pst-fractal

Plotting fractals; v.0.11a

Herbert Voß

March 16, 2020

Contents

The well known `pstricks` package offers excellent macros to insert more or less complex graphics into a document. `pstricks` itself is the base for several other additional packages, which are mostly named `pst-xxxx`, like `pst-fractal`.

This version uses the extended keyval package `xkeyval`, so be sure that you have installed this package together with the spcecial one `pst-xkey` for PSTricks. The `xkeyval` package is available at `CTAN:/macros/latex/contrib/xkeyval/`. It is also important that after `pst-fractal` no package is loaded, which uses the old keyval interface.

The fractals are really big, which is the reason why this document is about 15 MByte when you run it without using the external png-images.

All images in this documentation were converted to the `.jpg` format to get a small pdf file size. When using the pdf format for the images the file size will be more than 20 MBytes. However, having a small file size will lead into a bad image resolution. Run the examples as single documents to see how it will be in high quality.

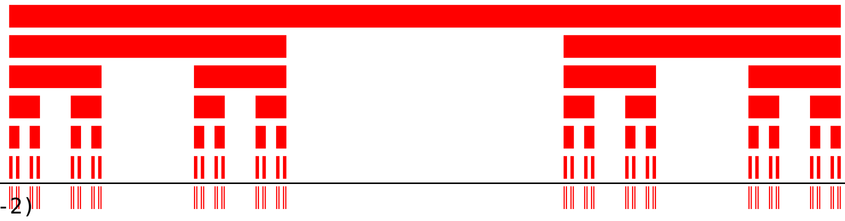
1 Cantor set

The set is always plotted from the origin down to into negative y values.

```
\psCantor [Options]
```

Possible optional arguments are `linewidth`, `linecolor`, `n` (recursion depth), `xWidth`, and `yWidth` (vertical increment). The defaults are 2mm, black, 5, 10cm, and 5mm.

```
\begin{pspicture}(10,-2)
  \psCantor
\end{pspicture}
```



```
\begin{pspicture}(10,-2)
  \psCantor[linewidth=3mm,linecolor=red,n=7,xWidth=11,yWidth=4mm]
\end{pspicture}
```

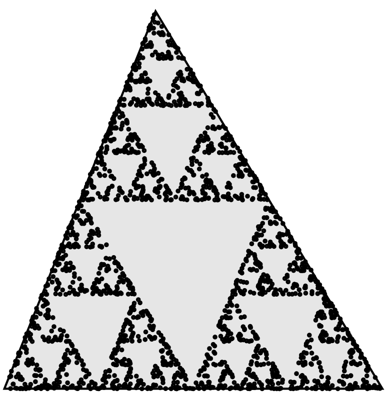
2 Sierpinski triangle and curve

The triangle must be given by three mandatory arguments. Depending to the kind of arguments it is one of the two possible versions:

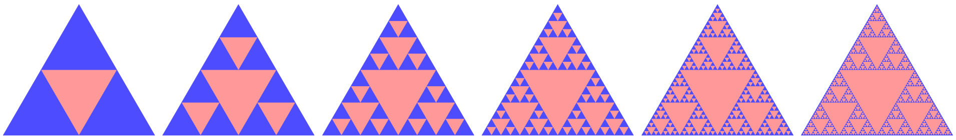
\psSier [Options] $(x_0,y_0)(x_1,y_1)(x_2,y_2)$
\psSier [Options] (x_0,y_0)\{*Base*\}\{*Recursion*\}
\psSier [Options]

2.1 Triangle

In difference to `\psfractal` it doesn't reserve any space, this is the reason why it should be part of a `pspicture` environment.



```
\begin{pspicture}(5,5)
  \psSier(0,0)(2,5)(5,0)
\end{pspicture}
```

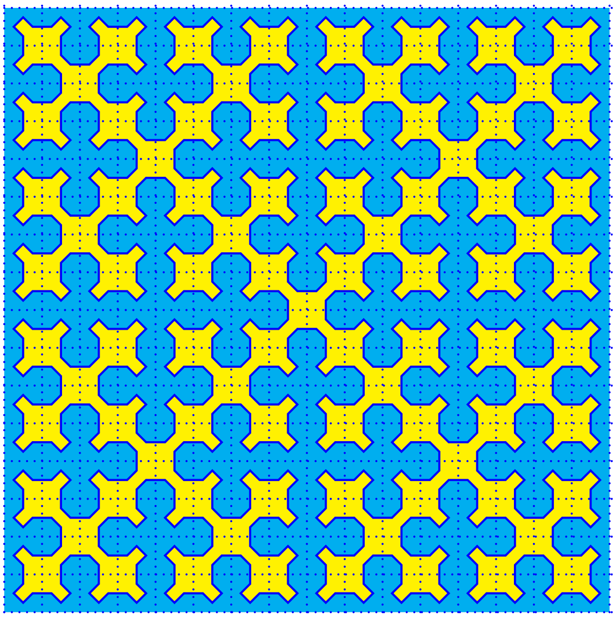


```
\multido{\iA=1+1}{6}{%
\begin{pspicture}(2,1.7)
\psSier[linecolor=blue!70,
  fillcolor=red!40](0,0){2cm}{\iA}
\end{pspicture} }
```

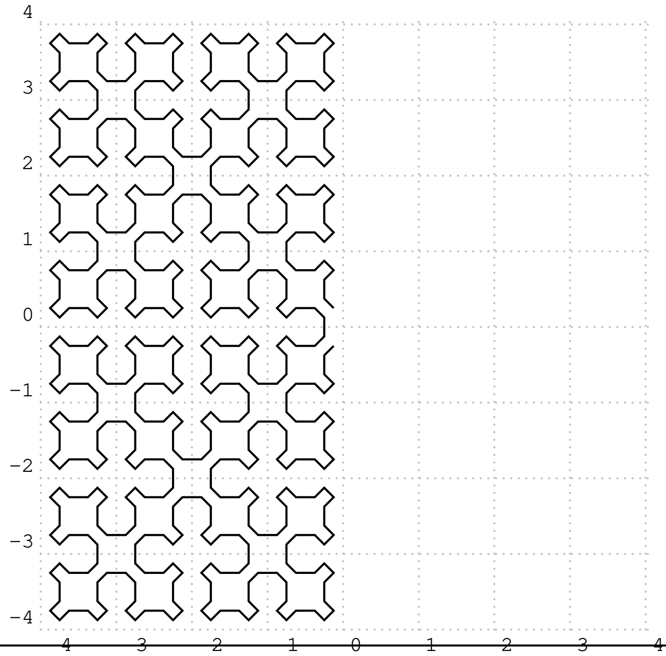
2.2 Curve

There are four special optional arguments for the Siepinski curve:

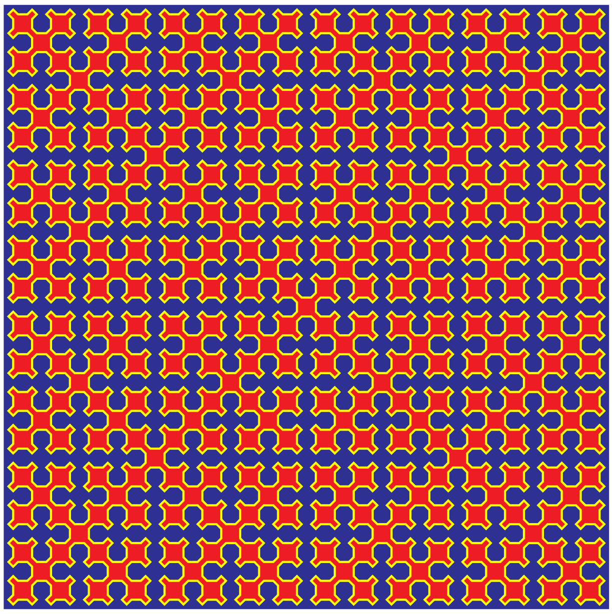
- [n=4] : number of iterations
- [N=all] : number of placed points (only of interest for an animation)
- [dotcolor=red] : in difference to linecolor for standard PSTricks
- [showpoints=false] : show calculated points



