NCEA Level 2 Mathematics
17. Number Sequences and Fractals

A number sequence is an arrangement of numbers in which each successive number follows the last according to a uniform rule. More precisely, a number sequence is a correspondence between the counting numbers and a set of numbers: $(a_n) = a_1, a_2, a_3, \ldots$. Our goal in this section is twofold:

- First, we will try to get through all of the boring computations first;
- Second, we will refresh ourselves in the oasis of geometry after a walk in the desert of analysis.

Arithmetic Sequences

Arithmetic sequences are the simplest kind of interesting sequence.

Definition. An arithmetic sequence is a number sequence in which each successive term may be found by adding the same number; formally, a sequence is arithmetic if $a_{n+1} = a_n + k$ for every $n > 1$ and for some constant k.

Example. In the following sequence, $a_1 = 2$ and $a_{n+1} = a_n + 3$.

$$2, 5, 8, 14, 17, \cdots$$

For arithmetic sequences, if we know the initial value and the constant difference then we can find each number in the sequence easily.

Theorem. *If $a_1, a_2, \cdots$ is an arithmetic sequence with constant difference k, then $a_n = a_1 + (n-1)k$. (This is called the* general term *of the sequence.)*

Proof. $a_n = a_{(n-1)} + k = a_{(n-2)} + 2k = \cdots = a_{(n-(n-1))} + (n-1)k$. □

Suppose we want to find the *sum* of the first n values of some sequence $a_1, a_2, \ldots$. For a simple example, we turn to a problem from last week.

Example. We will find the sum of the first n counting numbers. Behold:

$$\begin{array}{ccccccccccccccccc} 1 & + & 2 & + & 3 & + & 4 & + & \cdots & + & (n-3) & + & (n-2) & + & (n-1) & + & n \\ n & + & (n-1) & + & (n-2) & + & (n-3) & + & \cdots & + & 4 & + & 3 & + & 2 & + & 1 \end{array}$$

Adding the two rows together and dividing by two, we obtain the result that

$$1 + \cdots + n = \frac{n(n+1)}{2}.$$

(This is called the nth partial sum of the series $1 + 2 + 3 + \cdots + n$.)

We now turn our attention to finding the nth partial sum of the series $a_1 + a_2 + a_3 + \cdots + a_n$. By the theorem above, we can rewrite this as

$$\begin{aligned} a_1 + a_2 + a_3 + \cdots + a_n &= [a_1 + (1-1)k] + [a_1 + (2-1)k] + [a_1 + (3-1)k] + \cdots + [a_1 + (n-1)k] \\ &= [a_1 + \cdots + a_1] + [0k + 1k + 2k + \cdots + (n-1)k] \\ &= na_1 + k[0 + 1 + 2 + \cdots + (n-1)] \\ &= na_1 + k\frac{n(n-1)}{2}. \end{aligned}$$

Hence, we have proved the following

Theorem. *The nth partial sum of the series $a_1 + a_2 + a_3 + \cdots + a_n$ is given by*

$$na_1 + k\frac{n(n-1)}{2}$$

(but you should memorise the idea of the proof, not the formula.)

Geometric Sequences

The next simplest kind of sequence after arithmetic sequences (where you add a constant term) is a geometric sequence (where you multiply by a constant term).

Definition. A geometric sequence is a number sequence in which each successive term may be found by multiplying by the same number; formally, a sequence is geometric if $a_{n+1} = ka_n$ for every $n > 1$ and for some constant k.

Example.

1. $a_1 = 1$, $a_n = 2a_{n-1}$: $1, 2, 4, 6, 8, ...$ (the binary sequence).
2. $a_1 = 100$, $a_n = \frac{1}{10}a_{n-1}$: $100, 10, 1, 0.1, 0.01, ...$
3. $a_1 = 1$, $a_n = -1a_{n-1}$: $1, -1, 1, -1, ...$

Theorem. *If $a_1, a_2, \cdots$ is an geometric sequence with constant ratio k, then:*

1. *the general term of the sequence is $a_n = a_1 k^{n-1}$.*
2. *the nth partial sum of the sequence is $s_n = a_1 \frac{1-k^n}{1-k}$.*

Proof.

