Adventures in Ratsub II

Adam Ponting

Nov 2019

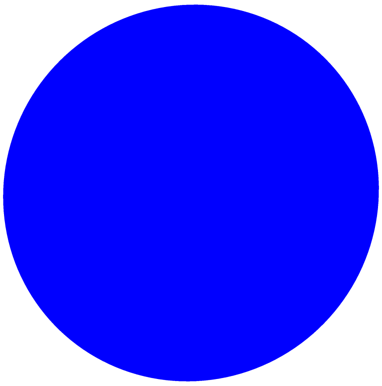
Contents

1 Circles **3**
Minsky method . . . 3
Bezier method . . . 4
2 Bezier curves **6**
3 Fractals **9**
IFS . . . 9
Self-replicating tiles . . . 9
Twindragon . . . 10
Square snowflake . . . 11
Classic fractals . . . 12
Lévy's dragon . . . 12
3D fractals . . . 14
4 Gallery **15**
Thue-Morse tilings . . . 17
Boot tiling . . . 18
5 Showing use of commands **21**
Random colour range . . . 21
par . . . 22
6 Misc **23**
7 Good bugs **24**
8 More tilings.. **25**
Scherer's L . . . 25
Vince-Barnsley paper . . . 25
Trapezoids . . . 25
C . . . 28
Notes and sources **29**

1. Circles

Minsky method

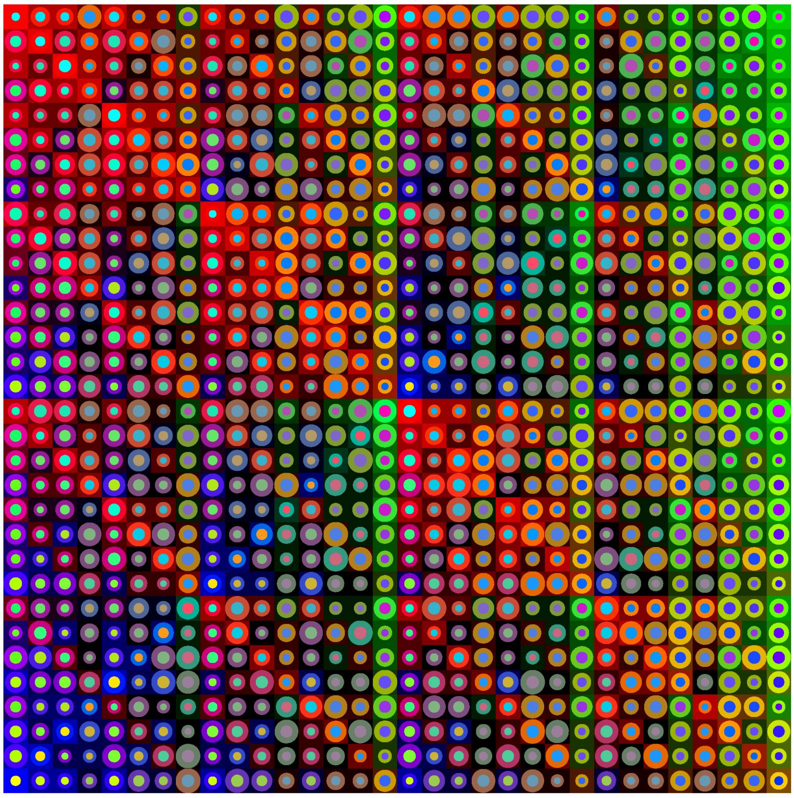
```
no edges
margin 51 51 1 51
pbox 50 0
def sq 4
  p4..5 p3 0..1/1 p2 left .04
  p6..7 p1 0..1/1 p5 left .04
  draw :027 blue
  sq p0,6,7,4
```



Level 200

Maybe use Minsky method only with colouring! like here use draw :027 + 0 0 .01 instead to get shaded circle...

```
no edges
def frame 4
  .5 .3 .5
  p4 c
  p5..8 p0.. .5 p1..0
  frame :5164 - -2 1 1
  frame :4627 - 1 -2 1
  frame :0548 - 1 1 -2
  frame :8473 - -1 -1 2
  >4
  sq :012347 =
def sq 6
  p6 p4 .5 p5
  p7 p6 rand p5
  p8 p4 .5 p7
  circ :4488 !
  circ :4477 =
  draw :0123 - 5 5 5
def circ 4
  p4..5 p3 0..1/1 p2 left .02
  p6..7 p1 0..1/1 p5 left .02
  draw :027 =
  circ :0674 =
```



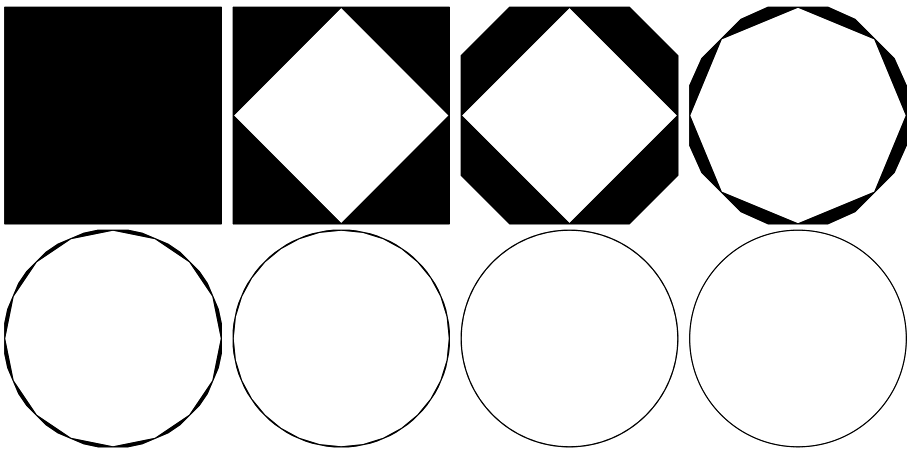
Level 316

Bezier method

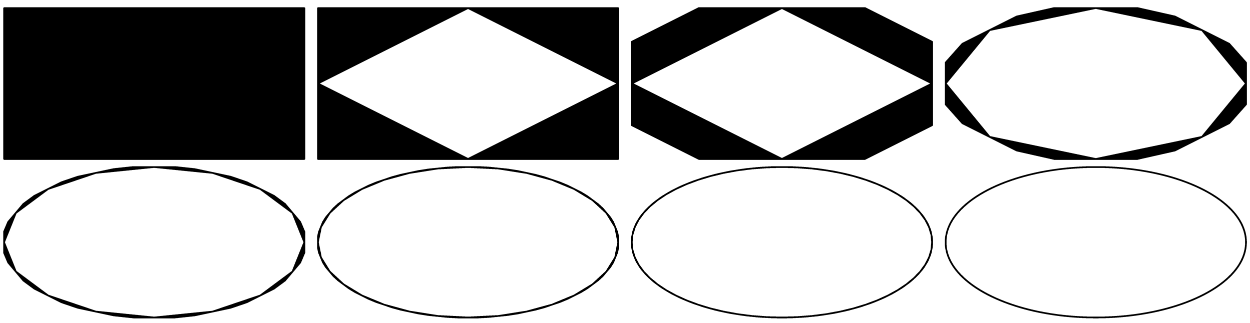
...for Bezier curve with n segments the optimal distance to the control points, in the sense that the middle of the curve lies on the circle itself, is $\frac{4}{3}\tan\frac{\pi}{2n}$. ...So for 4 points it is $\frac{4}{3}\tan\frac{\pi}{8} = \frac{4}{3}\left(\sqrt{2}-1\right) = 0.552284749831$[1]

Approximation to a circle with 4 points—

```
width .5
def sq 4
  p4..7 p0.. .5 p1..0
  q :415
  q :526
  q :637
  q :704
def q 3
  p3 p0 .55 p1
  p4 p2 " p1
  bez :0342
def bez 4
  p4..6 p0.. .5 p1..
  p7..8 p4.. .5 p5..
  p9 p7 .5 p8
  bez :0479
  bez :9863
```

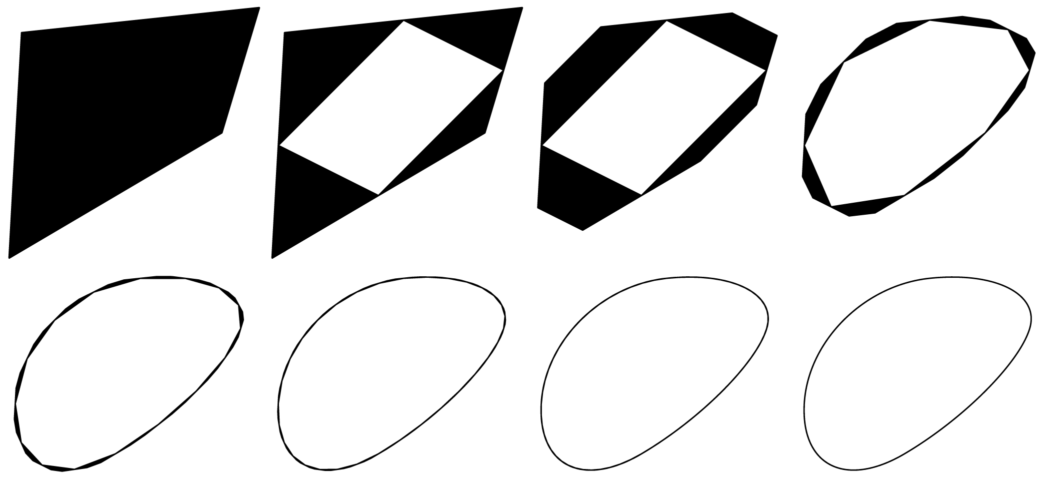


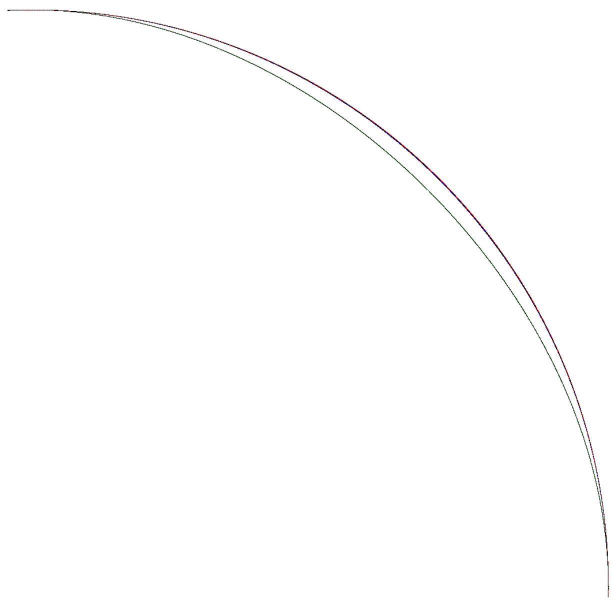
For an ellipse, just change the bounding box/initial points. The same program but with `pbox 100 50` added after the first line:



[1] https://stackoverflow.com/questions/1734745/how-to-create-circle-with-bézier-curves

With (`p0 0 0`), `p1 5 90`, `p2 100 100`, `p3 85 50`:

