An Origami-inspired Adventure in Number Theory and Programming

Jeanine Meyer

Abstract: *This paper describes an origami-inspired adventure. It will be a personal story, with attention to my history with mathematics and games. This focus is on the Dollar Bill Rosette model, created by Paul Jackson and modified by Martin Kruskal. The folding procedure is significant mathematically in [at least] two ways. It starts off with an iterative procedure that improves an original estimate, that is, decreases the amount of error. The folding procedure works, that is, goes through all the intermediate values, for a known class of numbers: reptend primes base 2. I came upon this class using programming in Python and online research. My proof that the numbers that work with the folding procedure are indeed the reptend primes base 2 is included. I term this an instance of "number theory in the wild".*

1 Background

My father always liked mathematical games and puzzles and, as a consequence, I did, also, because it was what we did. The family subscribed to **Scientific American**. I learned about origami from an article in the *Mathematical Games* section by Martin Gardner that featured the flapping bird. Later I met Lillian Oppenheimer, who taught me the Business Card Frog; her daughter-in-law, Laura Kruskal, teacher and origami inventor; and, more or less accidently, several of Lillian's grandchildren. I studied mathematics and computer science and worked at IBM Research in robotics and manufacturing research. When I came to academia, I used games and origami in my teaching examples. The model featured here inspired a book, **Origami with Explanations**, scheduled for publication Summer, 2020.

Several years ago, Mark Kennedy, master folder and teacher, organized informal folding events to take place while in line for New York City's Shakespeare in the Park. I have forgotten what play I saw, but one year I learned the dollar bill rosette. The model taught was the 22-panel rosette by Martin Kruskal.

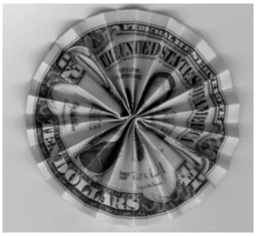

Figure 1: Dollar Bill Rosette.

At some point after learning the model, I showed it to a mathematician colleague at IBM Research and he said that he guessed that the numbers for which the procedure worked were a certain known class of primes. I now explain the folding of the model, which starts with making an estimate; show how the procedure improves the estimate; and then describe how I identified the class of primes using a program written in Python. I provide a proof connecting the class definition with the folding procedure. Lastly, I describe how this work inspired a new course and then a book project.

Note: I did not know Martin Kruskal (son of Lillian, husband of Laura, and mother of Clyde, one of my office mates when we were graduate students at NYU and highly regarded mathematician), but I would guess that he knew the mathematics, which I figured out on my own and will explain here.

2 Folding the Model

The first and main task in folding the rosette is to produce a fan consisting of 10 valley folds and 11 mountain folds. (If you want to do the Paul Jackson model, make 8 valley folds and then make a fan by putting mountain folds in-between the valley folds.) The valley folds divide the bill into 11 parts.

How do we make these 10 folds? First, estimate where the 1/11 position is on the dollar bill. The estimate is marked by putting a pinch on the side. Thinking of the pinch or mark as dividing the bill into 1-part and 10-part regions, we then divide the 10-part region in half by folding the end to the first mark, the one where we estimated one eleventh, and making a second pinch. The new pinch divides the bill into 6-part and 5-part regions. The pattern to note here is that there always will be two numbers, adding up to 11, with one even and one odd.

The next step is to divide (halve) the even portion of the two parts, setting a number N and set the other part to 11-N. This is repeated until you get back to 1 and 10. The sequence is

1 and 10
6 and 5
3 and 8
7 and 4

9 and 2
10 and 1
5 and 6
8 and 3
4 and 7
2 and 9
1 and 10

The last step produce a mark at the 1/11 area of the bill. Presumably it is close to the original mark. In fact, it will be an improved estimate! Yes, this is counter-intuitive. In the next section, I will explain how this process has improved the estimate.

Not counting the last row of the list, you see that 10 marks have been made on the dollar bill, all the intermediate positions. The dollar bill is divided into 11 equal size parts. Now, I will [quickly] describe the rest of the folding.