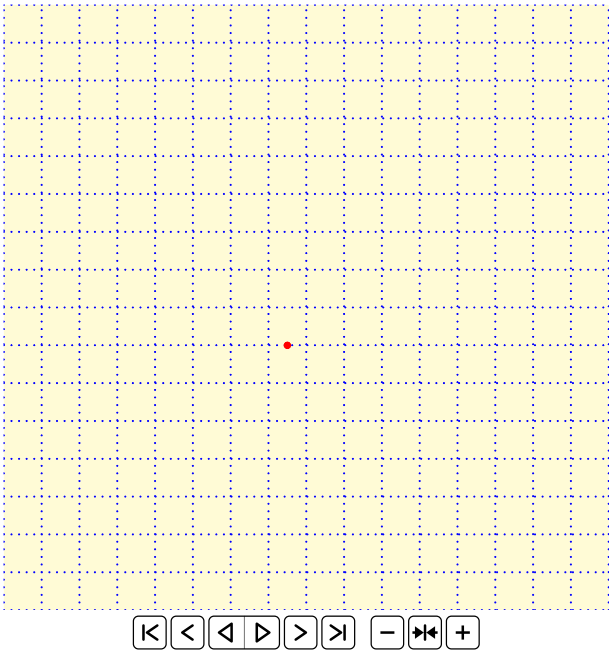
```
\begin{pspicture}(-4,-4)(4,4)
\psframe*[linecolor=cyan](-4,-4)(4,4)
\psSier[unit=0.25,n=4,fillstyle=solid,fillcolor=yellow,linecolor=blue]
\psgrid[subgriddiv=0,gridcolor=blue,griddots=5,gridlabels=0pt,unit=0.5](-8,-8)(8,8)
\end{pspicture}
```




```
\begin{pspicture}[showgrid](-4,-4)(4,4)
\psset{unit=0.25}
% n=4 => Nmax=4^(n+1)=1024
% ici on marque la moitie des points
\psSier[n=4,N=512]
\end{pspicture}
```



```
\begin{pspicture}(-4,-4)(4,4)
\psframe*[linecolor=-yellow](-4,-4)(4,4)
\psSier[n=5,unit=0.125,fillstyle=solid,fillcolor=-cyan,linecolor=-blue]
\end{pspicture}
```



```
\begin{animateinline}[controls,% palindrome,
                      begin={\begin{pspicture}(-4,-4)(4,4)},
                      end={\end{pspicture}}]{5}% 5 image/s
\multiframe{256}{i=1+1}{%
\psframe*[linecolor=yellow!20](-4,-4)(4,4)
\psgrid[subgriddiv=0,gridcolor=blue,griddots=5,gridlabels=0pt,unit=0.5](-8,-8)(8,8)
% n=3 => Nmax=4^(n+1)=256 points
\psSierpinskyCurve[linecolor=blue,linewidth=0.05,n=3,showpoints,dotsize=0.1,N=\i,unit=0.5]}
\end{animateinline}
```

3 Julia and Mandelbrot sets

The syntax of the `\psfractal` macro is simple

`\psfractal [Options]` $(x_0,y_0)(x_1,y_1)$

All Arguments are optional, `\psfractal` is the same as `\psfractal(-1,-1)(1,1)`. The Julia and Mandelbrot sets are a graphical representation of the following sequence x is the real and y the imaginary part of the complex number z. $C(x,y)$ is a complex constant and preset by $(0,0)$.

$$z_{n+1}(x,y) = (z_n(x,y))^2 + C(x,y) \tag{1}$$

3.1 Julia sets

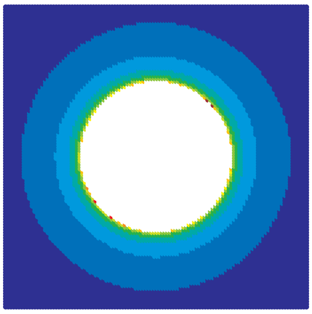
A Julia set is given with

$$z_{n+1}(x,y) = (z_n(x,y))^2 + C(x,y) \tag{2}$$

$$z_0 = (x_0; y_0) \tag{3}$$

$(x_0; y_0)$ is the starting value.

```
\psfractal
```



```
\psfractal[xWidth=4cm,yWidth=4cm, baseColor=white
    , dIter=20](-2,-2)(2,2)
```

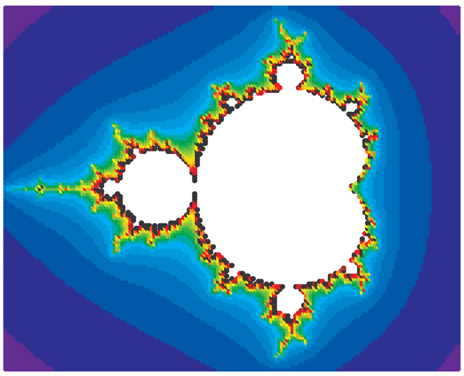
3.2 Mandelbrot sets

A Mandelbrot set is given with

$$z_{n+1}(x,y) = (z_n(x,y))^2 + C(x,y) \quad (4)$$
$$z_0 = (0;0) \quad (5)$$
$$C(x,y) = (x_0;y_0) \quad (6)$$

$(x_0;y_0)$ is the starting value.

```
\psfractal[type=Mandel]
```



```
\psfractal[type=Mandel, xWidth=6cm,
  yWidth=4.8cm, baseColor=white,
  dIter=10](-2,-1.2)(1,1.2)
```

3.3 The options

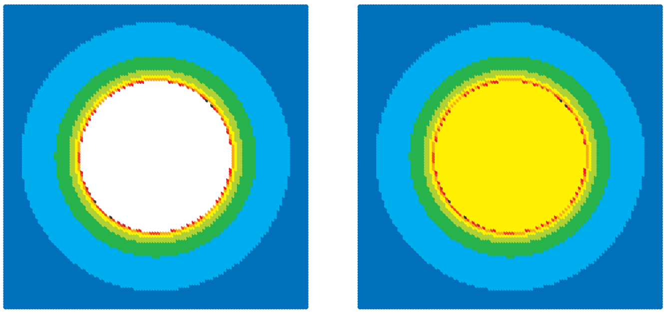
3.4 type

`txpe` can be of `Julia` (default) or `Mandel`.


```
\psfractal \qquad
\psfractal[type=Mandel]
```

3.5 baseColor

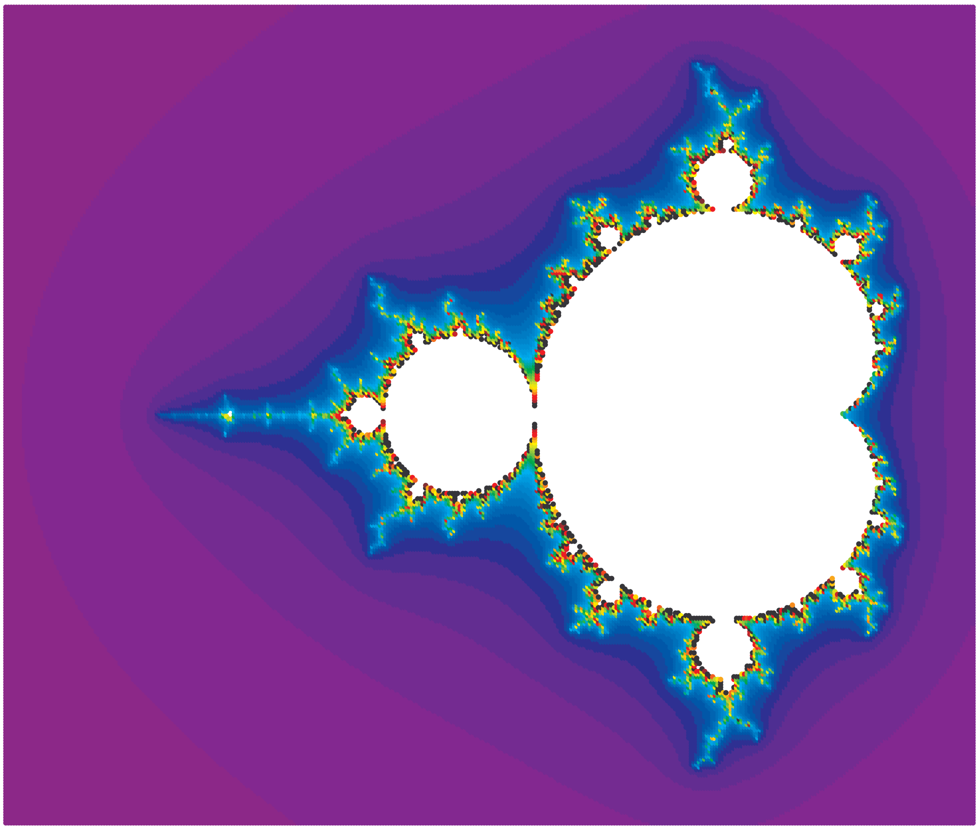
The color for the convergent part is set by `baseColor`.



```
\psfractal[xWidth=4cm,yWidth=4cm,dIter=30](-2,-2)(2,2) \qquad
\psfractal[xWidth=4cm,yWidth=4cm,baseColor=yellow,dIter=30](-2,-2)(2,2)
```

