Some fractals with Maple

(Article begins on next page)

Some Fractals with Maple

Miriam Ciavarella

Università degli Studi di Torino

Italy

miriam.ciavarella@unito.it

Marina Marchisio

Università degli Studi di Torino

Italy

marina.marchisio@unito.it

Introduction

This paper contains some procedures concerning the fractals.
In particular we describe the procedures in order to draw the triadic Cantor dust, the Koch curve, the Snowflake and the procedures in order to calculate the area and the perimeter of the Snowflake curve.

The idea of this procedures was born during a tutorial of Maple that we made for students in Mathematics of the first year of the University of Turin. They were interested in learning how to write procedures with Maple but also in working with objects different from the ones of the classical geometry. Fractals appeared to us as the right geometric objects with surprising properties.

We consider the fractals just as subsets of two-dimensional Euclidean space.

Triadic Cantor Dust

In the nineteenth century mathematician Georg Cantor became fascinated by the infinite number of points on a line segment and studied this new set.

To generate the triadic Cantor dust start with a closed interval [0,1] (includes the points 0 and 1). At the first iteration replace the interval with 3 equal length pieces and remove the middle third, or]1/3, 2/3[

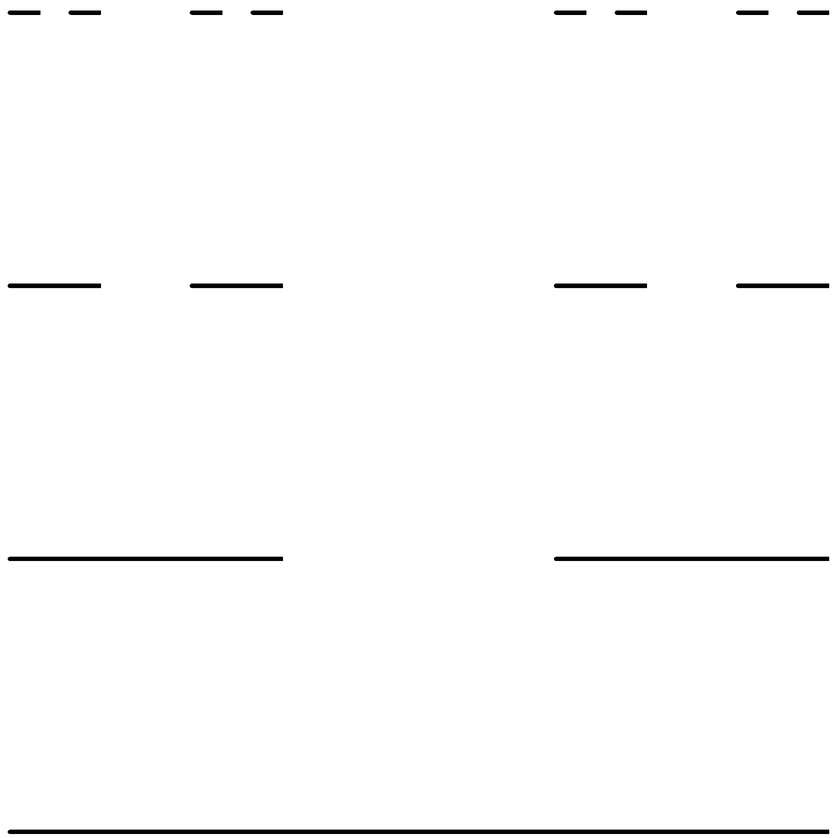
(excludes the points 1/3 and 2/3).
Subsequent iterations involve removing the middle portion of the remaining line segments.
The gap removed each time is usually called a trema from the Latin tremes = termite.
The triadic Cantor dust, as the name says, is the limit of this construction applying an infinite number of times, at the iteration *k* it is the remaining line segments after the *k* remotions.
We note that this fractal is defined recursively.

It follows immediately the procedure which draws the triadic Cantor dust giving as input the integer number *k* which is the step that we want to reach.