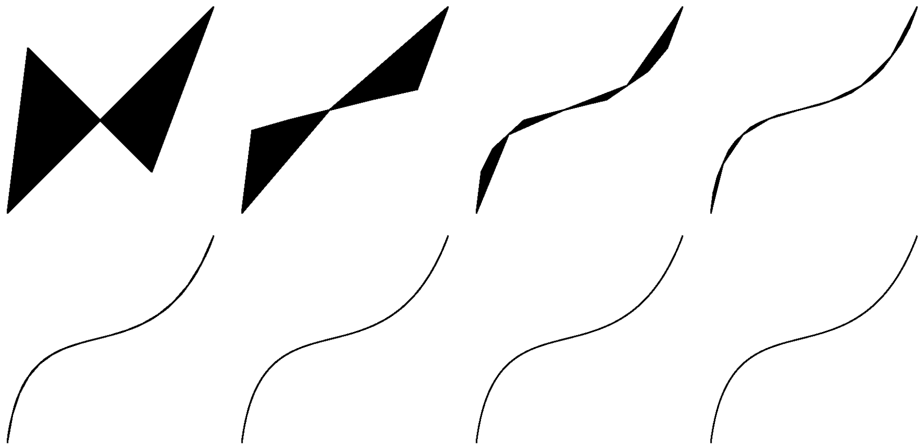




A comparison of circle approximations with control point values 0.5 (green), 0.55 (blue) and 0.5522847 (red) - the blue and red are indistinguishable. (Picture is level 12)

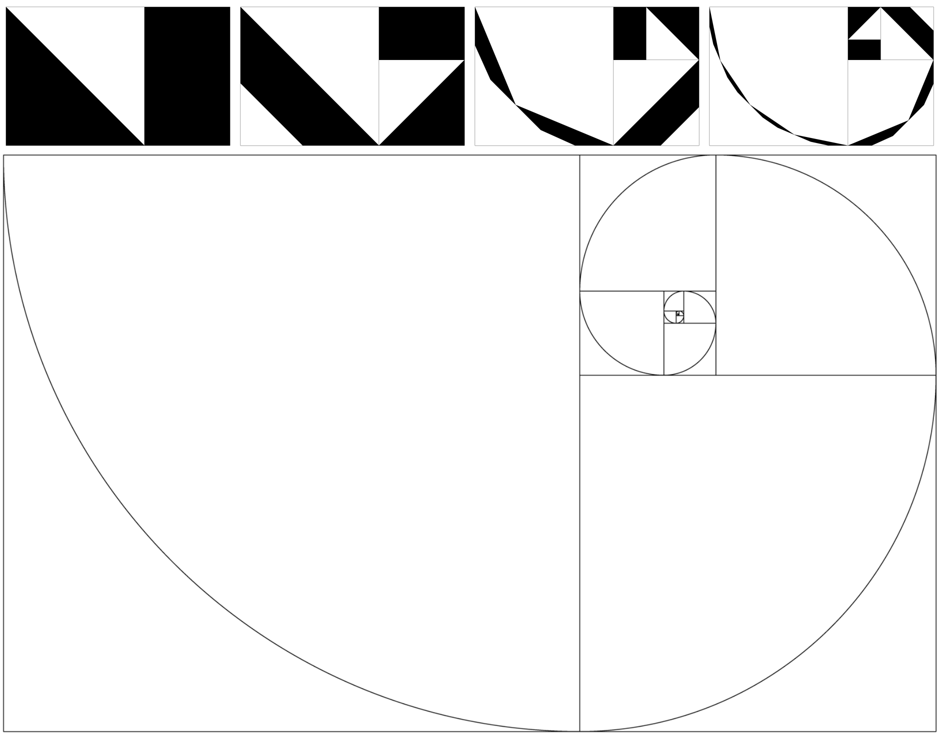
2. Bezier curves

```
p1 2 16
p2 14 4
p3 20 20
def bez 4
  p4..6 p0.. .5 p1..
  p7..8 p4.. .5 p5..
  p9 p7 .5 p8
  bez :0479
  bez :9863
```



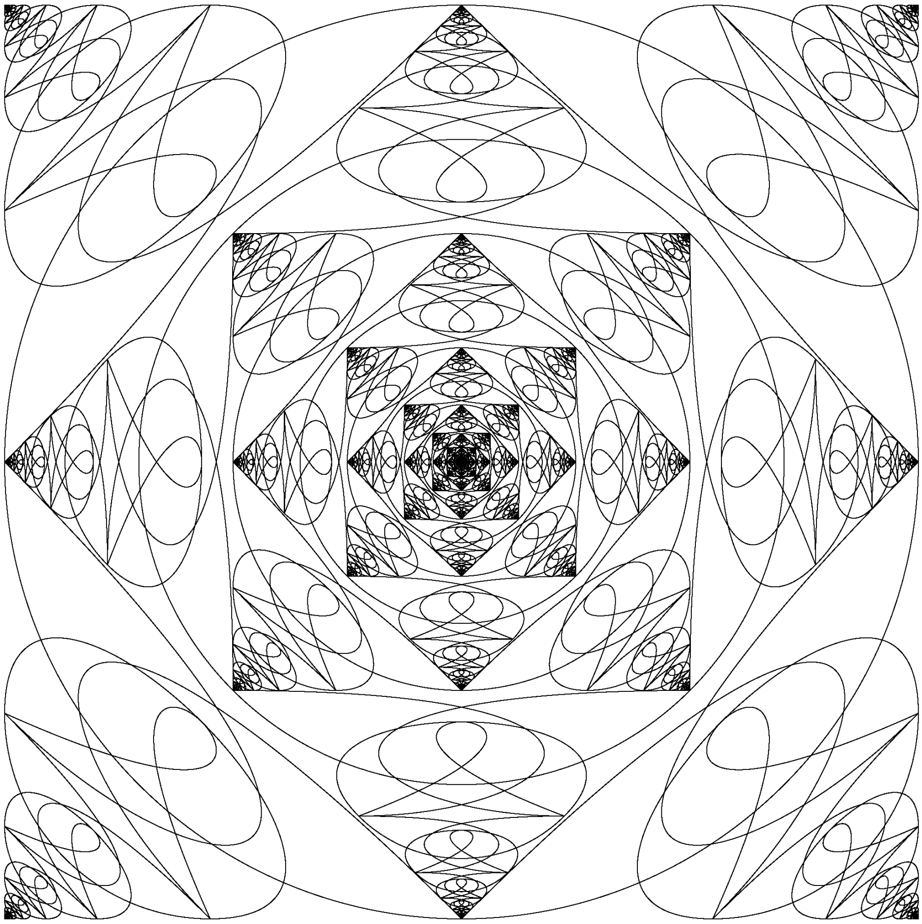
Fibonacci spiral

```
pbox 161.8 100
def rect 4
  p4 p1 .618 p2
  p5 p0 .618 p3
  arc :105
  rect :3542
  draw :0123 white
def arc 3
  p3 p0 .55 p1
  p4 p2 " p1
  arc :012
  >-9
  bez :0342
def bez 4
  p4..6 p0.. .5 p1..
  p7..8 p4.. .5 p5..
  p9 p7 .5 p8
  bez :0479
  bez :9863
```



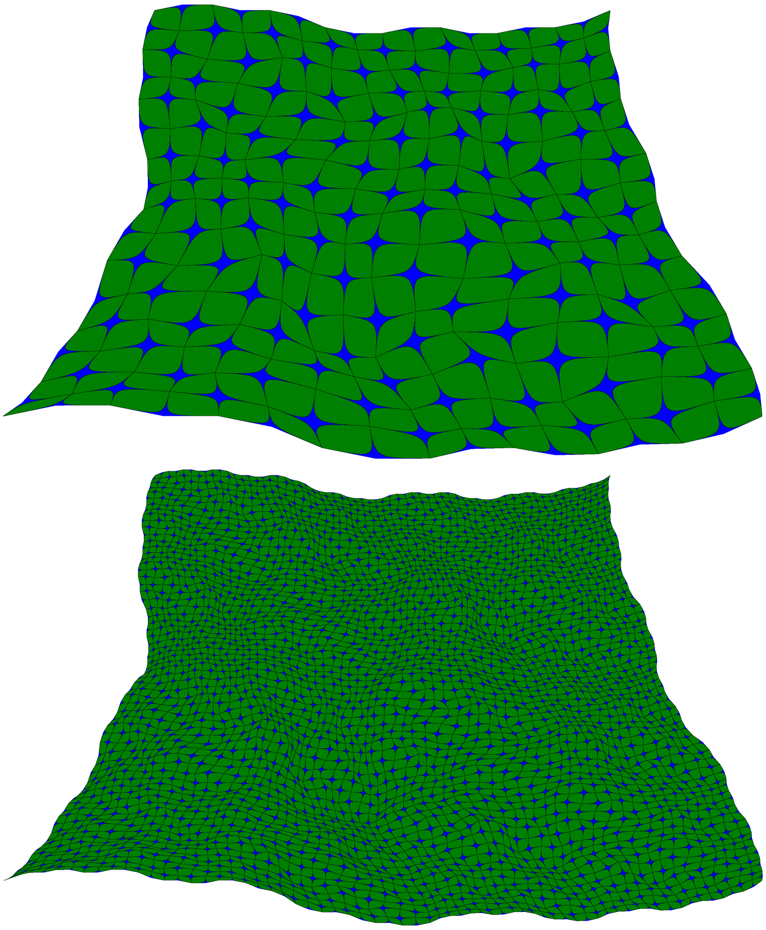
Levels 1-4 and 20.

```
width .05
def sq 4
  p4..7 p0.. .5 p1..0
  q :415
  q :526
  q :637
  q :704
  sq :4567
def q 3
  p3 p0 .55 p1
  p4 p2 " p1
  q :012
  >-10
  bez :3024
  bez :3204
  bez :3420
  bez :2034
  bez :0342
  q :314
def bez 4
  p4..6 p0.. .5 p1..
  p7..8 p4.. .5 p5..
  p9 p7 .5 p8
  bez :0479
  bez :9863
```



Level 20

```
srand 20
no edges
margin 20
p1 30 80
p2 120 80
p3 150 0
def sq 4
  p4..7 p0.. .4 p1..0
  p8 c
  p9..12 p0.. .6 p1..0
  p13..16 p4.. crand p9..
  p17..20 p0.. 1 p13.. right .4
  p21..24 p0.. 1 p13.. left .4
  p25..28 p17.. crand p21..
  sq p8,28,0,25
  sq p8,26,1,25
  sq p8,26,2,27
  sq p8,28,3,27
  >-6
  bez :0112
  bez :0332
def bez 4
  p4..6 p0.. .5 p1..
  p7..8 p4.. .5 p5..
  p9 p7 .5 p8
  bez :0479
  bez :9863
  draw :093 green
  draw :126874 blue
```



