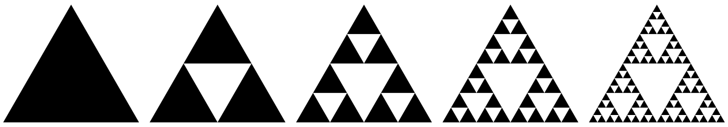
20 May 2025

Solutions 12 : Fractals

1 Similarity Dimension

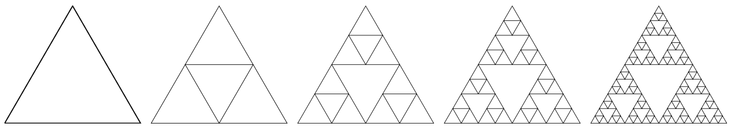
The Sierpinski triangle fractal is the object towards which the following iterations converge



(a) The side of the initial triangle has a length of 1. Compute the area of the Sierpinski triangle and deduce that its dimension is smaller than 2.

From an iteration to the next, a triangle is separated into four similar triangles, and one is taken out. This means at each iteration the area is multiplied by $3/4$. If the area of the n^{th} iteration is A_n then $A_n = A_0(3/4)^n$ which goes to zero as $n \to \infty$. Since it has an area of zero, the dimension must be smaller than 2.

(b) Now show that the "length" of the Sierpinski triangle is infinite, meaning that its dimension is greater than 1. To do so, notice how from one iteration to the next, the border of the shape stays inside the Sierpinski triangle and increases in size.



At the first iteration, the perimeter of the triangle is equal to 3. At the second iteration, 1 triangle of side 1/2 is added to the border, adding a length of 3/2. At the third iteration 3 triangles of side 1/4 are added to the border, adding a length of 9/4. Each time the number of triangles added is multiplied by 3 and the length of their sides is divided by 2. The length of this border is therefore

$$3 + \sum_{n=1}^{\infty} \frac{3^n}{2^n} = +\infty$$

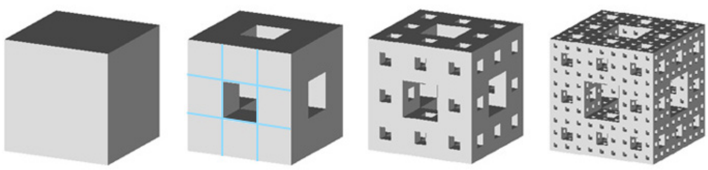
If the length of the border is infinite, then so is the "length" of the Sierpinski triangle. This means that the dimension of the fractal is greater than 1.

(c) Compute the similarity dimension of the Sierpinski triangle.

One can see that the Sierpinski triangle is composed of 3 copies of itself, and all 3 are reduced by a factor of 2. Therefore the similarity dimension is

$$d = \frac{\ln 3}{\ln 2} \approx 1.5850$$

The Menger sponge fractal is the fractal towards which the following iterations converge



We start with a cube, split it into 27 smaller cubes, remove the ones that form the "central cross", then repeat for all the smaller cubes left.

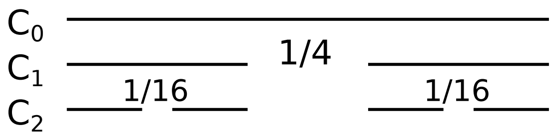
(d) Compute the similarity dimension of the Menger sponge.

The "central cross" that is removed is made up of 7 of the 27 smaller cubes, meaning that the Menger sponge contains 20 copies of itself, which are scaled down by 3. So the similarity dimension is

$$d = \frac{\ln 20}{\ln 3} \approx 2.7268$$

2 Fat Fractals

Fat fractals are fractals with a non-zero measure. Remember that for the Cantor middle set, S_0 starts with the interval $[0, 1]$ and then, from S_n to S_{n+1}, an open interval of size $1/3^{n+1}$ is removed from the middle of each sub-interval in S_n. The Cantor middle set is the limit S_∞. An example of a fat fractal, similar to the Cantor middle set, starts from C_0, which is the interval $[0, 1]$, and then, from C_n to C_{n+1}, intervals of size $1/4^{n+1}$ are removed.



(a) Briefly explain why this fractal is a topological Cantor set.

At each stage, there is a continuous transformation between S_n and C_n. The sub-intervals need to be streched and translated. Therefore they are topologically the same.