1. Exercise.
2. This proof uses a little trick:

$$\begin{aligned}
s_n &= a_1k^0 + a_1k^1 + a_1k^2 + \cdots + a_1k^{n-1} = a_1^n(1 + k + k^2 + \cdots + k^{n-1}) \\
(1-k)s_n &= a_1(1-k)(1 + k + k^2 + \cdots + k^{n-1}) \\
&= a_1[(1 + k + k^2 + \cdots + k^{n-1}) - k(1 + k + k^2 + \cdots + k^{n-1})] \\
&= a_1[(1 + k + k^2 + \cdots + k^{n-1}) - (k + k^2 + k^3 + \cdots + k^n)] \\
&= a_1[1 - k^n] \\
s_n &= a_1 \frac{1-k^n}{1-k}.
\end{aligned}$$

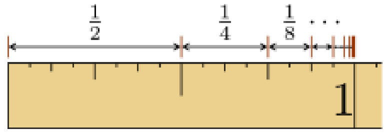
$\square$

Geometric sequences are slightly more interesting than arithmetic sequences; if we add up all the terms of an arithmetic sequence, the resulting partial sums always grow towards $\pm\infty$. On the other hand, it is possible for the sum of all the terms of a geometric sequence to tend to some finite value. One case in which this happens is the following example.

Example. Consider the geometric sequence given by $a_n = \frac{1}{2}^n$:

$$1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \ldots$$

The sum $1 + 1/2 + 1/4 + \cdots$ converges to 1, as shown graphically here.



On the other hand, it is very easy for series to diverge (not tend towards any finite value): just take $k > 1$. More interestingly, we can consider the following sequence:

Example. Consider the geometric sequence given by $a_n = (-1)^n$:

$$1, -1, +1, -1, +1, \ldots$$

The sum $1 - 1 + 1 - 1 + \cdots$ does not converge as it jumps between $+1$ and -1 infinitely often.

Let us try to work out under which conditions the geometric series *does* converge. We want to look at the behaviour of the quantity $a_1 \frac{1-k^n}{1-k}$ as n grows. We make the following observations:

1. If $k = 1$, then the quantity is undefined: but then, the sequence looks like $a_1 + a_1 + \cdots$, which grows arbitrarily large and so the sum diverges.

2. If $k > 1$, then the sum also grows arbitrarily large and the series diverges.

3. If $k = -1$, then the sum diverges but now oscillates around zero instead of growing to infinity.

4. If $k < -1$, then the sum grows arbitrarily large and oscillates between being positive and negative, so diverges.

5. If $-1 < k < 1$, then k^n gets smaller as n increases — so tends to zero, and the sum of the series as $n \to \infty$ is given by $\lim a_n = a_1 \frac{1}{1-k}$.

In summary, we have seen that:

- The general term of an arithmetic sequence is $a_n = a_1 + (n-1)k$.
- The nth partial sum of an arithmetic series is $s_n = na_1 + k\frac{n(n-1)}{2}$.
- The general term of a geometric sequence is $a_n = a_1 k^{n-1}$.
- The nth partial sum of a geometric series is $s_n = a_1 \frac{1-k^n}{1-k}$.
- If $-1 < k < 1$ is the ratio of a geometric series, then the series converges to $\lim a_n = a_1 \frac{1}{1-k}$

These five facts are the important things to remember.

