The Ticking Clock: A Geometric Interpretation of Modular Multiplicative Inverses

Stephen E. Erfle

International Business and Management, Dickinson College, Carlisle PA; erfle@dickinson.edu

Abstract

This paper introduces users to a class of images created using the Playing with Polygons' *String Art* Excel file and the companion web file which was developed to encourage exploration of the relation between geometric and numeric patterns for young learners. These *60-second images* are created using 60 cycles in which each cycle ends one vertex beyond the previous vertex, just like the second hand on a clock. This one vertex by one vertex clockwise movement is the geometric consequence of having two of the parameters being modular multiplicative inverses.

Introduction and Motivation

At one time or another, all of us have found ourselves looking at a clock, entranced by the simple movement of the second hand as it clicks off second by second movements around the clockface. This idle introspection led me to analyze a class of Playing with Polygons images I call "*60-second images*" because one can see the image emerge as a result of 60 cycles in which each cycle ends one vertex farther along the polygon than where the cycle started, just like with the second hand of a clock. Figure 1 shows three 180-line 60-second images with hyperlink (click link then click *Toggle Drawing*) with first cycle shown in red. One can enjoy exploring 60-second images (and other images) without being concerned with how the images are created. Put another way, this paper will be of interest to those who just want to play, but the goal is to explain why the images are created in this way. To accomplish this, we must first understand the parameters underlying the string art model. After that, we examine the attributes that produce 60-second images. Next, we explore some of the range of images possible in this situation. Finally, we examine the conditions required to consider time in an alternative universe.
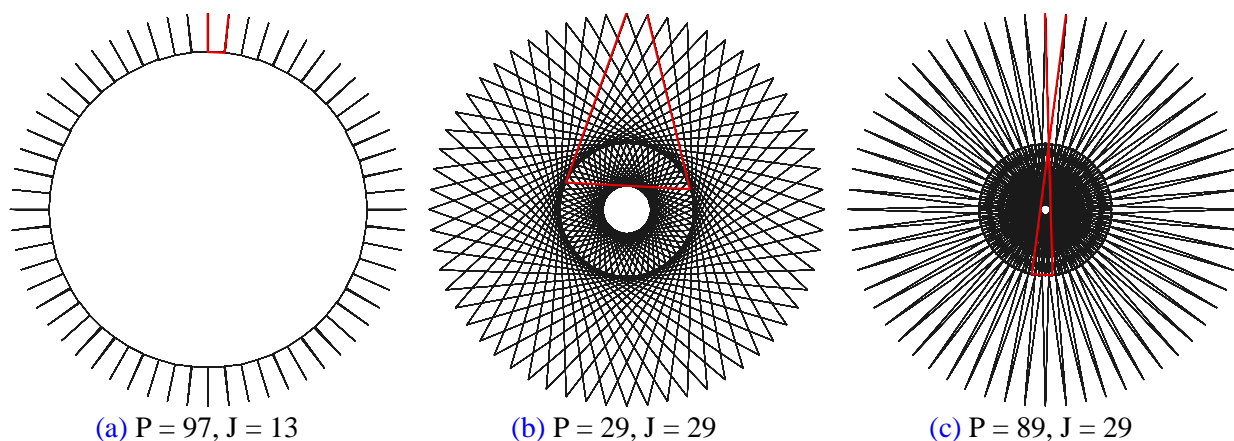

| (a) P = 97, J = 13 | (b) P = 29, J = 29 | (c) P = 89, J = 29 |

Figure 1: *Examples of 60-second images with n = 60 and S = 3.*

Exploring the Parameters Controlling the *String Art* Excel file

The paper's initial task is to get participants comfortable using the *String Art* file. This file uses the ideas set forth in Erfle and Erfle [1] with one important difference: rather than allowing users free reign in setting vertices in the (x, y) plane, vertices are restricted to be those of a regular polygon. This simplifies the file significantly because four parameters control the image and each parameter can be controlled via up and

down arrows, ⬍. This makes the file accessible to younger users because all they need do is click and watch what happens. Scroll arrows allow users to rapidly adjust values (and quickly test hypotheses, even if they are informal in nature), in much the same way I argued in [2] that one could teach fractions while creating spirals using three scroll-controlled parameters. I argue that such purposeful play can lead to learning, even if that learning is informal in nature [2, 3], and is consistent with Stein's *Triex: Explore, Extract, Explain* [7] view of how mathematics should be taught.