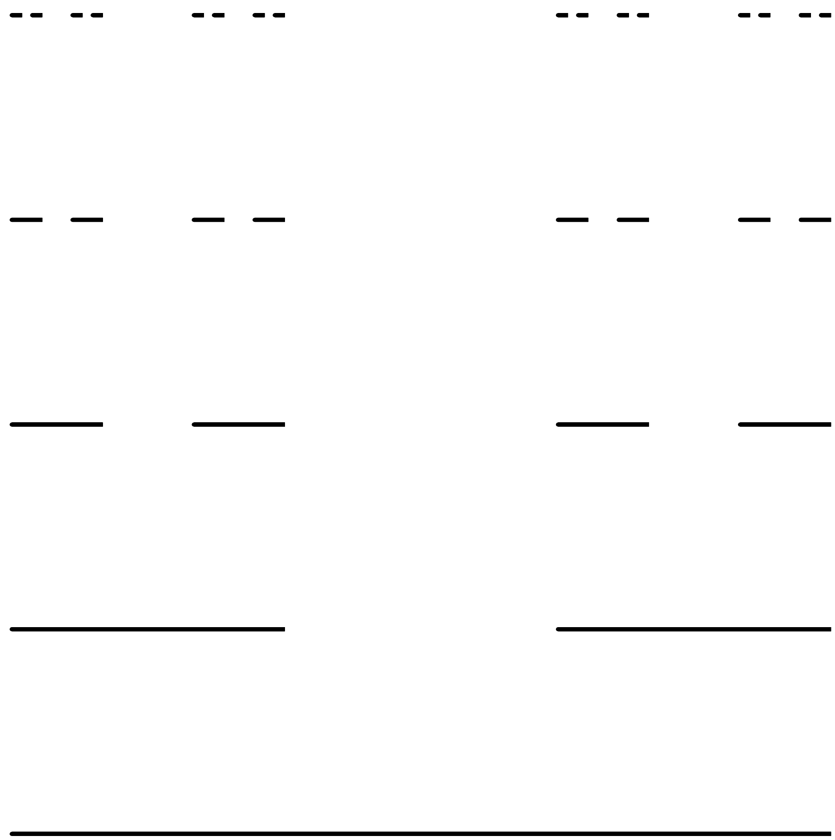
$with(plots):$

$dust := \mathbf{proc}(k)$
local $MS, MD, i, j, estremiSeg, a, b, t, n;$
$MS[1,1] := 0 : MS[2,2] := \frac{2}{3} : MS[2,1] := 0 : MD[1,1] := 1 : MD[2,1] := \frac{1}{3} : MD[2,2] := 1 :$
for i **from** 3 **to** $k+1$ **do** $MS[i,1] := 0 : MD[i,1] := \left(\frac{1}{3}\right)^{i-1} :$
for j **from** 2 **to** 2^{i-1} **do**
if $type(j, odd) = true$ **then** $MS[i,j] := MS\left[i-1, \frac{j+1}{2}\right] : MD[i,j] := MD\left[i-1, \frac{j+1}{2}\right] - \left(\frac{1}{3}\right)^{i-1} \cdot 2$
else $MS[i,j] := MS\left[i-1, \frac{j}{2}\right] + \left(\frac{1}{3}\right)^{i-1} \cdot 2 : MD[i,j] := MD\left[i-1, \frac{j}{2}\right]$ **end if**:**end do**: **end do**:
for a **from** 1 **to** $k+1$ **do**
for b **from** 1 **to** 2^{a-1} **do**
$estremiSeg[a,b] := [[MS[a,b], a-1], [MD[a,b], a-1]]$
end do end do:
$display\left(seq\left(seq\left(polygonplot(estremiSeg[t,n]), n = 1..2^{t-1}\right), t = 1..(k+1)\right), color = blue, axes = NONE, thickness = 2\right)$
end proc:

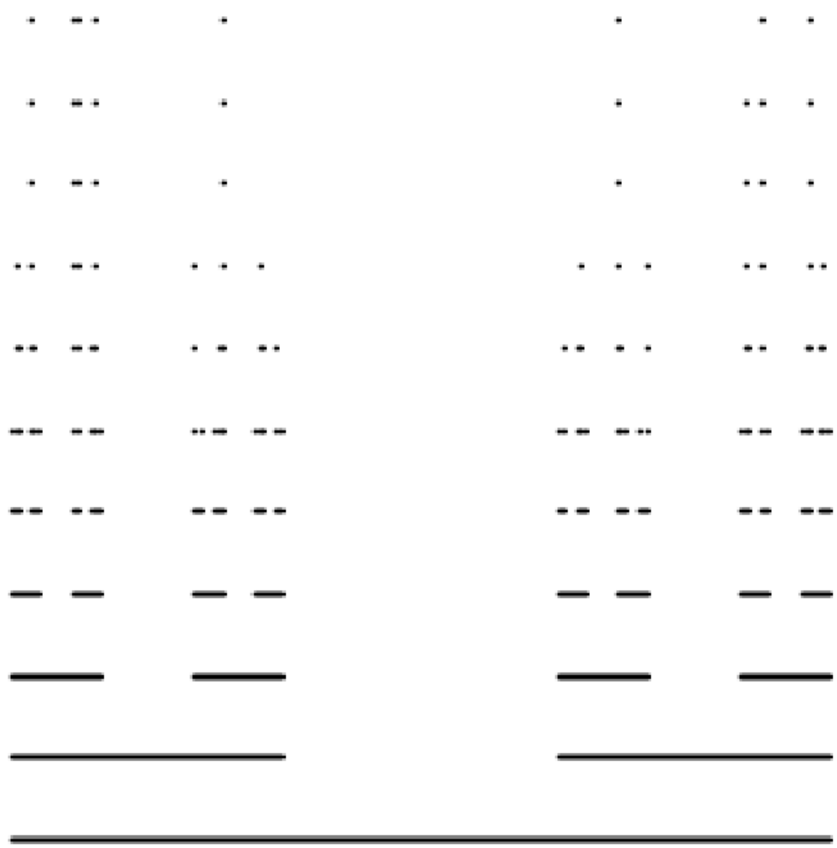
$dust(3)$



$dust(4)$



$dust(10)$



In the examples before we note that, already with $k = 10$, we obtain as representation of the fractal of Cantor a list of points, a "real dust". The total lenght of the Cantor dust itself is zero. The power of geometric visualization of Maple allow us, by this procedure, to draw easily this fractal easily at step $k >> 0$.

Koch Curve

The Koch curve appeared in a 1904 paper titled "*On a continuous curve without tangents, constructible from elementary geometry*" (original French title: "*Sur une courbe continue sans tangente, obtenue par une construction géométrique élémentaire*") by the Swedish mathematician Helge von Koch.

