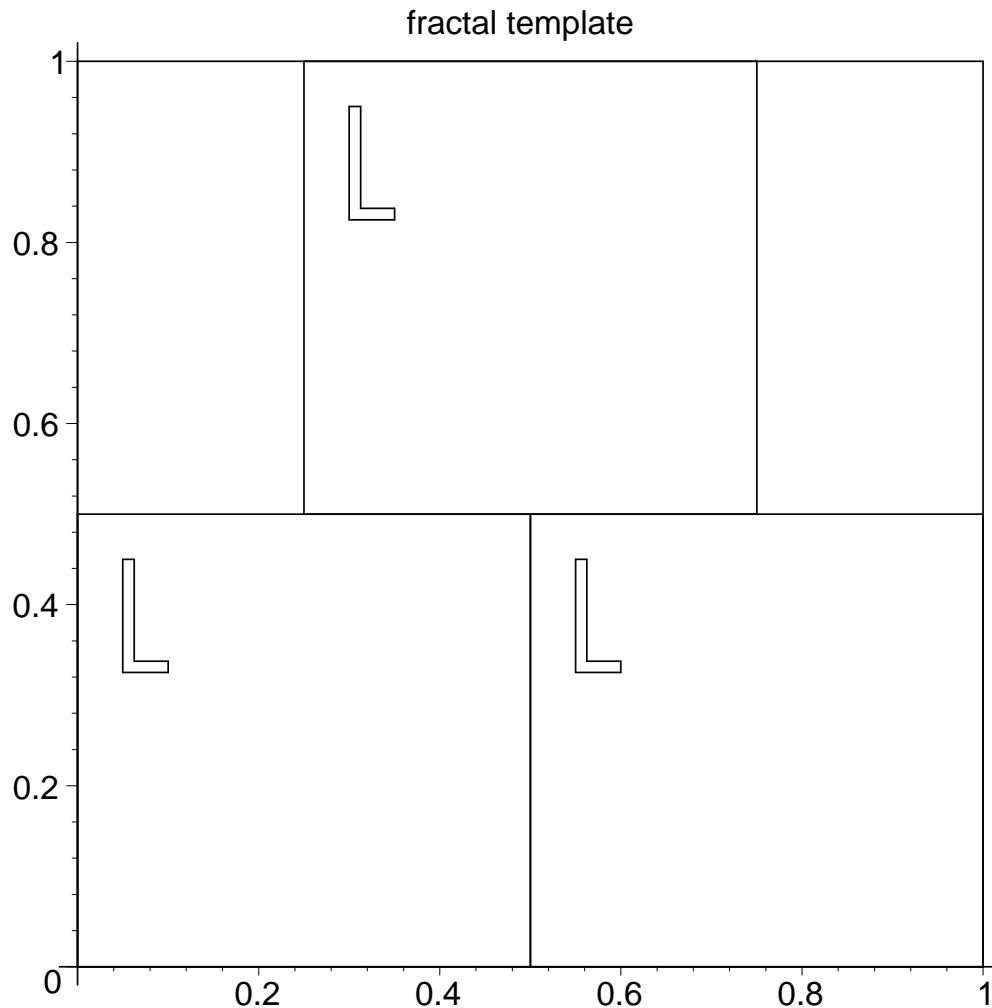


Sierpinski triangle

If you have loaded the procedures from the Lpictures.mws file, you can try defining some affine contractions, draw the L-picture, and then generate the fractal. We show it here for Sierpinski's triangle. (The isocetes version.)

```
> f1:=P->AFFINE1(P,.5,0,0,.5,0,0);  
    #shrink by .5 and don't translate  
f2:=P->AFFINE1(P,.5,0,0,.5,.5,0);  
    #same shrink, and translate 0.5 to the right  
f3:=P->AFFINE1(P,.5,0,0,.5,.25,.5);  
    #shrink, then displace by [.25,.5]  
    f1 := P → AFFINE1(P, 0.5, 0, 0, 0.5, 0, 0)  
    f2 := P → AFFINE1(P, 0.5, 0, 0, 0.5, 0.5, 0)  
    f3 := P → AFFINE1(P, 0.5, 0, 0, 0.5, 0.25, 0.5)  
> TESTMAP([f1,f2,f3]);
```



Since the template is correct, we may proceed.

