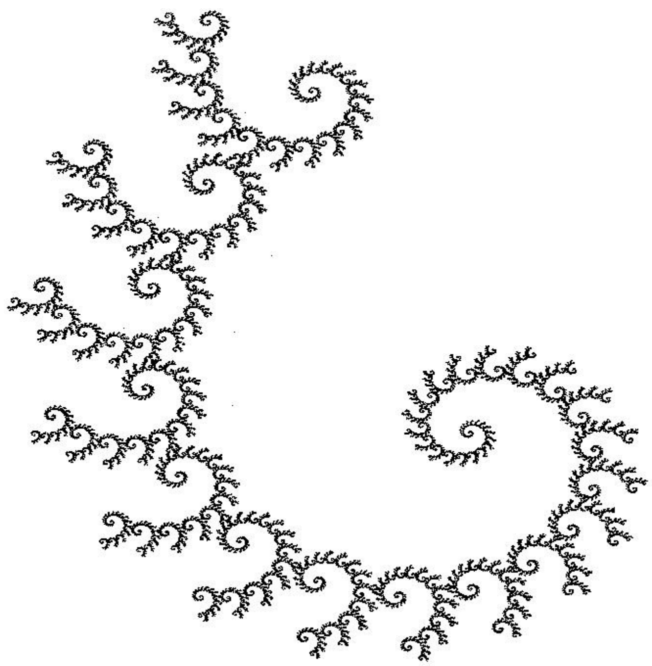
Fantastic Shapes and Where to Find Them: Generating Complexity from Linear Transformations

Igor Bezoyan

March 2026



Contents

1 Introduction

Have you ever been to a park? Likely not; us mathematicians spend most days inside, regardless of whether we are occupied or not. But after reading this essay, it will be surprisingly easy for you to pretend that you are, indeed, in touch with your primal roots, as I will show how present fractals and complex shapes are in nature, and moreover, everyday life!

2 Linear transformations

What do you get when you cross a mountain climber with a mosquito? The premise is ridiculous; you can't cross a scaler with a vector.

Hopefully my lack of humour won't discourage you from reading this section, I promise it's not contagious.

2.1 What are linear transformations?

A linear transformation is mathematically defined as a function mapping vectors from one vector space to another that preserves linear combinations. It must satisfy two key properties: additivity and homogeneity. Linear transformations keep grid lines parallel and evenly spaced, and the origin fixed.

Additivity:

$$T(\mathbf{u}+\mathbf{v}) = T(\mathbf{u}) + T(\mathbf{v}) \quad \forall\, \mathbf{u}, \mathbf{v}$$

Homogeneity:

$$T(k\mathbf{v}) = kT(\mathbf{v})\ \ \forall\ \ k \in \mathbb{R},\ \mathbf{v}$$

It is important to note that homogeneity applies to $k \in \mathbb{C}$, if and only if the vector $\mathbf{v}$ is also defined in the complex plane.

For some function $M(\mathbf{v})$, where $\mathbf{v}$ is a vector, we can say that, the vector $\mathbf{v}$, can be rewritten as a column vector.

$$\mathbf{v} = \begin{pmatrix} x \\ y \end{pmatrix}$$

Then, a linear transformation $M(\mathbf{v})$ in $\mathbb{R}^2$, will transform this point. if our transformation, or more specifically matrix M is defined by:

$$M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

Then any linear transformation by M upon a point (x, y) is given by:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} ax+by \\ cx+dy \end{pmatrix}$$

2.2 Geometric interpretations

We now describe several important classes of transformations.