(b) Compute the measure of the fractal, to prove it is indeed fat.

The first stage C_0 has a measure of 1. Then we remove an interval of size $1/4$ and leaving 2 sub-intervals. For the next stage intervals of size $1/4^2 = 1/16$ are removed from each of the 2 sub-intervals, leaving 4 sub-intervals. And so on...To get from C_n to C_{n+1} we remove 2^n intervals of size $1/4^{n+1}$. The measure of the fractal is therefore

$$1 - \sum_{i=0}^{\infty} 2^i \frac{1}{4^{i+1}} = 1 - \frac{1}{4}\sum_{i=0}^{\infty}\frac{1}{2^i} = 1 - \frac{1}{4}\frac{1}{1-1/2} = \frac{1}{2}$$

The fractal has a non-zero measure, namely it is fat.

(c) What is therefore the dimension of this fractal ?

Since it has a finite length, its dimension is 1. If one wants to compute it with the tools seen in class, the similarity dimension cannot be used since the fractal is not self similar (take one of the two sub-intervals of C_1, the gap that is removed from its middle at C_2 does not represent $1/4$ of its length). Box dimension can be used. C_n contains $N = 2^n$ intervals of equal size. For $n > 0$, the measure of C_n is $1 - \frac{1}{4}\sum_{i=0}^{n}\frac{1}{2^i}$, therefore this number divided by 2^n is the size of the sub-intervals of C_n. The box dimension is

$$\lim_{\epsilon\to 0}\frac{\ln[N(\epsilon)]}{\ln(1/\epsilon)} = \lim_{n\to\infty}\frac{\ln(2^n)}{\ln\left[\frac{1}{\left[1-\frac{1}{4}\sum_{i=0}^{n}\frac{1}{2^i}\right]\frac{1}{2^n}}\right]} = \lim_{n\to\infty}\frac{\ln(2^n)}{\ln\left[\frac{1}{\frac{1}{2}\frac{1}{2^n}}\right]} = \lim_{n\to\infty}\frac{n\ln 2}{(n+1)\ln 2} = 1$$

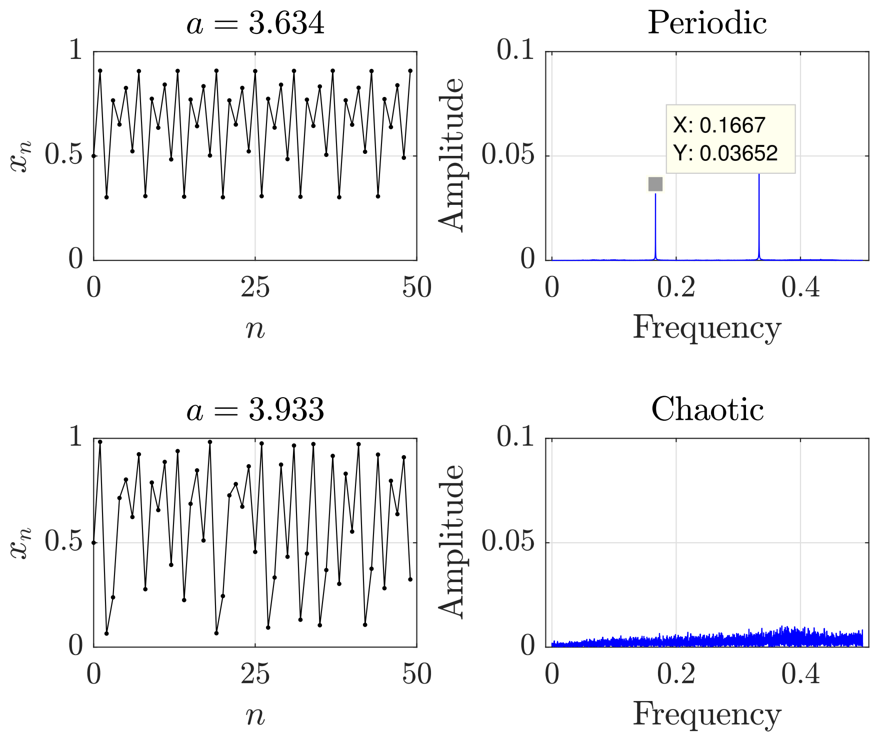
Fat fractals are linked to an important aspect of the logistic map. Farmer numerically proved in 1985 that the set of parameter values of a that lead to chaos is a fat fractal. He found that if a value of a is randomly chosen in the interval $[a_c \approx 3.57, 4]$ the resulting map is chaotic 89% of the time.