Going through the sequence again, we make full valley folds instead of pinches at each of the 10 positions. Next, make mountain folds in-between the valley folds to make a fan shape. These next steps are shown Figure 2: Completing the model. It works for the original 16 panel version as well as the 22-panel version. Divide the fan folds evenly into two parts and then unfold 3 segments on each side. Let's call these sections flaps. Fold the model over in the middle so that the two sets of flaps lie next to each other. Turn each combined flap into a tab shape and tuck each inside. Open up the fan to be a circle. The model is complete.

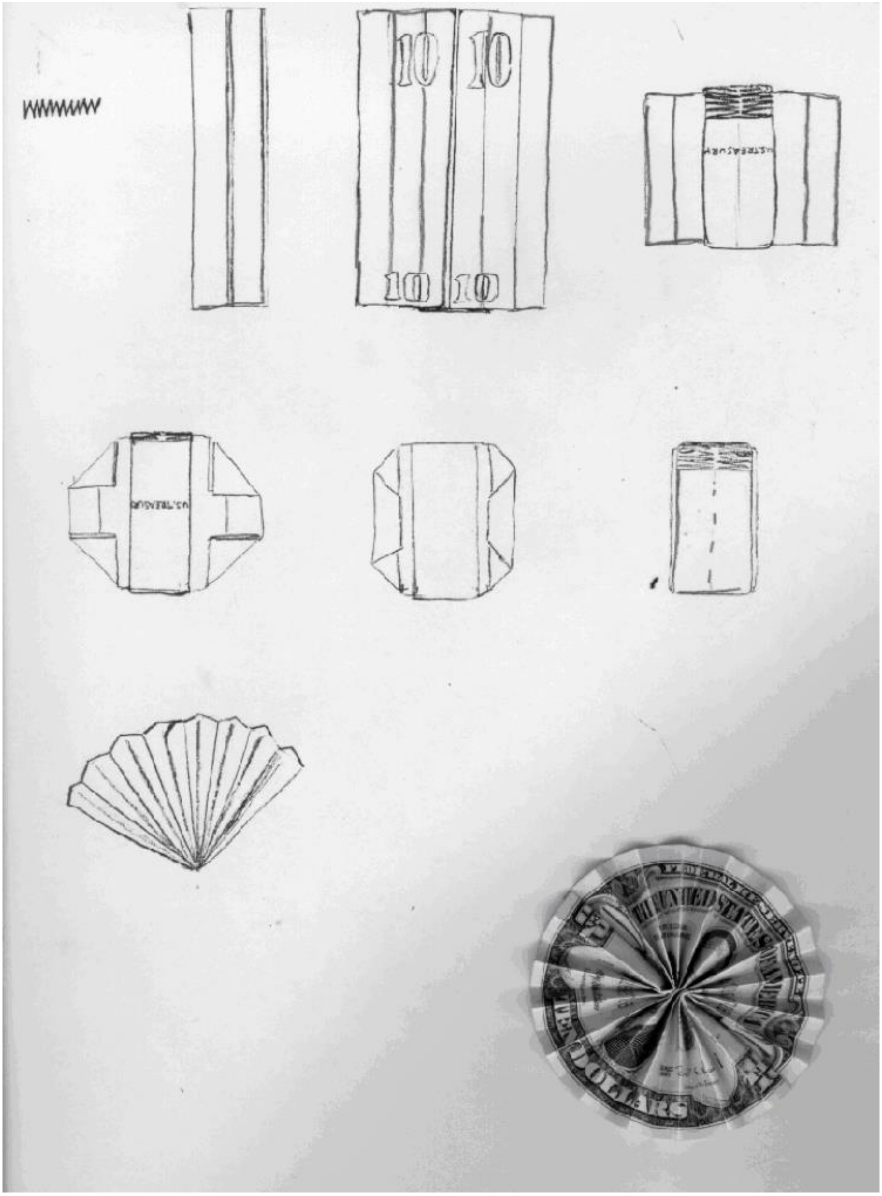

Figure 2: Completing the model.

