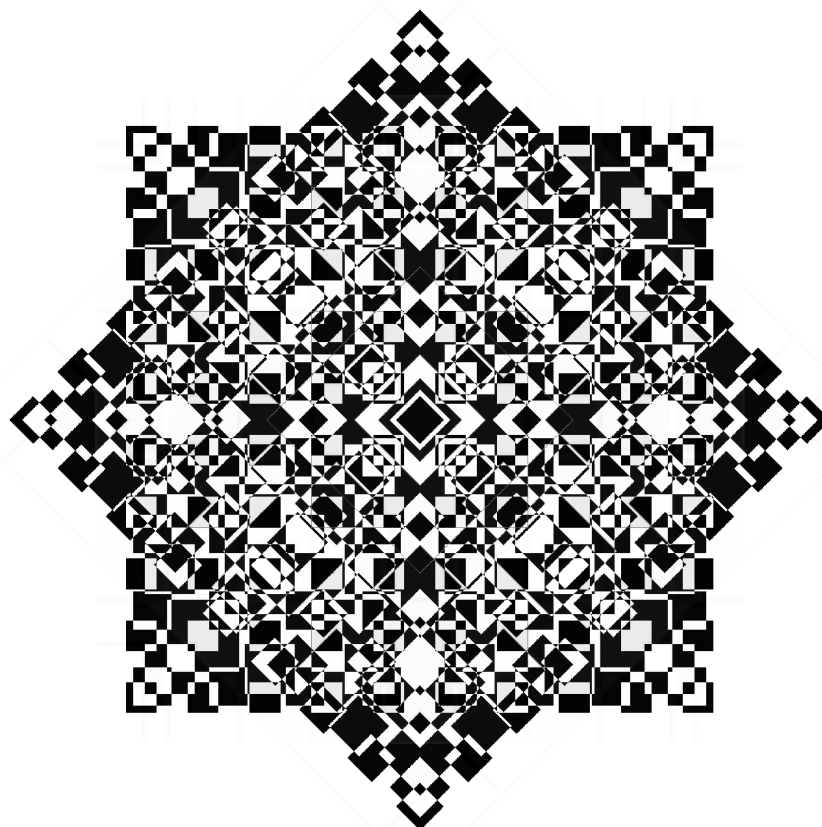


Marley Math: Cantor Sets, Sierpinski Carpets, Menger Sponges, And More

Carlo Mats Vincenti Mitchener

24 April 2022



Medium

This paper was originally published as a Medium article on the 24th of April 2022. This version has been modified to take advantage of L^AT_EX. In the process, many sections and images were left out, so feel free to give it a view.

Introduction

The whole wide world knows almost everything there is to know about the Ternary Cantor Set and its higher dimension analogues, the Sierpinski Carpet and the Menger Sponge. But the whole wide world seems fully unaware that each of these fractals have an infinite amount of sisters. The goal of this paper is to quickly introduce them so as to further our knowledge of fractal geometry.

Background

I would like to make it very clear right now that I am not a mathematician. The following will therefore be void of any formal definitions. Though mathematicians are my main audience here, I shall strive to write in a way that kids can understand. Alas, I'd have trouble writing any other way.

Prior Reading

The perusal of this paper does require some basic knowledge of fractal geometry. Feel free to get acquainted with Fractals, the Ternary Cantor Set, the Sierpinski Carpet and the Menger Sponge. I have linked their Wikipedia pages, but YouTube is full of information and awesome visualisations as well, so feel free to give it a search.

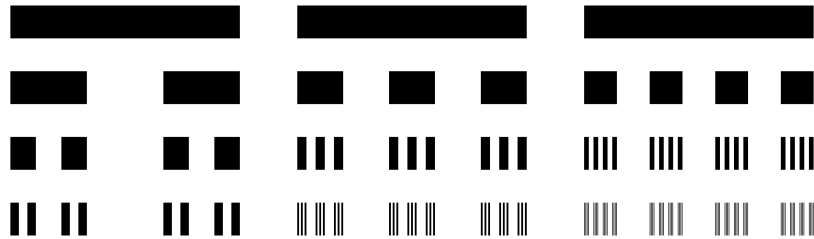
Terminologies

The perusal of this paper also requires a sound understanding of the terminologies used herein. Firstly, `mrly` stands for Marley. A `mrlyset` is a package of fractals. For example, the `mrlyset3` contains the `mrlytree3` (Cantor Ternary Set), the `mrlycarpet3` (Sierpinski Carpet) and the `mrlysponge3` (Menger Sponge). A `mrlytree` is the one-dimensional fractal in the `mrlyset`. A `mrlycarpet` is the two-dimensional fractal in the `mrlyset`. A `mrlysponge` is the three-dimensional fractal in the `mrlyset`. Grid Lines (`gl`), Grid Squares (`gs`) and Grid Cubes (`gc`) represent the initial spaces we need to build our fractals. Void Lines (`vl`), Void Squares (`vs`) and Void Cubes (`vc`) represent the spaces within our fractals that must be void. Fill Lines (`fl`), Fill Squares (`fs`) and Fill Cubes (`fc`) represent the spaces within our fractals that must be filled. The Tree Ratio (`tr`) represents the density of a `mrlytree`. The Carpet Ratio (`cr`) represents the area of

a mrlycarpet. The Sponge Ratio (sr) represents the volume of a mrly sponge. They each represent the fullness of our fractals, and are obtained by taking our filled stuff and dividing it by our grid stuff. Tree Dimension (td), Carpet Dimension (cd) and Sponge Dimension (sd) represent the Hausdorff Dimension of our fractals, which is also an important topic, so quickly read about it. The mrlyset0 represents the mrlytree0, the mrlycarpet0 and the mrly sponge0, which in turn represent the fractals for which the Number (n), or sequence term, is infinity. In mrlymath, iterations can happen in two ways. We can iterate through our mrlysets, so mrlyset3, mrlyset5, mrlyset7, and so on. We can also iterate within our mrlysets, so mrlytree3.1, mrlytree3.2, mrlytree3.3, and so on. So a mrly sponge3.5 represents the fifth iteration of a mrly sponge3, also called a Level 5 Menger Sponge, and is found within the second set, the mrlyset3. For now, this should do the trick. Before we begin our journey into the wonderful world of mrlymath, we must first put our math hats on. Okay, let's go!

