Tupper's Self-Referential Formula

Margaret Fortman

June 2, 2015

1 Introduction

Tupper's Self-Referential formula:

$$\frac{1}{2} < \left\lfloor \mathrm{mod}\left(\left\lfloor \frac{y}{17} \right\rfloor 2^{-17\lfloor x\rfloor - \bmod (\lfloor y\rfloor, 17)}, 2\right)\right\rfloor$$

is special because, at a certain distance up the y-axis, it plots itself. This works because the equation tells you which (x, y) coordinates are colored in and which are not. Therefore, between a certain k and k+17 on the y-axis and between 0 and 106 on the x-axis, the plot of the formula is itself. However, the key is the value of k, which is

k=4858450636189713423582095962494202044581400587983244549483093085061934
7047088099284506447698655243648499972470249151191104116057391774078569
1975432657185544205721044573588368182982375413963433822519945219165128
4348332905131193199953502413758765239264874613394906870130562295813219
4811136853395355652908500238750928568926945559742815463865107300491067
2305893358605254409666435126534936364395712556569593681518433485760526
6940161251266951421550539554519153785457525756590740540157929001765967
9654800644278291314885482599147212485063526866304763 00.

If you plot the equation and look at it between a height of k and $k + 17$ up the y-axis, it gives the plot of the equation (Figure 1):

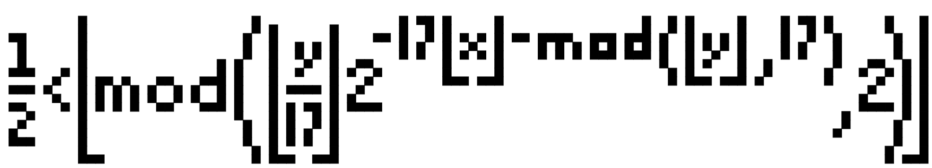

Figure 1: Tupper's Self-Referential Formula

However, Tupper's Self-Referential formula not only plots itself, it plots every 106×17 grid of white and black pixels. Therefore, we can find the value of k for anything that will fit in this size plot.

For example, when

k=1445202489708975828479425373371945674812777822151507024797188139685490
8873568298734888825132090576643817888323197692344001666776474924212512
8995265907053708020473915320841631792025549005418004768657201699730466
3833949016013743197155209961811452497819450190683595005106578043256408
0119786755686314228025969420625409608166564241736740394638417077453742
7319606443899923010379398938675025786929455234476319291860957618345432
2480049217280333494198162067498544720381939397385138489604767597826733
1343769705199458068186981933044633677404726886 4

the plot is (Figure 2):

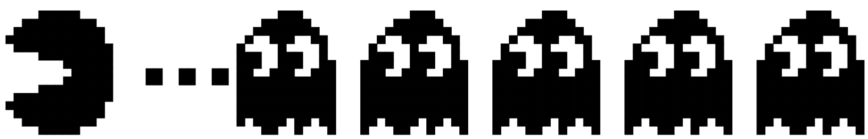

Figure 2: PacMan

When

k=2352035939949658122140829649197960929306974813625028263292934781954073
5954955446141406484573424615648873252234556208042047960114349551110223
7660163585321047663331899199046219268799910930820947231541971365223818
5967518731354596984676698288025582563654632501009155760415054499960

the plot is (Figure 3):

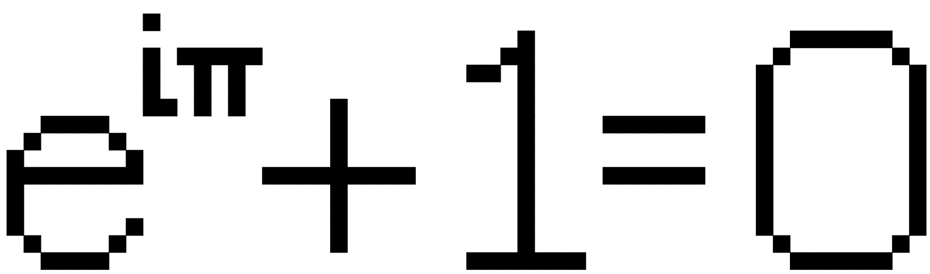

Figure 3: Euler's Identity

1.1 Calculating k

To figure out the k at which these, or any other 106×17 grid, occur you start at the top righthand corner and work your way down the columns and to the left, through each pixel. If you want there to be a black pixel, you write a 1 and if you want a white pixel you write a 0. At the end you will have a 1,802