3.6 xWidth and yWidth

xWidth and yWidth define the physical width of the fractal.



```
\psfractal[type=Mandel,xWidth=12.8cm,yWidth=10.8cm,dIter=5](-2.5,-1.3)(0.7,1.3)
```

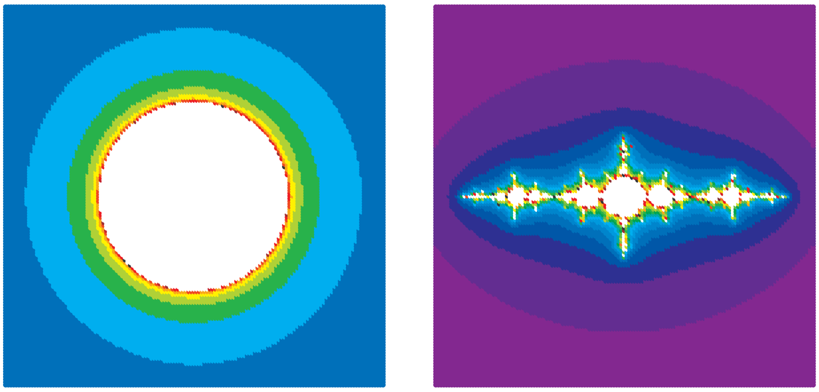
3.7 cx and cy

Define the starting value for the complex constant number C.

```
\psset{xWidth=5cm,yWidth=5cm}
\psfractal[dIter=2](-2,-2)(2,2) \qquad
\psfractal[dIter=2,cx=-1.3,cy=0](-2,-2)(2,2)
```

3.8 dIter

The color is set by wavelength to RGB conversion of the iteration number, where `dIter` is the step, predefined by 1. The wavelength is given by the value of `iter` added by 400.



```
\psset{xWidth=5cm,yWidth=5cm}
\psfractal[dIter=30](-2,-2)(2,2) \qquad
\psfractal[dIter=10,cx=-1.3,cy=0](-2,-2)(2,2)
```

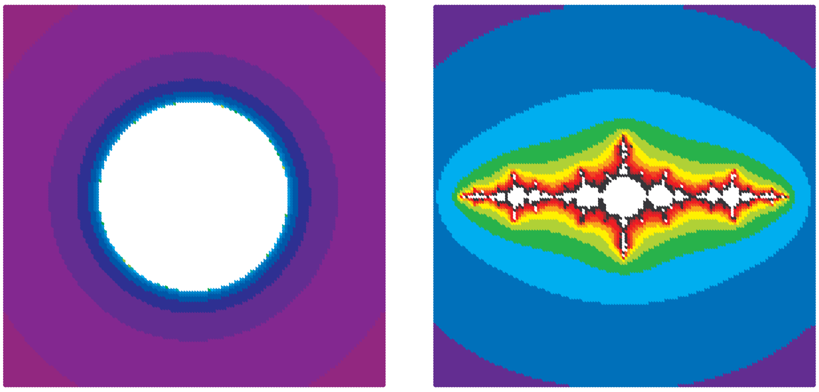
3.9 maxIter

`maxIter` is the number of the maximum iteration until it leaves the loop. It is predefined by 255, but internally multiplied by `dIter`.

```
\psset{xWidth=5cm,yWidth=5cm}
\psfractal[maxIter=50,dIter=3](-2,-2)(2,2) \qquad
\psfractal[maxIter=30,cx=-1.3,cy=0](-2,-2)(2,2)
```

3.10 maxRadius

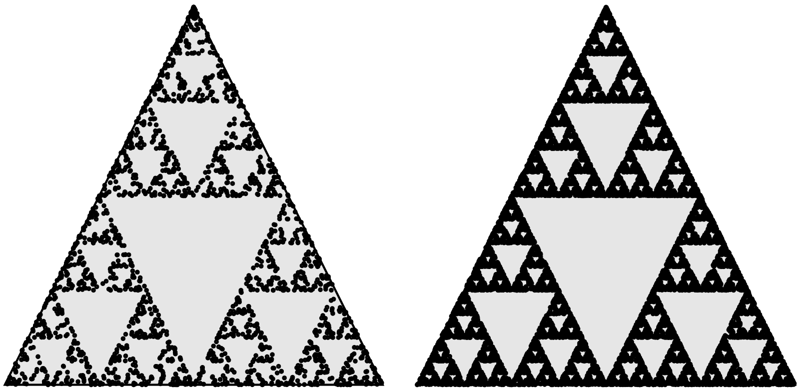
If the square of distance of z_n to the origin of the complex coordinate system is greater as `maxRadius` then the algorithm leaves the loop and sets the point. `maxRadius` should always be the square of the "‘real"’ value, it is preset by 100.



```
\psset{xWidth=5cm,yWidth=5cm}
\psfractal[maxRadius=30,dIter=10](-2,-2)(2,2) \qquad
\psfractal[maxRadius=30,dIter=30,cx=-1.3,cy=0](-2,-2)(2,2)
```

3.11 plotpoints

This option is only valid for the Sierpinski triangle and preset by 2000.



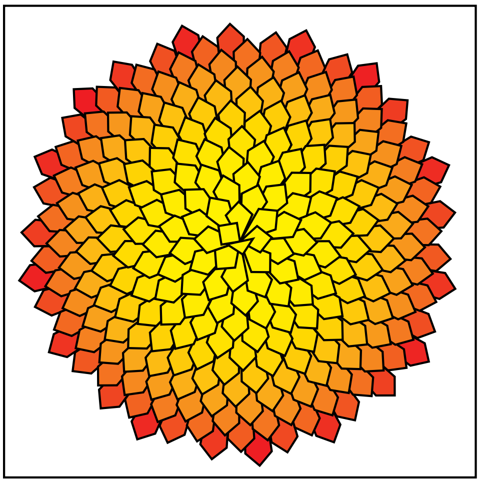
```
\begin{pspicture}(5,5)
  \psSier(0,0)(2.5,5)(5,0)
\end{pspicture} \quad
\begin{pspicture}(5,5)
  \psSier[plotpoints=10000](0,0)(2.5,5)(5,0)
\end{pspicture}
```

4 Phyllotaxis

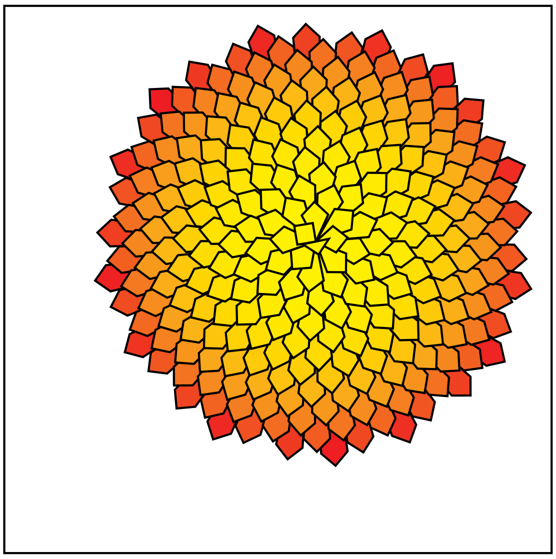
The beautiful arrangement of leaves in some plants, called phyllotaxis, obeys a number of subtle mathematical relationships. For instance, the florets in the head of a sunflower form two oppositely directed spirals: 55 of them clockwise and 34 counterclockwise. Surprisingly, these numbers are consecutive Fibonacci numbers. The Phyllotaxis is like a Lindenmayer system.