The First Dimension

Our journey begins in the very first dimension: the world of mrlytrees.



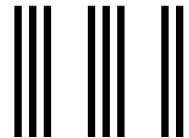
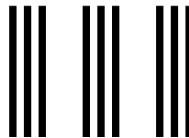
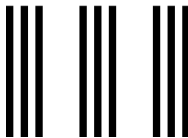
mrlytree3

To begin making our fractals, we need only a simple line. The well-known Ternary Cantor Set takes this line, divides it into 3 equal lines, and removes the middle one. We then have 2 lines, each of length $1/3$. We now divide both of these into 3 equal lines, and again remove the middle ones on each. We now have 4 lines, each of length $1/9$. We can continue this process forever.



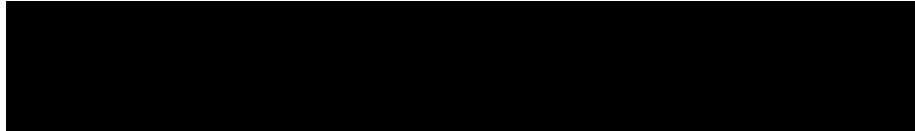
mrlytree5

The next fractal in line is the mrlytree5. As you can imagine, we just take a line, divide it by 5, and remove the 2 middle $1/5$. We now have 3 lines, each of length $1/5$. We then divide each of these by 5, and remove the 2 middle $1/25$ on each. We now have 9 lines, each of length $1/25$. Of course, we can do this forever.



mrlytree7

The next one is the mrlytree7. Once again, we just take a line, divide it by 7, remove the 3 even middle $1/7$, divide each of these by 7, and remove the 3 even middle $1/49$ of each, and continue forever.



This way, we can create an infinite set of one-dimensional fractals, which I am calling mrlytrees, because they branch in and out.

The very last fractal in line is the mrlytree0. Here, we just take a line and divide it by infinity. We now have infinite lines, and we remove the infinite even middle infinities. We then continue this process forever.

The basic rule for creating a mrlytree, which mathematicians will most surely make use of in a formal definition, is to take a line of odd length, divide it by that length, and remove all lines that are at an even position. So a mrlytree5.1 has 3 Fill Lines, each at an odd position (1, 3, 5), and 2 Void Lines, each at an even position (2, 4). A similar even/odd coordinate rule will be applied in other dimensions.

Let's take a look at the mrlytree formulas and the first couple values.

mrlytrees		1	3	5	7	9	11	13
Number		1	2	3	4	5	6	7
Grid Lines	gl = 2*n-1	1	3	5	7	9	11	13
2nd Iteration	gl2 = gl^2	1	9	25	49	81	121	169
3rd Iteration	gl3 = gl^3	1	27	125	343	729	1331	2197
Void Lines	vl = n-1	0	1	2	3	4	5	6
2nd Iteration	vl2 = gl2-fl2	0	5	16	33	56	85	120
3rd Iteration	vl3 = gl3-fl3	0	19	98	279	604	1115	1854
Fill Lines	fl = n	1	2	3	4	5	6	7
2nd Iteration	fl2 = fl^2	1	4	9	16	25	36	49
3rd Iteration	fl3 = fl^3	1	8	27	64	125	216	343
Tree Ratio	tr = fl/gl	1	0,6667	0,6000	0,5714	0,5556	0,5455	0,5385
2nd Iteration	tr2 = fl2/gl2	1	0,4444	0,3600	0,3265	0,3086	0,2975	0,2899
3rd Iteration	tr3 = fl3/gl3	1	0,2963	0,2160	0,1866	0,1715	0,1623	0,1561
Tree Dimension	td = log(fl/gl)	1	0,6309	0,6826	0,7124	0,7325	0,7472	0,7587

The Grid Lines are extremely important values and are used throughout mrlymath. It is from these values that all fractals get their names. For example, the third fractal has 5 Grid Lines, so we call it the mrlytree5, mrlycarpet5 and mrlysponge5.

The Tree Ratio is an interesting value, which basically calculates the proportion of Fill Lines over Grid Lines. Of course, as we iterate, there will always be more and more Fill Lines. But there will always be even more Grid Lines than there are Fill Lines. We see that each fractal starts with a line, which we can use to represent all real numbers between 0 and 1. So we always start with an infinite amount of real numbers. As we iterate through any mrlytree, we get less and less of these real numbers. So for example, the mrlytree5.0, like any other, starts with 100% of all real numbers. The mrlytree5.1 has around 60% of all real numbers. Yet, it still contains an infinite amount of numbers. The mrlytree5.2 has around 36% of all real numbers. The mrlytree5.3 has around 22% of all real numbers.