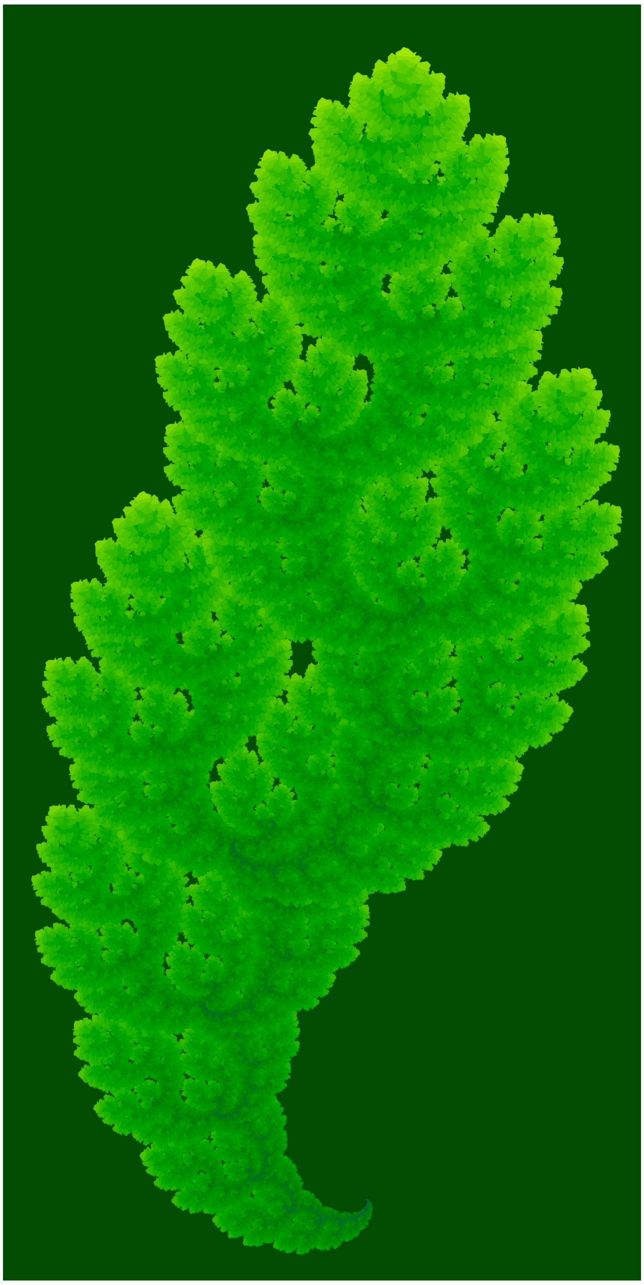
Levels 9 and 11

3. Fractals

IFS

```
no edges
background 0 .3 0
margin 30 10
pbox 60 180
def sq 4
  green
  p4 p0 .3 p3 right .05
  p5 p0 .6 p1 left .15
  p6 p0 .65 p1 right .1
  p7 p0 .9 p3 left .15
  p8 p0 .3 p1
  p9 p1 .5 p2
  p10 p2 .02 p3 left .15
  p11 p3 .3 p2 left .02
  sq p11,10,9,8 + .4 .3 -.5 flip
  sq p4,5,6,7 + -.5 -.1 .2
```

Level 16

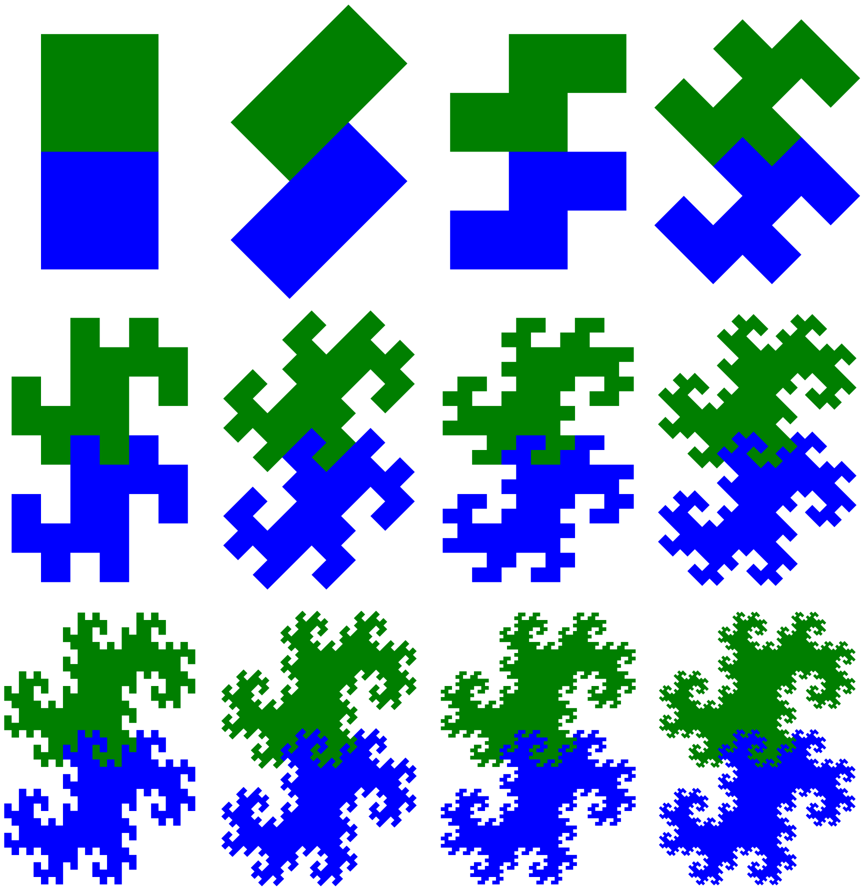


Self-replicating tiles

*** WHY are they called that?! Ordinary rep-tiles arguably might deserve that name, but these?!

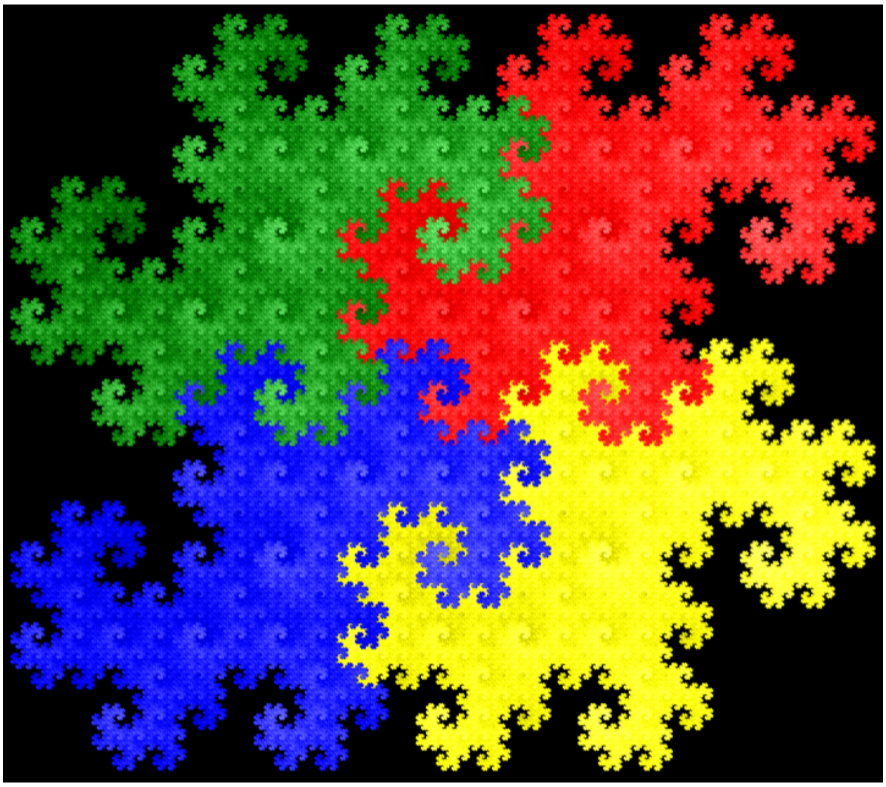
Twindragon

```
no edges
margin 20 13
pbox 50 100
def rect 4
  p4 p0 .5 p1
  p5 p2 .5 p3
  sq :0453 blue
  sq :4125 green
def sq 4
  p4 p1 .75 p0 right .25
  p5 p1 .75 p2 left .25
  p6 p2 .25 p3 left .25
  p7 p3 .75 p0 left .25
  p8..9 p1 1,3/4 p3
  sq :8569
  sq :4897
```



Levels 1-12

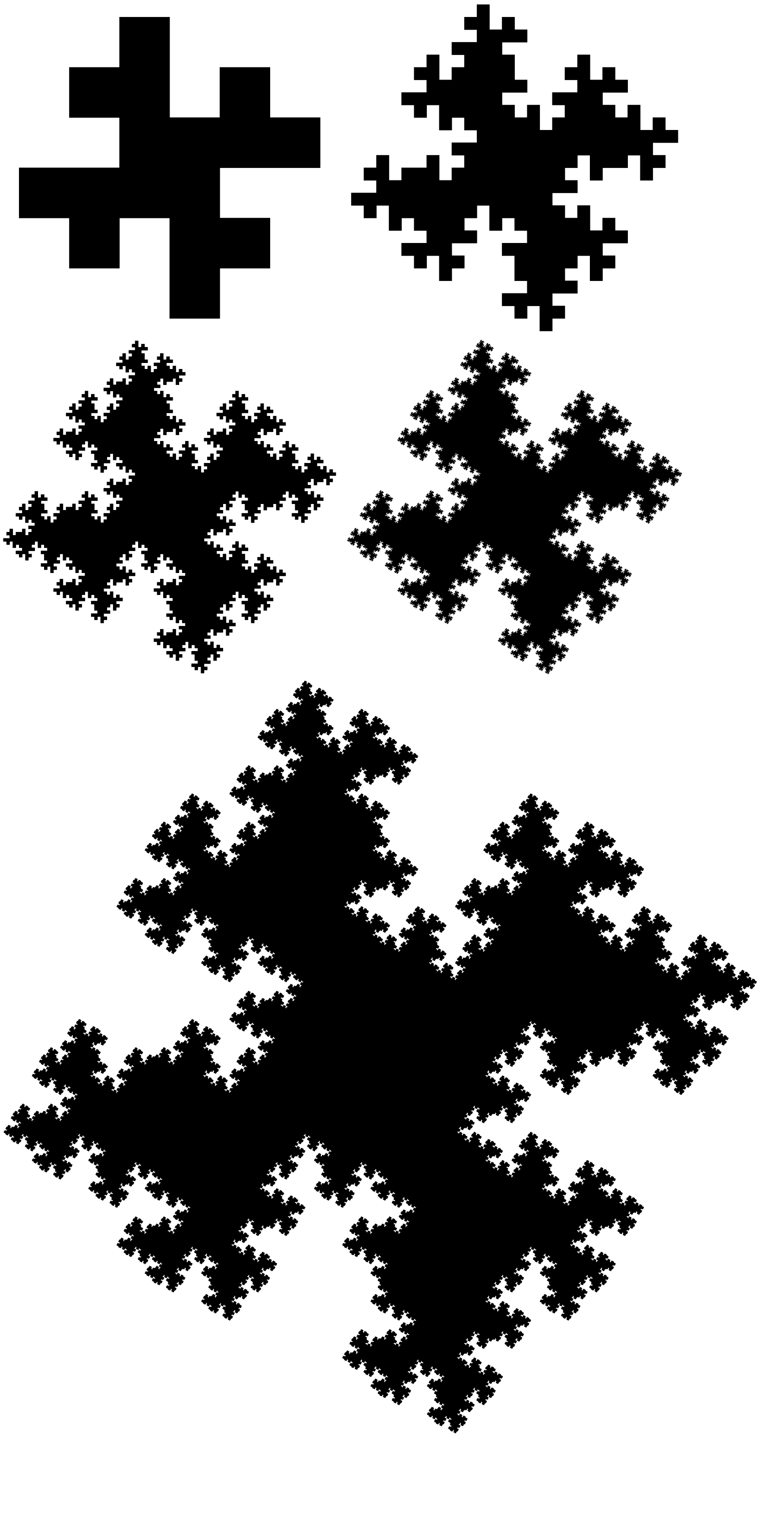
```
no edges
background black
margin 20 13
def init 4
  p4..7 p0.. .5 p1..0
  sq :415c green
  sq :04c7 blue
  sq :c526 red
  sq :7c63 yellow
def sq 4
  p4 p1 .75 p0 right .25
  p5 p1 .75 p2 left .25
  p6 p2 .25 p3 left .25
  p7 p3 .75 p0 left .25
  p8..9 p1 1,3/4 p3
  sq :8569 + .3
  sq :4897 - .3
```