<u>Horizontal shear:</u>

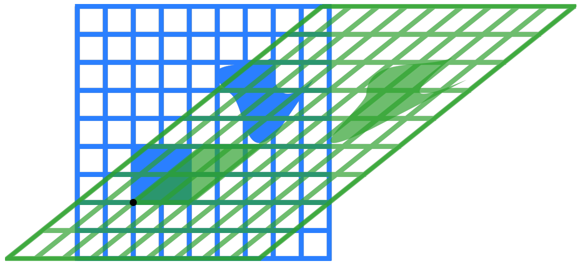


Figure 1: Horizontal shear

A horizontal shear with parameter $p \in \mathbb{R}$ is given by

$$S_h(p) = \begin{pmatrix} 1 & p \\ 0 & 1 \end{pmatrix}.$$

Which maps

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} x + py \\ y \end{pmatrix}$$

<u>Reflection:</u>

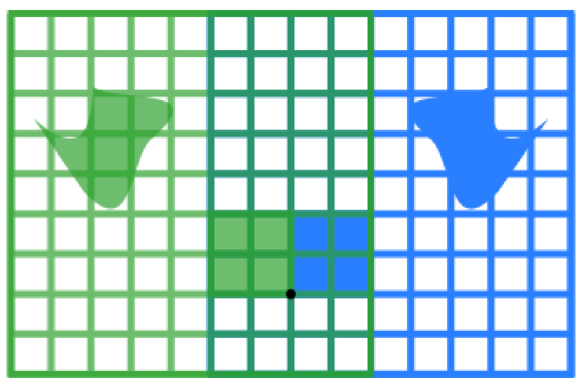


Figure 2: Reflection

Reflection in the y-axis:

$$R_y = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} -x \\ y \end{pmatrix}$$

Reflection in the x-axis:

$$R_x = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} x \\ -y \end{pmatrix}$$

Scaling dilation:

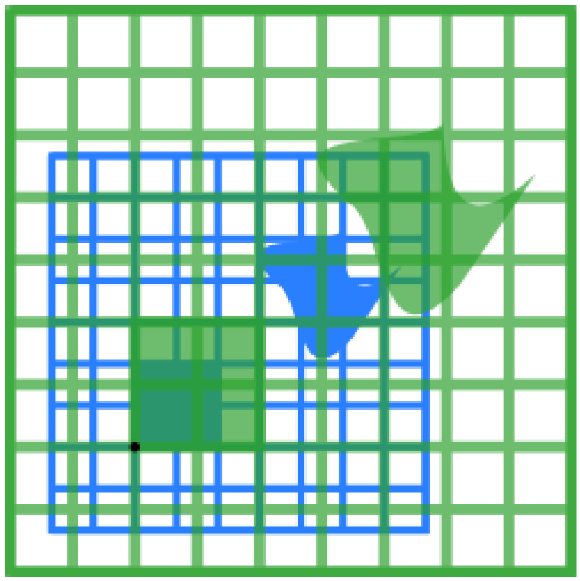


Figure 3: Scaling

Uniform scaling by factor $\phi \in \mathbb{R}$:

$$D(\phi) = \begin{pmatrix} \phi & 0 \\ 0 & \phi \end{pmatrix}.$$

Non-uniform scaling:

$$D\begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix} = \begin{pmatrix} \phi_1 & 0 \\ 0 & \phi_2 \end{pmatrix}.$$

This maps

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} \phi_1 x \\ \phi_2 y \end{pmatrix}$$

Squeeze mapping:

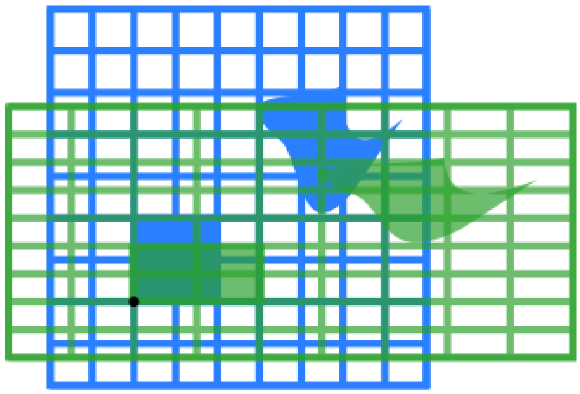


Figure 4: Squeeze mapping

A squeeze (area-preserving scaling) is given by, for a scalar $\Omega \in \mathbb{R}$

$$S(\Omega) = \begin{pmatrix} \Omega & 0 \\ 0 & \frac{1}{\Omega} \end{pmatrix}, \quad \Omega > 0.$$

This stretches in one direction and compresses in the other.