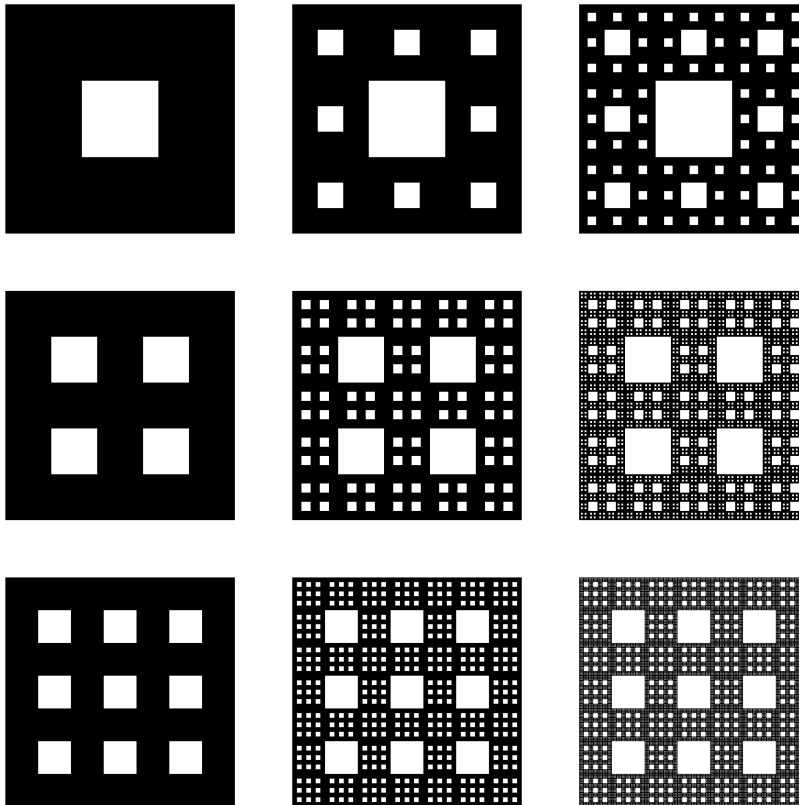
It is also interesting to calculate the limits of the Tree Ratio for any given iteration. We find that the limit for the 1st iteration is 1/2, so a mrlytree0.1

starts with 50% of all real numbers. The limit for the 2nd iteration is $1/4$, so a `mrlytree0.2` has 25% of all real numbers. The limit for the 3rd iteration is $1/8$, so a `mrlytree0.3` has 12,5% of all real numbers. And so on. In other words, the limit of the Tree Ratio for any given iteration (i) is simply $(1/2)^i$.

By comparing the dimensions of each of these fractals, we see that the `mrlytree` dimensions tend towards one, which is the dimension of a simple line. So 1 is the limit for `td`. It will, of course, never be reached. For someone living on a `mrlytree0`, their world must look pretty one-dimensional, even though it's not. For now, that's all I have to say about `mrlytrees`. We can now enter the second dimension.

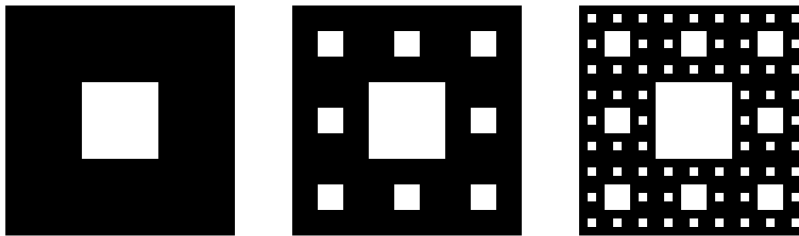
The Second Dimension

Things get a bit trickier in higher dimensions. Luckily, our designs are very simple, so it's not that bad. In two dimensions, our `mrlytrees` become `mrlycarpets`.



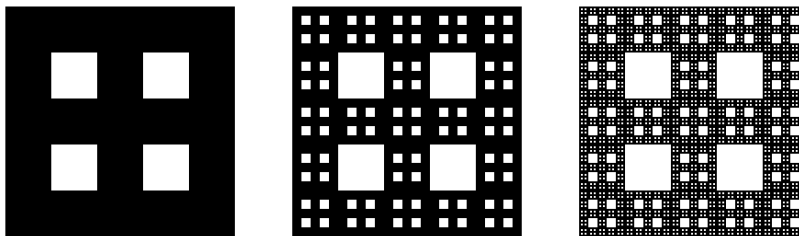
mrlycarpet3

The first two-dimensional fractal is the well-known Sierpinski Carpet, which I call the mrlycarpet3. To make one, we take our 1st iteration mrlytree3 Grid Lines, which is 3, and square to get 9 Grid Squares. We then punch out the middle square. We now have 8 Fill Squares and 1 Void Square. We can take this shape, and make another one from it. In fact, we can do this forever. The Sierpinski Carpet is pretty well documented, so feel free to look it up.



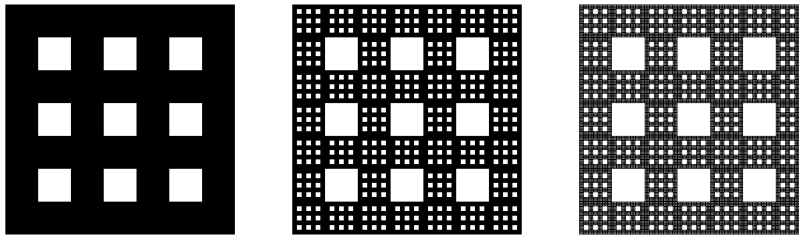
mrlycarpet5

Of more interest to us is the next fractal, the mrlycarpet5. To make this one, we take our 5 Grid Lines, and square to get 25 Grid Squares. We then take our 2 Void Lines, and square to get 4 Void Squares. We just punch out the 4 even middle holes, and we are left with 21 Fill Squares. We can take this shape, and do that forever.



mrlycarpet7

We then have the mrlycarpet7. Once again, we just take our mrlytree7 Grid Lines and Void Lines and square each to find 49 Grid Squares and 9 Void Squares. Thus, our 1st iteration mrlycarpet7 has 40 Fill Squares.



Once again, we can find a simple coordinate rule for building mrlycarpets. We simply state that if a square has 2 even coordinates, it must be void. This rule, of course, works for all mrlycarpets, and all their iterations. A similar rule will again be applied in three dimensions.