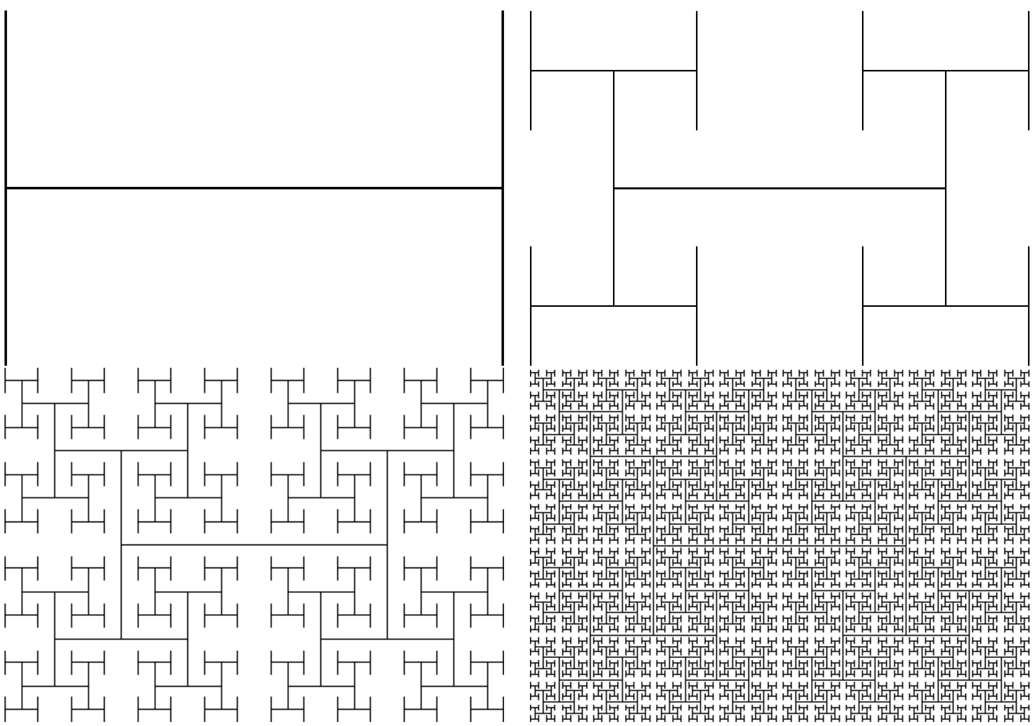
Fractals

We now move from sequences of numbers to sequences of geometric objects, with an very brief overview of fractals. Informally, a fractal is a geometric figure with the property of self-similarity: if you zoom in, it looks 'the same'. Many examples of fractal geometry can be found in nature; the traditional example is a coastline:

As we zoom in, the coastline reveals new 'wiggles' that we couldn't see at a larger scale — but it always looks wiggly in the same way, no matter how far you zoom in.

Construction 1 (H-fractal)**.** Begin at step one with a line segment of length 1. Then at each step n, add a line segment of length $\left(\frac{1}{\sqrt{2}}\right)^{n-1}$ to the endpoints of each line segment created at step $n-1$. (The lengths are purely aesthetic.)

The curve produced is the H-fractal; a Python script to draw it (`hcurve.py`) is given in the appendices. Below we have the curve after the second, fourth, eighth, and twelfth steps.



How many segments are present at the nth step? At the first step we have one, at the second three, at the third seven, at the fourth fifteen, and so on: the number seems to be $2^n - 1$ segments. How do we prove this? Well, suppose we have k endpoints at the $n-1$th step. Then at the nth step, we add k lines and hence $2k$ endpoints — so the number of endpoints doubles each time. Initially we have two endpoints, so the number of endpoints added at each step follows a geometric sequence with initial term 2 and ratio 2: at step n, we add 2^n endpoints. The sum of all these endpoints is simply $2\frac{1-2^n}{1-2} = 2(2^n - 1)$ (using our formula for the partial sum of a geometric series); and every line has two endpoints, so we must divide by two.

An interesting property of this curve is that, as well as having 'infinite length', it is space-filling — when we complete the construction at infinity, the curve will cover every point in the rectangle that it is bounded by.*

Other methods of producing fractals involve removing portions of a figure rather than adding portions.

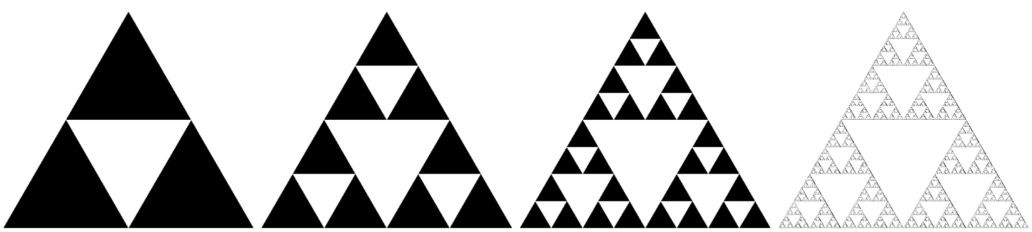
Construction 2 (Cantor set)**.** At step 1, begin with a unit segment; then at step n, remove the middle third of each of the segments remaining from the previous step.