Rotation:

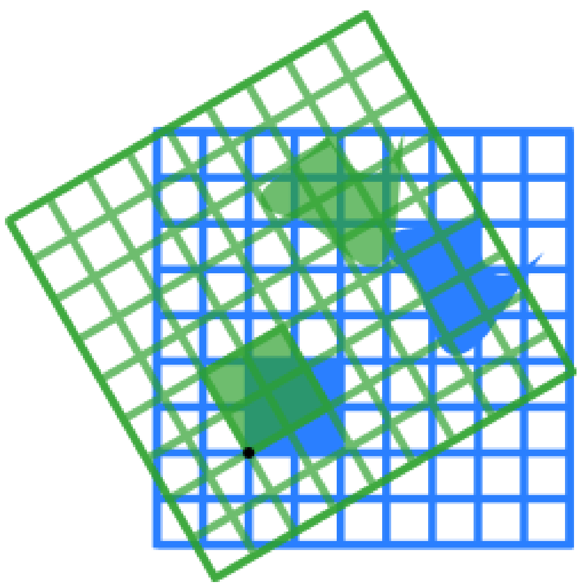


Figure 5: Rotation

A rotation by angle θ about the origin is given by

$$R(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}.$$

This maps

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} x\cos\theta - y\sin\theta \\ x\sin\theta + y\cos\theta \end{pmatrix}$$

3 Iterations

3.1 What are iterative sequences?

Iteration refers to repeatedly applying a function, such that the previous output is the subsequent input.

$$S_{n+1} = f(S_n) \quad f : \mathbb{R} \mapsto \mathbb{R}$$

Examples include:

1. The Fibonacci sequence
$$F_{n+1} = F_n + F_{n-1}$$
2. Geometric series
$$S_{n+1} = kS_n\ ,\ k \in \mathbb{R}$$
3. Arithmetic series
$$S_{n+1} = S_n + k\ ,\ k \in \mathbb{R}$$

As mentioned previously, a matrix F represents a linear transformation, which can be viewed as a function $f : \mathbb{R}^2 \mapsto \mathbb{R}^2$, mapping a point $\mathbf{x}_n = (x_n, y_n)$ to a new point $\mathbf{x}_{n+1}$ This allows us to cleanly transition from common iterative sequences with scalar constants, to iterative sequences with linear transformations, and allowing us to analyse their long-term behaviour under repeated application.

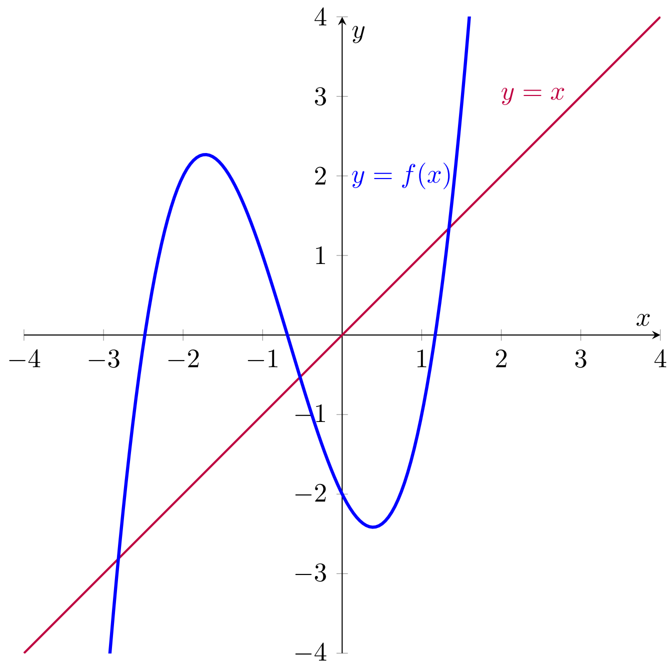
3.2 Behaviour under iteration

The first key behaviour to mention is the nature of fixed points. Some of you may link to this to invariant lines/points, which is an area of linear transformations.