Let's now take a look at the mrlycarpet formulas and values.

mrlycarpets		1	3	5	7	9	11	13
Number		1	2	3	4	5	6	7
Grid Squares	$gs = (2*n-1)^2$	1	9	25	49	81	121	169
2nd Iteration	$gs2 = gs^2$	1	81	625	2401	6561	14641	28561
3rd Iteration	$gs3 = gs^3$	1	729	15625	117649	531441	1771561	4826809
Void Squares	$vs = (n-1)^2$	0	1	4	9	16	25	36
2nd Iteration	$vs2 = gs2 - fs2$	0	17	184	801	2336	5425	10872
3rd Iteration	$vs3 = gs3 - fs3$	0	217	6364	53649	256816	886825	2474172
Fill Squares	$fs = 3*n^2 - 2*n$	1	8	21	40	65	96	133
2nd Iteration	$fs2 = fs^2$	1	64	441	1600	4225	9216	17689
3rd Iteration	$fs3 = fs^3$	1	512	9261	64000	274625	884736	2352637
Carpet Ratio	$cr = fs/g_s$	1	0,8889	0,8400	0,8163	0,8025	0,7934	0,7870
2nd Iteration	$cr2 = fs2/g_s2$	1	0,7901	0,7056	0,6664	0,6440	0,6295	0,6193
3rd Iteration	$cr3 = fs3/g_s3$	1	0,7023	0,5927	0,5440	0,5168	0,4994	0,4874
Carpet Dimension	$cd = \log(fs/g_l)$	2	1,8928	1,8917	1,8957	1,8998	1,9035	1,9066

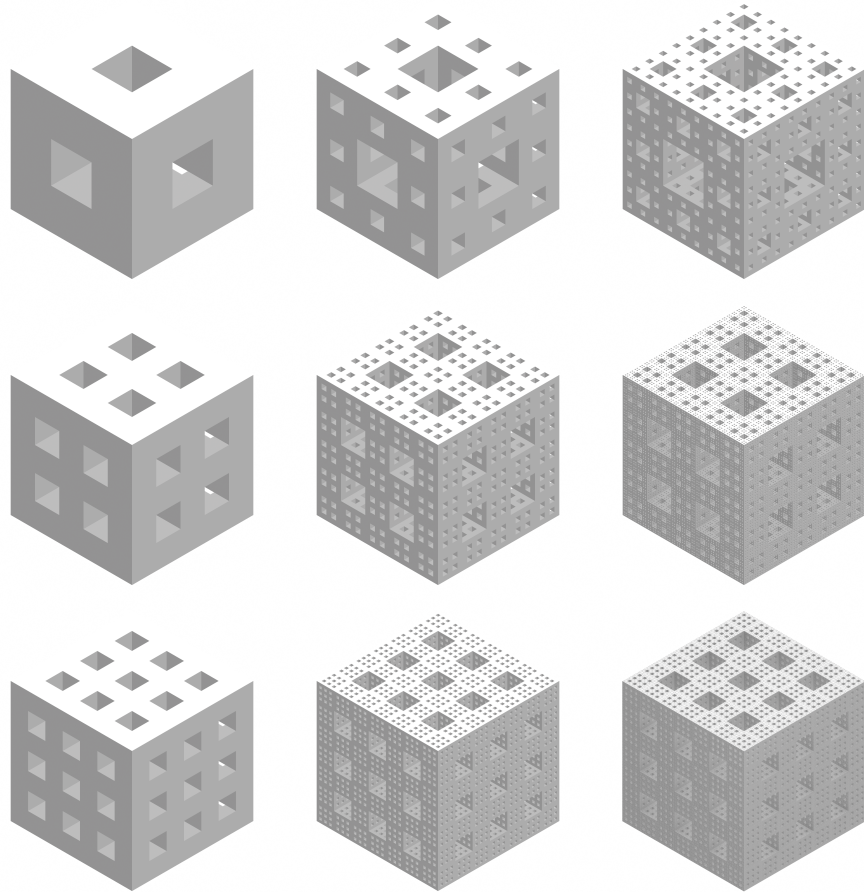
If you're a complete nerd, you may have noticed that the mrlycarpet formula for calculating Fill Squares is identical to the formula for calculating Octagonal Numbers. Why in the world are mrlycarpets connected to octagons, I do not yet know. There must be some obvious visual explanation, which I have yet to find. I did find this YouTube video which makes it easy to visualise. But this still does not explain why octagons and fractals go hand in hand. The only logical explanation I can find is that a square has 4 sides, and an octagon has 8, so that's already a major connection. To get the Fill Squares formula, we simply calculate the Grid Squares minus the Void Squares.

Similar to the limits of the mrlytrees, we find some unintuitive properties within our mrlycarpets. We calculate that the 1st iteration Carpet Ratio tends towards $3/4$. So a mrlycarpet0.1 is 75% filled, which is a bit bizarre, since we're punching an infinite amount of holes into it. We find that the Carpet Ratio for the 2nd iteration tends towards $9/16$. So a mrlycarpet0.2 is around 56% filled. We calculate that the Carpet Ratio for the 3rd iteration tends towards $27/64$. So a mrlycarpet0.3 is around 42% filled. We can, of course, see a pattern here. The formula for calculating the Carpet Ratio limit of any iteration (i) is simply $(3/4)^i$.

As expected, we find that the mrlycarpet dimensions tend towards 2, which is the dimension of a square. So our mrlycarpet0, with infinite Grid, Void and Fill Squares is practically, well, a square. For someone living on a mrlycarpet0, their world must look pretty two-dimensional, even though it's not. That's enough carpet business for now. Let's have some fun with sponges.

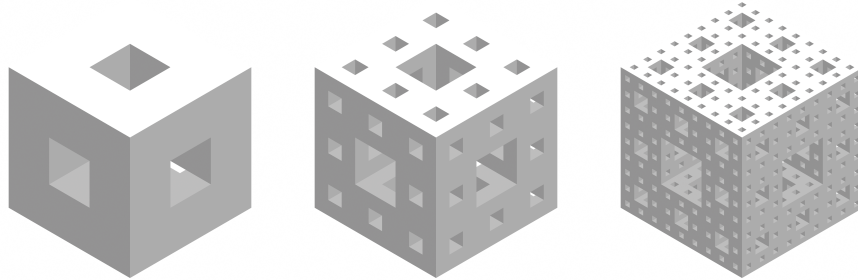
The Third Dimension

In three dimensions, our mrlycarpets become mrlysporges. Things are getting ever more trickier. Not only do we have to calculate the volume of any iteration of any mrly sponge, we also need to calculate the surface area of any iteration of any mrly sponge. Boy, oh, boy. This is no easy feat. Make sure your math hat is nice and tight.