3 Improving an Estimate

Most origami folders are familiar with what is termed the S method for dividing something into thirds. In the S method, you estimate what 1/3 would be and either make a mark or remember the position of your estimate. Orient the paper to look down on the edges and manipulate it into an S shape and then carefully turn the curves of the S into a Z and make the 3 parts the same size. A more systematic variation is to make a mark at what you think is one third on the edge of the paper. This divides the paper into one-part and two-part areas. Let's call the length of the edge L, the length of a true third t and where the mark was made t+e. The e is the amount of error. The length of the two-part area will be

$$L - (t+e).$$

(If you think of e as positive, this assumes that the original error was an over-estimate of a the third. The reasoning applies to an under-estimate.)

Fold the paper to the mark to divide the two-part area in two. We make the assumption that this fold is accurate. Make a mark. The distance from the edge to this second mark is half of L – (t+e).

$$(L-(t+e))/2$$

Making some re-arrangement of terms, this distance is

$$(L-t)/2 - e/2.$$

Since

$$t = L/3$$

Substituting for t we get

$$L/3 - e/2.$$

This shows that the error for the second mark is half of the original error. You can repeat the process is many times as you want to improve the estimate; that is, shrink the error.

The same phenomenon occurs when the rosette procedure is done. Assuming dividing a portion in half and the folding to a mark is accurate, the error amount is halved each time a section is divided into two parts. For the rosette model which involves 10 steps, the cumulative effect is to halve the error term 10 times! This means the original e is shrunk to $e/2^{10}$. The value of 2^{10} is 1024 so the final error is very small. To use mathematical language, the value of the error term e has limit zero. In practical terms, you can make it as small as you want.

3 The Number 11 and What Else

The next question relates to the number 11. The procedure of dividing the dollar bill edge into two parts and then dividing the even part in half goes on 10 times. Each of the intermediate points is hit. Does this work for all numbers? It does work for 3 but what other numbers?

The first observation is that the number must be odd so that any partition into two parts yields one odd and one even part.

NOTE: Following the practice in programming, I use the asterisk for multiplication.

A next observation is that the number must be prime. Consider the case of 15. The procedure would start with 1 and 14 and then continue as follows

1 and 14
8 and 7
4 and 11
2 and 13
1 and 14

We note that this sequence does not hit all the intermediate points. Here is an informal proof:

Suppose P is not prime, say it is equal to M * N, where M and N each >1. Note that neither M or N can be even.
In the rosette procedure, at some point, the sizes for the two portions must be M and (N-1)*M. What is the next step? M is not even, so the next step would be

M+((N-1)/2)*M and ((N-1)/2)*M

Continuing the folding process, each of the pair of numbers would have a factor of M. That is, it would not continue to a pair with one of the two equal to 1. One way to make this more concrete is to consider the number 9. In this case, M and N are each equal to 3. Assuming the process works, applying the folding procedure to 3 and 6 results in 6 and 3 and then 3 and 6. The procedure gets stuck and never reaches 1 and 8.

The procedure does not work for all primes. Consider the situation with 17. Here are the successive pairs produced when we start with 1 and 16. The procedure ends, that is, returns to 1 and 16, but does not hit all the intermediate points.

- 1 and 16
- 9 and 8
- 13 and 4
- 15 and 2
- 16 and 1
- 8 and 9
- 4 and 13
- 2 and 15
- 1 and 16

At this point, I recalled the conversation from many years ago that there is a certain class of primes that may correspond to those satisfying the origami procedure. I decided to investigate.

4 Write a program and Search the Web

Python is the language I used to check if numbers work using the folding procedure. I chose Python because it is the language we use in our Number Theory course, which makes use of a book, **Elementary Number Theory with Programming**. We made that decision because Python has arbitrary precision for integers.

[Aside: JavaScript is used in the book because the authors believed it to be an easier language for people less familiar with programming to follow. In contrast to most colleges, Purchase College/SUNY offers only a joint Mathematics/Computer Science major, so the students taking the Number Theory course have had at least one programming course. The Number Theory course provides us a way to introduce another programming language, Python. Our students can appreciate the advantage that the arbitrary precision provides for number theory and can appreciate when it is critical to avoid leaving the integer domain for floating point numbers.]