Formally, a fixed point is a value such that it is mapped onto itself by some function, or remains fixed under a transformation. $f : \mathbb{R} \mapsto \mathbb{R}$. In other words, for some fixed point x_i, $f(x_i) = x_i$.

Below is a graph showing this visually for the function:

$$f(x) = x^3 + 2x^2 - 2x - 2$$



As seen in the graph, the points of intersection of $y = f(x)$ and $y = x$, are the invariant/fixed points.

If the linear transformation that we iteratively apply to some point reaches a fixed point x_i, such that for some matrix M, $M(x_i) = x_i$, we know that the sequence will converge. This can be a limiting factor to the complexity of the fractal/shape generated by our sequence.

An iterative sequence converges if its values approach a fixed point. On the other hand, an iterative sequence diverges if its values approach $\pm\infty$, or move away from the fixed point. So, for a converging iterative sequence:

$$x_{n+1} = f(x_n)$$

We say that, for some fixed point x_{fix}:

$$\lim_{n\to\infty} x_n = x_{fix}$$

It is also useful to note that all sequences that converge have a fixed point, but not all sequences with fixed points converge. This is due to fixed points having an either attracting or repelling nature.

A fixed point is said to be attracting, or stable, if when you choose a point x_i, which is within the set $\mathbf{N}$, the 'neighbouring set' to x_{fix}, then after repeated iteration, the point x_i converges to x_{fix} and is in other words, 'attracted' by the fixed point. The opposite can be said for an unstable/repelling point, as points $x_i \in \mathbf{N}$, after repeated iteration, move further from the point x_{fix}. This is important to constructing complex shapes, as we will have to select a point to be the x_0 in our iteration. If we choose poorly, this may lead to divergence, causing a chaotic shape to be produced. Below is how we identify whether a fixed point is attracting or repelling. For any unfamiliar readers, I recommend looking at differentiation before continuing. For the following section, let $f : \mathbb{R} \mapsto \mathbb{R}$ be our chosen function, and x^* be our fixed point of $f(x)$.

First, let's introduce a small error, let it be called $\varepsilon \in \mathbb{R}$.

$$x_n = x^* + \varepsilon$$

Let's use a topic called linear approximation, which is where we model a segment of a function as a straight line using this definition:

$$f(a+h) = f(a) + f'(a)(h)$$

For some point $(a, f(a))$ on the curve, and a small deviation h. We can now smoothly apply our function $f(x)$ to our definition of x_n.

$$f(x_n) = f(x^* + \varepsilon).$$

Now, we know by the properties of our function that $f(x_n) = x_{n+1}$, and hence we can use linear approximation, where $a = x^*$ and $h = \varepsilon$ to obtain:

$$x_{n+1} = f(x^*) + f'(x^*)\varepsilon$$

And as x^* is a fixed point:

$$x_{n+1} = x^* + f'(x^*)\varepsilon$$

Now we can see that, as we keep iterating, our values will either converge to x^* or diverge from it. As ε and x^* are constant, the divergence/convergence is caused by the magnitude of $f'(x^*)$. Therefore, we can say that if $|f'(x^*)| < 1$ then the fixed point is stable, and if $|f'(x^*)| > 1$, then it is unstable.

Certain sequences can also introduce complexity, for instance certain functions can cause oscillation between two unique points. Likewise, a certain iterative sequence may be periodic, meaning that after some period length n, $n \in \mathbb{N}$, the values x_i of the sequence will repeat.