$(106 \times 17 = 1802)$ digit long binary number. Then put that number into base 10. Every 0 or 1 in the binary number corresponds to a power of two. The first digit corresponds to 2^0, the second 2^1, then 2^2, and so on. Multiply each power of two with its corresponding 0 or 1 and then add up each product to convert to base 10. For example $1011 = (1\times2^0)+(1\times2^1)+(0\times2^2)+(1\times2^3) = 1+2+0+8 = 11$. Then multiply the number in base 10 by 17, and you will get the k-value where the plot appears.

2 Project

Using Tupper's Self-Referential Formula, my main project was to write a program that would be able to find the position where the equation plots any three-line phrase in the 106×17 plot on command.

2.1 Letters

First, I wrote formulas for each lowercased letter, each capital letter, a space, a hashtag, an exclamation point, a heart, and a smiley face.

The binary number for each character is either 15 digits long or 25 digits long, depending on whether the character fits in a 3×5 area or a 5×5 area. For example, the binary number for the letter a is 11101 10101 11111 in a 3×5 area.

Each column translates to a sum such as

$$(1 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3 + 1 \cdot 2^4)$$

if every pixel is black or

$$(0 \cdot 2^0 + 0 \cdot 2^1 + 0 \cdot 2^2 + 0 \cdot 2^3 + 0 \cdot 2^4)$$

if every pixel is white.

The same is done for every column in the 3×5 or 5×5 area, however each sum gets multiplied by 2^{17i}, where i is the number of columns it gets moved to the left.

For example, the formula for the lowercased a is:

```
f["a"] = 17*((1 + 2 + 4 + 16) + (1 + 4 + 16)*2^{17} + (1 + 2 + 4 + 8 + 16) 2^{34})
```

2.2 Speller

In order to combine letters to write a word, sum the characters, allowing 5 spaces per character, to get the formula:

$$\sum_{k=1}^{Length(w)} 2^{17\cdot5(k-1)} f(w(k))$$

Then, use this sum to create a function that will spell anything by just defining the word.

```
Speller[word_] := Module[{w = Characters[word]},
  Sum[2^(17*5*(k - 1))*f[w[[k]]], {k, 1, Length[w]}]]
```

Now, I can write any word in the function and it will calculate the k-value for that word. Then, I created a function that will plot any k, called tupperPlotter. Therefore, the command tupperPlotter[Speller["word"]] will plot the word. For example, tupperPlotter[Speller["celebrate"]] will plot (Figure 4):

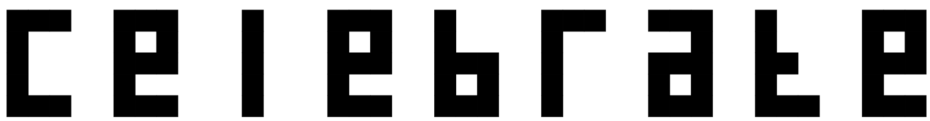

Figure 4: tupperPlotter[Speller["celebrate"]]

2.3 Writing Multiple Lines

In order to write multiple lines, I changed the formula for Speller to add 6 if the length of the sentence is more than 21 characters long (since we are allowing 5 spaces per character and $21 \times 5 = 105$). The sum 2^{17i+j} will move any character up j spaces, therefore, if $j = 6$ it would move the characters up to the second line. If the length of the sentence is more than 42 characters, it will add 12 to move them up 12 rows. Therefore, the new sum is

```
Sum[2^{17*5*(k - 1) + 12}*f[w[[k]]], {k, 1, Min[21, Length[w]]}] +
  If[Length[w] > 21,
   Sum[2^{17*5*(k - 22) + 6}*f[w[[k]]], {k, 22, Min[42, Length[w]]}],
   0] +
  If[Length[w] > 42,
   Sum[2^{17*5*(k - 43)}*f[w[[k]]], {k, 43, Length[w]}], 0]].
```