Level 22

Square snowflake

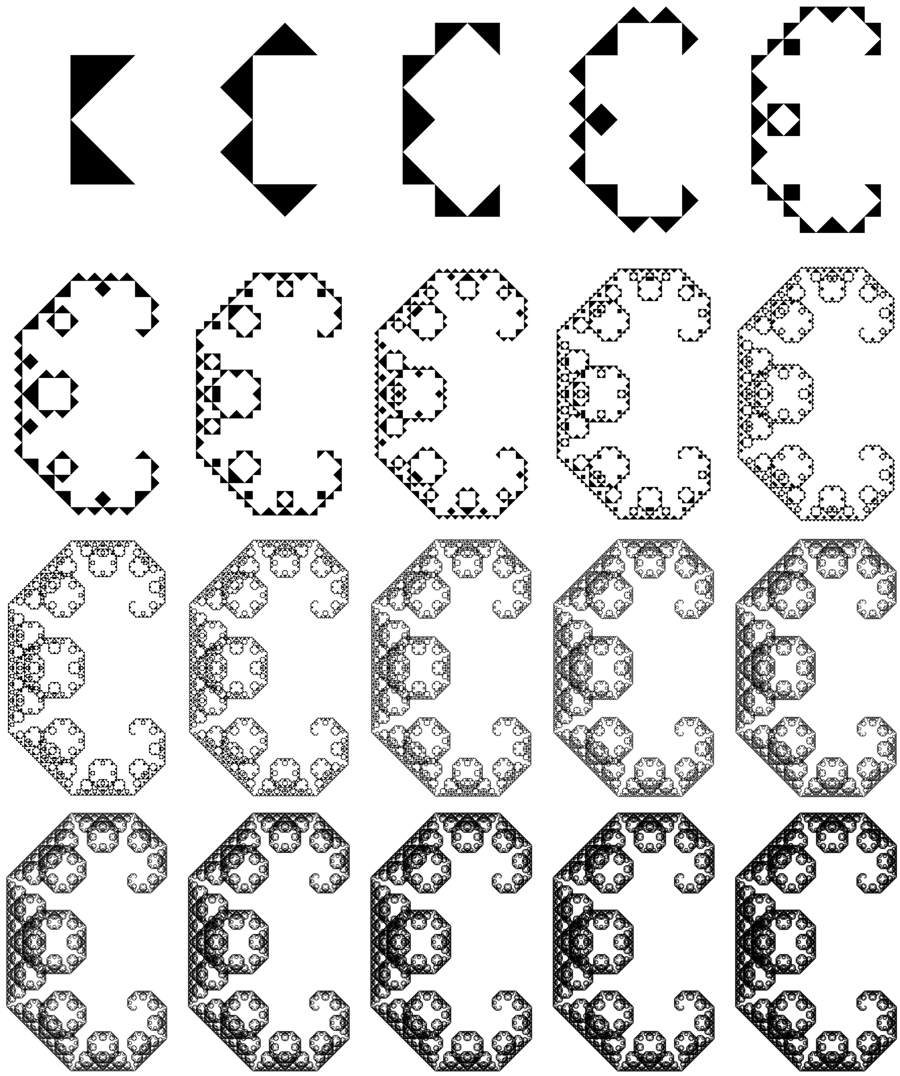
```
margin 34
def sq 4
  p4..7 p0.. 1/4 p1..0
  p8..11 p0.. 2/4 p1..0
  p12..15 p0.. 3/4 p1..0
  p16..18 p12 1..3/4 p6
  p19..21 p8 1..3/4 p10
  p22..24 p4 1..3/4 p14
  p25 p22 2 p4
  p26 p19 2 p8
  p27 p16 2 p5
  p28 p17 2 p9
  p29 p18 2 p6
  p30 p21 2 p10
  p31 p24 2 p7
  p32 p23 2 p11
  sq p5,27,28,9
  sq p12,1,5,16
  sq p16,5,9,17
  sq p18,13,2,6
  sq p19,16,17,20
  sq p20,17,18,21
  sq p21,18,6,10
  sq p10,6,29,30
  sq p25,26,8,4
  sq p4,8,19,22
  sq p22,19,20,23
  sq p23,20,21,24
  sq p0,4,22,15
  sq p11,23,24,7
  sq p7,24,14,3
  sq p32,11,7,31
```



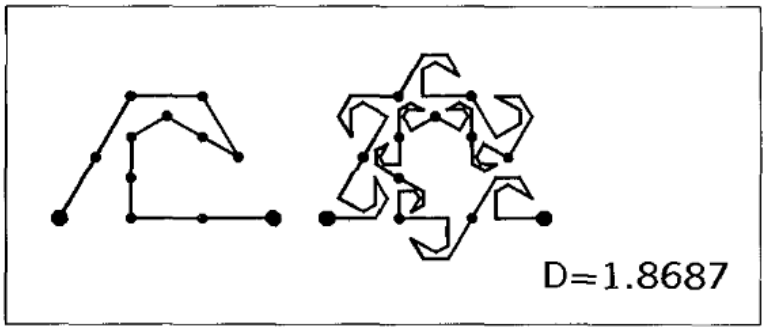
Classic fractals

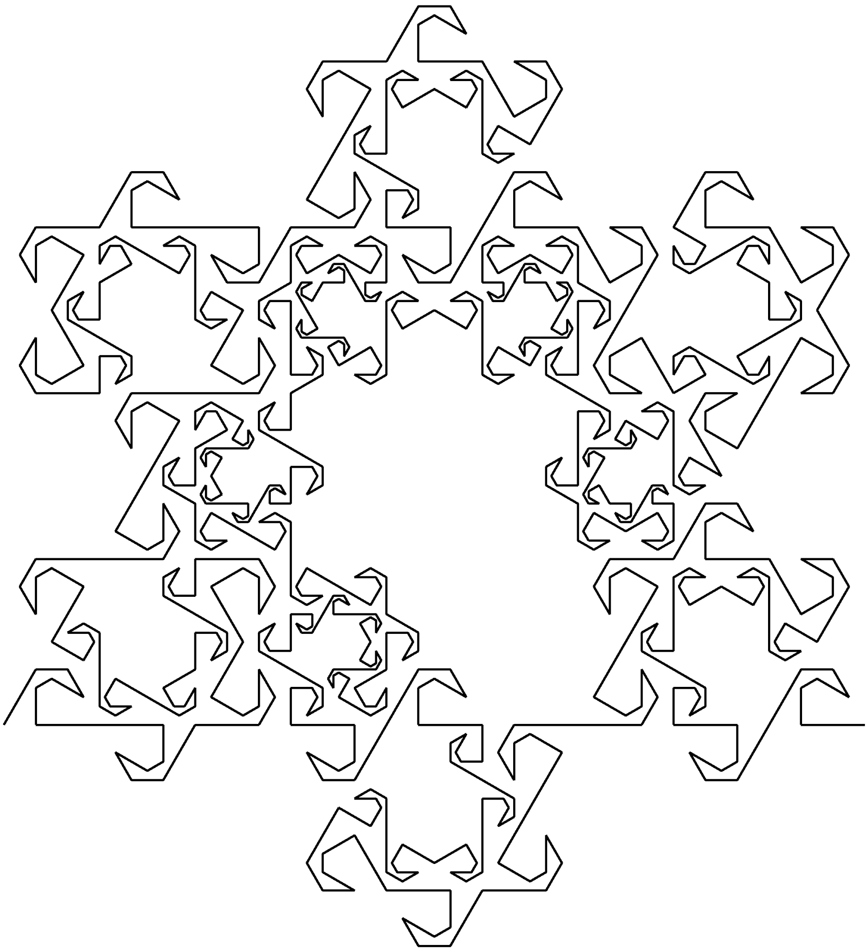
Lévy's dragon

```
margin 55 55 30 55
p1 -50 50
p2 0 100
def tri 3
  p3 p2 1 p0 right .5
  p4 p0 1 p2 left .5
  tri :142
  tri :031
```



```
width .2
margin 1 88 1 30
p1 100 0
def line 2
  p2..3 p0 1..2/3 p1
  p4 p0 1 p2 left 1.732
  p5 p2 1 p3 left 1.732
  p6..7 p2 1..2/3 p4
  p8 p0 2 p6
  p9 p4 .5 p8
  p10 p0 .5 p4
  p11 p9 2 p8
  line p0,10 flip
  line p4,10 flip
  line p4,5
  line p11,5 flip
  line p8,11 flip
  line p8,9 flip
  line p7,9
  line p6,7
  line p6,2
  line p3,2
  line p3,1
```





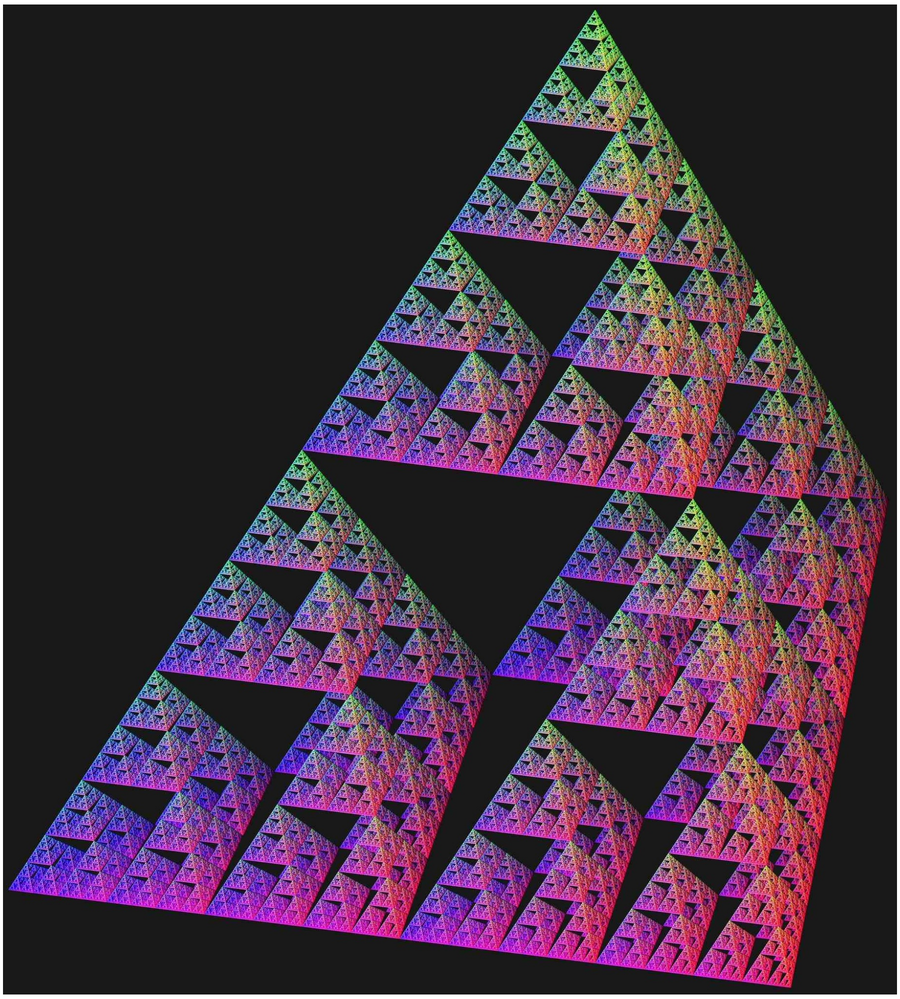
Level 3

But the Koch curve/snowflake is a better, cleaner, nicer-looking, more famous example of this...