4 Fractals

If you have read this far, you have just entered the most exciting part of this essay - the one with lots of pictures!

4.1 Defining fractals and self-similarity

What is a fractal? Simply put, a fractal is a geometric object that typically exhibits self similarity at multiple scales, often arising from the repeated application of simple rules.

The property of self similarity is defined as the replication of similar patterns at increasingly small scales. Take for instance, the fractal below, called a Menger sponge, which is affine self-similar.

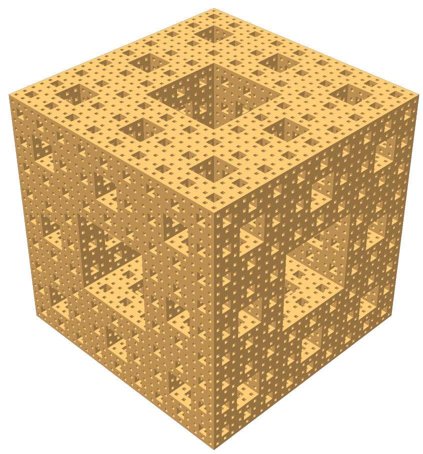


Figure 6: Menger Sponge

I also promised in the beginning to help you pretend you go outside! Here is another fractal called the Barnsley Fern:

Figure 7: Barnsley Fern

Though I won't go into much detail around these, they are still highly fascinating and you should check them out in your free time. Also yes, I indeed dramatically foreshadowed the Barnsley Fern with its non-fractal counterpart in the **Introduction**.

4.2 A brief tribute to Sierpiński

One of the most famous fractals is the work of Sierpiński, with infinite perimeter, yet no area...

I introduce, the Sierpiński triangle! - *red carpet rolls out.*

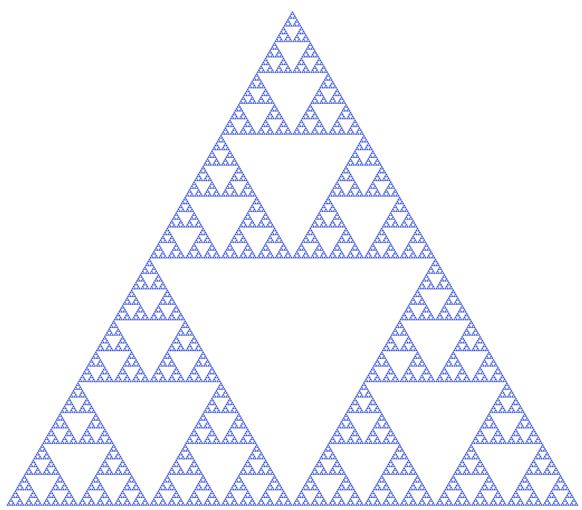


Figure 8: Sierpiński triangle

The process of generating the following fractal is in line with the definition - repetition of simple rules. It involves removing the central equilateral triangle repeatedly, and although it may seem like it has area, that's only due to the line thickness.

Here is how we generate this shape. First, let us understand visually. At each step, some triangle T_i is being shrunk by a factor half, and then being moved to the either the bottom left/right, or top.

So our 3 pairs of matrices (A_i), and translation vectors ($\mathbf{b}_i$) are as follows:

Pair 1:

$$A_1 = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \quad \mathbf{b}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Pair 2:

$$A_2 = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \quad \mathbf{b}_2 = \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix}$$

Pair 3:

$$A_3 = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \quad \mathbf{b}_3 = \begin{pmatrix} \frac{1}{4} \\ \frac{\sqrt{3}}{4} \end{pmatrix}$$

And so our 'iterative sequence' is given by:

$$\mathbf{x}_{n+1} = A_i\mathbf{x}_n + b_i \quad i \in \{1, 2, 3\},\ n \in \mathbb{N}_0$$

Where now you can cycle through the values of i, which will, as the number of repetitions approaches infinity, create the Sierpiński triangle. Since each transformation contracts space, the system converges to a limiting set, known as the attractor of the iterated function system.