\psPhyllotaxis [Options] (x,y)

The coordinates of the center are optional, if they are missing, then $(0,0)$ is assumed.

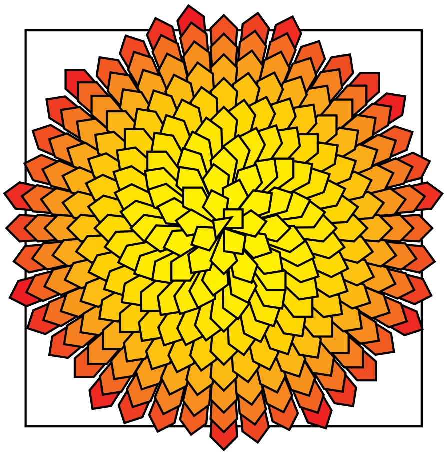


```
\psframebox{%
\begin{pspicture}(-3,-3)(3,3)
  \psPhyllotaxis
\end{pspicture}}
```



```
\psframebox{%
\begin{pspicture}(-3,-3)(4,4)
  \psPhyllotaxis(1,1)
\end{pspicture}}
```

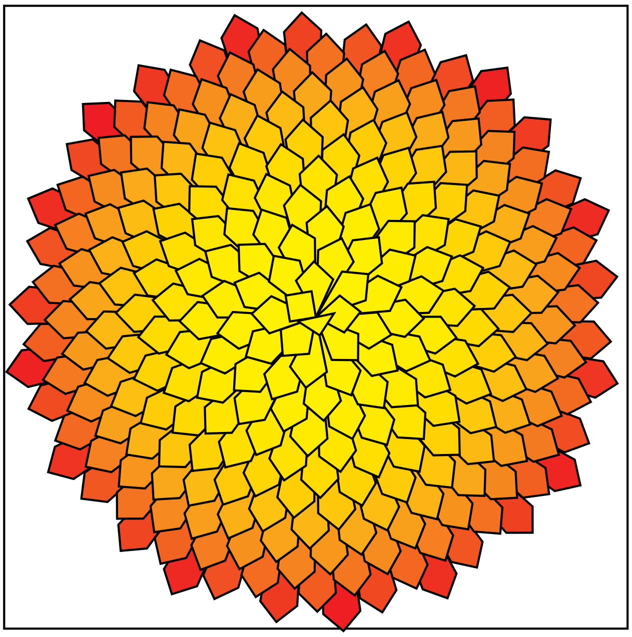
4.1 angle



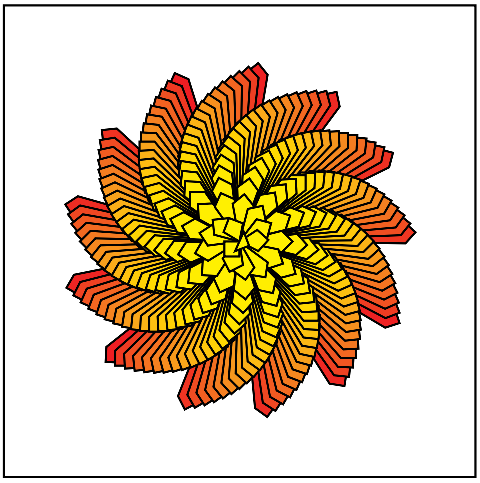
```
\psframebox{%
\begin{pspicture}(-2.5,-2.5)(2.5,2.5)
  \psPhyllotaxis[angle=99]
\end{pspicture}}
```

4.2 c

This is the length of one element in the unit pt.



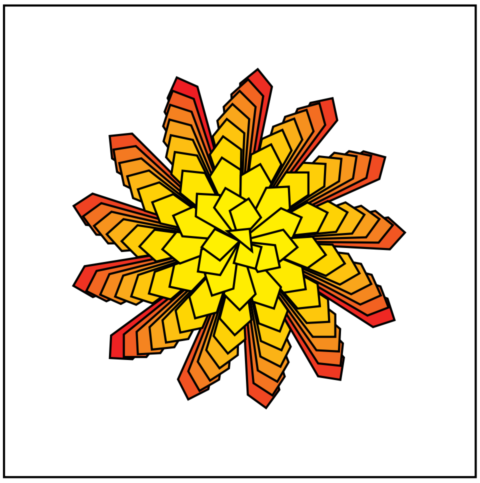
```
\psframebox{%
\begin{pspicture}(8,8)
  \psPhyllotaxis[c=7](4,4)
\end{pspicture}}
```



```
\psframebox{%
\begin{pspicture}(-3,-3)(3,3)
  \psPhyllotaxis[c=4,angle=111]
\end{pspicture}}
```

4.3 maxIter

This is the number for the iterations.

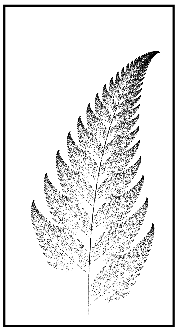


```
\psframebox{%
\begin{pspicture}(-3,-3)(3,3)
  \psPhyllotaxis[c=6,angle=111,maxIter=100]
\end{pspicture}}
```

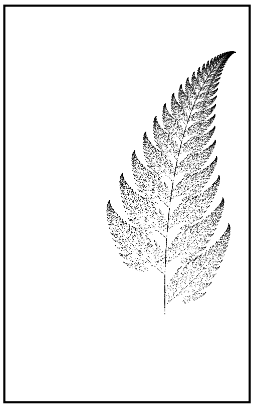
5 Fern

\psFern [Options] (x,y)

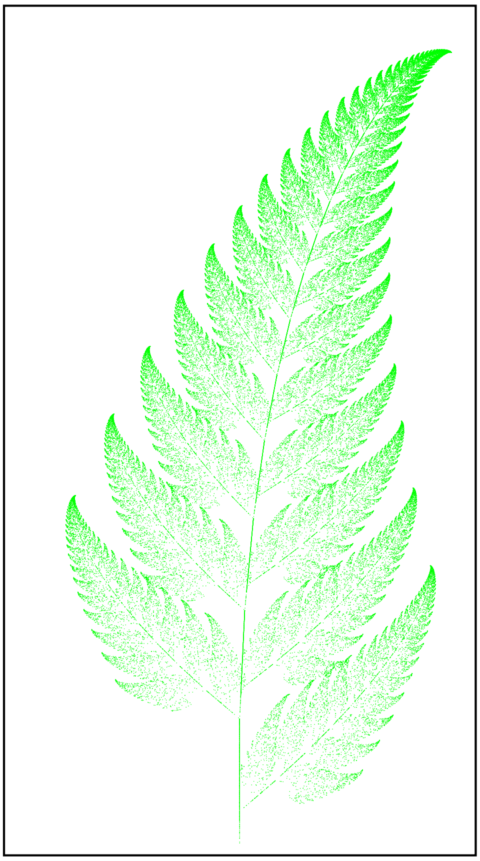
The coordinates of the starting point are optional, if they are missing, then $(0,0)$ is assumed. The default `scale` is set to 10.



```
\psframebox{%
\begin{pspicture}(-1,0)(1,4)
  \psFern
\end{pspicture}}
```



```
\psframebox{%
\begin{pspicture}(-1,0)(2,5)
  \psFern(1,1)
\end{pspicture}}
```

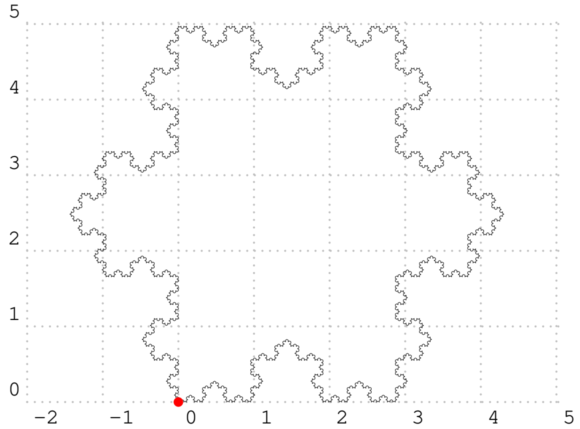


```
\psframebox{%
\begin{pspicture}(-3,0)(3,11)
  \psFern[scale=30,maxIter=100000,linecolor=green]
\end{pspicture}}
```