My Python program is shown below and is, hopefully, readable. It is my only Python program. Comments start with #. Indentation is required to indicate the content of functions and clauses. The function, `tryProcedure`, is invoked with a value `N` as argument. The variable `count` keeps track of the number of steps. The variables `currentpos` and `remainder` describe the pair of numbers (parts). The # symbol indicates a comment for the rest of the line.

```
def tryProcedure(N):
   count = 1          # start with 1 and N-1
    currentpos = 1
    remainder = N-currentpos
    while True:
      if (isEven(currentpos)): #  determine even side
           currentpos = currentpos//2
             # the // forces integer division
           remainder = N - currentpos
      else:
           currentpos = currentpos + remainder//2
           remainder = N-currentpos
     count = count + 1

     if (currentpos==1):   #at 1, leave loop
           break
```

```
# outside of the while loop
   if (count==N):
       print(" ",N,end="")    #This is a good value
   return
```

The operation of integer division, indicated by the //, is important for keeping everything integers.

The program prints out a good number, that is, the numbers that go through N steps before returning to the pair 1 and N-1. Invoking the `tryProcedure` function from 3 to 1000 produced the following list of numbers:

3 5 11 13 19 29 37 53 59 61 67 83 101 107 131 139 149 163 173 179 181 197 211 227 269 293 317 347 349 373 379 389 419 421 443 461 467 491 509 523 541 547 557 563 587 613 619 653 659 661 677 701 709 757 773 787 797 821 827 829 853 859 877 883 907 941 947

Since I did not remember what my colleague said some years ago, I attempted to consult the institutional memory of the web by putting this whole set of numbers into the Google search field. I was not that optimistic, but it was successful! I reached https://en.wikipedia.org/wiki/Full_reptend_prime

5 Proof the two Classes are the Same

[Note: Repeat: I do not claim to be the first person to prove that the folding procedure for the rosette and the reptend prime base 2 procedure are the same. I did not find a proof, but I did not look very hard because I liked thinking about it myself.]

The reptend prime base 2 class is defined as follows:

> A number P for which 2 raised to the power N, N going from 0 to P-2, produces the numbers 1 to P-1, modulo P, is a reptend prime base 2.

If the P-2 seems strange, do note that the process starts with 0, not 1.

Here is the reptend procedure for 11:

2^0 is 1 = 1 mod 11
2^1 is 2 = 2 mod 11
2^2 is 4 = 4 mod 11
2^3 is 8 = 8 mod 11
2^4 is 16 = 5 mod 11
2^5 is 32 = 10 mod 11
2^6 is 64 = 9 mod 11
2^7 is 128 = 7 mod 11

2^8 is 256 = 3 mod 11
2^9 is 512 = 6 mod 11
2^{10} is 1024 = 1 mod 11

This sequence, that is, the defining characteristic of reptend primes base 2, resembles the folding sequence for 11, but in reverse order. This certainly is not a proof since it is just the one number, 11, but it is encouraging.

To prove that the numbers that can work using the folding procedures are the reptend primes base 2, one needs to prove that the numbers for which the folding procedure hits all the intermediate numbers are the same as the numbers for which the reptend process, raising 2 to powers from 0 to the number -1, hits all the intermediate numbers. I decided to try for a <u>stronger</u> result: the two procedures are the same procedure, with the folding procedure done in reverse order. Proving the bigger thing seemed easier to me than proving the smaller thing. That is, if N and P-N are pairs in reverse folding, then I will show that

$N = 2^k \bmod P$

for all k steps starting from 0, for all primes P.

So how to define the reverse folding process? There are several ways to approach this challenge. If (F and P-F) goes to (G and P-G) in the normal folding procedure, I need to define F in terms of G. I can consider cases of if F was odd and if it were even. Instead, consider the following. Either F was halved or P-F was halved. So either F is equal to 2*G, or P-F is equal to 2* (P-G). Which one happened? The answer is to consider if 2*G is greater than P or not. Keep in mind that P is prime so 2*G cannot be equal to P. Also, since the pair of numbers, G and P-G add up to P, one is less than ½ of P and one is greater. So doubling one will be greater than P and doubling the other will be less.