I provide a series of *explainers* in File 2 of [3] that may be consulted for further reference. Explainers are short documents that annotate specific aspects of each file. Here, I provide a short overview of the landscape detailed there. The file is controlled by four parameters, n, S, P, and J which can be thought of in two groups. First are n and J which determine the number of vertices in the polygon, n, and the number vertex jumps between successive lines on the vertex frame, J. For example, there are two distinct vertex frame stars for n = 7: 7&0 – 2 – 4 – 6 – 1 – 3 – 5 – 7&0 if J = 2 (the – means connect vertices in the frame); and 7&0 – 3 – 6 – 2 – 5 – 1 – 4 – 7&0 if J = 3. When J = 1 or J = n-1 the vertex frame is a polygon. Vertices are enumerated clockwise starting at the top (e.g., 7&0) so that the only difference between these two polygons is the order in which sides are drawn (clockwise for J = 1, counterclockwise for J = n-1). All endpoints of lines in the image are on the vertex frame, shown in blue in Figure 2(a) for n = 4 and J = 1.

If 1 < J < n-1 the vertex frame will be an n-point star if n and J are coprime. If $GCD(n, J) > 1$ (GCD is the *Greatest Common Divisor*), the vertex frame uses only n/GCD vertices (so, for example, there is only one <u>distinct</u> 12-point star (J = 5) but there are four 11-point stars (J = 2, 3, 4, 5). Three points are worth noting: First, only continuously drawn images are considered here (when n = 6 and J = 2, the vertex frame is a triangle); Second, the only difference in vertex frame between J and n-J is the direction in which the star is drawn (compare Figure 3(a) and 3(b)); Third, we call GCD(n, J) = *VCF*, the *Vertex Common Factor*.

Next, we create S equal subdivisions on each of the n/VCF lines of the vertex frame (S = 2 in Figure 2). This produces S·n/VCF subdivision endpoints (red dots in Figure 2(a)). The image is created from these endpoints using rules (a) – (e). (a) The top of the image is the start of the first line in the image. (b) Count P subdivisions on the vertex frame (P = 3 noted in Figure 2(a)). (c) Draw a line from the top to Pth endpoint (this first line is shown in red in Figures 2 and 3). (d) Count out P more subdivisions and draw a line from the end of the previous endpoint to the new endpoint. (e) Repeat (d) until the endpoint is once again the top of the figure. The final image will use all subdivision endpoints if P and S·n/VCF are coprime. The number of lines, L, is L = (S·n/VCF)/SCF where *SCF* = GCD(S·n/VCF, P) is the *Subdivision Common Factor*.