4.3 Fractals from linear transformations

Now that we have looked at how to construct a famous example, lets aim to generalise and reach a conclusion about which linear transformations lead to the generation of fractals.

I have previously introduced you to the idea of *affine transformations*, in other words:

$$\mathbf{x}_{n+1} = A\mathbf{x}_n + \mathbf{b}$$

An affine linear transformation is a combination of one of the five previously mentioned transformations (A), and a translation ($\mathbf{b}$), which follow the conditions necessary for linear transformations. We have noticed with the Sierpiński triangle that, these affine transformations, when iterated, can produce complex structures.

In order for fractals to form, rather than erupt and diverge indeterminantly, it is necessary that the points are drawn together, meaning that the applied transformations need to contract space. This behaviour is determined by the eigenvalues of the transformation matrix. Generally, though not universally, we can say that if:

$$|\lambda_i| < 1$$

Then the transformations contract to a particular attractive fixed point x^*, and if:

$$|\lambda_i| > 1$$

Then the transformations expand and diverge. Thus, we can conclude that the ability of linear transformation to form fractals depends on their eigenvalues, which indicate the movement of points under repeated application of the linear transformation. More formally, a collection of such linear transformations is referred to as an *Iterated Function System* (IFS), which is used to generate certain fractals like the Sierpiński triangle, though it is one of multiple methods. Thus, fractal

structure emerges from the repeated combination of contractive transformations, rather than their individual complexity.

4.4 Why does complexity occur?

It may seem surprising to the reader why complexity occurs, even with simple linear transformations, and there are three key reasons.

Since $\det(A^n) = (\det A)^n$, repeatedly applying a transformation leads to an exponential scaling of the area. If $|\det A| < 1$, this leads to the points contracting exponentially towards the limiting set.

Secondly, consider contraction and repetition. If, such as in the case of the Sierpiński, we apply multiple linear transformations, a pattern in the value of our points should appear. Even thinking as plainly as in one dimension, what would happen if we scaled by $\frac{1}{2}$, then $\frac{1}{3}$, then $\frac{1}{4}$, in that periodic loop? This can be visualised on a number line.

Lastly, think about the idea of having multiple distinct transformations. Interactions between distinct transformations can produce competing geometric effects.

5 Case study: Constructing and analysing a fractal.

5.1 Constructing a fractal using matrices

We construct an IFS using three affine transformations satisfying $|\lambda| < 1$. This guarantees convergence to a unique limiting set (the attractor) under repeated iteration.

This was simulated in Python using the following IFS:

$$\mathbf{x}_{n+1} = A_i\mathbf{x}_n + b_i \quad i \in \{1,2,3\},\ n \in \mathbb{N}_0$$

At each step, a transformation pair $(A_i, \mathbf{b}_i)$ is selected at random.
<u>Transformation 1, non uniform scaling:</u>

$$A_1 = \begin{pmatrix} 0.6 & 0 \\ 0 & 0.35 \end{pmatrix} \quad \lambda_1 = 0.6 \quad \lambda_2 = 0.35 \therefore |\lambda_2| < |\lambda_1| < 1$$

$$\mathbf{b}_1 = \begin{pmatrix} -0.2 \\ 0 \end{pmatrix}$$

<u>Transformation 2, Rotation + uniform contraction:</u>

$$A_2 = \begin{pmatrix} 0.5 & -0.3 \\ 0.3 & 0.5 \end{pmatrix} \quad \lambda = 0.5 \pm 0.3i \quad \therefore |\lambda| < 1$$

$$\mathbf{b}_2 = \begin{pmatrix} 0.2 \\ 0 \end{pmatrix}$$

Transformation 3, Shear + Contraction

$$A_3 = \begin{pmatrix} 0.4 & 0.2 \\ 0 & 0.4 \end{pmatrix} \qquad \lambda_1 = \lambda_2 = 0.4 \therefore |\lambda_2| = |\lambda_1| < 1$$

$$\mathbf{b}_3 = \begin{pmatrix} 0 \\ -0.25 \end{pmatrix}$$

As $\forall \quad \lambda_i, |\lambda_i| < 1$, all transformations are contractive, ensuring convergence to a bounded attractor. The complex eigenvalue of *Transformation 2* indicates a rotational component in the transformation.