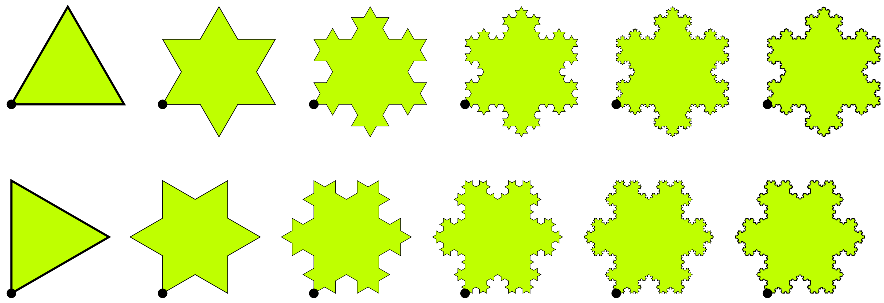
6 Koch flake

```
\psKochflake [Options] (x,y)
```

The coordinates of the starting point are optional, if they are missing, then $(0, 0)$ is assumed. The origin is the lower left point of the flake, marked as red or black point in the following example:




```
\begin{pspicture}[showgrid=true](-2.4,-0.4)(5,5)
  \psKochflake[scale=10]
  \psdot[linecolor=red,dotstyle=*](0,0)
\end{pspicture}
```



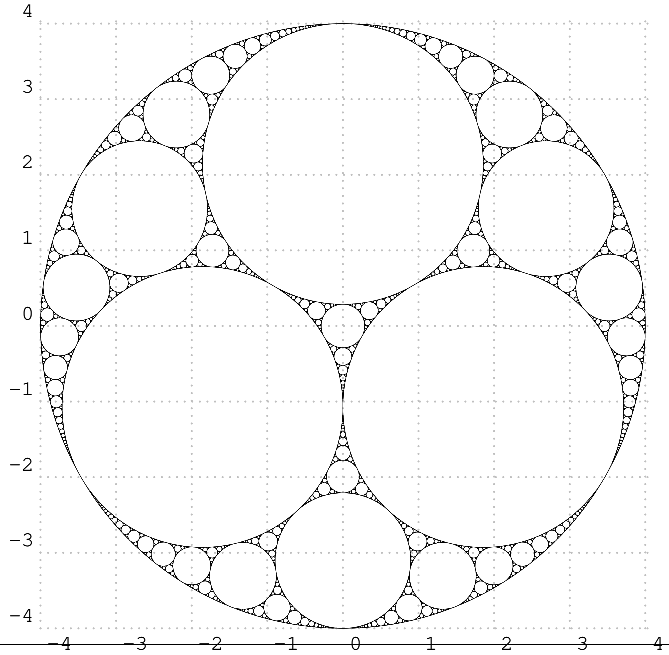
```
\begin{pspicture}(-0.4,-0.4)(12,4)
  \psset{fillcolor=lime,fillstyle=solid}
  \multido{\iA=0+1,\iB=0+2}{6}{%
    \psKochflake[angle=-30,scale=3,maxIter=\iA](\iB,2.5)\psdot*(\iB,2.5)
    \psKochflake[scale=3,maxIter=\iA](\iB,0)\psdot*(\iB,0)}
\end{pspicture}
```

Optional arguments are `scale`, `maxIter` (iteration depth) and `angle` for the first rotation angle.

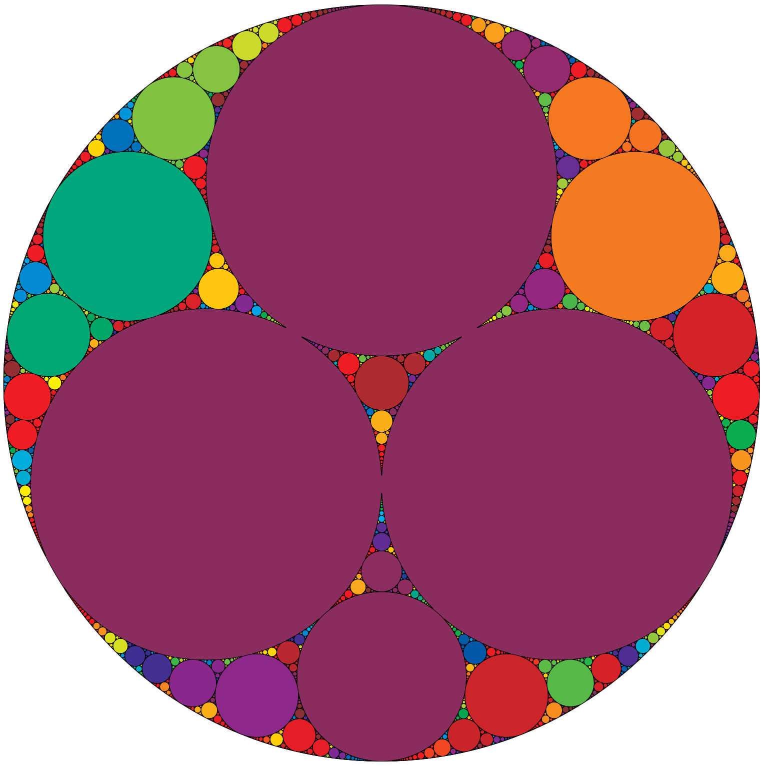
7 Apollonius circles

```
\psAppolonius [Options] (x,y)
```

The coordinates of the starting point are optional, if they are missing, then $(0,0)$ is assumed. The origin is the center of the circle:




```
\begin{pspicture}[showgrid=true](-4,-4)(4,4)
  \psAppolonius[Radius=4cm]
\end{pspicture}
```



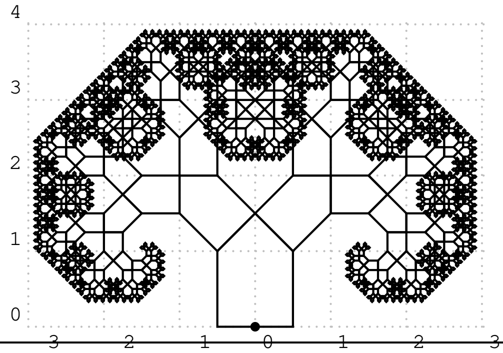
```
\begin{pspicture}(-5,-5)(5,5)
  \psAppolonius[Radius=5cm,Color]
\end{pspicture}
```

8 Trees

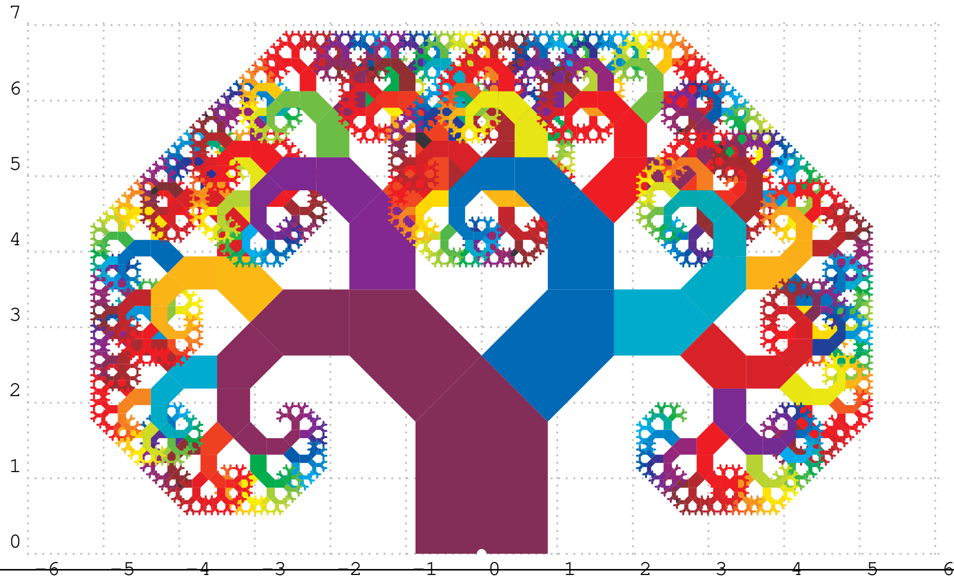
`\psPTree [Options] (x,y) \psFArrow [Options] (x,y){fraction}`