Let us recall the construction of the Koch curve. Start with a segment of lenght 1; divide it into three equal segments and replace the middle segment by the two sides of an equilateral triangle of the same length as the segment being removed. Now repeat this construction, taking each of the four resulting segments, dividing them into three equal parts and replacing each of the middle segments by two sides of an equilateral triangle as before. Continue this construction: the Koch curve is the limiting curve obtained by applying this construction an infinite number of times.

The following procedure, returns the picture of the curve obtained by applying a finite number of times the construction above.

Input datum is an integer number which is the step of the finite construction.

restart

with(*plots*) : *with*(*Student*:-*Precalculus*) :

Koch ≔**proc** (*m* :: *integer*)

local *a, A, k, n, l, r, M, C, j, Sx, P, Q, Sy, retta, S, Coef, Coefd, Coefs, rettad, rettas, rette, segmento, frat*;

$A[0,1] := [0,0] : A[0,2] := [1,0] : l[0] := 1 :$

if $m > 0$ **then**

for n **from** 1 **to** m **do**

$l[n] := \frac{1}{3^n} :$

$r[n] := \frac{1}{3^n \cdot 2} :$

for j **from** 0 **to** $4^{(n-1)}$ **do**

$A[n, 1 + j \cdot 4] := A[n-1, j+1]$

end do:

for j **from** 0 **to** $4^{n-1} - 1$ **do**

$M[n, j+1] := \left[\frac{A[n, 1+j\cdot 4][1] + A[n, (j+1)\cdot 4 + 1][1]}{2}, \frac{A[n, 1+j\cdot 4][2] + A[n, (j+1)\cdot 4 + 1][2]}{2} \right]$

end do:

for j **from** 0 **to** $(4^{n-1} - 1)$ **do**

$C[n, j+1] := (x - M[n, j+1][1])^2 + (y - M[n, j+1][2])^2 = (r[n])^2 :$

if $A[n, 1+j\cdot 4][1] - A[n, (j+1)\cdot 4 + 1][1] = 0$ **then**

$Sx[n, j+1] := solve([C[n, j+1], x = A[n, 1+j\cdot 4][1]], [x, y]) :$

$assign(Sx[n, j+1][1]) : P[n, j] := [x, y] :$

unassign('x') : *unassign*('y') :

$assign(Sx[n, j+1][2]) : Q[n, j] := [x, y] :$

unassign('x') :

unassign('y')

end if:

if $A[n, 1+j\cdot 4][2] - A[n, (j+1)\cdot 4 + 1][2] = 0$ **then**

$Sy[n, j+1] := solve([C[n, j+1], y = A[n, 1+j\cdot 4][2]], [x, y]) :$

$assign(Sy[n, j+1][1]) : P[n, j] := [x, y] :$

unassign('x') : *unassign*('y') :

$assign(Sy[n, j+1][2]) : Q[n, j] := [x, y] :$

unassign('x') : *unassign*('y')

end if:

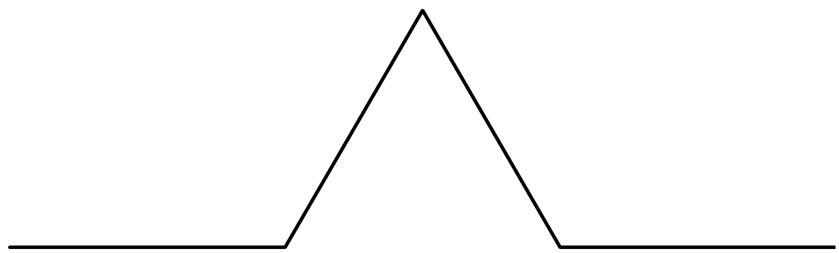
if $A[n, 1+j\cdot 4][1] - A[n, (j+1)\cdot 4 + 1][1] \neq 0$ **and** $A[n, 1+j\cdot 4][2] - A[n, (j+1)\cdot 4 + 1][2] \neq 0$ **then**

$retta[n, j+1] := \frac{x - A[n, (j+1)\cdot 4 + 1][1]}{A[n, 1+j\cdot 4][1] - A[n, (j+1)\cdot 4 + 1][1]} = \frac{y - A[n, (j+1)\cdot 4 + 1][2]}{A[n, 1+j\cdot 4][2] - A[n, (j+1)\cdot 4 + 1][2]} :$

