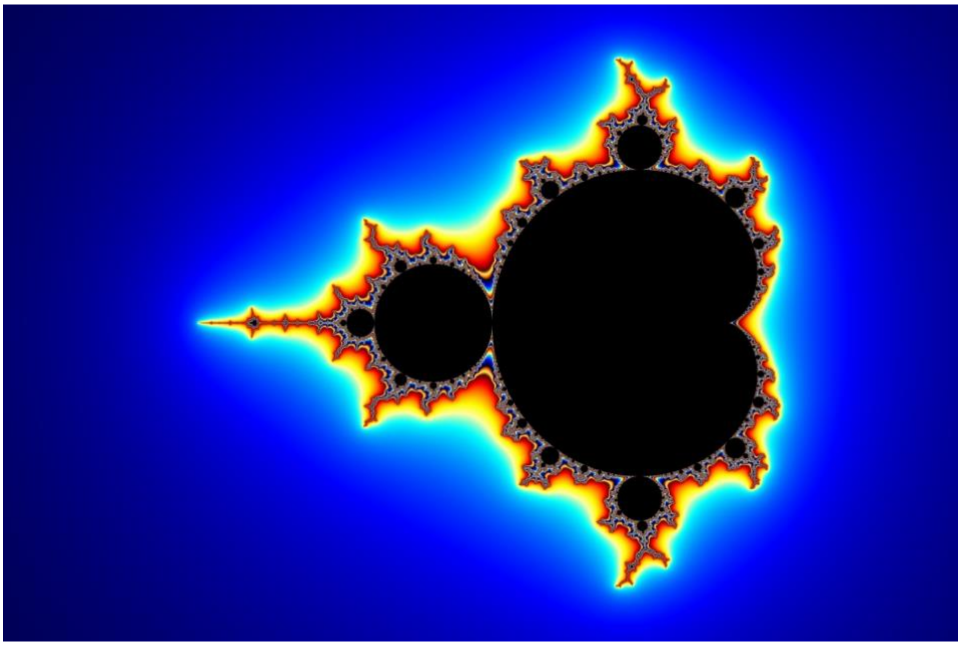
Manchester Metropolitan University

The Royal Institution
Science Lives Here

An Introduction to Fractals with python™



Dr Stephen Lynch NTF FIMA SFHEA

https://www.mmu.ac.uk/computing-and-maths/staff/profile/dr-stephen-lynch

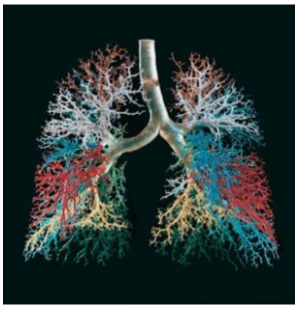
Definition 1. A *fractal* is an image repeated on an ever-reduced scale.

Definition 2. A *fractal* is an object with non-integer dimension.

