

## Testing the L picture for fractal generation

In order to make fractals with iterated function systems it is nice to have a test procedure to make sure you have picked your affine maps correctly (and to help you adjust them later if necessary.) The procedure TESTMAP below, takes a list of affine functions as its input, and the result is a mapping -L picture like the ones in the fractal blueprints from Peitgen's book.

Before trying to generate fractal examples you should load the subroutines in this file, either by hitting <enter> through the entire document, or by choosing Edit/Execute/Worksheet from the menu toolbar.

```
> restart:with(plots):Digits:=4:
#every time you start a new fractal save and close your
#previous fractals work. Then return to this window
#and restart; otherwise you are likely to overwhelm
#Maple and crash! Also, save frequently.
Warning, the name changecoords has been redefined

> TESTMAP:=proc(funclist) #this procedure lets you test a list of
#functions in your iterated function system
local num,#the number of functions
      i, #dummy index
      F, #current function in list
      S, #corners of unit square
      L, #corners of letter L
      Sq, #unit square
      Llet, #letter L
      AS, #transf of square corners
      ASq,#transf of square
      AL, #transf of L corners
      ALlet, #transf of letter L
      Pics; #a list of pictures
S:=[[0,0],[0,1],[1,1],[1,0]];
L:=[[.1,.9],[.1,.65],[.2,.65],[.2,.675],[.125,.675],[.125,.9]]:
Sq:=polygonplot(S): #polygonplot connects the dots!
Llet:=polygonplot(L):
display({Sq,Llet});
num:=nops(funclist):
for i from 1 to num do
  F:=funclist[i]: #select ith map
  AS[i]:=map(F,S):
  AL[i]:=map(F,L):
  ASq[i]:=polygonplot(AS[i]):
  ALlet[i]:=polygonplot(AL[i]):
  #a plot of the transformed square and letter:
  Pics[i]:=display({ASq[i],ALlet[i]}):
od:
#finally, display the unit square and all its images:
display({Sq, seq(Pics[i],i=1..num)},scaling=constrained,
```

```

title='fractal template');
end:

```

Here is the standard affine map, which encodes

$$\text{AFFINE1} \begin{pmatrix} x \\ y \end{pmatrix} = x \begin{bmatrix} a \\ b \end{bmatrix} + y \begin{bmatrix} c \\ d \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}$$

[ or, in the alternate matrix representation:

$$\text{AFFINE1} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} a & c \\ b & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}$$

```

> AFFINE1:=proc(X,a,b,c,d,e,f)
  RETURN(evalf([a*X[1]+c*X[2]+e,
                b*X[1]+d*X[2]+f]));
end:
[ >
[ >

```

Now if you open a new file you can use these procedures to test a template and generate a fractal. Every time you restart you will need to return to this window to re-enter the procedures you need.