(a) Vertex frame in blue and first line in red   (b) Completed image with first line overlay
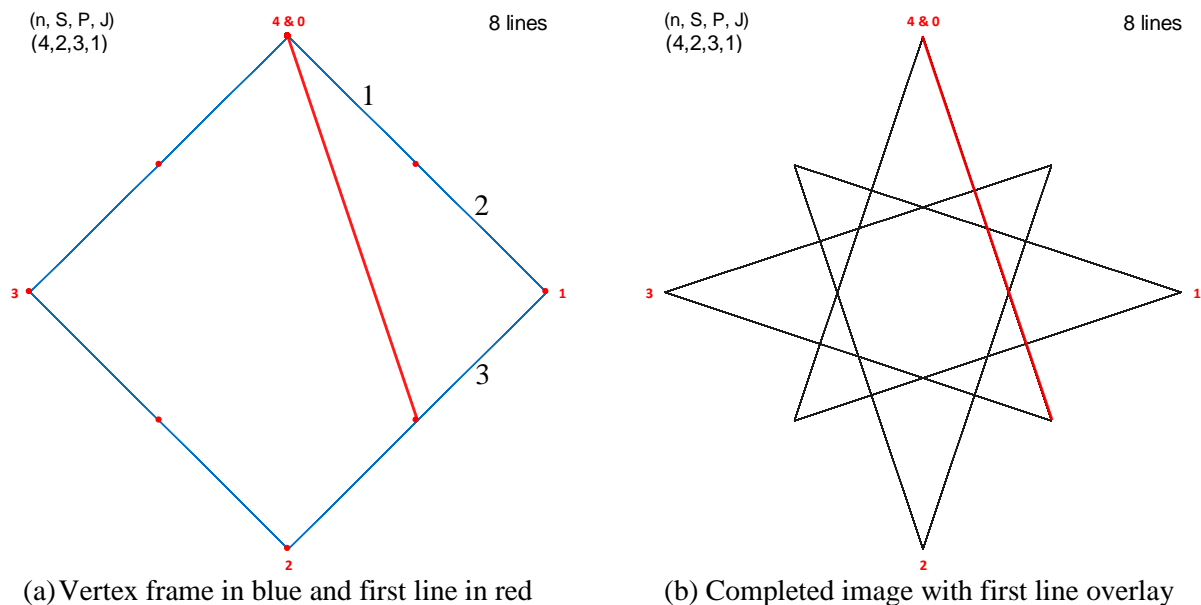
Figure 2: *Building the n = 4, S = 2, P = 3, J = 1 image using the String Art Excel file.*

Figure 3 examines the consequence of changing three of the four parameters, one at a time. The vertex frame is a triangle regardless of J (because n = 3) but changing J from 1 to 2 in moving from 3(a) to 3(b) reverses the order that the vertex frame is drawn (3&0 to 1 to 2 to 3&0 for panels 3(a), 3(c) and 3(d) but 3&0 to 2 to 1 to 3&0 for panel 3(b)) and therefore changes the order in which the lines are drawn in 3(a) and 3(b). Comparing 3(a) and 3(c) shows what happens when S changes from 3 to 5 (when S = 4 the image is the triangular vertex frame). When S > P the vertex frame is part of the image as we see in 3(c) and 3(d) but not if P > S (like in 3(a) and 3(b), as well as 1 and 2(b)). Both 3(a) and 3(d) require 9 connected line segments to create the image but those segments are more apparent in 3(a) than 3(d). After all, it looks like you could draw 3(d) with 6 segments, but this is only true if you lift your pencil (meaning draw disconnected segments). String art images are created by following a counting rule using connected segments.
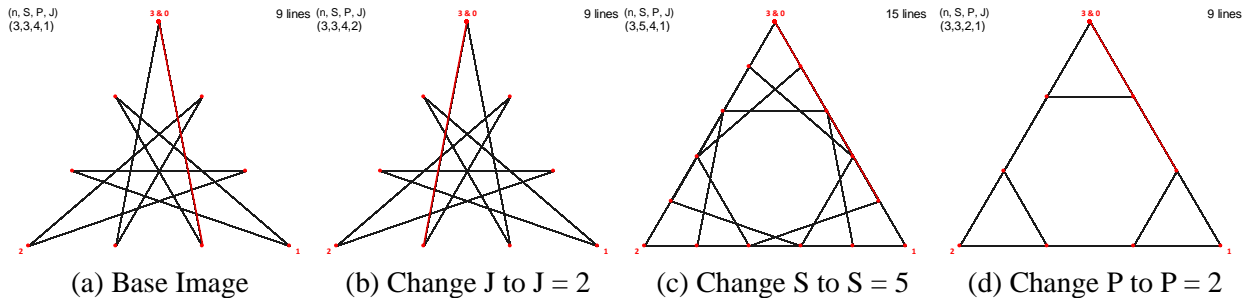

(a) Base Image   (b) Change J to J = 2   (c) Change S to S = 5   (d) Change P to P = 2

Figure 3: *Varying one parameter at a time given Base Image values of n = 3, J = 1, S = 3, and P = 4.*