This construction, pictured below, was first discovered by Henry John Stephen Smith in 1874 and entered mainstream mathematical knowledge in 1883 due to Georg Cantor (the father of set theory). The Cantor set itself, produced by continuing the construction to infinity, still has infinitely many points but has (in a precise sense) zero length. A Python script to draw it (`cantor.py`) is given in the appendices.

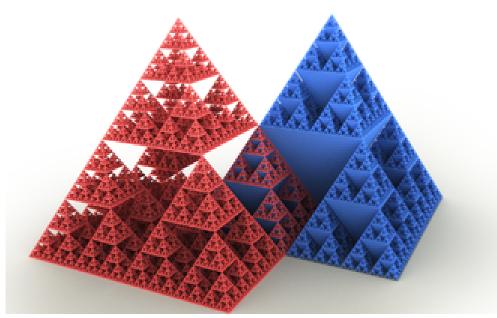
Our initial length is 1; at each step we remove precisely 1/3 of the remaining length, or equivalently we keep 2/3 of the remaining length; so the length of the set at step n is given by the geometric sequence with general term $\left(\frac{2}{3}\right)^{n-1}$. In particular, since 2/3 is less than 1, if we take $n \to \infty$ the sequence goes to zero.

Construction 3 (Sierpinski triangle)**.** At step 1, we start with a filled equilateral triangle. At the nth step, split each filled triangle into four equilateral triangles and remove the central triangle.

If this construction is carried on forever, the result is the Sierpinski triangle. This fractal was first described by a Polish mathematician, Waclaw Sierpinski, in 1916, and is a generalisation of the Cantor construction to two dimensions. Similarly to the Cantor set, the measure of the Sierpinski triangle (in this case the area) is zero but it still contains infinitely many points! A Python script to draw it (`sierp.py`) is given in the appendices; below, we have the triangle after two, three, four, and nine iterations.



One possible generalisation to three dimensions is the Sierpinski tetrahedron, produced by removing pyramids from a pyramid; two are pictured below.†



*More precisely, we can pick some number N such that after the Nth step, the curve comes within any distance δ of any point within the rectangle that we want. Obviously if δ is small then we need N to be (very) large, but the point is that it's theoretically possible!

†CC BY-SA 3.0, `https://commons.wikimedia.org/w/index.php?curid=647064`

Questions

Sequences and Series

1. The following are snippets of sequences that are either arithmetic or geometric. Give the general term of each. (In each case, $a_1 = 1$.)
   (a) $\ldots, \frac{1}{27}, \frac{1}{81}, \frac{1}{243}, \ldots$
   (b) $\ldots, 5, 7, 9, \ldots$
   (c) $\ldots, -10, 100, -1000, \ldots$
   (d) $\ldots, 0.02116, 0.0097336, 0.004477456, \ldots$

2. Prove that the general term of a geometric sequence (a_n) with constant ratio k is $a_1 k^{n-1}$.

3. Compare the formulae that give the general term of an arithmetic sequence and a geometric sequence.

4. The Fibonacci sequence is the sequence defined by $f_1 = 1$, $f_2 = 1$, and $f_n = f_{n-2} + f_{n-1}$. The first few numbers in the sequence are $1, 1, 2, 3, 5, 8, 13, 21, \cdots$.
   (a) The Fibonacci sequence was studied by Leonardo of Pisa (Fibonacci) in the middle ages, in his book *Liber Abaci* ("Book of Calculation"). In this book appeared the following problem:

   A pair of rabbits one month old are too young to produce more rabbits, but suppose that in their second month and every month thereafter they produce a new pair. If each new pair of rabbits does the same, and none of the rabbits die, how many pairs of rabbits will there be at the beginning of each month?

   Show that the solution is given by the Fibonacci sequence.
   (b) Prove that the Fibonacci sequence is neither an arithmetic nor a geometric sequence.
   (c) It turns out that, despite not being a geometric sequence, the ratios of adjacent Fibonacci numbers tend to a constant value. Taking this to be true without proof, we will calculate what this ratio is.
      i. Justify why, if n is very large, $\frac{f_{n+1}}{f_n} \approx \frac{f_n}{f_{n-1}}$.
      ii. Show that, if $x = \frac{f_n+1}{f_n}$, then $x \approx 1 + \frac{1}{x}$.
      iii. Show that $x = \frac{1}{2} + \frac{\sqrt{5}}{2}$. (This value is usually called φ, the golden ratio.)
      iv. Experimentally verify that this is the approximate ratio between adjacent values for the first few values of the Fibonacci series.
      v. Explain why we have *not* proved that this is the eventual ratio between adjacent values of the Fibonacci sequence.