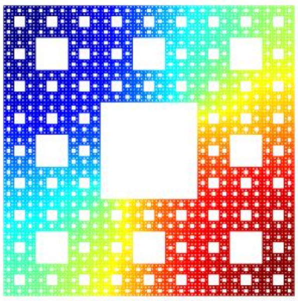
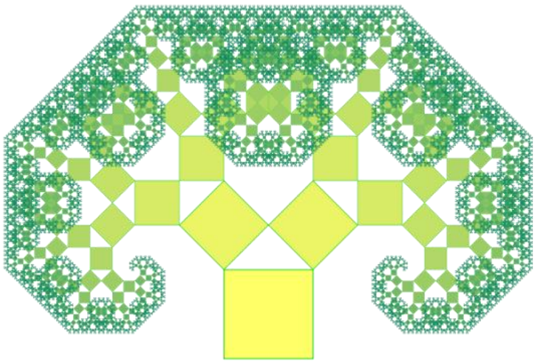
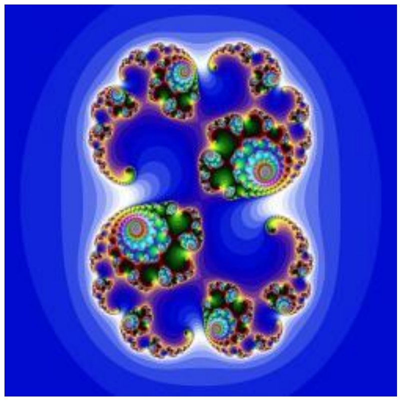
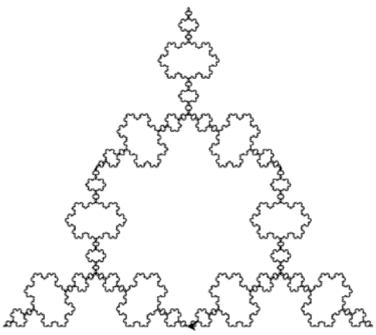
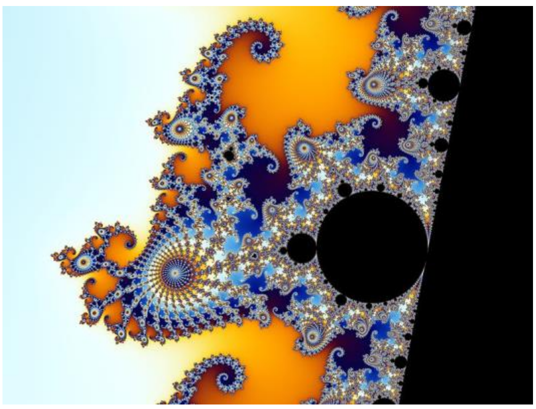
Fractals in Nature








Mathematical Fractals


Example 1. The *Cantor set* (1870).

Start with a unit line segment and at each stage remove the middle third segment and replace one segment with two segments each of length one third of the previous segment.

```python
# Program 1: Cantor Fractal.
# Run Module and type >>> cantor(-400, 200 , 400) in Python Shell.
from turtle import *          # Load functions from turtle module.
speed(0)                      # Fastest speeed of turtle.
def cantor(x, y, length):
    if length >= 5:           # Stop when length < 5.
        penup()
        pensize(3)
        pencolor('blue')
        setpos(x , y)
        pendown()
        fd(length)
        y -= 60               # Height difference.
        cantor(x , y, length / 3)
        cantor(x + 2 * length / 3, y, length / 3)
        penup()
        setpos(x , y + 60)
```

Figure 1: The Cantor set to stage 4.

To understand the program, set speed(1) and plot the Cantor set to stage 2. Mimic by-hand what the program is doing.

Problem 1: Edit Program 1 to plot a variation of the Cantor set, where two segments (each one-fifth the length of the previous segment) are removed at each stage.

Example 2. The *Koch curve* (1904).

Start with a unit line segment, remove the middle third segment and replace one segment with four segments each of length one third the previous segment, as illustrated below.

```python
# Program 2: Koch Curve Fractal
# Run Module and type >>> Koch_Curve(300,4) in Python Shell.
from turtle import *
speed(0)                           # Fastest speed.
def KochCurve(length,level):
    if level==0:
        fd(length)
        return
    KochCurve(length/3,level-1)
    lt(60)                         # Left turn 60 degrees.
    KochCurve(length/3,level-1)
    rt(120)                        # Right turn 120 degrees.
    KochCurve(length/3,level-1)
    lt(60)
    KochCurve(length/3,level-1)
```

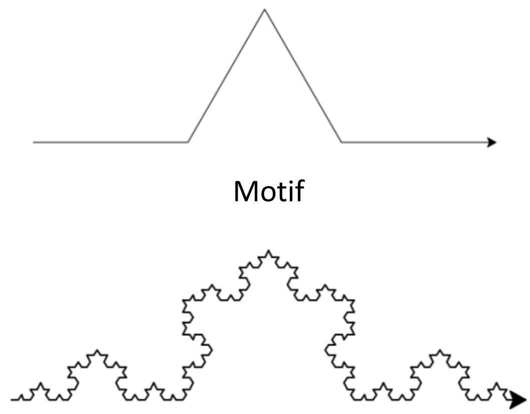


Figure 2: The Koch curve.