Cycles

The final image is created from connected lines. At least one of the endpoints must be the vertex of the parent polygon (because the top is always an endpoint) but other polygonal vertices may be excluded from the final image as Figures 4(a) and 4(c) show. If more than one vertex is used in the final image, then that image will have rotational symmetry equal to the number of vertices used. This means that to understand the image, one need only focus on the part of the image that is created from one vertex to the next time a polygonal vertex is used. *We call this part of the image a **cycle**.* The first cycle, noted in red in Figure 4, ends at vertex 2 in 4(a), 1 in 4(b), 4&0 in 4(c), and 1 in 4(d). In each panel of Figure 1, the first cycle ends at vertex 1. In Figure 2, the first cycle ends at vertex 3 and in Figure 3, the first cycle ends at vertex 1 in 3(a), 2 in 3(b), 1 in 3(c), and 2 in 3(d). Note that the fraction of subdivisions used in each image is 1/SCF.
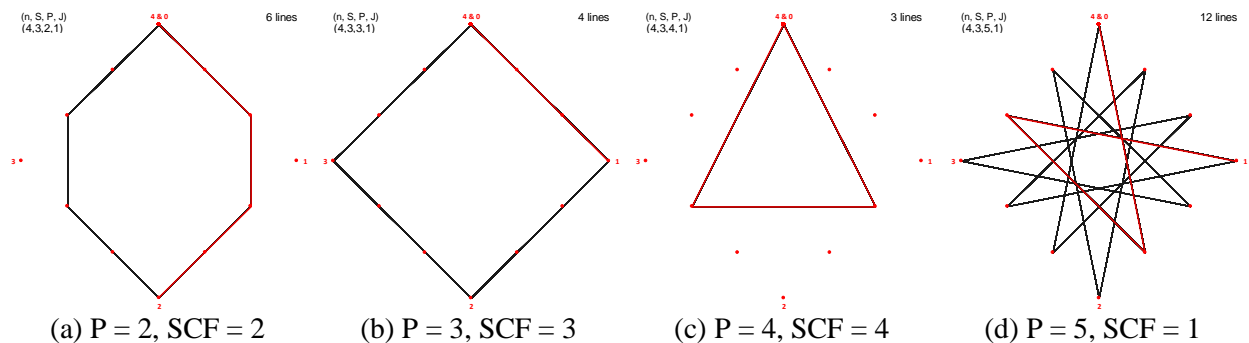

(a) P = 2, SCF = 2   (b) P = 3, SCF = 3   (c) P = 4, SCF = 4   (d) P = 5, SCF = 1

Figure 4: *Images with n = 4, S = 3 and J = 1 showing first cycle in red.*

The number of lines per cycle, C, is given by C = S/GCD(S, P) and the number of cycles in the image is L/C. Figure 4(a) has two 3-line cycles, 4(b) has four 1-line cycles, 4(c) has one 3-line cycle and 4(d) has four 3-line cycles. Each panel in Figure 1 has sixty 3-line cycles. Figure 2 has four 2-line cycles and all panels in Figure 3 have three 3-line cycles except for 3(c) which has three 5-line cycles.