5. According to legend, the game of chess was invented by an ancient Indian minister for his ruler; the ruler was impressed, and asked the minister what reward he wanted, and the minister requested that the ruler take a chessboard and give him one grain of wheat on the first square, two grains on the second, four on the third, eight on the fourth, and so on. The ruler laughed it off as a meager prize for such a brilliant invention.
   (a) How much wheat did the minister ask for?
   (b) The weight of a grain of wheat is around 65 mg. Compare the weight of the wheat on the first half of the chessboard to the weight on the second half.

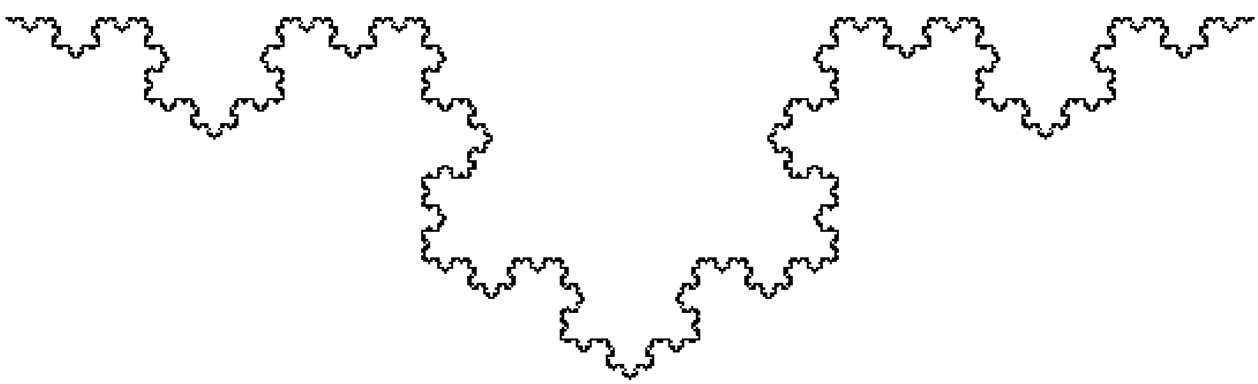
Fractals

6. What is the total length of the H-fractal after n steps?

7. Consider the figure given of the Cantor set, where each iteration is added on below the previous terms in order to form a fractal figure. Supposing that we continue this on for n steps (so the figure above is of step 6 of this process), and the height of each term is constant at 1 unit.
   (a) What is the total area of the fractal after n steps?
   (b) What happens to the area as n tends towards infinity? Does the area converge to some finite number?
8. Show that the area of the Sierpinski triangle is zero, by calculating the total area of triangles that are removed.
9. The Sierpinski triangle is very amenable to generalisation.
   (a) Give a construction for a fractal produced in a similar way to the Sierpinski triangle, but instead of subdividing a triangle into triangles subdivide a square into squares. (The resulting fractal is known as the Sierpinski carpet.) What is the area of this figure after n steps?
   (b) Generalise part (a) to three dimensions, by removing cubes from a cube. (The resulting cube is known as the Menger sponge.) Find the volume after n steps.
10. The Koch curve is produced by continuing following construction to infinity:

    Construction 4. At step 1, we have a unit segment. At the nth step, we replace the middle third of each segment from the previous step with the upright sides of an equilateral triangle, so the segment is replaced by four segments with total length $4/3$ of the segment length.

    (a) Below is the seventh iteration of the Koch curve (see `koch.py` in the appendices). Draw the first few iterations.



    (b) Show that the length of the curve as $n \to \infty$ becomes infinite.
    (c) This curve is an example of a **continuous** curve that has **no tangent line anywhere**. Justify both bolded claims.
    (d) Show that, if we start with an equilateral triangle and then add our equilateral triangles always on the outside, then the area enclosed by the final figure is precisely $8/5$ of the original triangle area. The resulting figure, the Koch snowflake, is therefore a finite area bounded by a curve of infinite length.

