Project 20

Koch’s Fractal

For this project you will need familiarity with the commands:

\[
\begin{array}{c}
\text{plot} & \text{proc} & \text{display} \\
\text{for} & \text{from} & \text{to} & \text{do} & \text{end do} \\
\end{array}
\]

In addition, you will need to use the \textit{style} and \textit{numpoints} options of \texttt{plot} and the \textit{two argument} form of \texttt{arctan}.

Background

Helga von Koch described a continuous curve that has come to be called a \textit{Koch Snowflake} (1904). The enclosed area is called the \textit{Koch Island}. The method of generating the snowflake is simple to describe. Start with an equilateral triangle.

\begin{verbatim}
> r3 := sqrt(3)
> snow0 := [[0, r3/6], [1/2, r3\times2/3], [1, r3/6], [0, r3/6]]
> plot(snow0, scaling=constrained)
\end{verbatim}

Remove the middle third of each side and insert two sides of a smaller equilateral triangle as shown below.

\begin{verbatim}
> snow1 := [[0, r3/6], [1/6, r3/3], [1/2, r3/2], [1, r3/6], [0, r3/6],
>           [1/3, r3/6], [2/3, r3/3], [1, r3/6], [2/3, r3/3], [1/2, 0],
>           [1/3, r3/6], [0, r3/6]]
> plot(snow1, scaling=constrained)
\end{verbatim}

There are two observations to make:

1. Since the length of each side has increased by one third, the total perimeter has increased by one third.

2. The island is still contained in a rectangle of size \(1 \times 2\sqrt{3}/3\).

\textbf{Question 1} \textit{Calculate:}

1. The perimeters of the original and first snowflakes.
2. The areas of the original and first islands.

3. The distance from a new vertex in the first snowflake to the edge of the original snowflake.

The second iteration is shown below. Try it—if you’re up to the typing.

```
snow2 := [ [0, r3/6], [1/18, r3*2/9], [0, r3*5/18], [1/9, r3*5/18],
            [1/6, r3/3], [1/9, r3*7/18], [0, r3*7/18], [1/18, r3*4/9],
            [0, r3/2], [1/9, r3/2], [1/6, r3*5/9], [2/9, r3/2],
            [1/3, r3/2], [7/18, r3*5/9], [1/3, r3*11/18], [4/9, r3*11/18],
            [1/2, r3*2/3], [5/9, r3*11/18], [2/3, r3*11/18], [11/18, r3*5/9],
            [2/3, r3/2], [7/9, r3/2], [5/6, r3*5/9], [8/9, r3/2],
            [1, r3/2],
            [17/18, r3*4/9], [1, r3*7/18], [8/9, r3*7/18], [5/6, r3/3],
            [8/9, r3*5/18], [1, r3*5/18], [17/18, r3*2/9], [1, r3/6],
            [8/9, r3/6], [5/6, r3/9], [7/9, r3/6], [2/3, r3/6],
            [11/18, r3/9], [2/3, r3/18], [5/9, r3/18], [1/2, 0],
            [4/9, r3/18], [1/3, r3/18], [7/18, r3/9], [1/3, r3/6],
            [2/9, r3/6], [1/6, r3/9], [1/9, r3/6], [0, r3/6] ]
```

> plot(snow2, scaling=constrained)

For those who aren’t up to the typing, a printout of a Maple session with the routines for generating and animating Koch snowflakes appears at the end of this project.

**Question 2** Calculate:

1. The perimeter of the second snowflake.

2. The area of the second island.

3. The distance from a new vertex in the second snowflake to the edge of the first snowflake.

For each successive iteration, repeat the procedure of inserting smaller and smaller triangles as seen in the figure below.

**Question 3** How many line segments are in the $n$th iteration’s curve.
The Koch Snowflake is the limit curve of this procedure. The snowflake has the remarkable property that at no point does the curve possess a tangent line. To see the curve develop, run the animation using the Maple procedures given in the listing at the end of this project.

**Question 4** At which iteration can you no longer see a difference in the image displayed on the screen? On a laser printout?

**Project Report**

1. Develop a formula $L_n$ that gives the perimeter of $S_n$, the $n$th snowflake. Find the limit of the sequence $L_n$, that is find
   \[ \lim_{n \to \infty} L_n. \]
2. Develop a formula $A_n$ that gives the area enclosed by $S_n$. Find
   \[ \lim_{n \to \infty} A_n. \]
3. Develop a formula $D_n$ that gives the distance from a new vertex of $S_n$ to the edge of $S_{n-1}$. Find
   \[ \lim_{n \to \infty} D_n. \]
4. Discuss the apparent paradox.

**Extension**

There are many unusual curves that have properties similar to the snowflake. Find a reference to Hilbert’s Peano curve. This is Hilbert’s example of a space-filling curve. Modify the Maple procedures for Koch’s Snowflake so as to produce Hilbert’s curve. Describe the behavior of this curve.

**Report Requirements**

A minimal project report will include:

- Answers to Questions 1 through 4.
- Full responses to Problems 1 and 2 with supporting arguments.
Maple Code to Animate the Koch Snowflake

++++++++++++++++++++++
## Animating the Koch Snowflake ##
+++++++++++++++++++++++

# setup plotting
> plots[setoptions](axes=none, scaling=constrained)

+++++++++++++++++++++++
## A proc to "connect the dots" to plot a path
## (mainly for Maple V Release 2)
+++++++++++++++++++++++

> graph := proc() local pts;
    pts := [args];
    plot(pts, style= LINE, numpoints=nargs)
end:

+++++++++++++++++++++++
## Rotate point A about the origin through angle p
+++++++++++++++++++++++

> rotate := proc(a, p);
    [p[1]*cos(a)-p[2]*sin(a), p[1]*sin(a)+p[2]*cos(a)];
end:

+++++++++++++++++++++++
## Insert three points of a Koch triangle between pts A & B
+++++++++++++++++++++++

> Koch := proc(A, B)
    local pA, pB, pC, dx, dy, len, alpha, delta;
    dx := B[1]-A[1];
    dy := B[2]-A[2];
    len := sqrt(dx^2+dy^2)/3;
    alpha := arctan(dy, dx);  # Note the comma
    delta := rotate(alpha, [len/2, sqrt(3)*len/2]);
    pA, pB, pC;
end proc:

+++++++++++++++++++++++
## Take a path and iterate with inserted Koch triangles

```wolfram
> path := proc() local pa, pth, i;
   pa := args;
   pth := pa[1];
   for i from 2 to nops([pa])
       do
           pth := (pth, Koch(pa[i-1], pa[i]), pa[i]);
       end do;
   pth:
end proc:
```

## Start with an equilateral triangle#

```wolfram
> s := sqrt(3)/6:
p := [0, s], [1/2, 4*s], [1, s], [0, s]:
snow[0] := graph(p):
```

## Now begin the iterations - go up to 6
## (make sure to use the colons after the last two lines)

```wolfram
> N := 6:
for i from 1 to N
   do
       p := path(p);
       snow[i] := graph(p);
   end do:
snow := convert(snow, list):
```

## and animate

```wolfram
> plots[display](snow, insequence = true)
```

```wolfram
#################################################
```