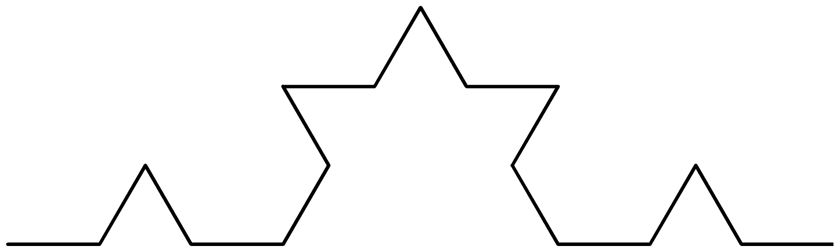
```
S[n, j + 1] := solve( [C[n, j + 1], retta[n, j + 1]], [x, y]) :
assign(S[n, j + 1][1]) : P[n, j] := [x, y] :
unassign('x') : unassign('y') :
assign(S[n, j + 1][2]) : Q[n, j] := [x, y] :
unassign('x') : unassign('y')
end if:
if evalf (Distance(A[n, 4·j + 1], P[n, j])) < evalf (Distance(A[n, 4·j + 1], Q[n, j])) then
A[n, 2 + 4·j] := P[n, j] : A[n, 4 + 4·j] := Q[n, j] else
A[n, 2 + 4·j] := Q[n, j] : A[n, 4 + 4·j] := P[n, j]
end if end do ;
unassign('j') :
for j from 0 to (4^(n - 1) - 1) do
Coef[n, j] := (A[n, 4 + 4·j][2] - A[n, 2 + 4·j][2]) / (A[n, 4 + 4·j][1] - A[n, 2 + 4·j][1]) :
Coefd[n, j] := - (sqrt(3) + Coef[n, j]) / (sqrt(3) ·Coef[n, j] - 1) :
Coefs[n, j] := - (-sqrt(3) + Coef[n, j]) / (-sqrt(3) ·Coef[n, j] - 1) :
rettad[n, j] := y - A[n, 2 + 4·j][2] = Coefd[n, j]· (x - A[n, 2 + 4·j][1]) :
rettas[n, j] := y - A[n, 4 + 4·j][2] = Coefs[n, j]· (x - A[n, 4 + 4·j][1]) :
rette[n, j] := solve( {rettad[n, j], rettas[n, j]}, [x, y]) :
assign(rette[n, j]) : A[n, 3 + 4·j] := [x, y] :
unassign('x') : unassign('y') :
end do  end do  end if:
for k from 1 to 4^m do
 segmento[m, k] := [A[m, k], A[m, k + 1]] :
frat[m, k] := polygonplot(segmento[m, k], axes = NONE, scaling = constrained) :
 end do:
display( seq( frat[m, k], k = 1 ..4^m ) )
 end proc:

Koch(0)
```

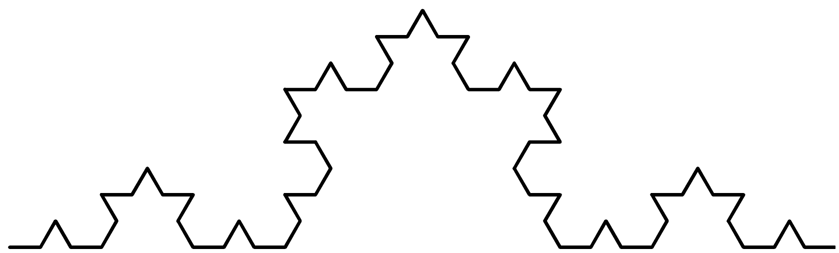
$Koch(1)$



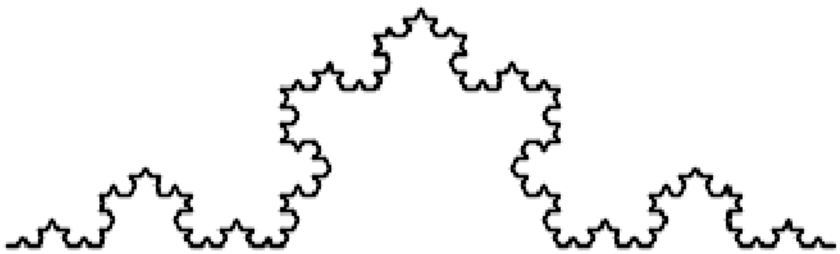
$Koch(2)$



$Koch(3)$



$Koch(4)$



We can note that already at the step 4 we obtain a nice and rich poligonal!

Snowflake Curve

The Koch snowflake is a simple close curve and one of the earliest fractal curves to have been described. It can be constructed by starting with an equilateral triangle, then recursively on each line segment by drawing the Koch curve. Sometimes it is used as example of a curve of infinite length surrounding a finite area.

The following procedure allow us to draw it at the step *m*, the integer number we have to give as input.

```
restart
with(plots) : with(Student:-Precalculus) :
Snowflake := proc (m :: integer)
 local a, A, k, n, l, r, M, C, j, Sx, P, Q, Sy, retta, S, Coef, Coefd, Coefs, rettad, rettas, rette, segmento, frat,
      E, F, fratE, fratF, segmentoE, segmentoF;
  A[0, 1] := [0, 0] : A[0, 2] := [1, 0] :
 l[0] := 1 :
```