(d) On Matlab, generate a random, uniformly distributed, parameter value a inside $[a_c, 4]$, then iterate the map and determine whether the behaviour is chaotic or periodic. Average over multiple examples to compute the fraction at which chaos occurs.
Help : To generate a uniformly distributed number use command `rand`. To see if a map is periodic, iterate for long enough and apply the discrete Fourier transform. To do so use commands

```
DFT = fft(x);
DFT = abs(DFT/N);
DFT = DFT(2:N/2);
```

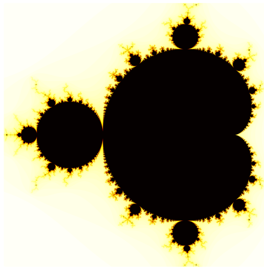
where `x` is the vector of size `N` containing the map. The absolute value is taken to get the amplitudes of each frequency. The third line is to get the single-sided spectrum. If you want to plot this, the amplitudes of `DFT` correspond to frequencies `freq=(1:N/2-1)/N`. You can then say that the map is periodic if the maximum value of `DFT` is above a certain threshold.

See the code for solutions. Since `rand()` generates a uniformly distributed variable on interval $[0, 1]$, just rescale and shift the interval using command `a = 3.57+0.43*rand()`. The threshold value above which the map is considered periodic is empirically found to be 0.03. The program given as solution finds that maps are chaotic around 81% of the time. Plots show an example of a map with a 6-cycle and a chaotic map.



3 Mandelbrot set

We consider the following sequence :



$$f : z_{n+1} = z_n^2 + c$$

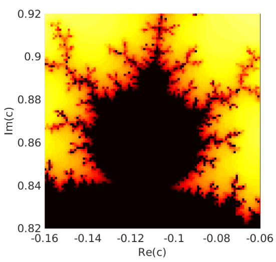
where $z, c \in \mathbb{C}$, $c = x + iy$, and $z_0 = 0$. The Mandelbrot set is the set of complex numbers c such that f is bounded when $n \to \infty$. This set has been defined, when discovered, as "the most complex object in Mathematics ". It is one of the best known examples of mathematical beauty. The boundary of this set in the complex space is a fractal.

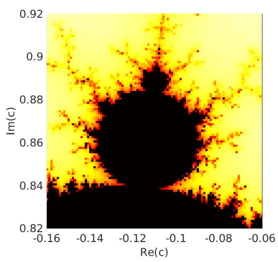
(a) On Matlab, write a script that shows graphically the Mandelbrot set in the complex plane. Define a rectangle of 2000×2000 points in the (x, y) plane. Consider $-1.5 \leq x \leq 0.5$ and $-1.0 \leq y \leq 1.0$. Define a maximum absolute value z_{max} (e.g. 4) such that the sequence f is considered as bounded if :

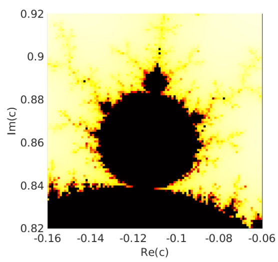
$$\lim_{n \to \infty} |z_{n+1}| < z_{max}$$

Now iterate the sequence f on the defined space, starting from $z_0 = (0, 0)$. For each iteration, assign a different value k (for example the number of the iteration itself) to the set of points where f is bounded, then plot k as a function of x and y. Find an appropriate number of iterations in order to converge to the Mandelbrot set.

The script `Mandelbrot_set.m` finds and plots the Mandelbrot set in the complex plane.
In the iteration process, it is useful to decompose complex numbers in real and imaginary parts, and then apply f, so that $Re(z_{n+1}) = (Re(z_n))^2 - (Im(z_n))^2 + x$ and $Im(z_{n+1}) = 2Re(z_n)Im(z_n) + y$.
In order to choose the maximum number of iterations, let's consider three cases corresponding to 64, 128 and 256 iterations, and zoom around a small region ($-0.16 \leq x \leq -0.06$ and $0.82 \leq y \leq 0.92$). The next figures show these three cases from left (64 iterations) to right (256 iterations).