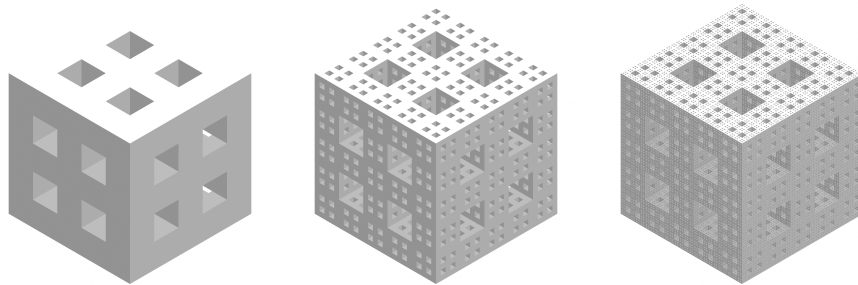
mrly sponge3

The first fractal is the mrly sponge3, also called the Menger Sponge. To build this one, we simply take our mrlytree3 Grid Lines, and cube to find 27 Grid Cubes. We then punch holes to make each face look like the 1st iteration of a mrlycarpet3, remove the centre cube, and end up with 7 Void Cubes and 20 Fill Cubes. We can then use this structure to make another one that looks just like it, and do it forever.



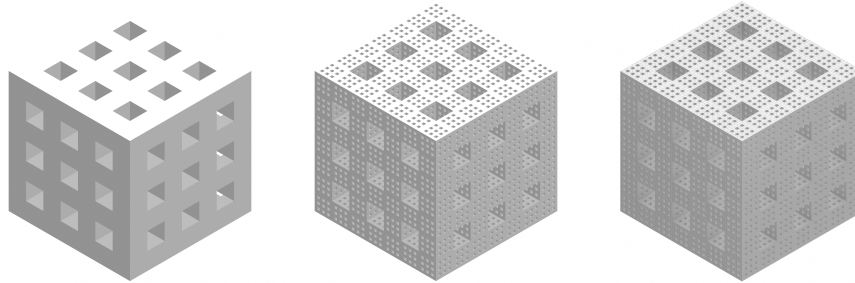
mrly sponge5

My beloved brick, the mrly sponge5, is the first one that I discovered, and holds a special place in my heart. To make one, we take the mrlytree5 Grid Lines, and cube to find 125 Grid Cubes. We then punch holes in each face to make them look like 1st iteration mrlycarpet5s, remove some more cubes, and eventually end up with 44 Void Cubes and 81 Fill Cubes. Once again, we can use this thing to make more.



mrlysponge7

The third fractal is the mrlysponge7. Here, we take our mrlytree7 Grid Lines, and cube to find 349 Grid Cubes. We do what we have to do, and end up with 135 Void Cubes and 208 Fill Cubes. We can, of course, make a mrlysponge7 out mrlysponge7s, forever.



As you can probably imagine, we can iterate through our mrlytree Grid Lines, and create an infinite amount of mrlysponges. Of what practical use could an arbitrarily chosen mrlysponge3579 ever be to us? I do not yet know. For now, this is all purely mathematical. Let's now take a look at the mrlysponge formulas and values.

mrlysponges		1	3	5	7	9	11	13
Number		1	2	3	4	5	6	7
Grid Cubes	$gc = (2*n-1)^3$	1	27	125	343	729	1331	2197
2nd Iteration	$gc2 = gc^2$	1	729	15625	117649	531441	1771561	4826809
3rd Iteration	$gc3 = gc^3$	1	19683	1953125	40353607	387420489	2357947691	10604499373
Void Cubes	$vc = 4*n^3-9*n^2+6*n-1$	0	7	44	135	304	575	972
2nd Iteration	$vc2 = gc2-fc2$	0	329	9064	74385	350816	1200025	3326184
3rd Iteration	$vc3 = gc3-fc3$	0	11683	1421684	31354695	310654864	1925866475	8766233748
Fill Cubes	$fc = 4*n^3-3*n^2$	1	20	81	208	425	756	1225
2nd Iteration	$fc2 = fc^2$	1	400	6561	43264	180625	571536	1500625
3rd Iteration	$fc3 = fc^3$	1	8000	531441	8998912	76765625	432081216	1838265625
Sponge Ratio	$sr = fc/gc$	1	0,7407	0,6480	0,6064	0,5830	0,5680	0,5576
2nd Iteration	$sr2 = fc2/gc2$	1	0,5487	0,4199	0,3677	0,3399	0,3226	0,3109
3rd Iteration	$sr3 = fc3/gc3$	1	0,4064	0,2721	0,2230	0,1981	0,1832	0,1733
Sponge Dimension	$sd = \log(fc/gl)$	3	2,7268	2,7304	2,7430	2,7544	2,7641	2,7723

As one might not expect, as we iterate through our mrlysponges, we find that the Sponge Ratio, which is basically the volume coefficient, tends towards 1/2. So the volume of the mrlysponge0.1 is only half that of a standard cube. This is a bit absurd, since we're punching an infinite amount of holes into it. One would naturally expect the volume of such an infinitely porous structure to be very close to zero, but it's not. For the 2nd iterations, we find that the Sponge Ratio tends towards 1/4. So a mrlysponge0.2 is around 25% filled. In fact,

we find exactly what we had with the Tree Ratio. So the Sponge Ratio limit for any given iteration (i) is simply $(1/2)^i$. So a true mrlysponge0 is pretty perfectly void. If the one-dimensional limit is equal to the three-dimensional limit, would the two-dimensional limit be equal to the four-dimensional limit? Can't wait to find out. As one would expect, we find that as we iterate through our mrllysponges, the Sponge Dimension tends towards 3, which is the dimension of a cube. For someone living in a mrlysponge0, their world must look pretty three-dimensional, even though it's not.