if $m > 0$ **then**
for n **from** 1 **to** m **do**
$l[n] := \frac{1}{3^n} : r[n] := \frac{1}{3^n \cdot 2} :$
for j **from** 0 **to** $4^{(n-1)}$ **do**
$A[n, 1 + j \cdot 4] := A[n-1, j+1]$
end do:
for j **from** 0 **to** $4^{n-1} - 1$ **do**
$M[n, j+1] := \left[\frac{A[n, 1+j \cdot 4][1] + A[n, (j+1) \cdot 4 + 1][1]}{2}, \frac{A[n, 1+j \cdot 4][2] + A[n, (j+1) \cdot 4 + 1][2]}{2} \right]$
end do:
for j **from** 0 **to** $\left(4^{n-1} - 1\right)$ **do**
$C[n, j+1] := (x - M[n, j+1][1])^2 + (y - M[n, j+1][2])^2 = (r[n])^2 :$
if $A[n, 1+j \cdot 4][1] - A[n, (j+1) \cdot 4 + 1][1] = 0$ **then**
$Sx[n, j+1] := solve([C[n, j+1], x = A[n, 1+j \cdot 4][1]], [x, y]) :$
$assign(Sx[n, j+1][1]) : P[n, j] := [x, y] :$
$unassign('x') : unassign('y') :$
$assign(Sx[n, j+1][2]) : Q[n, j] := [x, y] :$
$unassign('x') : unassign('y')$
end if:
if $A[n, 1+j \cdot 4][2] - A[n, (j+1) \cdot 4 + 1][2] = 0$ **then**
$Sy[n, j+1] := solve([C[n, j+1], y = A[n, 1+j \cdot 4][2]], [x, y]) :$
$assign(Sy[n, j+1][1]) : P[n, j] := [x, y] :$
$unassign('x') : unassign('y') :$
$assign(Sy[n, j+1][2]) :$
$Q[n, j] := [x, y] :$
$unassign('x') : unassign('y')$
end if:
if $A[n, 1+j \cdot 4][1] - A[n, (j+1) \cdot 4 + 1][1] \neq 0$ **and** $A[n, 1+j \cdot 4][2] - A[n, (j+1) \cdot 4 + 1][2] \neq 0$ **then**
$retta[n, j+1] := \frac{x - A[n, (j+1) \cdot 4 + 1][1]}{A[n, 1+j \cdot 4][1] - A[n, (j+1) \cdot 4 + 1][1]} = \frac{y - A[n, (j+1) \cdot 4 + 1][2]}{A[n, 1+j \cdot 4][2] - A[n, (j+1) \cdot 4 + 1][2]} :$
$S[n, j+1] := solve([C[n, j+1], retta[n, j+1]], [x, y]) :$
$assign(S[n, j+1][1]) : P[n, j] := [x, y] :$
$unassign('x') : unassign('y') :$
$assign(S[n, j+1][2]) : Q[n, j] := [x, y] :$
$unassign('x') : unassign('y')$
end if:
if $evalf(Distance(A[n, 4 \cdot j + 1], P[n, j])) < evalf(Distance(A[n, 4 \cdot j + 1], Q[n, j]))$ **then**
$A[n, 2 + 4 \cdot j] := P[n, j] : A[n, 4 + 4 \cdot j] := Q[n, j]$ **else**
$A[n, 2 + 4 \cdot j] := Q[n, j] : A[n, 4 + 4 \cdot j] := P[n, j]$
end if end do ;
$unassign('j') :$
for j **from** 0 **to** $\left(4^{n-1} - 1\right)$ **do**
$Coef[n, j] := \frac{A[n, 4 + 4 \cdot j][2] - A[n, 2 + 4 \cdot j][2]}{A[n, 4 + 4 \cdot j][1] - A[n, 2 + 4 \cdot j][1]} :$

$Coefd[n, j] := -\frac{\sqrt{3} + Coef[n, j]}{\sqrt{3} \cdot Coef[n, j] - 1}$:

$Coefs[n, j] := -\frac{-\sqrt{3} + Coef[n, j]}{-\sqrt{3} \cdot Coef[n, j] - 1}$:

$rettad[n, j] := y - A[n, 2+4 \cdot j][2] = Coefd[n, j] \cdot (x - A[n, 2+4 \cdot j][1])$:
$rettas[n, j] := y - A[n, 4+4 \cdot j][2] = Coefs[n, j] \cdot (x - A[n, 4+4 \cdot j][1])$:
rette[*n*, *j*] := *solve*({*rettad*[*n*, *j*], *rettas*[*n*, *j*]}, [*x*, *y*]) :
assign(*rette*[*n*, *j*]) : $A[n, 3+4 \cdot j] := [x, y]$:
unassign('*x*') : *unassign*('*y*') :
end do end do end if:
for *k* **from** 1 **to** $4^m + 1$ **do**

$E[m, k][1] := \cos\left(-\frac{\pi}{3}\right) \cdot A[m, k][1] - \sin\left(-\frac{\pi}{3}\right) \cdot (-A[m, k][2]);$

$E[m, k][2] := \sin\left(-\frac{\pi}{3}\right) \cdot A[m, k][1] + \cos\left(-\frac{\pi}{3}\right) \cdot (-A[m, k][2]);$

$E[m, k] := [E[m, k][1], E[m, k][2]]$:

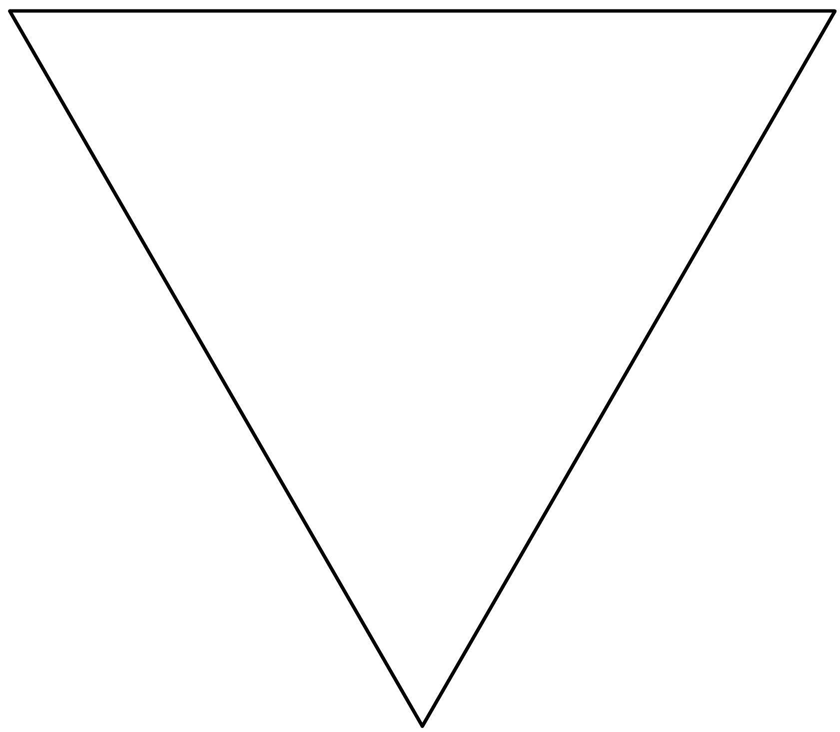
$F[m, k][1] := \left(\cos\left(-\frac{2 \cdot \pi}{3}\right) \cdot (A[m, k][1]) - \sin\left(-\frac{2 \cdot \pi}{3}\right) \cdot A[m, k][2]\right) + 1;$