The coordinates of the starting point are optional, if they are missing, then $(0,0)$ is assumed. The origin is the center of the lower line, shown in the following examples by the dot. Special parameters are the width of the lower basic line for the tree and the height and angle for the arrow and for both the color option. The color step is given by `dIter` and the depth by `maxIter`. Valid optional arguments are

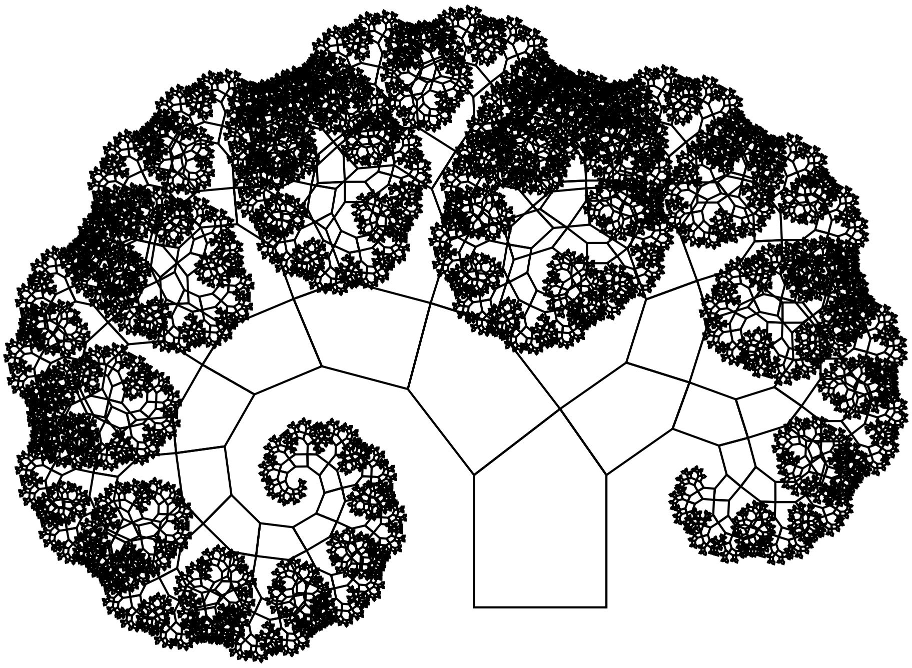
Name	*Meaning*	*default*
`xWidth`	first base width	1cm
`minWidth`	last base width	1pt
`c`	factor for unbalanced trees (0<c<1)	0.5
`Color`	colored tree	false




```
\begin{pspicture}[showgrid=true](-3,0)(3,4)
  \psPTree
  \psdot*(0,0)
\end{pspicture}
```

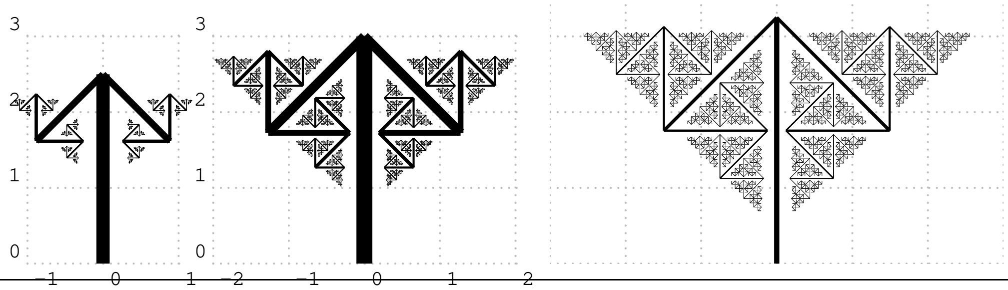



```
\begin{pspicture}[showgrid=true](-6,0)(6,7)
  \psPTree[xWidth=1.75cm,Color=true]
  \psdot*[linecolor=white](0,0)
\end{pspicture}
```



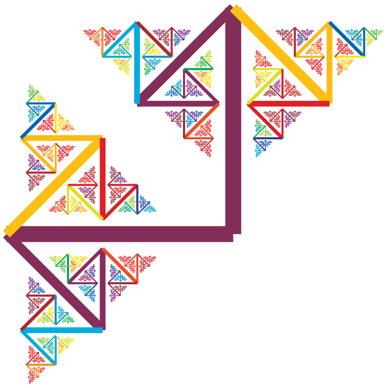
```
\begin{pspicture}(-7,-1)(6,8)
  \psPTree[xWidth=1.75cm,c=0.35]
\end{pspicture}
```

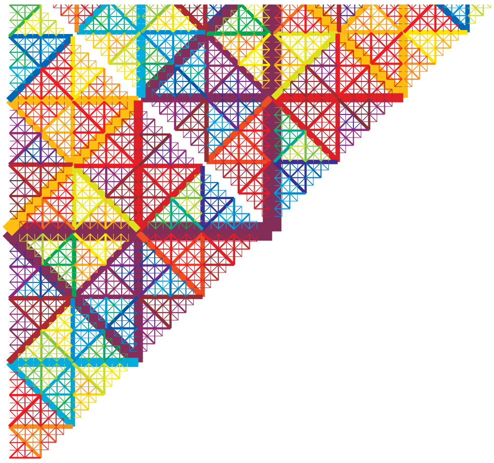
```
\begin{pspicture}(-5,-1)(7,8)
  \psPTree[xWidth=1.75cm,Color=true,c=0.65]
\end{pspicture}
```




```
\begin{pspicture}[showgrid=true](-1,0)(1,3)
  \psFArrow{0.5}
\end{pspicture}
\quad
\begin{pspicture}[showgrid=true](-2,0)(2,3)
  \psFArrow{0.6}
\end{pspicture}
\quad
\begin{pspicture*}[showgrid=true](-3,0)(3,3.5)
  \psFArrow[linewidth=3pt]{0.65}
\end{pspicture*}
```

```
\begin{pspicture}(-1,0)(1,3)
  \psFArrow[Color]{0.5}
\end{pspicture}
\quad
\begin{pspicture}(-2,0)(2,3)
  \psFArrow[Color]{0.6}
\end{pspicture}
\quad
\begin{pspicture*}(-3,0)(3,3.5)
  \psFArrow[Color]{0.65}
\end{pspicture*}
```





```
\begin{pspicture}(-3,-3)(2,3)
  \psFArrow[Color]{0.6}
  \psFArrow[angle=90,Color]{0.6}
\end{pspicture}
\quad
\begin{pspicture*}(-4,-3)(3,3)
  \psFArrow[Color]{0.7}
  \psFArrow[angle=90,Color]{0.7}
\end{pspicture*}
```

9 Fibonacci fractals

There are seven different commands which are all defined by Manuel Luque (for more informations see http://pstricks.blogspot.de):

```
\psFibonacciWord [Options] (x,y)
\psFibonacci [Options]
\psNewFibonacci [Options]
\psiFibonacci [Options]
\pskFibonacci [Options] (x,y)
\psBiperiodicFibonacci [Options] (x,y)
\psFibonacciPolyominoes [Options] (x,y)
```

- \psFibonacciWord A Fibonacci word after n iterations
- \psFibonacci Draw the fractal curve of a Fibonacci word
- \psNewFibonacci Draw a bunch of curves obtained from the “Dense Fibonacci Word” (DFW) by substitutions.
- \psiFibonacci In the article [4] a new family of curves in a row is called “i-Fibonacci Word Fractal”.
- \pskFibonacci study the following k-Fibonacci and the curves associated with words in the article “*On the k-Fibonacci words*[1]”, this command allows to represent these curves.