Let's now take a look at the mrlysponge surface area formulas.

mrllysponges		1	3	5	7	9	11	13
Number		1	2	3	4	5	6	7
Fill Squares	$fs = 3*n^2 - 2*n$	1	8	21	40	65	96	133
Fill Cubes	$fc = 4*n^3 - 3*n^2$	1	20	81	208	425	756	1225
1st Iteration								
Total Faces	$tf = 24*n^3 - 18*n^2$	6	120	486	1248	2550	4536	7350
Hidden Faces	$hf = 12*n^3 - 12*n^2$	0	48	216	576	1200	2160	3528
Visible Faces	$vf = 12*n^3 - 6*n^2$	6	72	270	672	1350	2376	3822
Surface Area	$sa = vf/gs$	6	8,0000	10,8000	13,7143	16,6667	19,6364	22,6154
2nd Iteration								
Total Faces	$tf2 = fc*vf$	6	1440	21870	139776	573750	1796256	4681950
Hidden Faces	$hf2 = fs*hf$	0	384	4536	23040	78000	207360	469224
Visible Faces	$vf2 = tf2 - hf2$	6	1056	17334	116736	495750	1588896	4212726
Surface Area	$sa2 = vf2/gs2$	6	13,0370	27,7344	48,6197	75,5601	108,5237	147,4992
3rd Iteration								
Total Faces	$tf3 = fc*vf2$	6	21120	1404054	24281088	210693750	1201205376	5160589350
Hidden Faces	$hf3 = fs*hf2$	0	3072	95256	921600	5070000	19906560	62406792
Visible Faces	$vf3 = tf3 - hf3$	6	18048	1308798	23359488	205623750	1181298816	5098182558
Surface Area	$sa3 = vf3/gs3$	6	24,7572	83,7631	198,5524	386,9174	666,8124	1056,2221

Calculating the surface areas is more or less straightforward. Admittedly, I did get some expert help. In essence, to calculate the Surface Area (sa), we divide the amount of Visible Faces (vf) by the Grid Squares. So the 1st iteration of a mrlysponge3 has 72 Visible Faces, divided by its 9 Grid Squares, we find a surface area coefficient of 8. So if we have a mrlysponge3.1 of side length 3cm, we find it has a surface area of $8 * 3^2$, which is $72cm^2$. At the moment, I can't explain in full detail what's going on here. But basically, to get the surface area coefficient, we need the amount of Visible Faces. To get the amount of Visible Faces, we calculate the Total Faces (tf) minus the Hidden Faces (hf). The 1st iteration generating functions are given in the table above. I have not yet found a generating function to calculate the surface area of any sponge for any iteration, only a recurrence formula, given below.

$$vf(i + 1) = fc * vf(i) - fs * hf(i)$$

With this recurrence formula, and the 1st iteration Total Faces and Hidden Faces formulas, we have everything we need to calculate the surface areas of any iterated mrlysponge. There is surely room for improvement here.

Sequences

Here, I have assembled a list of all OEIS sequences I could find that already exist, but have not been linked to mrlymath. For example, Fill Squares (1, 8, 21, 40, etc.) is the well known Octagonal Numbers Set, whereas Void Cubes (0, 7, 44, 135, etc.) is unknown. I find it amazing that such simple shapes can represent a wide variety of sequences. Perhaps a good tool for teaching basic math in school? This list only contains 1st iteration sets.

Number — $n(n + 1) = n + 1$
1, 2, 3, 4, 5, 6, 7... Positive Integers, A000027

1D mrllyset

Grid Lines — $gl = 2n - 1$
1, 3, 5, 7, 9, 11, 13... Odd Numbers, A005408
Void Lines — $vl = n - 1$
0, 1, 2, 3, 4, 5, 6... Nonnegative Integers, A001477
Fill Lines — $fl = n$
1, 2, 3, 4, 5, 6, 7... Positive Integers, A000027

2D mrllyset

Grid Squares — $gs = (2n - 1)^2$
1, 9, 25, 49, 81, 121, 169... Odd Squares, A016754
Void Squares — $vs = (n - 1)^2$
0, 1, 4, 9, 16, 25, 36... Squares, A000290
Fill Squares — $fs = 3n^2 - 2n$
1, 8, 21, 40, 65, 96, 133... Octagonal Numbers, A000567

3D mrllyset

Grid Cubes — $gc = (2n - 1)^3$
1, 27, 125, 343, 729, 1331, 2197... Odd Cubes, A016755
Void Cubes — $vc = 4n^3 - 9n^2 + 6n - 1$
0, 7, 44, 135, 304, 575, 972... UNKNOWN
Fill Cubes — $fc = 4n^3 - 3n^2$
1, 20, 81, 208, 425, 756, 1225... Number of divisors of 240^n , A103532
Total Faces — $tf = 24n^3 - 18n^2$
6, 120, 486, 1248, 2550, 4536, 7350... UNKNOWN
Hidden Faces — $hf = 12n^3 - 12n^2$
0, 48, 216, 576, 1200, 2160, 3528... UNKNOWN
Visible Faces — $vf = 12n^3 - 6n^2$
6, 72, 270, 672, 1350, 2376, 3822... UNKNOWN

Questions

As you can see, and imagine, there is still lots to discover. I'd appreciate any form of help. You can send me an email.

Media

You can find compressed and uncompressed imagery on my Google Drive.

Definitions

I would love to see formal mrlymath definitions.

Functions

Where missing, I would also love to see generating functions. For example, the mrlly sponge surface area formulas are totally recursive at the moment.