3D fractals

Sierpinski tetrahedron

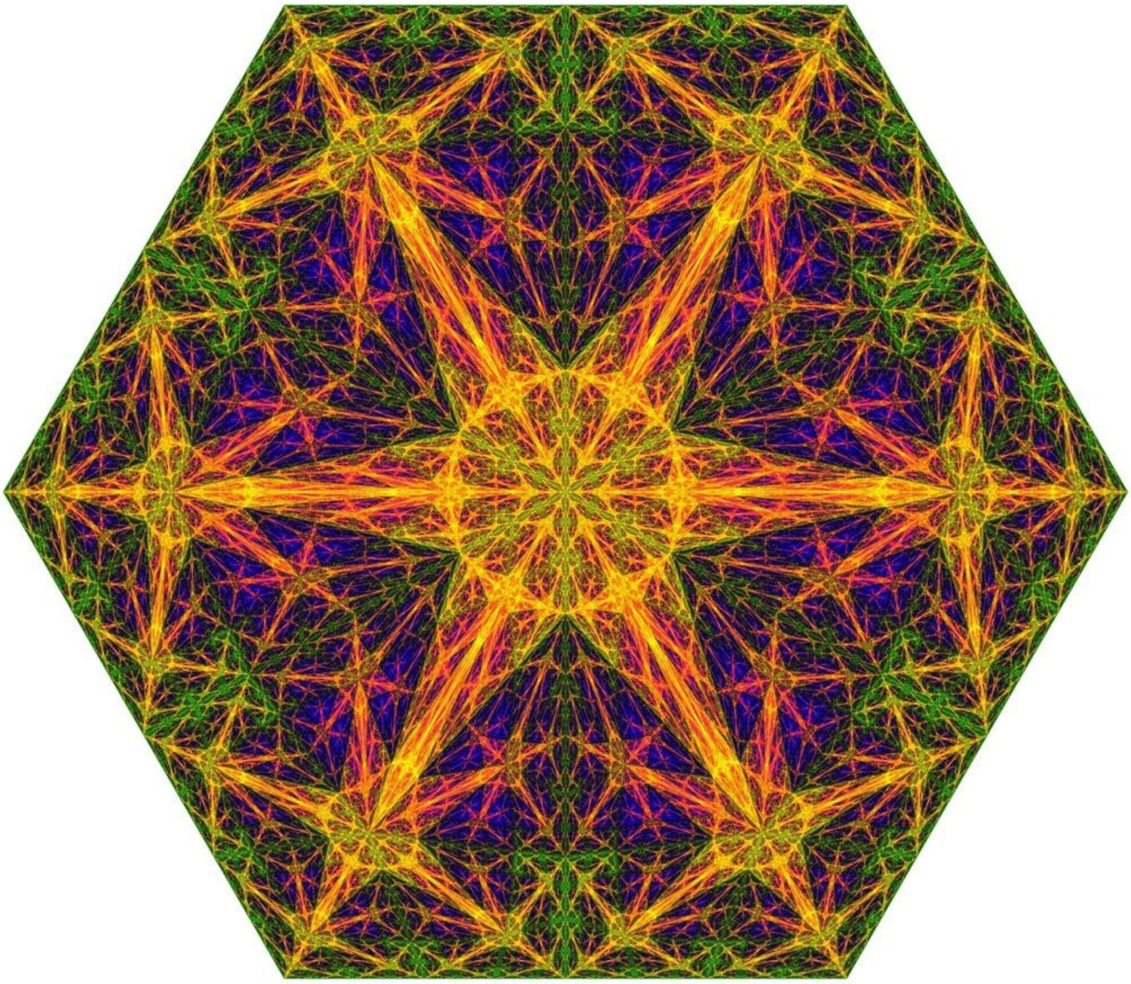
```
no edges
background .1
p1 90 40
p2 80 -10
p3 60 90
def pyr 4
  .7 .4 .7
  p4 p0 .5 p3
  p5 p1 .45 p3
  p6 p2 .5 p3
  p7 p0 .5 p2
  p8 p0 .55 p1
  p9 p1 .45 p2
  pyr :7926 - -1 .5 .5
  pyr :4563 - .5 -1 .5
  pyr :0874 - .5 .5 -1
  pyr :8195 - 1
  =-1
  draw :032 + 2
  draw :132 - 3
```



4. Gallery

A variant of a CFP basic hex division...from ratsub.pdf I think

```
noedges
width .02
pgon 6 50 0
def hex 6
  p6..11 p0.. .5 p1..0
  p12 c
  p13..18 p6.. .5 p12
  hex2 p0,11,18,12,13,6
  hex2 p1,6,13,12,14,7
  hex2 p2,7,14,12,15,8
  hex2 p3,8,15,12,16,9
  hex2 p4,9,16,12,17,10
  hex2 p5,10,17,12,18,11
def hex2 6
  p6..11 p0.. .5 p1..0
  p12 p0 .3 p3
  p13..18 p6.. .5 p12
  hex2 p0,11,18,12,13,6 + -1 .2 -1
  hex2 p1,6,13,12,14,7 + -1 .2 -1
  hex2 p2,7,14,12,15,8 + -2 -2 .5
  hex2 p3,8,15,12,16,9 + 7 1.5 -1
  hex2 p4,9,16,12,17,10 + -2 -2 .5
  hex2 p5,10,17,12,18,11 + -1 .2 -1
```

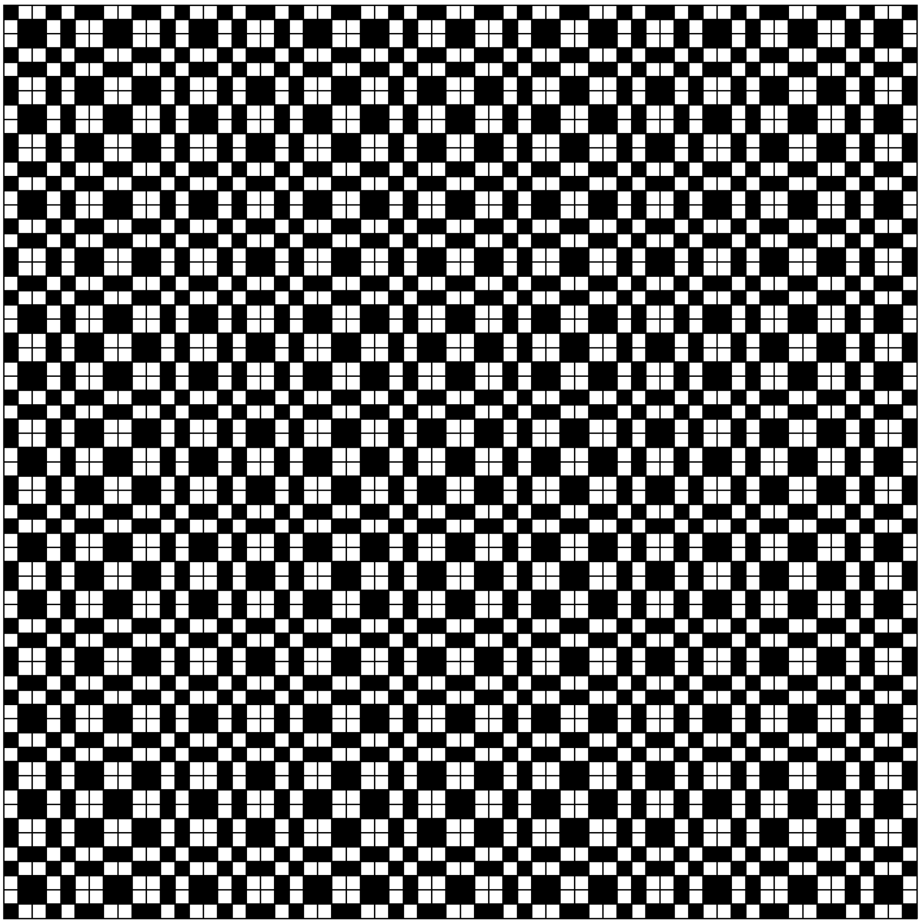


Level 9

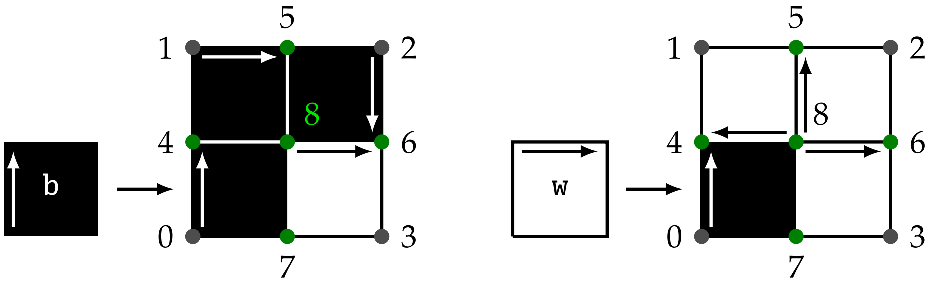
Thue-Morse tilings

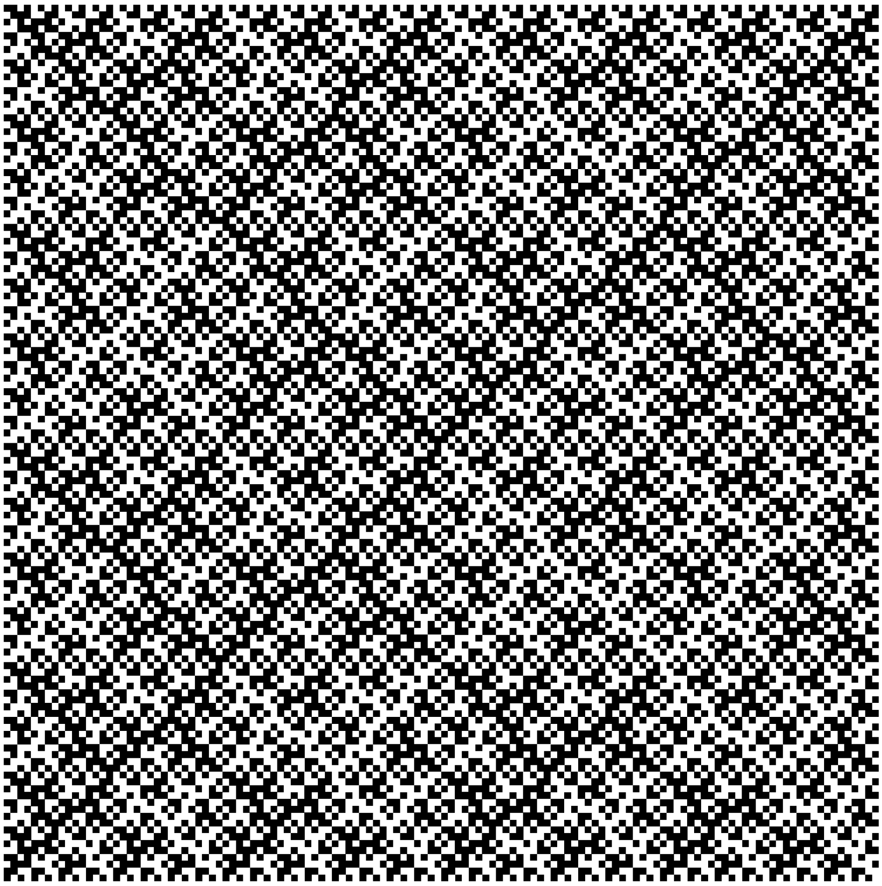
```
def t 4
  p4..7 p0.. .5 p1..0
  t :415c
  t :c526 !
  t :04c7 !
  t :7c63
```