Now, using tupperPlotter[Speller["sentence"]] with this new sum, it will plot any sentence that is 63 characters or less. However, to make it even nicer, I added Speller into the definition of tupperPlotter, therefore, you can simply define the word and tupperPlotter[word] will plot it. For example, if you define word= "With this new sum thefunction can plot anysentence you want! :", then tupperPlotter[word] plots (Figure 5):

3 Repeating the Formula

I also looked at whether there was a reason the formula is plotted in a 106×17 area, and what would happen if I expanded the width. I found that the formula will plot for any distance on the x-axis. I also looked into finding a value of k

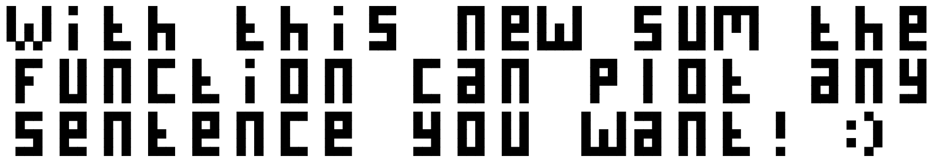

Figure 5: word="With this new sum thefunction can plot anysentence you want! :"

that would plot the formula twice, horizontally, in a 212×17 area. I found that the k-value that would achieve this is

k=13885130983892771457071560494795753867604613853153329935880431599397536165913984571317936193177670012322316137866930002556337933968807951480269830492732162639083378961130121392250156268332224872590915922728871868424322879096740204594081697861245691897143564311449621124024641474797847223246759394504949391213201084048626206631183936855773280754111507384415700389649263117659643153694621616620022591605484342479978281282158423812271641701329411936511745395231394197268240500175391126299885407176127927898616219211890107119784328614332907337716845878197446472010431303798437694022612567090668199565016886421373029096462176340087739850107832032589888502654425405875777141089269439514651866811509251704077197803323577302355330142622794638544010613289015689946657333585544427830289816892288584100839336006357616256396097454923042033785160553832776300918730803630933784933653384282431669362912065633844184645887587515472103174437014842286619065313882557417919886346218535975348936841957778302713186098456434604006742219932514369741820566961226517615438503459915386394311849886932798983192315000.

That plot is shown in (Figure 6):

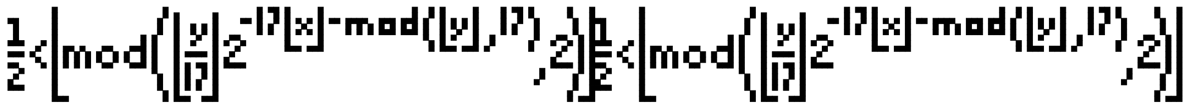

Figure 6: The formula is repeated

This also means that to repeat the formula, the original k-value is simply multiplied by $2^{17\times 106} + 1$ to make the formula repeat again.

4 Common Mistake

Many people reference Tupper's formula incorrectly because in computer science positive y often goes downwards, however in mathematics positive y goes upwards. Therefore, the k-value that most people say works for self-referencing is actually incorrect. The value of k that is given in most websites is

k=9609393799189588849716729621278527547150043396601293066515055192717028023952664246896428421743507181212671537827706233559932372808741443078913259639413377234878577357498239266297155171737169951652328905382216124032388558661840132355851360488286933379024914542292886670810961844960917051834540678277315517054053816273809676025656250169814820834187831638491155902256100036523513703438744618483787372381982248498634650331594100549747005931383392264972494617515457283667023697454610146559979337985374831437868418065934222278983887229800007484047 19. When the incorrect k-value is used the plot looks like (Figure 7):

Figure 7: The incorrect k-value

Therefore, we had to correct the grapher we were using to plot the formula, and had to find the correct k. I also had to correct the formulas for each of the characters for this corrected version.

5 Bibliography

The 'Everything' Formula. *YouTube*. Numberphile, 15 Apr. 2015. Web. 30 May 2015.

"How Does Tupper's Self-referential Formula Work?" *The Lumber Room*. N.p., 12 Apr. 2011. Web. 30 May 2015.