60-Second Images and Modular Multiplicative Inverses

We are now able to analyze 60-second images. We require 60 vertices to be used so we restrict our analysis to n = 60 and J coprime to n, VCF = GCD(n, J) = 1. Additionally, we want S to be coprime with P so that cycles have S lines per cycle and a total of S·n lines in the image. Figure 5 provides two candidate images with S = 5 and J = 29 with hyperlinks to the web version at labels (a) and (b) [5]. If the first cycles were not shown in red, you would be hard-pressed to see a difference between the two images.
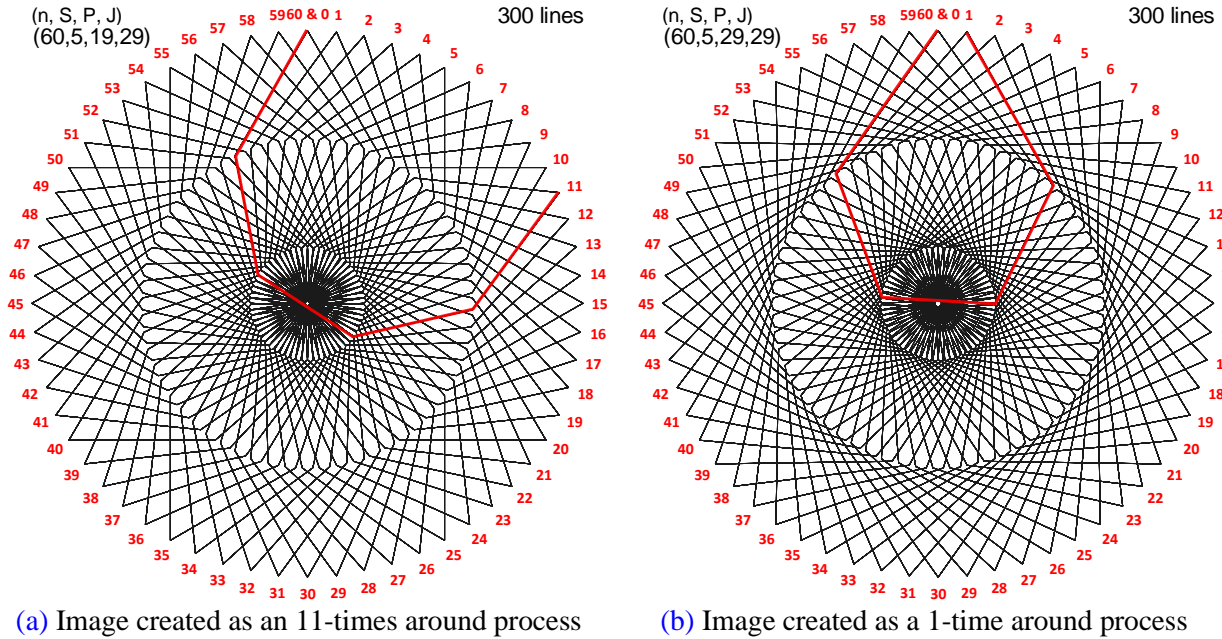

(a) Image created as an 11-times around process   (b) Image created as a 1-time around process

Figure 5: *Comparing two n = 60, J = 29, S = 5, 300-line images.*

Consider how Figure 5(a) is drawn. The first cycle ends at 11 so subsequent cycles end at 22, 33, 44, 55, 6, 17, … (66 MOD 60 = 6 and 77 MOD 60 = 17). Note that each time "around" uses one more of the vertices from 1 to 10 that were skipped because the end of the first cycle was 11. The image appears to be a rotating 5 petal flower (thanks to Liam Myles [5], you can see this happen by clicking the hyperlink at (a) and then clicking *Toggle Drawing*) that is completed once all vertices from 1 to 10 are filled in with the last line connecting to the first at vertex 60&0. By contrast, 5(b) is completed as a 1-time around process because each cycle ends one vertex after the prior cycle until the image is complete after sixty 1-vertex forward cycles when viewed dynamically via the hyperlink at (b).

You may have noticed that 5(b) is the second 60-second image shown with J = 29 and P = 29 (the first being 1(b) where S = 3). Additional versions occur as long as GCD(S, P) = 1 because one cycle is S lines or P·S subdivisions, but every S subdivisions is a vertex that is J away from the prior vertex so that P jumps of J is the ending vertex relative to the starting vertex. The ending vertex is 1 larger than the starting vertex if P·J is 1 larger than a multiple of 60. Put another way, **P and J are multiplicative inverses modulo 60**. Modular multiplicative inverses have been used to create art [6] as well as cryptography [4]. Table 1 shows the 16 values of J that are coprime with 60 and the paired values of P producing 60-second images. These pairs are not as hard to find as you might imagine because P·J must have a last digit of 1. This requires that the last digits of J and P are both 1, or both 9, or that one has last digit 3 and the other 7.