Level 6



Boot tiling





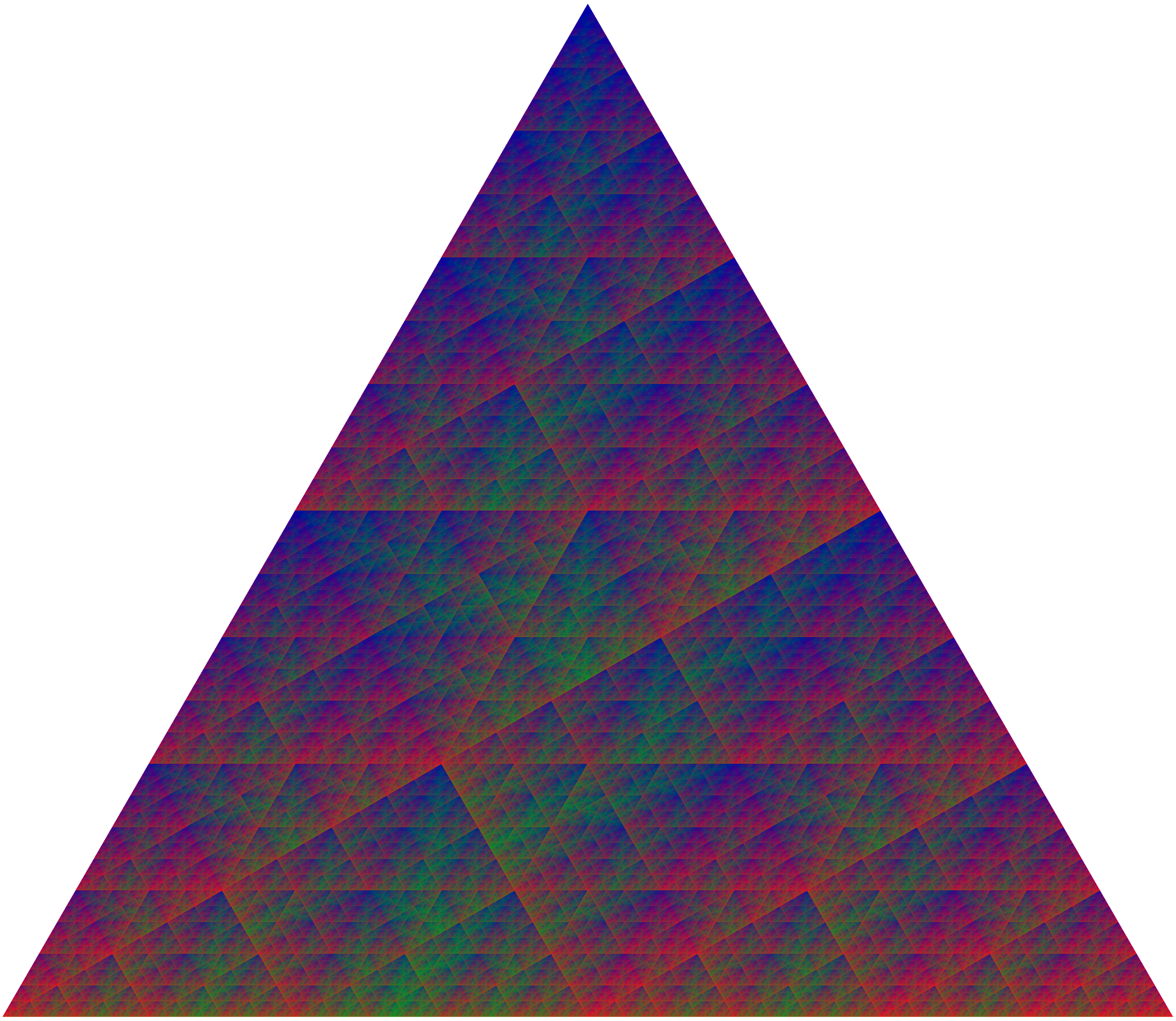
```
no edges
def b 4
  black
  p4..7 p0.. .5 p1..0
  b :15c4
  b :26c5
  b :04c7
  w :7c63
def w 4
  white
  p4..7 p0.. .5 p1..0
  w :5c41
  w :6c52
  b :04c7
  w :7c63
```

Level 8

NB!! The white tiles dont correspond to what the program does, must be rotated 1. so change the program or the diagram...

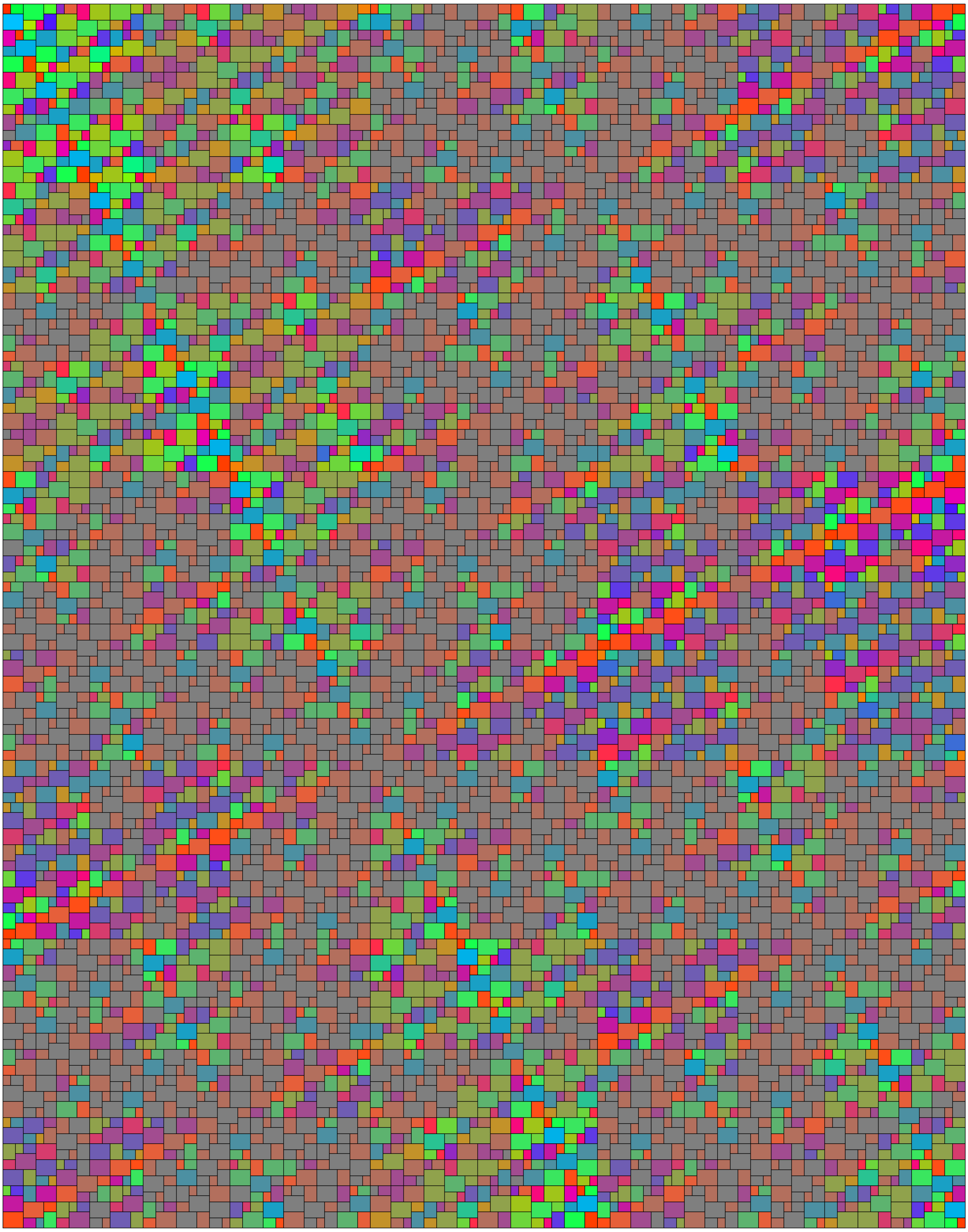
```
no edges
pgon 3 50 90
def tri 3
  black
  p3..4 p0.. .5 p2
  tri :031 + 0 0 .5
  tri :341 + 0 .6 0
  tri :342 + 1 0 0
```

Level 13



```
pbox 261.803 333.019
def rect 4
  p4 p0 0.618 p1
  p5 p3 0.618 p2
  p6 p1 0.382 p2
  p7 p4 0.382 p5
  rect :4167 - -2 .64 1.36 wait 3
  rect :6257 rot wait 2
  rect :4530 !
```

Level 17

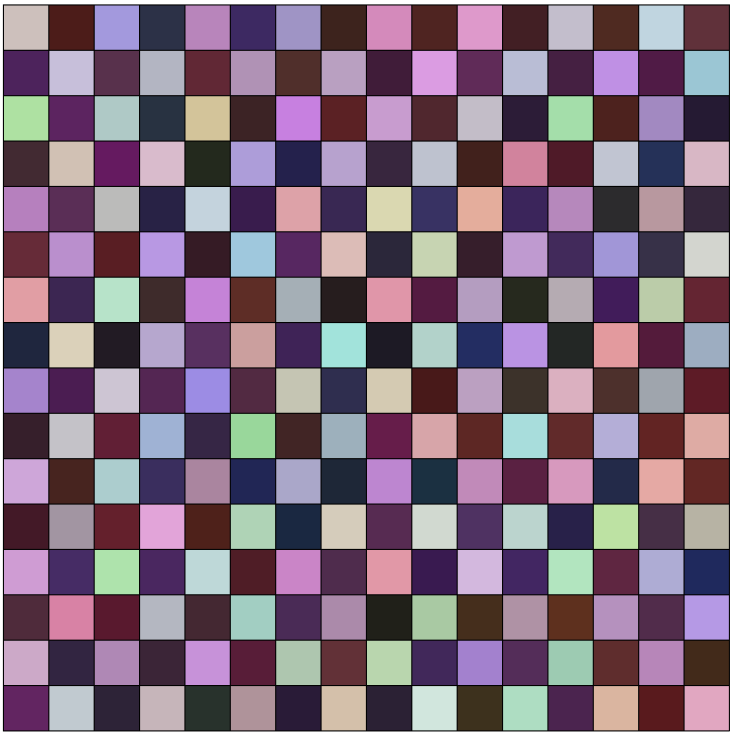


5. Showing use of commands

Random colour range

```
def sq 4
  p4..7 p0.. .5 p1..0
  sq1 :415c
  sq2 :c526
  sq2 :04c7
  sq1 :7c63
def sq1 4
  rand .6 .9 .5 .9 .6 .9
  sq :0123
def sq2 4
  rand .1 .4 .1 .2 .1 .4
  sq :0123
```