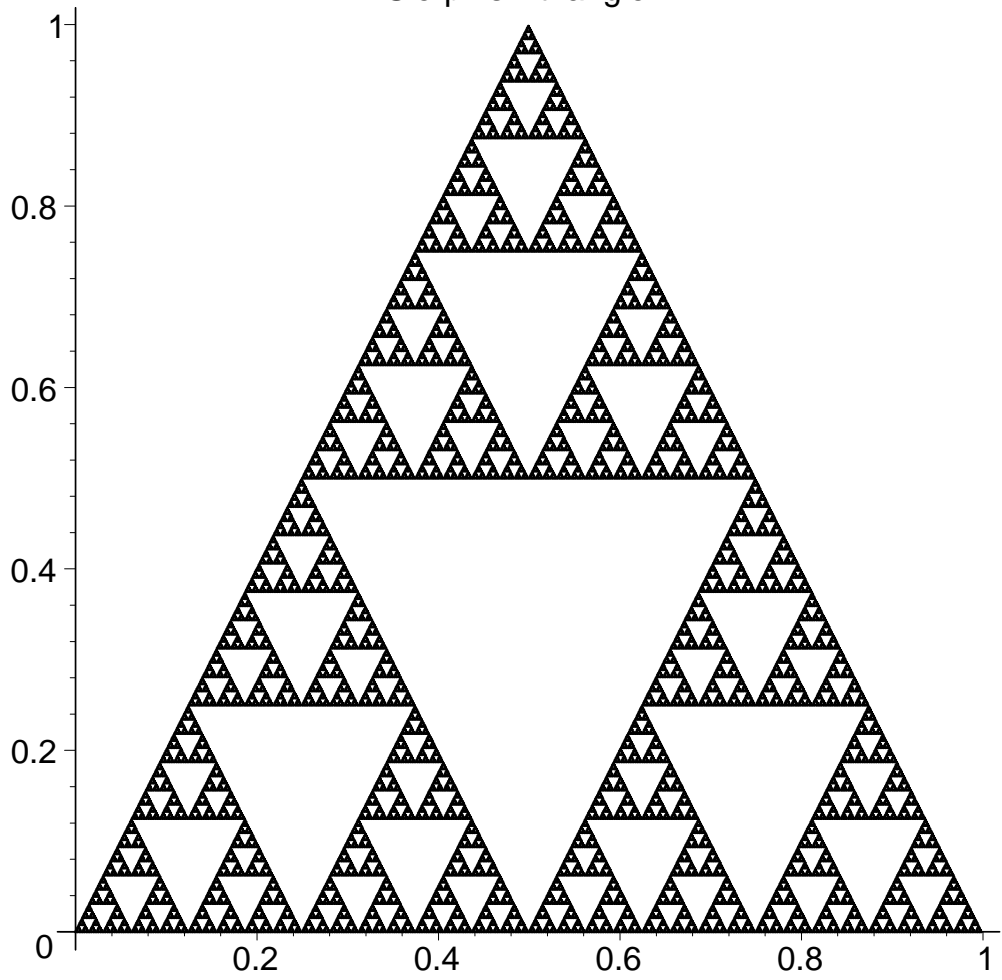
```
[ > S:={ [0,0] }:#initial set consisting of one point
[ > 3^9; #good to keep point numbers well below 100,000,
      #so as not to strain Maple's memory
```

19683

Based on the computation above I will do nine iterations below:

```
[ > for i from 1 to 9 do
  S1:=map(f1,S);
  S2:=map(f2,S);
  S3:=map(f3,S);
  S:='union'(S1,S2,S3);
od:
[ > pointplot(S,symbol=point,scaling=constrained,
  title='Sierpinski triangle');
```

Sierpinski triangle



[>