$F[m, k][2] := \sin\left(-\frac{2 \cdot \pi}{3}\right) \cdot (A[m, k][1]) + \cos\left(-\frac{2 \cdot \pi}{3}\right) \cdot A[m, k][2]$:

$F[m, k] := [F[m, k][1], F[m, k][2]]$
end do:
for *k* **from** 1 **to** 4^m **do**
segmento[*m*, *k*] := [*A*[*m*, *k*], *A*[*m*, *k* + 1]] :
segmentoE[*m*, *k*] := [*E*[*m*, *k*], *E*[*m*, *k* + 1]] :
segmentoF[*m*, *k*] := [*F*[*m*, *k*], *F*[*m*, *k* + 1]] :
frat[*m*, *k*] := *polygonplot*(*segmento*[*m*, *k*], *axes* = *NONE*, *scaling* = *constrained*) :
fratE[*m*, *k*] := *polygonplot*(*segmentoE*[*m*, *k*], *axes* = *NONE*, *scaling* = *constrained*) :
fratF[*m*, *k*] := *polygonplot*(*segmentoF*[*m*, *k*], *axes* = *NONE*, *scaling* = *constrained*)
end do:
display(*seq*({ *frat*[*m*, *k*], *fratE*[*m*, *k*], *fratF*[*m*, *k*]}, $k = 1..4^m$))
end proc:

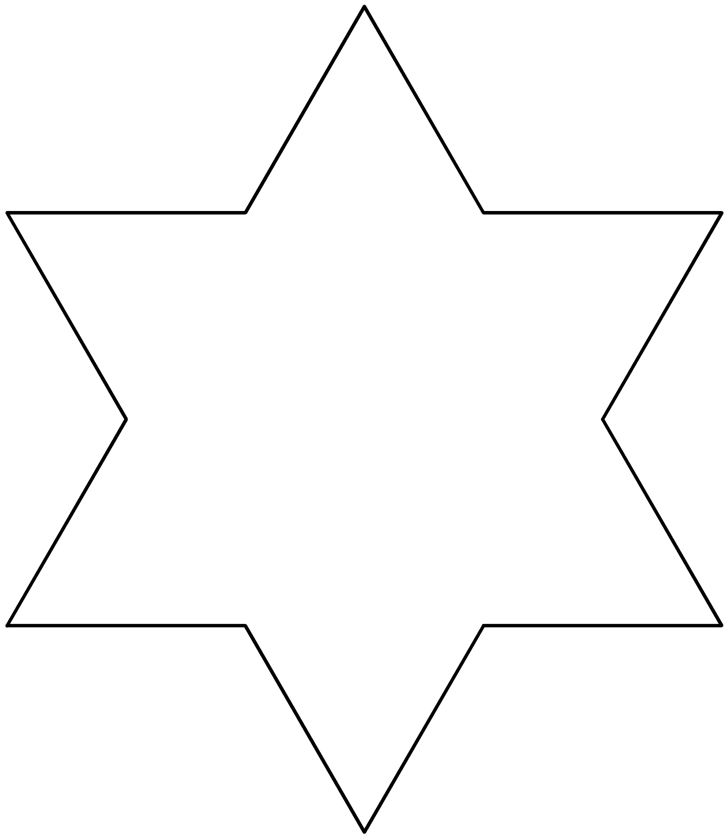
At step *m=0*, the Koch snowflake is an equilateral triangle.

Snowflake(0)

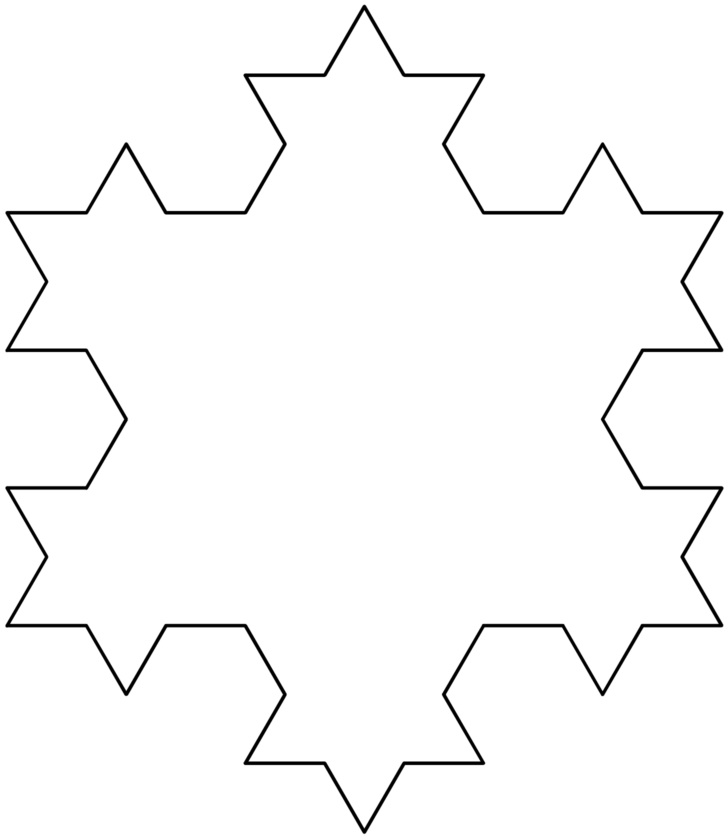


After one iteration, the result is a shape similar to the Star of David.