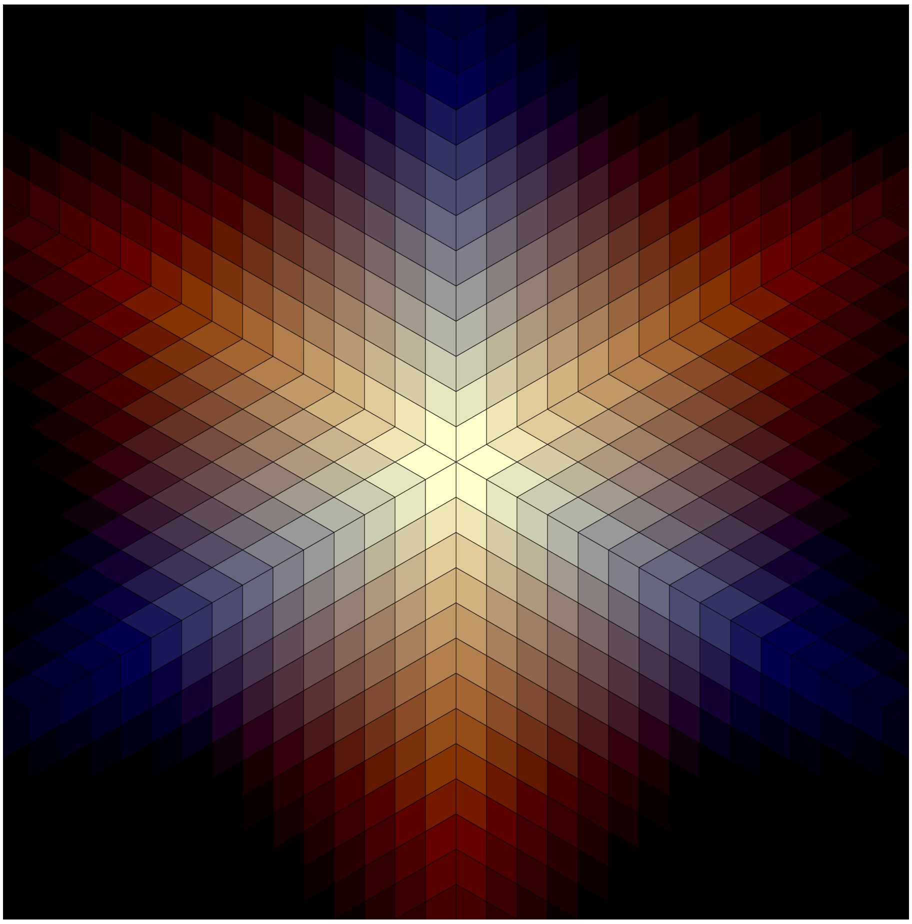
Level 8



par

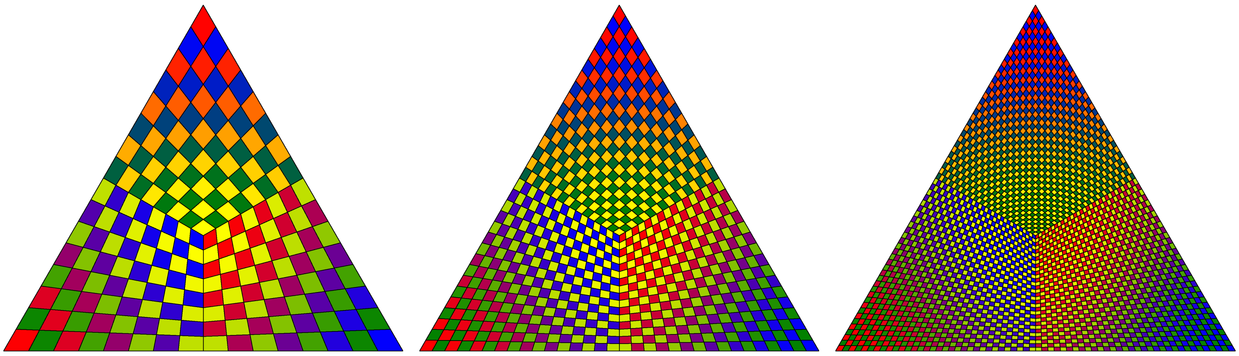
```
margin 120
pgon 6 10 30
def sh 6
  1 1 .8
  p6 c
  third :6012
  third :6234
  third :6450
def third 4
  double :021
  double :023
def double 3
  p3 par p0 p1 p2
  p4..5 p0 2 p1..
  double :235 - .6 1 1
  single :135 - 1 1 .5
  draw :0132
def single 3
  p3 par p0 p1 p2
  p4 p0 2 p2
  single :234 - 1 1 .5
  draw :0132
```

Level 23



6. Misc

```
pgon 3 50 90
p3 0 0
def tri 3
  p3..5 p0.. .5 p1..0
  sq1 :305c
  sq2 :425c
  sq3 :413c
def sq1 4
  p4..7 p0.. .5 p1..0
  ip3 yellow ip0 red
  sq1 :415c
  sq1a :c526
  sq1a :04c7
  sq1 :7c63
def sq1a 4
  p4..7 p0.. .5 p1..0
  ip3 green ip0 blue
  sq1 :415c
  sq1a :c526
  sq1a :04c7
  sq1 :7c63
def sq2 4
  p4..7 p0.. .5 p1..0
  ip3 red ip2 blue
  sq2 :415c
  sq2a :c526
  sq2a :04c7
  sq2 :7c63
def sq2a 4
  p4..7 p0.. .5 p1..0
  ip3 yellow ip2 green
  sq2 :415c
  sq2a :c526
  sq2a :04c7
  sq2 :7c63
def sq3 4
  p4..7 p0.. .5 p1..0
  ip3 blue ip1 red
  sq3 :415c
  sq3a :c526
  sq3a :04c7
  sq3 :7c63
def sq3a 4
  p4..7 p0.. .5 p1..0
  ip3 yellow ip1 green
  sq3 :415c
  sq3a :c526
  sq3a :04c7
  sq3 :7c63
```



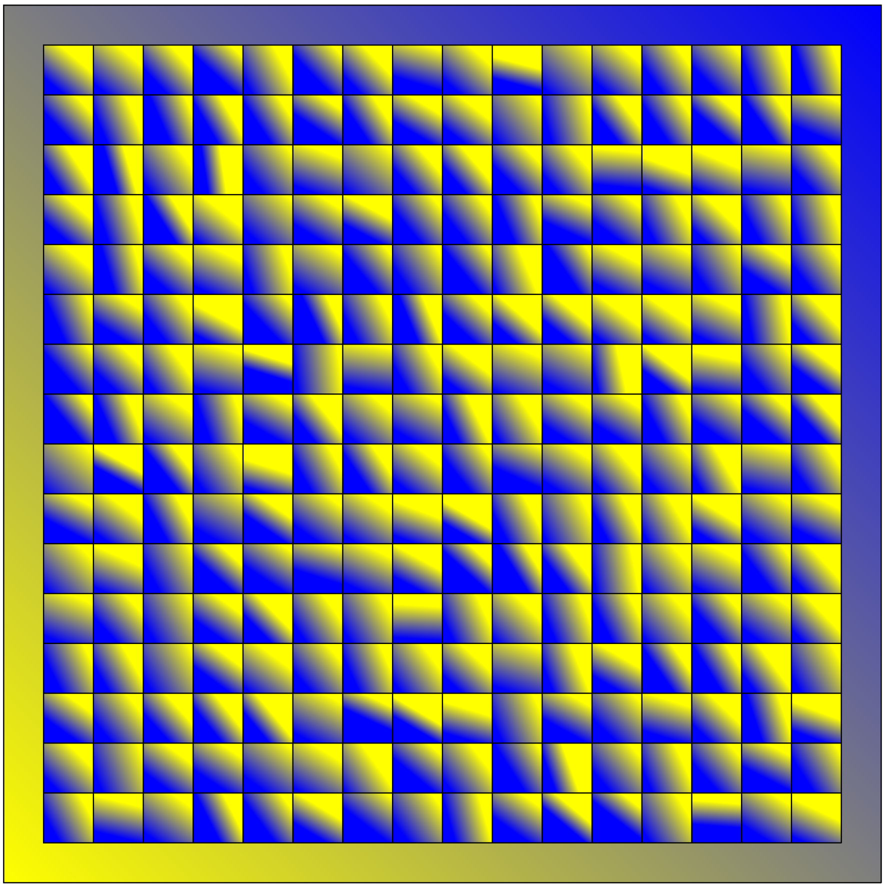
Levels 4, 5, 6.

7. Good bugs

This was supposed to be a shortened version of the 'random 4gons inside a grid of squares'.. but `randquad` is drawn with points 0,1,2,3, not the 4,5,6,7 making the random shapes! Whoops. But the shading works with the old points.. for an effect of random shading in squares. I like it :-)

```
margin 6
def frame 4
  p4 c
  p5..8 p4 1.1 p0..
  sq :0123
  backgr :5678
def backgr 4
  p0 yellow p2 blue
def sq 4
  p4 c
  p5..8 p0.. .5 p1..0
  sq :5164
  sq :4627
  sq :0548
  sq :8473
  =-2
  randquad :0123
def randquad 4
  p4..7 p0.. rand p1..0
  p8 p7 .5 p4
  p9 p5 .5 p6
  p8 blue p9 yellow
```

Level 7



8. More tilings..

Scherer's L

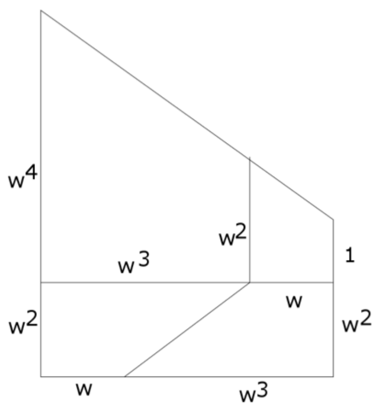
Figure 8.1 is a version of a tiling in Scherer's irreptile book, according to Michael Reid in *Tiling With Similar Polyominoes*. Well, that is a tiling of a rectangle by six hexominos. This is a rectangle divided into 6 Ls, each of which is divided into 3 rectangles, etc. I like it how the tiling is alternately all rectangle then all L each generation.

Vince-Barnsley paper

i.e. *Self-Similar Polygonal Tiling*

Trapezoids

The trapezoid tiling is in Scherer's book. It's Vince and Barnsley's example 5.2.