Plotting 40000 points produces the following fractal:

Figure 9: The Flaming Dorito

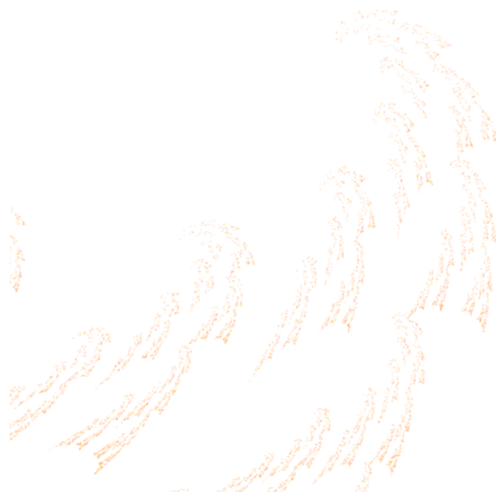


Figure 10: The Flaming Dorito: Zoomed In

Although Figure 10 is fairly opaque, Figure 9 bears an unfortunate resemblance to a flaming Dorito, suggesting that affine transformations may be more relevant to snack geometry than I had anticipated... And you thought I would it make it appear like mathematicians go outside, rather than spending far too long indoors, fuelled by fiery snacks.

5.2 Changes and Limitations

If the values of $|\lambda_i|$ had increased, the graph would be more dispersed, a similar effect to the increase in $|\mathbf{b}_i|$. This is because an increase in both the translation vector and factor of scaling would cause the subsequent points to disperse more from the starting point.

If we take a look at the complex eigenvalue of Transformation 2, we can understand that, if the $Im(\lambda_2)$ were to increase, the magnitude by which the rotation matrix rotates the points would increase, causing our fractal to bend more and look like a wave, but I don't believe in coincidences...

This system is highly sensitive to parameter choice, and our Dorito was a very lucky construction... or was it? Additionally, as it was constructed using Python, the accuracy of the model is limited by the number of points plotted, plotting density and visualisation settings. Lastly, the system provides little analytical insight, as even though the attractor can be generated computationally, the geometric properties are difficult to describe and prove.

6 Conclusion

So, after this long journey, did you, reader, find out where to find fantastic shapes, and how to generate complexity from linear transformations? If you just said no, then I must be a terrible teacher, and I should never be allowed to write again. If you said yes, I hope you instantly thought that, for linear transformations to generate complex shapes, we must ensure that they are contractive, and that we iterate by alternating between multiple distinct transformations, such as rotations, scaling and the shear. Overall, this highlights that complexity can arise from the simplest of algorithms, and that fantastic shapes can be found (even household snacks), given that you know how to look for them.

7 Bibliography and References

1. Diagrams for linear transformations. Wikipedia. Available at: https://en.wikipedia.org/wiki/Matrix_(mathematics)#Linear_transformations
2. Menger sponge diagram. Wikimedia. Available at: https://commons.wikimedia.org/wiki/Category:Menger_sponge_diagonal_section
3. Sierpiński triangle diagram. Wikipedia. Available at: https://en.wikipedia.org/wiki/Sierpiski_triangle
4. Barnsley Fern diagram. Wikipedia. Available at: https://en.wikipedia.org/wiki/Barnsley_fern
5. Formula for Linearisation of a function. Math LibreTexts. Available at: https://math.libretexts.org/Bookshelves/Calculus/Map%3A_University_Calculus_(Hass_et_al)/3%3A_Differentiation/3.11%3A_Linearization_and_Differentials