Problem 2: Edit Program 2 to plot a Koch square curve, where one segment is replaced with five segments each one-third the length of the segment before.

Example 3. The *Sierpiński triangle* (1915).

Start with a solid equilateral triangle and at each stage remove the middle-inverted triangle. Complete the construction up to stage 4.

```python
# Program 3: Sierpinski triangle - save file as Sierpinski.py.
# Run the Module (or type F5).
# In the Python shell, type >>> Sierpinski(400 , 5)
from turtle import *
def Sierpinski(length, level):
    speed(0)      # Fastest speed.
    if level==0:
        return
    begin_fill()  # Fill shape.
    color("red")

    for i in range(3):
        Sierpinski(length/2,level-1)
        fd(length)
        lt(120)   # Left turn 120 degrees.
    end_fill()
```

Motif

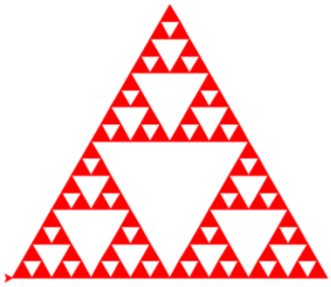


Figure 3: The Sierpiński triangle.

Problem 3: Edit Program 3 to plot a Sierpiński square, where the middle square is removed at each stage. See one face of the Menger Sponge in Figure 4b.

Example 4. Use train tickets to construct stage 1 of the *Menger Sponge (see Figures 4a and 4b).*

Problem 4: Given that you need 6 tickets to make one small block:

How many tickets do you need for the stage 1 construction? (See Figure 4a).

How many tickets would be required for the stage 2 construction?

How many tickets would be needed for the stage 4 construction? (See Figure 4b).

Figure 4a: Menger sponge, stage 1.

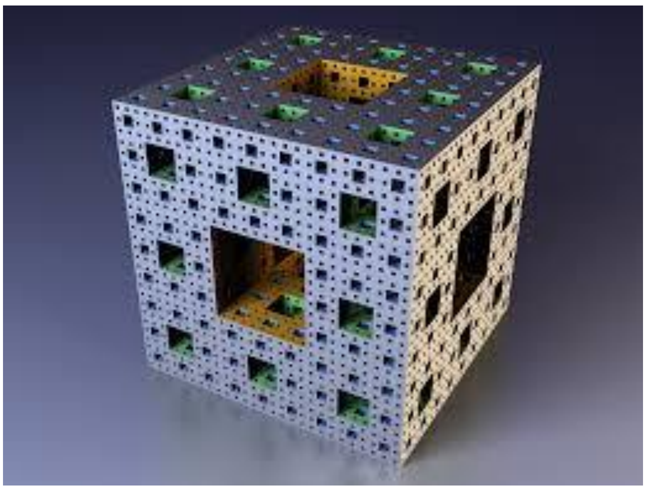


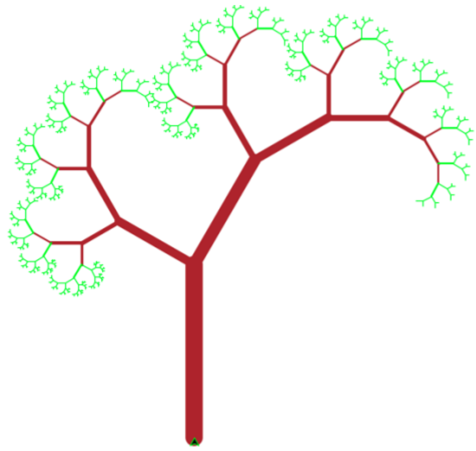
Figure 4b: Menger sponge, stage 4.