Table 1: *Jump, J, and smallest subdivisions between points, P, pairings producing 60-second images.*

J	**1**	7	**11**	13	17	**19**	23	**29**	**31**	37	**41**	43	47	**49**	53	**59**
smallest P	**1**	43	**11**	37	53	**19**	47	**29**	**31**	13	**41**	7	23	**49**	17	**59**

Additional P possibilities exist for a given J. Distinct 60-second images occur for P + 60·k for $0 \leq k < S$. Figures 1(b) and 1(c) are the k = 0 and k = 1 versions of J = 29 and P = 29. You should verify that k = 2 (i.e., P = 149) produces a third version but P = 209 is an identical image to P = 29. The reason for this identity is straightforward: there are S·n = 180 possible subdivisions so counting 209 is the same as counting all of them once then the first 29 a second time.

Spinning Stars, Pulsing Polygons, Donuts, and Stacked Circles

As you play with the file you will find a wide array of 60-second images. Some of the images are similar to those found in File 2 of Playing with Polygons although those images were, generally based on smaller n than 60 [3]. Figure 6 shows some of the possibilities with links to the image on the companion website for each panel (click *Toggle Drawing* to see the image emerge in 60 cycles). This list is by no means exhaustive and other examples are readily available (for example change S = 24 which produced 6 pulsing rectangles in 6(b) to S = 27 and see 9 pulsing triangles or S = 30 to see 6 pulsing pentagrams). And by changing J (and hence P) additional images emerge like this chrysanthemum. (The Excel file and the companion website will show greater detail than possible in Figure 6 due to the size constraints here.)
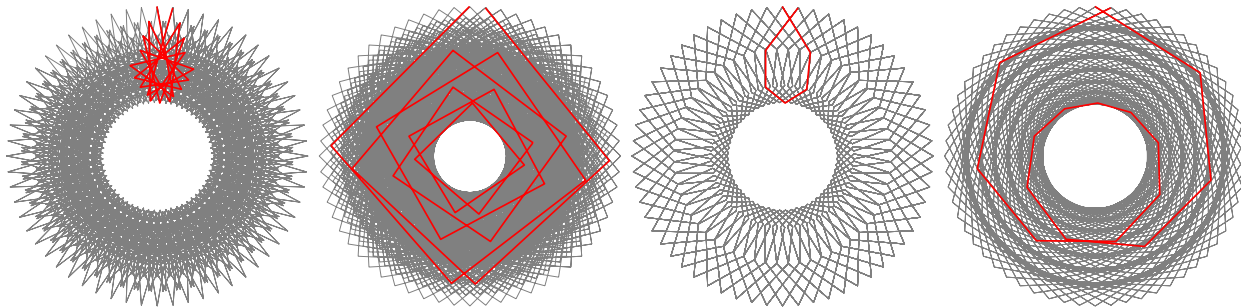
(a) Spinning Stars (S = 18) (b) Pulsing Polygons (S = 24) (c) Donuts (S = 6) (d) Stacked Circles (S = 16)
Figure 6: *Examples 60-second image types using n = 60, J = 23, and P = 47 with first cycle in red.*

Alternative Universes and Turning Back Time

Sixty is a nice number to work with because its many factors mean that we can consider a variety of fractions of an hour that have whole number solutions (1/2, 1/3rd, 1/4th, 1/6th, 1/10th, 1/12th, 1/15th, 1/20th of an hour are all recognizable units of time). The downside for our purposes is that it reduces the number of J, P pairs we can consider that create 60-second images (to 16 as seen in Table 1) because we require that J and 60 be coprime. Imagine we lived in an alternative universe that had a bit smaller day. Suppose there are 24 hours in a day but only 59 minutes in an hour and 60 seconds per minute. One way to reconfigure our units of time in this instance is to say that each hour has 60 minutes, but each minute only has 59 seconds.

In this happy universe there are 58 59-second J and P pairings because 59 is prime, so there is one pair for each J from 1 to 58. These are shown in Table 2. As with Table 1 (and as will always be the case) 1 is paired with 1 and n-1 is paired with n-1. However, the rest of the pairings are distinct in Table 2 meaning J ≠ P, unlike in Table 1 where 6 of the remaining 14 pairs were J = P (noted in bold and symmetric, as are other (J, P) and (n-J, n-P) pairs). The closest we get are the consecutive number swaps highlighted in yellow.