1 urlhttp://www.acta.sapientia.ro/acta-info/C5-2/info52-4.pdf

- `\psBiperiodicFibonacci` it is still José L. Ramírez and Gustavo N. Rubiano who in the article "*Biperiodic Fibonacci Word and Its Fractal Curve*"[2] extend the notion of Fibonacci sequence with 2 parameters (a, b). This command draws the associated fractal curves.
- `\psFibonacciPolyominoes` this command draws a Fibonacci tile, also called a Fibonacci flake and allows you to pave the plane in two ways, following the rules established by A. Blondin-Massé, S. Labbé, S. Brlek and M. Mendès-France in their article "*Fibonacci snowflakes*[3]".

The valid optional arguments with its default values:

1. `[n=10]` : number of iterations;
2. `[k=5]` : k-Fibonacci series;
3. `[a=5,b=5]` : Biperiodic-Fibonacci series;
4. `[angle=90]` : turn right (-) or left (+) an angle of this value (see examples in the article of José L. Ramírez et Gustavo N. Rubiano.
5. `[i=6]` : sets the follow-up nature of generalized Fibonacci;
6. `morphism=(0) (1) (2)` : for the command `\psNewFibonacci`, we will write in the 3 pairs of parentheses the substitutions to be performed (see section 10)).
7. `[PSfont=Times-Roman]` : PostScript font;
8. `[fontscale=8` : fontscale;
9. `[colorF]` : curve color `n-1` for construction by juxtaposition;
10. `[juxtaposition=false]` allows the juxtaposition of the `n` and `n-1` curves to bring up the `n+1` curve by simply writing `[juxtaposition]` in the options.
11. `[DFW=false]` to display the "*Dense Fibonacci Word*" (DFW) with `\psFibonacciWord[DFW]` ;
12. `[iFibonacci=false]` to display the word "*i-Fibonacci*" with `\psiFibonacciWord[iFibonacci]`, obtained with the `[i]` parameter after `[n]` iterations.

The color and the thickness of the line of the fractal curve `n` are fixed with the usual parameters of PSTricks: `linewidth` and `linecolor`. The starting point of the curve is in $(0, 0)$ and the unit is set by the PSTricks `unit =` option.

This package does not pretend to exhaust the subject on the continuation of Fibonacci, the word of Fibonacci and the various fractals which are inspired by it. The subject is very vast and the studies very numerous. For those who discover the subject here are some tracks.

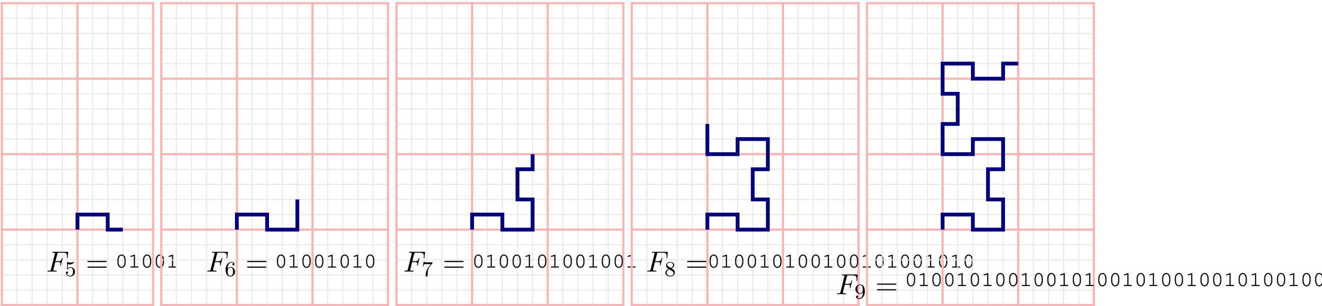
The number 478 of the August 2017 issue of *Pour la Science* contains an article by Jean-Paul Delahaye "*The following of Fibonacci . . . and its consequences*" whose title sums up the content of the article with, as usual, detailed explanations and beautiful illustrations.

Concerning all the variations on the fractal curve of the Fibonacci word, Alexis Monnerot-Dumaine's article entitled "*The Fibonacci Word fractal*" is the reference[4].

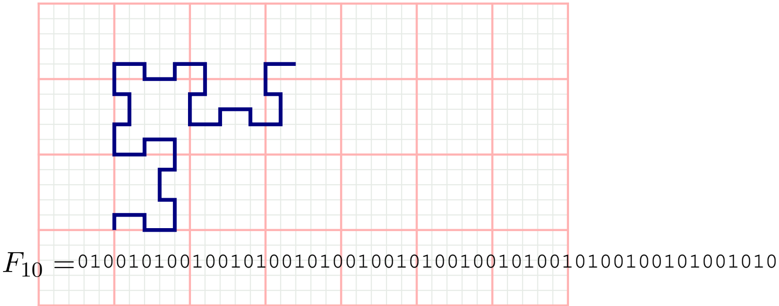
2 https://www.researchgate.net/publication/276406650_Biperiodic_Fibonacci_word_and_its_fractal_curve
3 www.slabbe.org/Publications/2011-fibo-snowflakes.pdf
4 https://hal.archives-ouvertes.fr/hal-00367972

The site https://fr.wikipedia.org/wiki/Fractale_du_mot_de_Fibonacci is also very rich in information.

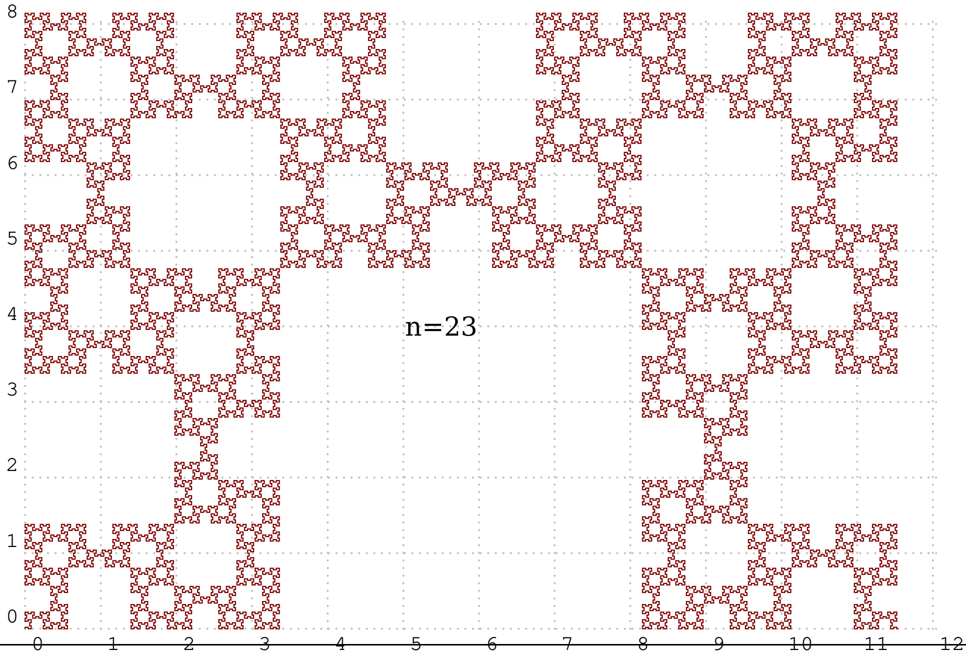
The command `\pSTilingsFibonacci` allows the tiling of the plane with the n order Fibonacci tile.




```
\begin{pspicture}[showgrid=false](-1,-1)(1,3)
\psgrid[style=gridstyleA]
\psFibonacci[unit=0.2,linecolor={[rgb]{0 0 0.5}},linewidth=0.05cm,n=5]
\rput(0,-0.45){$F_{5}=$}
\psFibonacciWord[n=5](0.5,-0.5)
\end{pspicture}
\begin{pspicture}[showgrid=false](-1,-1)(2,3)
\psgrid[style=gridstyleA]
\psFibonacci[unit=0.2,linecolor={[rgb]{0 0 0.5}},linewidth=0.05cm,n=6]
\rput(0,-0.45){$F_{6}=$}
\psFibonacciWord[n=6](0.5,-0.5)
\end{pspicture}
\begin{pspicture}[showgrid=false](-1,-1)(2,3)
\psgrid[style=gridstyleA]
\psFibonacci[unit=0.2,linecolor={[rgb]{0 0 0.5}},linewidth=0.05cm,n=7]
\rput(-0.5,-0.45){$F_{7}=$}
\psFibonacciWord[n=7](0,-0.5)
\end{pspicture}
\begin{pspicture}[showgrid=false](-1,-1)(2,3)
\psgrid[style=gridstyleA]
\psFibonacci[unit=0.2,linecolor={[rgb]{0 0 0.5}},linewidth=0.05cm,n=8]
\rput(-0.4,-0.45){$F_{8}=$}
\psFibonacciWord[n=8](0,-0.5)
\end{pspicture}
\begin{pspicture}[showgrid=false](-1,-1)(2,3)
\psgrid[style=gridstyleA]
\psFibonacci[unit=0.2,linecolor={[rgb]{0 0 0.5}},linewidth=0.05cm,n=9]
\psFibonacciWord[n=9](-0.5,-0.75)
\rput(-1,-0.75){$F_{9}=$}
\end{pspicture}
```