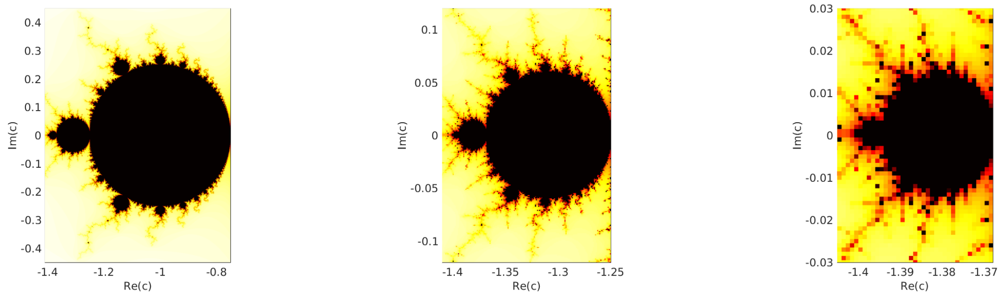




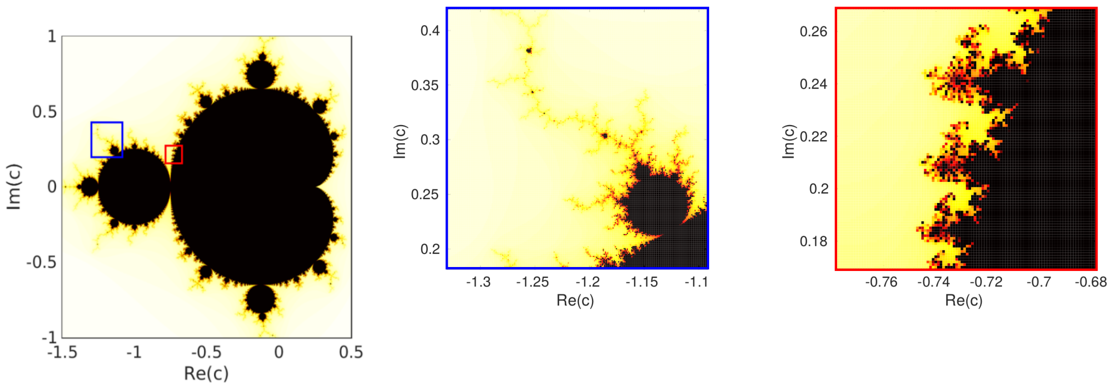
Looking at the base of the circle, we can see that the solution is most accurate in the 256 iterations case. As expected, the highest number of iterations corresponds to the smallest final set of points. However, the fine structures, also called *tendrils*, are more visible in the 64 iterations case (because of the bigger color contrast). 128 iterations are a good compromise between the resolution of the structures, and the visibility of the set.

(b) Discuss the self-similarity of the boundary of the Mandelbrot set. You can refine the resolution of your box, in order to see finer structures.

Zooming in certain regions of the boundary, one can notice the repetition of very similar structures. This is well visible looking, for example, at the approximately circular structures on the negative part of real axis (notice the different scales of the figures).



Nevertheless, the self-similarity is not perfect, as some structures appear distorted or slightly modified at smaller scales at other locations, as we can see from the figures below, which show a zoom from different regions of the set.



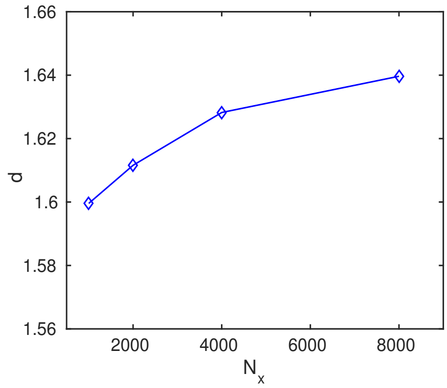
Therefore similarity dimension cannot be used to estimate the dimension of the boundary.

(c) Estimate the fractal dimension of the boundary of the Mandelbrot set with the box counting method. Use the Matlab function `contour`, to trace a contour at a specific value of k. For sufficiently high k, the points found by the `contour` command approximate well the boundary of the Mandelbrot set. In order to calculate the fractal dimension, evaluate the number of points found by `contour` as function of the number of the (x, y) grid points.