Table 2: *Jump, J, and smallest subdivisions between points, P, pairings producing 59-second images.*

J	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
smallest P	1	30	20	15	12	10	17	37	46	6	43	5	50	38	4	48	7	23	28	3	45	51	18	32	26	25	35	19	57
Note: Bottom half is 59-J and 59-P from top half so smallest to largest J is from right to left in bottom half of the table.																													
n-J = 59-J	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30
n-P = 59-P	58	29	39	44	47	49	42	22	13	53	16	54	9	21	55	11	52	36	31	56	14	8	41	27	33	34	24	40	2

A couple of additional points are worth noting. First, you may have noted in Table 1 that if J = a and P = b then J = b and P = a both produce 60-second images. J and P are modular multiplicative inverses so that which is J and which is P does not matter. This same pattern appears in Table 2 (J =2, P = 30 so J = 30, P = 2, for example). Second, additional P values occur but care must be taken to add 59 each time not 60 to find the next P that produces a 59-second image. Third, there is nothing special about 59: other values of n can also produce one-time around images. Fourth, the resulting images are easy to visualize both from the Excel file (using the *Show first k lines* Toggle) and from the companion website (using *Toggle Drawing*).

One final point to make applies to all of the images discussed above. The string art file produces images with vertical symmetry. Suppose you find an image you like based on $(n, S, P, J) = (n_0, S_0, P_0, J_0)$ that is a clockwise one-time around image as discussed above. The image $(n, S, P, J) = (n_0, S_0, P_0, n_0 - J_0)$ will look the same as the initial image. But if you click *Toggle Drawing* you will find that the image is now a counterclockwise one-time around image. Put another way, if you start with a 60-second image and change J to 60-J and you can turn back time!

Summary and Conclusions

This paper introduces users to string art on a regular polygon in order to understand why some images appear that they were created like the second hand of a clock (click here then click *Toggle Drawing* to see a simple example). Such images are created in 60 repetitive cycles. Each cycle ends one vertex clockwise farther around the 60-gon just like the ticking second hand on a clock. This one-step by one-step forward movement occurs because two of the parameters are multiplicative inverses modulo 60. Two numbers a and b are modular multiplicative inverses modulo c if their product has remainder 1 upon division by c, (this is denoted a·b MOD c = 1; in Excel this is MOD(a·b,c) = 1). Put another way, 60-second images provide a novel geometric interpretation of modular multiplicative inverses. For additional string art examples of modular multiplicative inverses, see the Introduction to Modular Multiplicative Inverses, Backtracking Euclid, and Three *Shape-shifting* Triangles explainers in [3].

Acknowledgements

References

[1] S. E. Erfle and K. A. Erfle, "Exploring Symmetry Using Aestheometry in Classrooms and Beyond." *Bridges Conference Proceedings*, Online, Aug. 1-5, 2020, pp. 547–554. https://archive.bridgesmathart.org/2020/bridges2020-547.html.

[2] S. E. Erfle, "Using Archimedean Spirals to Explore Fractions." *Bridges Conference Proceedings*, Online, Aug. 1-3, 2021, pp. 397–402. https://archive.bridgesmathart.org/2021/bridges2021-397.html.

[3] S. E. Erfle, "Playing with Polygons." https://blogs.dickinson.edu/playing-with-polygons/.

[4] S. Flannery with D. Flannery, *In Code: A Mathematical Journey*. Chapel Hill, N.C: Algonquin Books, 2002.

[5] L. Myles, "Playing with Polygons." https://www.playingwithpolygons.com/.

[6] D. Spector, "Images Produced via Modular Multiplicative Inverse Filters." *Bridges Conference Proceedings*, Linz, Austria, Jul. 16-20, 2019, pp. 367–370. https://archive.bridgesmathart.org/2019/bridges2019-367.html.

[7] S. K. Stein, "The Triex: Explore, Extract, Explain." *Humanistic Mathematics Network Journal*, vol. 1, no. 14, pp. 6–8, 1996. DOI: 10.5642/hmnj.199601.14.04.