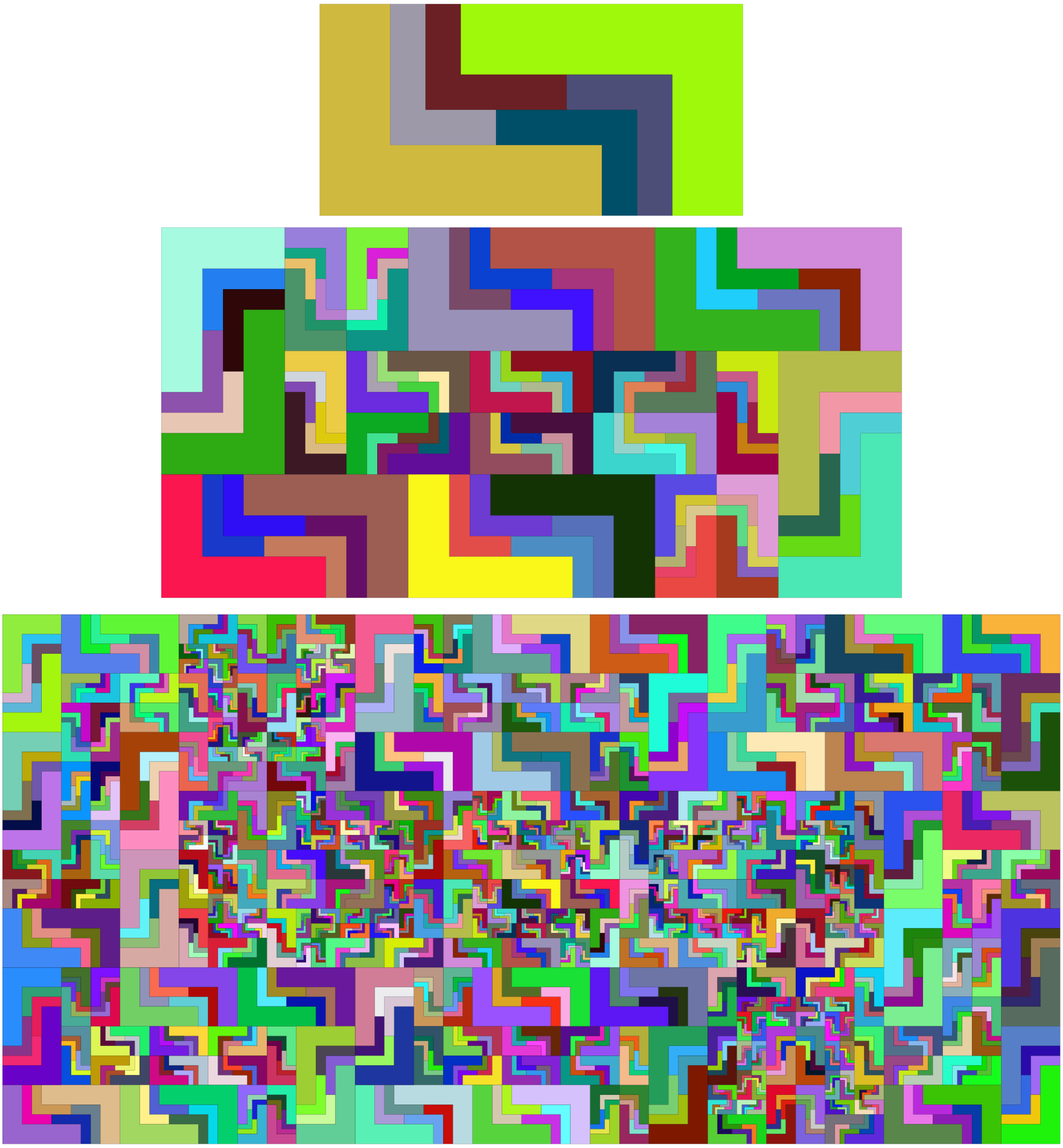
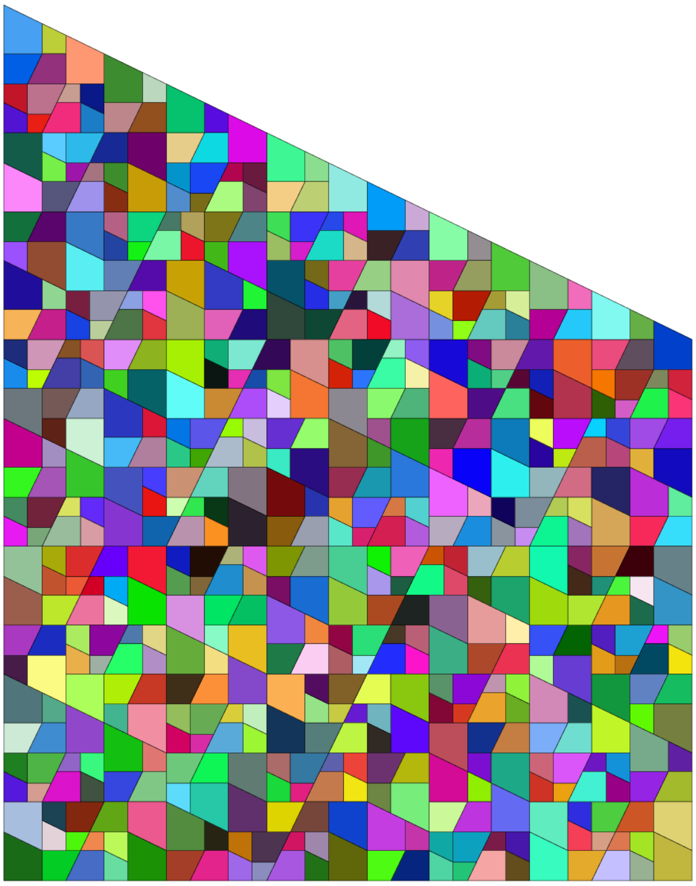


Figure 8.1: Levels 1,3,5 of Scherer's L.

```
margin 0.5
p0 0 0
p1 0 42.36
p2 33.3019 26.18
p3 " 0
def trap 4
  rand
  p4 p0 0.38197 p1
  p5 p2 " p3
  p6 p5 " p4
  p7 p2 " p1
  p8 p0 " p3
  w :4176
  w2 :4680
  w2 :3865
  w3 :6725
def w3 4
  rand
  w2 :0123
def w2 4
  rand
  w :0123
def w 4
  rand
  trap :0123
```

Level 12:

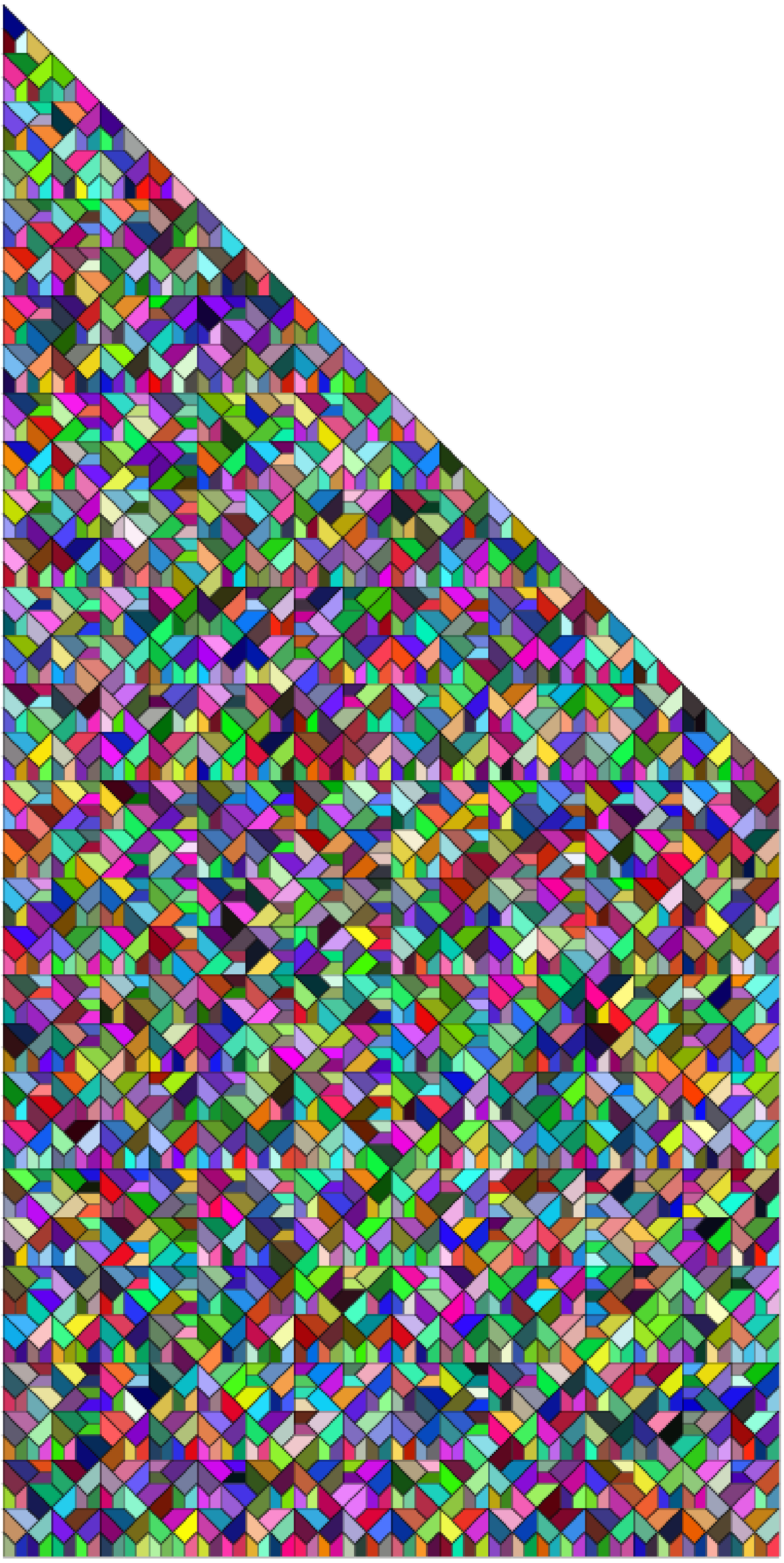


Version with the `wait` command:

C

```
width 0.001
p0 0 0
p1 0 40
p2 20 20
p3 20 0
def trap 4
  rand
  p4 p0 1/2 p1
  p5 p0 " p3
  p6 p0 " p2
  trap :0465
      wait
  trap :3265
      wait
  trap :2146
```

levels 11



Notes and sources

p3. Minsky circles: See HAKMEM, *Minskys & Trinskys* etc

p9. Idea from Barnsley, *Superfractals*, p129.

p9. Twin twindragon construction from Jack Giles Jr, *Construction of Replicating Super-figures*, 1977, Figure 1. ”I have recently learned that this same rep-2 figure was discovered independently by Robert Ammann”

p11. Square snowflake from Mandelbrot, *The Fractal Geometry of Nature* p50 "A quadric Koch island"

p13. Generator picture from Mandelbrot, *The Fractal Geometry of Nature*, 1982. *Monkeys tree* is rounded version of this; Mandelbrot calls level 4 *Split Snowflake Halls* and has a very fanciful story on p146.

p16. Mapping from Joan Taylor `http://taylortiling.com/square.html`

p18. from Joan Taylor `http://taylortiling.com/d15boottiling.pdf`