $f(n)=\cos(A*n) - \cos(B*n)$ where n takes on natural numbers from 1 to infinity. The turtle is also made to slowly curve (rather than meander about a straight line) by offsetting each turn by an angle C (here $C=35$) and the form is fine-tuned through the use of a rational scalar as discussed earlier. This approach is very similar to that used by David Chappell [8] to create beautifully sinuous meandering patterns but builds off the turtle geometry algorithm described in this paper to produce forms that are a little less *bouba* (sinuous) and a little more *kiki* (spikey) [9].

Any integer sequence will do. Moving beyond our specific generating function, any sequence of integers can act as the basis for a turtle dance. Any periodic sequence has the potential to churn out a surprising and beautiful pattern. Curious mathematicians on a hunt for interesting integer sequences should be directed to the extraordinary trove of sequences in the On-line Encyclopedia of Integer Sequence [10].

References

- [1] Papert, Seymour. *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc., 1980.
- [2] Abelson, Harold, and Andrea A. DiSessa. *Turtle geometry: The computer as a medium for exploring mathematics*. MIT press, 1986.
- [3] Turtle Art. <https://turtleart.org/> (as of January 30, 2017)
- [4] Gailiunas, P. (2005). Meanders. In *Renaissance Banff: Mathematics, Music, Art, Culture*, pages 25-30. Bridges Conference. Available online at <http://archive.bridgesmathart.org/2005/bridges2005-25.pdf>
- [5] Zantema, Hans. Turtle graphics of morphic sequences. *Fractals* 24.01 (2016): 1650009.
- [6] Verhoeff, Tom. 3D turtle geometry: artwork, theory, program equivalence and symmetry. *International Journal of Arts and Technology* 3.2-3 (2010): 288-319.
- [7] Reas, Casey, and Ben Fry. *Processing: a programming handbook for visual designers and artists*. No. 6812. MIT Press, 2007.
- [8] Chappell, David. Sinuous Meander Patterns in Natural Coordinates. In Robert Bosch, Douglas McKenna and Reza Sarhangi, editors, *Proceedings of Bridges 2012: Mathematics, Music, Art, Architecture, Culture*, pages 183-190, Phoenix, Arizona, 2012, Tessellations Publishing. Available online at <http://archive.bridgesmathart.org/2012/bridges2012-183.html>
- [9] Ramachandran, Vilayanur S., and Edward M. Hubbard. "Synaesthesia--a window into perception, thought and language." *Journal of consciousness studies* 8.12 (2001): 3-34.
- [10] N. J. A. Sloane. *The On-Line Encyclopedia of Integer Sequences*. <http://oeis.org> (as of January 30, 2017)