H-fractal: hcurve.py

```
import tkinter as tk
from math import sqrt

def hcurve(x, y, length, scale, horizontal, iterations, canvas):
  if(iterations == 0):
    return

  if horizontal == True:
    x0 = x - length/2
    x1 = x + length/2
    y0 = y1 = y
  else:
    y0 = y - length/2
    y1 = y + length/2
    x0 = x1 = x

  canvas.create_line(x0, y0, x1, y1)

  hcurve(x0, y0, length*scale, scale, not(horizontal), iterations - 1, canvas)
  hcurve(x1, y1, length*scale, scale, not(horizontal), iterations - 1, canvas)

w = tk.Canvas(width=500,height=500)
hcurve(250, 250, 220, 1/sqrt(2), True, 12, w)
w.grid()
w.mainloop()
```

Cantor set: cantor.py

```
import tkinter as tk
from math import sqrt

def comb(x, y, length, height, iterations, canvas):
  print(length)
  if(iterations == 0):
    return

  x0 = x - length/2
  x1 = x + length/2 + 1
  y0 = y - height/2
  y1 = y + height/2 + 1

  w.create_rectangle(x0, y0, x1, y1, fill='black')

  comb(x - length/3, y + height + 10, length/3, height, iterations - 1, canvas)
  comb(x + length/3, y + height + 10, length/3, height, iterations - 1, canvas)


w = tk.Canvas(width=1400,height=500)
comb(700, 20, 1300, 20, 6, w)
w.grid()
w.mainloop()
```

Sierpinski triangle: sierp.py

```
import tkinter as tk
from math import sqrt

def sierp(x, y, length, iterations, canvas):
  if(iterations == 0):
    return

  xleft = x - length/2
  xright = x + length/2
  yleft = yright = y + length*sqrt(3)/2

  canvas.create_polygon(x, y, xleft, yleft, xright, yright, fill = 'black')
  canvas.create_polygon((x+xleft)/2, (y+yleft)/2,
                        x, yleft, (x+xright)/2, (y+yright)/2, fill = 'white')

  sierp((x+xleft)/2, (y+yleft)/2, length/2, iterations - 1, canvas)
  sierp((x+xright)/2, (y+yright)/2, length/2, iterations - 1, canvas)
  sierp(x,y, length/2, iterations - 1, canvas)

w = tk.Canvas(width=600,height=600)
sierp(300, 20, 550, 5, w)
w.grid()
w.mainloop()
```

Koch curve: koch.py

```
import tkinter as tk
from math import sqrt, atan2, cos, sin

def koch(coords, where, iterations):
  x0 = coords[where][0]
  y0 = coords[where][1]
  x1 = coords[where + 1][0]
  y1 = coords[where + 1][1]
  length = sqrt((x1 - x0)**2 + (y1 - y0)**2)
  r = length*1/((sqrt(3)))
  theta = atan2(y1 - y0, x1 - x0)
  phi = atan2(1, sqrt(3))
  xnew1 = x0 + length/3 * cos(theta)
  ynew1 = y0 + length/3 * sin(theta)
  xnew2 = x0 + r * cos(theta + phi)
  ynew2 = y0 + r * sin(theta + phi)
  xnew3 = x0 + 2*length/3 * cos(theta)
  ynew3 = y0 + 2*length/3 * sin(theta)

  coords.insert(where + 1, [xnew1, ynew1])
  coords.insert(where + 2, [xnew2, ynew2])
  coords.insert(where + 3, [xnew3, ynew3])

  if(iterations > 1):
    coords = koch(coords, where, iterations - 1)
    coords = koch(coords, where + 4**(iterations-1), iterations - 1)
    coords = koch(coords, where + 2*4**(iterations-1), iterations - 1)
    coords = koch(coords, where + 3*4**(iterations-1), iterations - 1)
  return coords

w = tk.Canvas(width=600,height=600)
coords = koch([[30, 300],[600 - 30, 300]], 0, 6)
for i in range(0, len(coords) - 1):
  w.create_line(coords[i][0], coords[i][1], coords[i + 1][0], coords[i + 1][1])
w.grid()
w.mainloop()
```