Initial case: k=0

- Reptend and reverse folding start out with
  $2^0 = 1$, so $2^0 = 1 \bmod P$

Induction step

- Can assume $G = 2^k \bmod P$ meaning
  $G = 2^k + a*P$
- Two cases: 2*G< P and 2*G > P.
  Case 2*G< P.
  So F = 2*G. Substituting the expression for G
  $F=2*(2^k + a*P) = 2^{k+1} + 2*a*P$ so $F = 2^{k+1} \bmod P$

- Case 2*G>P
  F = P-2*(P-G)
  F = P – 2*P + 2*G Rearranging terms

$F = 2*G – P$ Substituting the expression for G
$F = 2*(2^k + a*P) – P$
$F = 2^{k+1} + 2*a*P – P$
$F = 2^{k+1} + (2*a-1)*P$
$F = 2^{k+1} \bmod P$

To recap: Because both processes yield the same results, both either satisfy both the reptend AND the folding criteria of hitting all the intermediate points between 1 and P-1 or neither do.

To put it another way, the sequences of numbers are the same starting at $k = 0$ and continuing for all integers! However, we only consider the values up to $k = P – 2$ for each P.

6 Reflection

This paper describes exploring an origami model, the dollar bill rosette. The model provided opportunities to touch on topics in basic algebra, limits, programming, and number theory. It also demonstrates what is a proof and the benefits and the limitations of web searches. A talk on this process, which we refer to as an *adventure in origami*, has been given several times to our Number Theory and Senior Seminar classes and the response from the students is strongly positive.

In fact, my chair, upon hearing about my adventure, suggested designing a general education course based on origami. As one of many colleges that require everyone to take a math class, we always are looking for new courses. My first reaction was that the mathematics associated with origami was too difficult for most students. However, late one night, I was inspired and came out with a plan, making use of origami models to inspire topics in basic algebra, geometry and trigonometry. For example, final dimensions of the model can be computed in terms of the size of the (flat) paper. Students can think about the change from 2D to 3D. We can compare crease patterns, folding sequences and final models.

The dollar bill rosette model is taken up after simpler dollar bill folds. I don't expect the students to understand every aspect. It does seem that most of the students in the two classes to date:

1) are initially surprised, but then understand how the procedure improves the initial estimate (the initial surprise is important)
2) accept that seeing that the numbers that work for the folding procedure match this specific class of primes up to 1000 does not prove that the two classes are the same; and
3) appreciate that my proof is stronger than just proving the definitions produce the same numbers…but sometimes stronger is easier.

In addition, I hope they observe for all the models my excitement and delight at the beauty, structure, and patterns of the origami and role of mathematics. The course also includes peeks at origami mathematics topics such as tessellations, flat-foldability and fold-and-cut. I now am working with a former student, now colleague and co-author, on a book project taking this approach. The models are traditional and modern, including action and modular models. Many models are made from squares, such as Japanese kami. However, inspired by Laura Kruskal, who favoured using so-called found paper, in addition to the dollar bill rosette, there are other dollar bill models, and also models from business cards and copier paper. See Figure 3 for some of the models. We appreciate the permissions granted by the designers and the general support. The adventure continues.

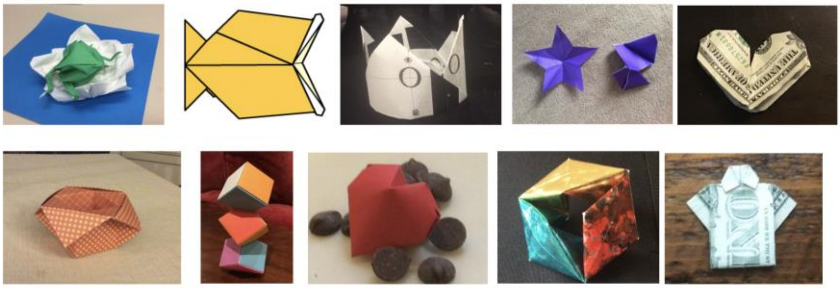

Figure 3: Selection from models used in course and book.

References

Full Reptend Prime, **Wikipedia**, https://en.wikipedia.org/wiki/Full_reptend_prime, last edited March 22, 2017.

Jeanine Meyer and Takashi Mukoda, **Origami with Explanations**, World Scientific Publishing, Summer/Fall 2020.

Lewinter, Marty and Jeanine Meyer, **Elementary Number Theory with Programming**, Wiley, June, 2015, http://www.wiley.com/WileyCDA/WileyTitle/productCd-1119062764,subjectCd-CS00.html