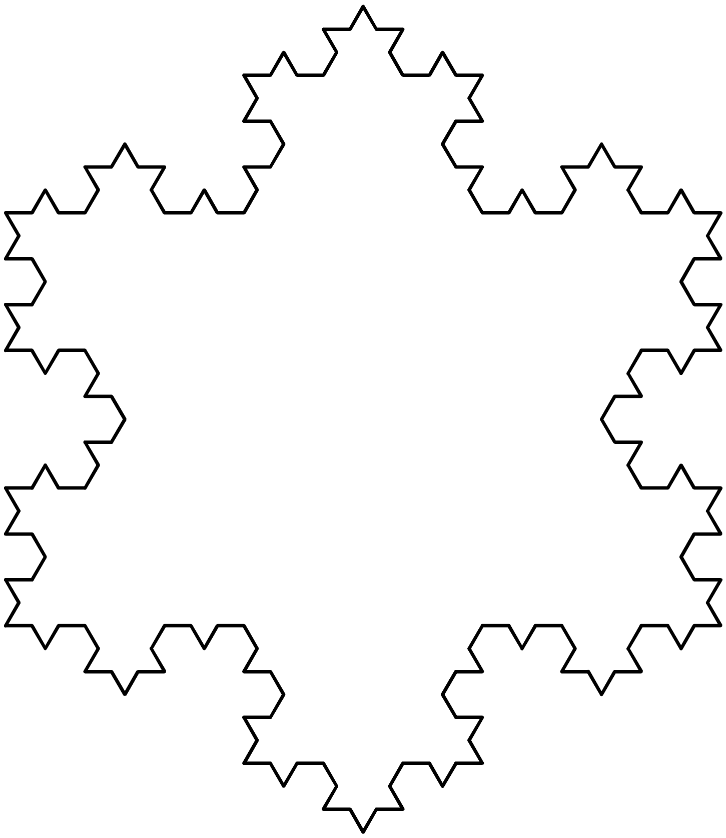
$Snowflake(1)$



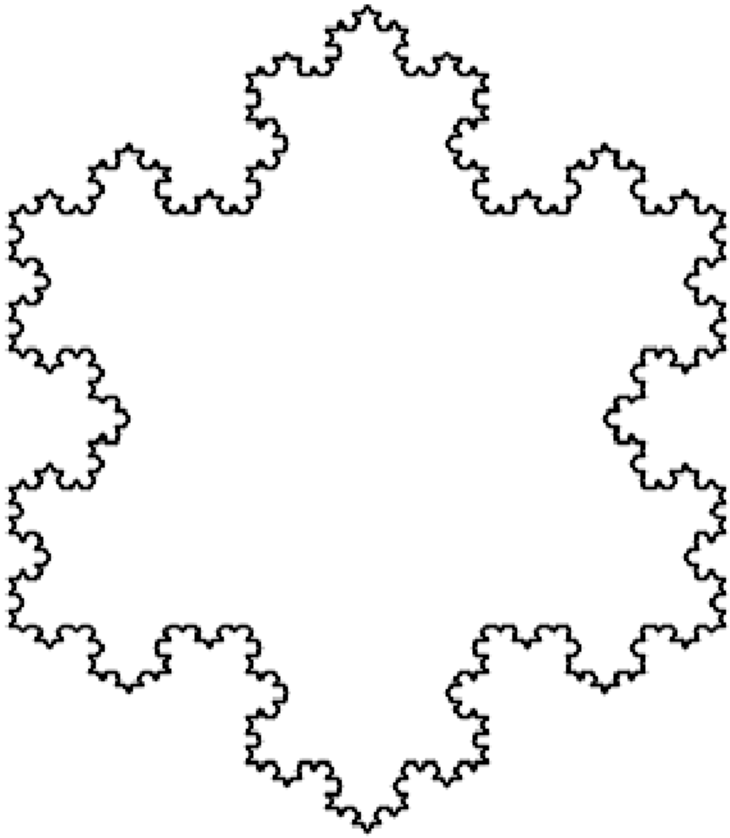
$Snowflake(2)$



$Snowflake(3)$



$Snowflake(4)$



We are interested to write simple procedures that calculate the area of the region inside the snowflake curve and its perimeter. Of course we expect that the first one tend to a finite value and the second one tends to infinite.

Area of the Region Inside the Fractal Snowflake Curve

The imput of the following procedure is again a positive integer *n* which tells to Maple how many steps calculate.
The output is the area at each step and it is approximated at 7 decimal digits.