Dimensions

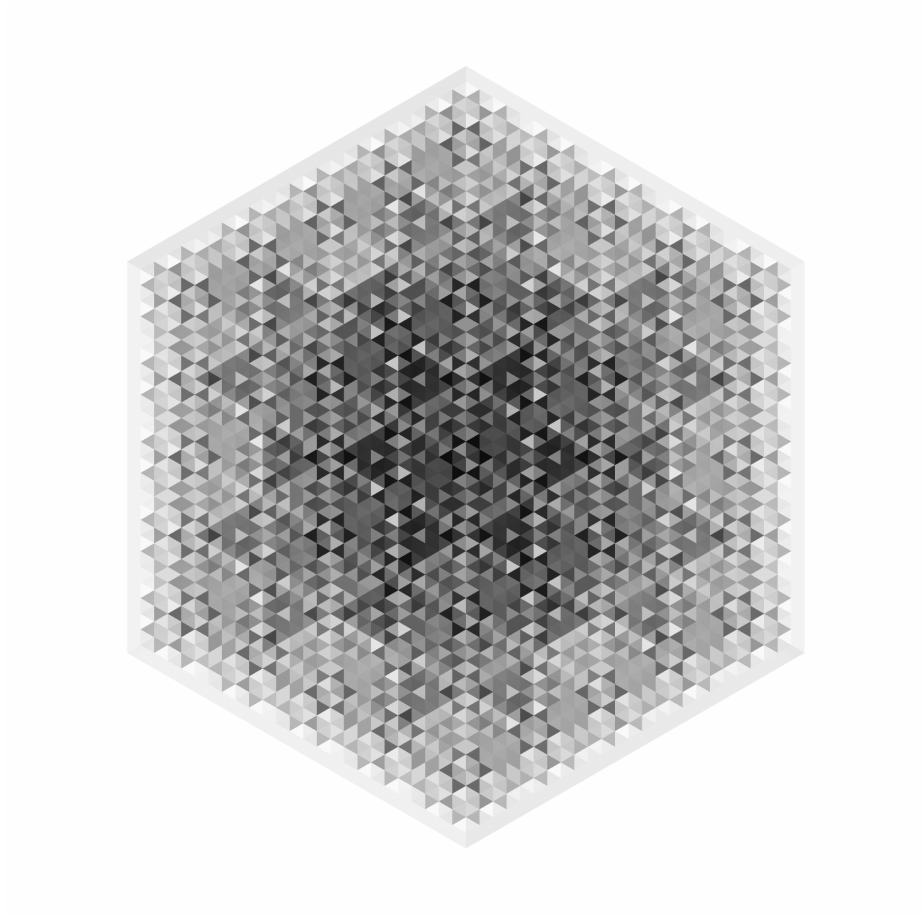
As with the cube and the tesseract, there may be higher-dimensional mrlymath to discover. Perhaps here we will find interesting answers to some of the mrlmysteries.

Animations

If you are great with animation software, I would very much appreciate mrllyset renderings made publicly available on YouTube. I have made quite a few GIFs, which you can find on my GIPHY.

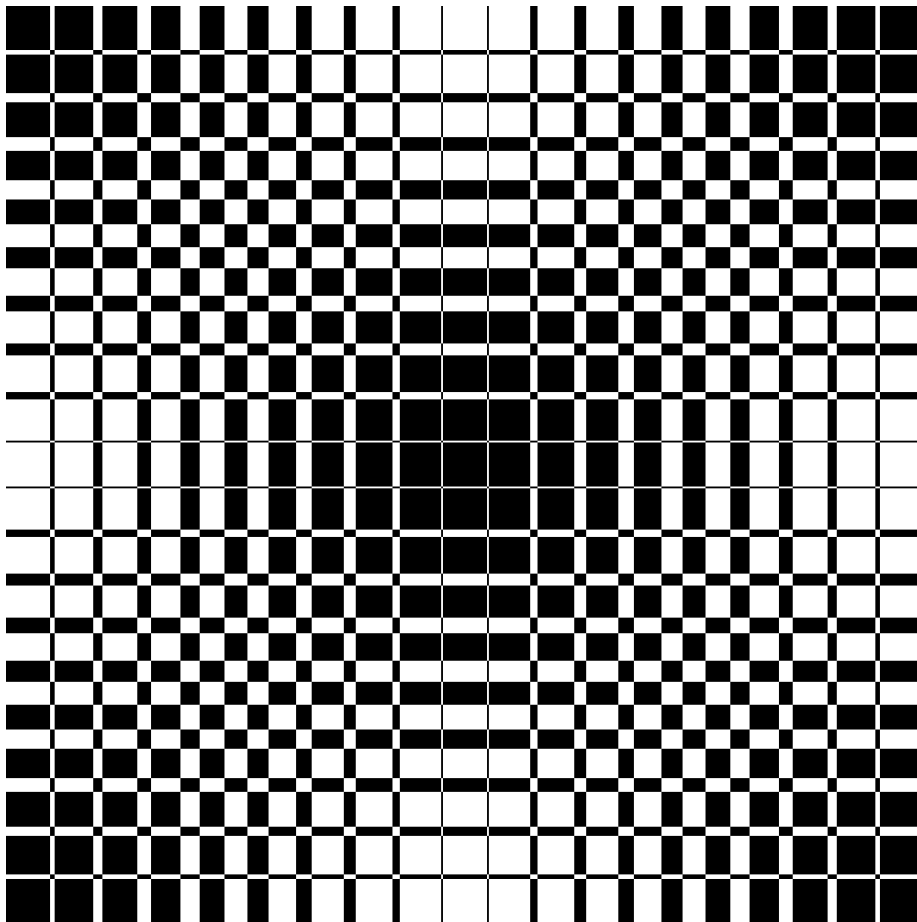
Slices

Similar to the wonderful way in which a Menger Sponge can be sliced open to reveal six-fold hexagonal symmetry, I am absolutely sure that mrllysporges can be sliced open so as to reveal even more wondrous beauty. Believe it or not, I've never sliced a mrllysporge. The only thing I could do is to drop the opacity, which reveals some beautiful symmetry. Pictured below is a transparent mrllysporge5.2.