Example 5. A Python program for plotting a fractal tree.

```
# Program 3: A program to plot a color fractal tree.
# Run Module and type >>> FractalTreeColor(200,10) in Python Shell.
from turtle import *
setheading(90)              # The turtle points straight up.
penup()                     # Lift the pen.
setpos(0,-250)              # Set initial point.
pendown()                   # Pen down.

def FractalTreeColor(length, level):
    """
    Draws a fractal tree
    """
    pensize(length/10)      # Thickness of lines.
    if length < 20:
        pencolor("green")
    else:
        pencolor("brown")

    speed(0)
    if level > 0:
        fd(length)          # Forward.
        rt(30)              # Right turn 30 degrees.
        FractalTreeColor(length*0.7, level-1)
        lt(90)              # Left turn 90 degrees.
        FractalTreeColor(length*0.5, level-1)
        rt(60)              # Right turn 60 degrees.
        penup()
        bk(length)          # Backward.
        pendown()
```



Problem 5: Edit Program 4 to plot a trifurcating tree with three branches at each level.

The formula for working out the fractal dimension D_f, say, is

$$D_f = -\frac{ln(\text{Number of segments})}{ln(\text{Length scale})}.$$

The fractal dimension of the Cantor set is $D_f = \frac{ln2}{ln3} \sim 0.6309$.

The fractal dimension of the Koch curve is $D_f = \frac{ln4}{ln3} \sim 1.2619$.

The fractal dimension of the Sierpiński triangle is $D_f = \frac{ln3}{ln2} \sim 1.5850$.

The fractal dimension of the Menger sponge is $D_f = \frac{ln20}{ln3} \sim 2.7268$.

Problem 6: Work out the fractal dimensions of the other fractals.

Further Information:

URL to download IDLE Python (which is free):

https://www.python.org/downloads/

URL for an introduction to the Python Turtle module:

https://docs.python.org/3/library/turtle.html

Python for A-Level Maths, undergraduate Maths and employability:

https://www.mathscareers.org.uk/python-for-a-level-maths-undergraduate-maths-and-employability/

Python for A-Level Maths and Beyond:

http://www.doc.mmu.ac.uk/STAFF/S.Lynch/Python_for_A_Level_Mathematics_and_Beyond.html

The Mandelbrot Set

URL for the Mandelbrot Set (deep zoom):

https://www.youtube.com/watch?v=LhOSM6uCWxk

URL for the Mandelbrot Set song:

https://www.youtube.com/watch?v=aIj30SOoIDM

IMA Workshops and My Python Book

1. One-day interactive workshops: Python for A-Level Mathematics and Beyond!

   https://ima.org.uk/events/

2. Stephen Lynch, *Dynamical Systems with Applications using Python*, Springer International Publishing, Switzerland, 2018.

Springer International Publishing:

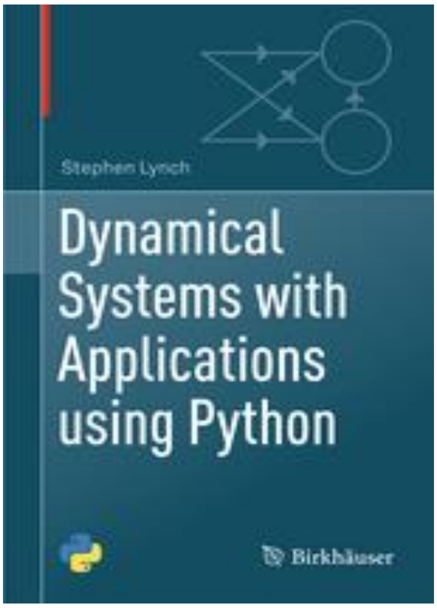
https://www.springer.com/us/book/9783319781440

Jupyter Notebook:

http://www.doc.mmu.ac.uk/STAFF/S.Lynch/DSAP_Jupyter_Notebook.html

GitHub:

https://github.com/DrStephenLynch/Tekbac



NEW BOOK: **Python for Scientific Computing and TensorFlow for Artificial Intelligence**

This book will be published by CRC Press in 2023.