$ASnowflake := \textbf{proc}(n :: integer)$
$\textbf{local}\ a, i;$
$a := \frac{\sqrt{3}}{4} : print(A[0] = a, evalf[7](a)) :$
$\textbf{for}\ i\ \textbf{from}\ 1\ \textbf{to}\ n - 1\ \textbf{do}$

$a := a + \frac{1}{3}\left(\frac{4}{9}\right)^{i-1}\frac{\sqrt{3}}{4}$:

print($A[i] = a$, *evalf*[7](a)) **end do**

end proc:

ASnowflake(20)

$$A_0 = \frac{1}{4}\sqrt{3}, 0.4330128$$

$$A_1 = \frac{1}{3}\sqrt{3}, 0.5773503$$

$$A_2 = \frac{10}{27}\sqrt{3}, 0.6415004$$

$$A_3 = \frac{94}{243}\sqrt{3}, 0.6700115$$

$$A_4 = \frac{862}{2187}\sqrt{3}, 0.6826831$$

$$A_5 = \frac{7822}{19683}\sqrt{3}, 0.6883150$$

$$A_6 = \frac{70654}{177147}\sqrt{3}, 0.6908180$$

$$A_7 = \frac{636910}{1594323}\sqrt{3}, 0.6919305$$

$$A_8 = \frac{5736286}{14348907}\sqrt{3}, 0.6924248$$

$$A_9 = \frac{51642958}{129140163}\sqrt{3}, 0.6926446$$

$$A_{10} = \frac{464852158}{1162261467}\sqrt{3}, 0.6927423$$

$$A_{11} = \frac{4183931566}{10460353203}\sqrt{3}, 0.6927858$$

$$A_{12} = \frac{37656432670}{94143178827}\sqrt{3}, 0.6928050$$

$$A_{13} = \frac{338912088334}{847288609443}\sqrt{3}, 0.6928135$$

$$A_{14} = \frac{3050225572222}{7625597484987}\sqrt{3}, 0.6928173$$

$$A_{15} = \frac{27452097258862}{68630377364883}\sqrt{3}, 0.6928190$$

$$A_{16} = \frac{247069143765214}{617673396283947}\sqrt{3}, 0.6928199$$

$$A_{17} = \frac{2223623367628750}{5559060566555523}\sqrt{3}, 0.6928201$$

$$A_{18} = \frac{20012614603626046}{50031545098999707}\sqrt{3}, 0.6928202$$

$$A_{19} = \frac{180113548612503598}{450283905890997363}\sqrt{3}, 0.6928204 \quad \textbf{(1)}$$

Perimeter of the Fractal Snowflake Curve

The imput of the following procedure is again a positive integer *n* which tells to Maple how many steps calculate.
The output is the perimeter at each step and it is approximated again at 7 decimal digits.

restart

```
PSnowflake := proc(n :: integer)
local p, i;
 for i from 0 to n − 1 do
 p := 3·(4/3)^i :
 print(P[i] = p, evalf[7](p)) end do
 end proc:
PSnowflake(20)
```

$$P_0 = 3, 3.$$

$$P_1 = 4, 4.$$

$$P_2 = \frac{16}{3}, 5.333333$$

$$P_3 = \frac{64}{9}, 7.111111$$

$$P_4 = \frac{256}{27}, 9.481481$$

$$P_5 = \frac{1024}{81}, 12.64198$$

$$P_6 = \frac{4096}{243}, 16.85597$$

$$P_7 = \frac{16384}{729}, 22.47462$$

$$P_8 = \frac{65536}{2187}, 29.96616$$

$$P_9 = \frac{262144}{6561}, 39.95488$$

$$P_{10} = \frac{1048576}{19683}, 53.27318$$

$$P_{11} = \frac{4194304}{59049}, 71.03091$$

$$P_{12} = \frac{16777216}{177147}, 94.70788$$

$$P_{13} = \frac{67108864}{531441}, 126.2772$$

$$P_{14} = \frac{268435456}{1594323}, 168.3696$$

$$P_{15} = \frac{1073741824}{4782969}, 224.4927$$

$$P_{16} = \frac{4294967296}{14348907}, 299.3237$$

$$P_{17} = \frac{17179869184}{43046721}, 399.0982$$

$$P_{18} = \frac{68719476736}{129140163}, 532.1309$$

$$P_{19} = \frac{274877906944}{387420489}, 709.5079 \qquad \textbf{(2)}$$

At step n

the perimeter is: $P(n) = 3 \cdot \left(\frac{4}{3}\right)^n$

and the area is: $A(n) = \frac{\sqrt{3}}{4} \cdot \left(1 - \frac{3}{5}\left(\left(\frac{4}{9}\right)^n - 1\right)\right)$

We can define the functions in x:

$$P := x \rightarrow 3 \cdot \left(\frac{4}{3}\right)^x$$

$$x \rightarrow 3 \left(\frac{4}{3}\right)^x \qquad \textbf{(3)}$$

$$A := x \rightarrow \frac{\sqrt{3}}{4} \cdot \left(1 - \frac{3}{5}\left(\left(\frac{4}{9}\right)^x - 1\right)\right)$$

$$x \rightarrow \frac{1}{4}\sqrt{3}\left(\frac{8}{5} - \frac{3}{5}\left(\frac{4}{9}\right)^x\right) \qquad \textbf{(4)}$$

and if we calculate the limits for x tending to infinite we obtain as we expected:

$$\lim_{x \to +\infty} P(x)$$

$$\infty \tag{5}$$

$$\lim_{x \to +\infty} A(x)$$

$$\frac{2}{5}\sqrt{3} \tag{6}$$

Reference

Gerald A.Edgar, *Measure, Topology, and Fractal Geometry*, Springer-Verlag, New York, Berlin, 1990.