The second part of the Matlab script calculates the contour relative to the highest value of k used in the construction of the Mandelbrot set. Then, the fractal dimension is calculated as :

$$d = \frac{\ln\left[N(\epsilon)\right]}{\ln(1/\epsilon)}$$

where $N(\epsilon)$ is the number of points in the contour object, and ϵ is the measure of the side of the grid boxes. The next figure shows this calculation for grids with different resolutions.



The results converge to values ≈ 1.64 for a fine grid (8000×8000). A small correction is applied in the Matlab script, in order to exclude the points found by `contour` function which are out of the box.

4 Lorenz Attractor Dimension

In this exercise, we evaluate the correlation dimension of the Lorenz attractor. The correlation dimension is a measure of the fractal structure of an attractor. In particular, consider the set of points $\{\vec{x}_i, i = 1, 2, ...N\}$ on an attractor, obtained from a sequence with a fixed time increment $\tau : \vec{x}_i \equiv \vec{x}(t + i\tau)$. In *Grassberger and Procaccia (1983)*, the correlation integral is defined as

$$C(l) = \lim_{N\to\infty} \frac{1}{N^2} \times \{number\ of\ pairs\ (i,j) : |\vec{x}_i - \vec{x}_j| < l\}\,, \tag{1}$$

where $l > 0$. One finds that, for small values of l,

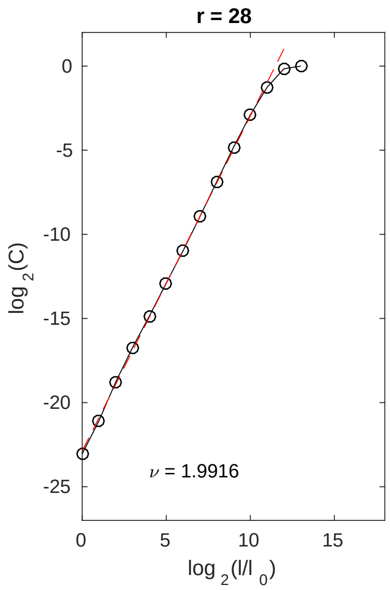
$$C(l) \sim l^{\nu}\,, \tag{2}$$

where the exponent ν is the correlation dimension.

(a) On Matlab, write a program that calculates the correlation dimension of the Lorenz attractor. First integrate the Lorenz system with parameters $\sigma = 10$ and $b = 8/3$. As a first step, choose $r = 28$, a value that leads chaotic evolution. As in *Grassberger and Procaccia (1983)*, use $N = 15000$ points in the integration of the Lorenz system. Count all the pairs of points (i, j) that satisfy the condition $|\vec{x}_i - \vec{x}_j| < l$, for several values of l, with l in the interval $[l_{\text{min}}, l_{\text{max}}]$, with l_{max} such that $C(l_{\text{max}}) = 1$, i.e. all pairs of points have a relative distance lower than l_{max}, and l_{min} given by the minimum distance between two points.

The file `lor_attr_dim.m` contains a script to compute the correlation dimension of the Lorenz attractor. After having integrated the Lorenz system over the established number of time steps, we calculate the maximum and the minimum distances among every couple of points, and choose $l_{\max} > max(|\vec{x}_i - \vec{x}_j|)$. A series of values l_j is chosen such that

$l_j = l_{\max}/2^j$, with $j = 1, 2, ...j_{\max}$. We call l_0 the minimum l value such that $C(l) > 0$. We count then the number of pairs in the trajectory which have a relative distance shorter than each value l_j, and calculate $C(l)$. The figure below shows the correlation integral C in function of l/l_0. Notice that with the choices that we made, the number of l values are the same as the ones used in *Grassberger and Procaccia (1983)*.



We use the Matlab function `polyfit` or `cftool` to fit $\log_2(C)$ as function of $\log_2(l/l_0)$. The resulting interpolating curve is superimposed to the calculated points, showing good agreement. The obtained correlation dimension is slightly lower than the value of 2.05 claimed in *Grassberger and Procaccia (1983)*.

(b) Study how the correlation dimension varies with the r parameter.

In a non-chaotic regime, using for example $r = 10$, we obtain a correlation dimension $\nu \sim 0$. In fact, all trajectories converge to a single point. Increasing r we observe a progressive increase in the correlation dimension. The correlation dimension rises quickly after the onset of chaos ($r \sim 21$), reaches the value ~ 2.05 in chaotic state, and saturates for higher r.