XPPAUT Commands used to generate the Sierpinski Triangle

The Sierpinski triangle is a fractal obtained by dividing a triangle into 4 equal pieces and removing the middle piece. This is repeated in the three smaller triangles which remain, and then repeated again and again. The result is reminiscent to a two-dimensional Cantor set.

This set can also be generated as explained in Section 14.1 of your text book. This procedure will work with any triangle, starting the iteration from any initial point. The program provided does this for a right triangle with vertices at $(0,0),(1,0)$, and $(0,1)$. Zoom in to observe the self-similarity.

```
#sierpinski.ode
#drawing the Sierpinski triangle
#
par c0=0,c1=.5,c2=0
par d0=0,d1=0,d2=.5
p=flr(ran(1)*3)
dx/dt=.5*x+shift(c0,p)
dy/dt=.5*y+shift(d0,p)
@ xp=cx,yp=cy
@ maxstor=20000000,total=20000,meth=discrete
@ xlo=0,xhi=1,ylo=0,yhi=1,lt=0
@ xp=x,yp=y
aux pp=p
done
```

flr(arg) – gives the largest integer less than arg.

ran(c) – generates a uniformly distributed random variable from the interval $(0, c)$.

flr(ran(1)*3) – This can be used to simulate throwing a dice with the numbers 0, 1, and 2.

shift(c0,p) – will be c0, c1, or c2 according as whether p is 0, 1, or 2, respectively.

shift(d0,p) – evaluates the variable dp, similarly.

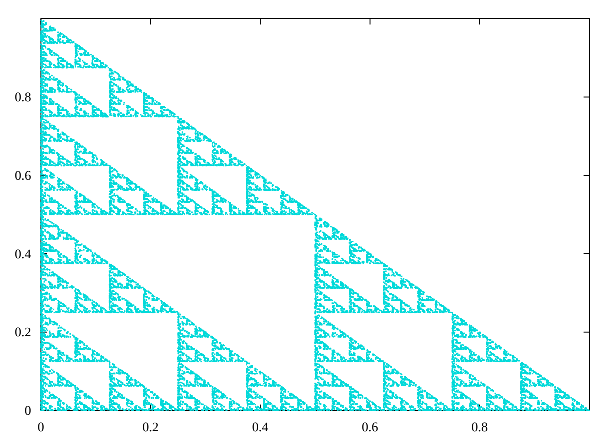

Figure 1: The Sierpinski Triangle.