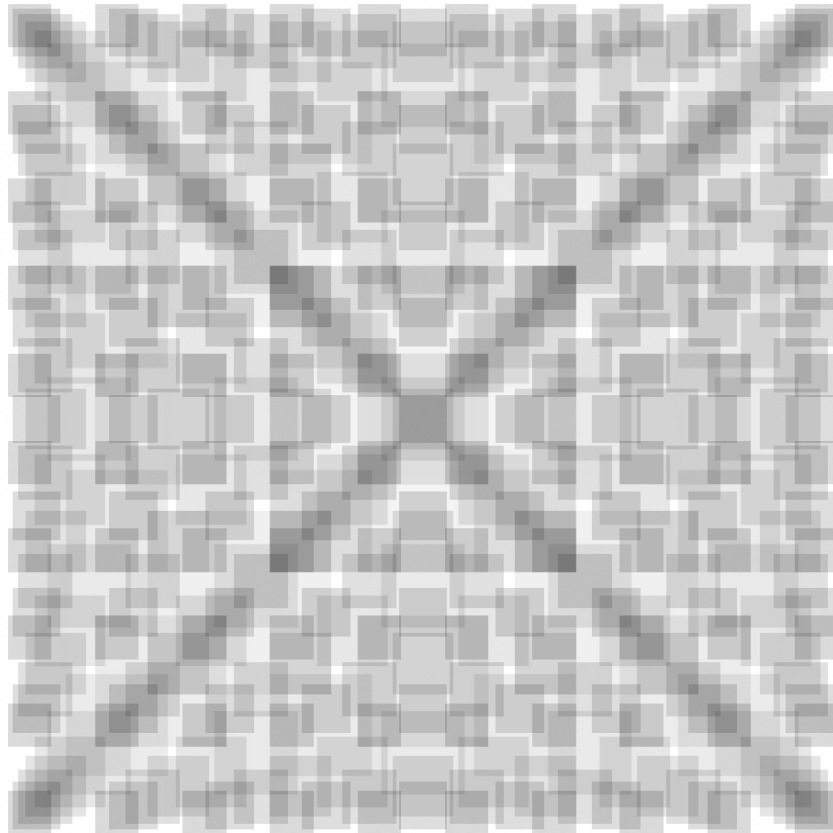
Pi

Inspired by the Wallis Sieve, I tried to find pi (π) hidden within the mrlyset. Unfortunately, I did not succeed, probably because I'm not good enough a mathematician. I remain confident that pi can, one way or another, be calculated. While investigating this, I came across some very bizarre things. Like with the Wallis Sieve, I tried to create a shape from mrlycarpets that would have the same surface area as a circle. Playing around in Photoshop, some spectacular designs suddenly emerged. Most animations clearly show that parity alternates from one mrlyset to another. For example, the centre of the mrlycarpet3 must always be void, whereas the centre of the mrlycarpet5 must always be filled, and so on. The picture below is a mrlycarpet23.1 on top of a mrlycarpet21.1 with difference filter enabled. I wonder what a mrlycarpet2323.1 on top of a mrlycarpet2321.1 would look like. Such visualisations must be done programatically, and will most certainly leave the developer completely bewildered.



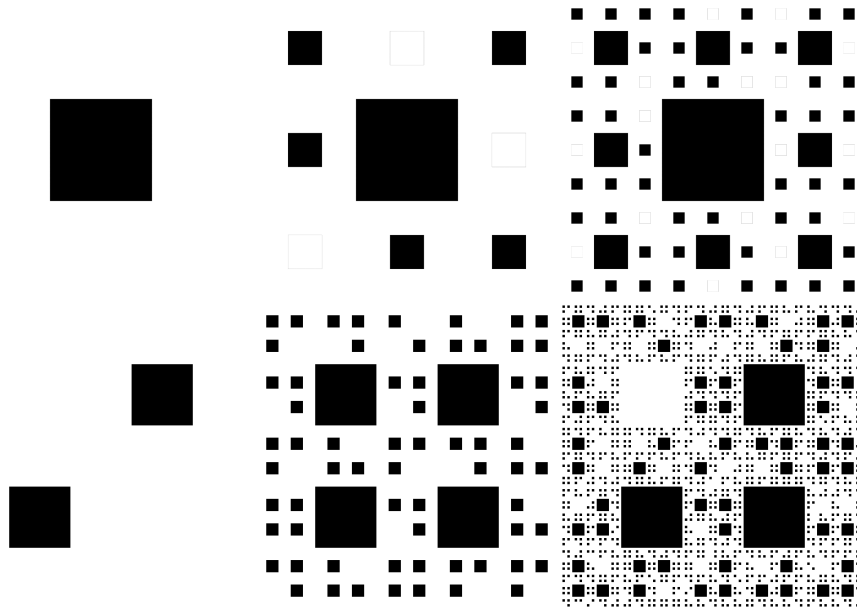
Octagons

It's been quite some time now that I've known that mrlycarpets are linked to octagons, but I could never quite explain why. While having fun in Photoshop, an octagon suddenly appeared! Actually, it's two tilted squares, that together form an octagon. I assume that with an infinite amount of mrlycarpets superimposed one on top of the other, we could draw out a perfect octagon.



Codes

Using our fractals, it is possible to create a wide variety of data matrices, which I am calling mrlycodes. These can be one-dimensional (like barcodes), two-dimensional (like QR codes) or three-dimensional. Pictured below are some mrlycarpetcodes, with the Void Squares switched on and off. One could also leave the Void Squares as they are, and instead flick on and off the Fill Squares. One could also use colors. For example, a simple mrlycarpetcode7.1 where the 9 Void Squares are either Red, Green or Blue would result in 19683 (3^9) possible combinations. This is a rather vast topic, perhaps of interest to computer scientists.



Conclusion

I'm not a mathematician, but I like math, so I thought I might give it a go. All I want is for mathematicians to wake up to this new reality, and dive deeper into the wonderful world of mrlymath. Once mathematicians figure out what there is to figure out, I'm sure mrlymath will find itself to be applied pretty much everywhere.

Dedications

This paper is dedicated to Georg Cantor, Waclaw Sierpinski and Karl Menger. They're all dead, but I'm sure they're celebrating in paradise.

Further Reading

If you haven't already, feel free to read my mrlystarter paper, in which I unleash my mrllybeast, and apply it to construction and manufacturing.

Acknowledgements

As always, many thanks to the universe and everything within it.

Peace and Love,
Carlo Mats

The End