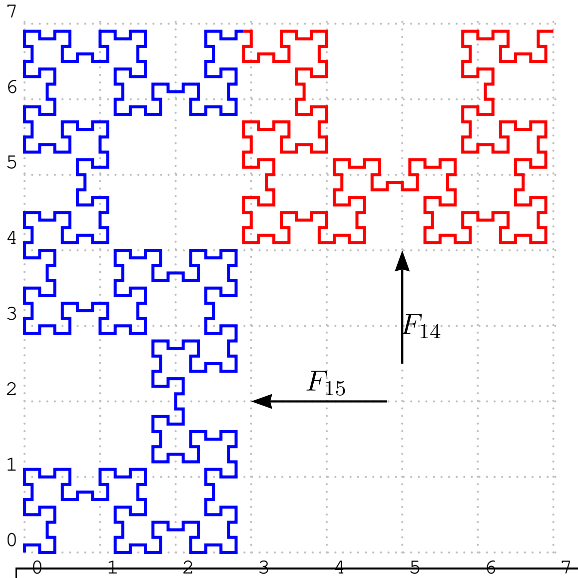


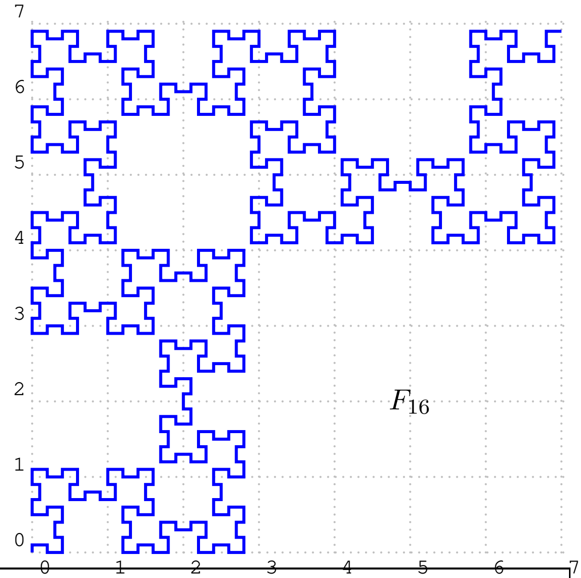

```
\begin{pspicture}[showgrid=false](-1,-1)(6,3)
\psgrid[style=gridstyleA]
\psFibonacci[unit=0.2,linecolor={[rgb]{0 0 0.5}},linewidth=0.05cm,n=10]
\psFibonacciWord[n=10](-0.5,-0.5)
\rput(-1,-0.45){$F_{10}=$}
\end{pspicture}
```



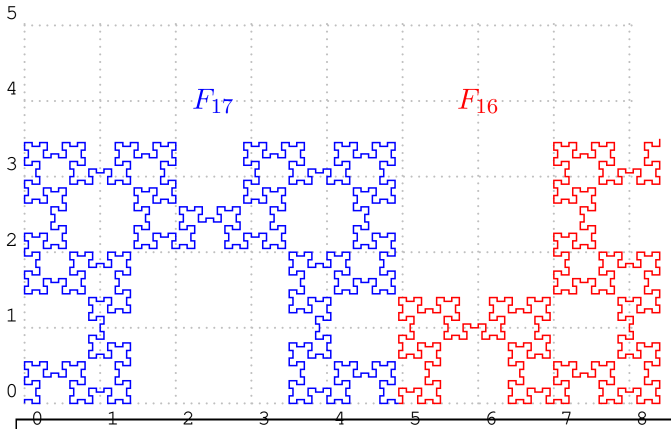

```
\begin{pspicture}[showgrid](0,0)(12,8)
\psFibonacci[unit=0.02,linecolor={[rgb]{0.5 0 0}},n=23,linewidth=0.015cm]
\rput(5.5,4){n=23}
\end{pspicture}
```

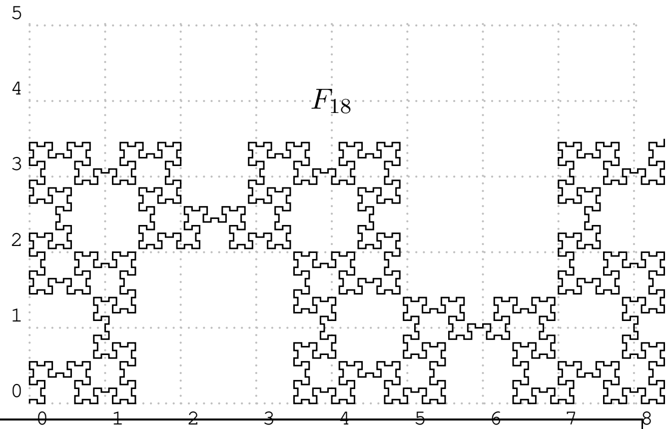
9.1 Fractal curves with juxtaposition



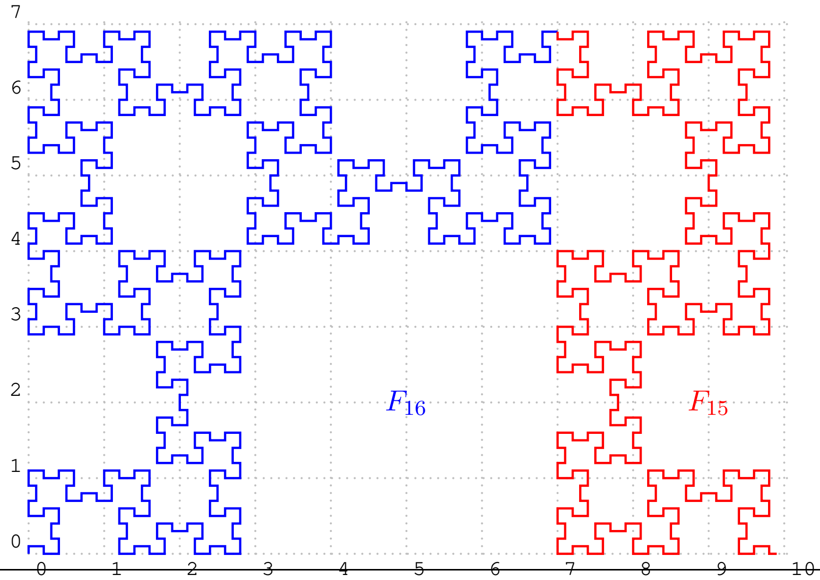



```
\begin{pspicture}[showgrid](0,0)(7,7)
\psFibonacci[unit=0.1,linecolor=blue,n=15,linewidth=0.04cm,juxtaposition]
\rput(4,2.25){$F_{15}$}
\rput(5.25,3){$F_{14}$}
\psline[arrowinset=0.1,arrowsize=0.2]{->}(4.8,2)(3,2)
\psline[arrowinset=0.1,arrowsize=0.2]{->}(5,2.5)(5,4)
\end{pspicture}
\hfill
\begin{pspicture}[showgrid](0,0)(7,7)
\psFibonacci[unit=0.1,linecolor=blue,n=16,linewidth=0.04cm]
\rput(5,2){$F_{16}$}
\end{pspicture}
```

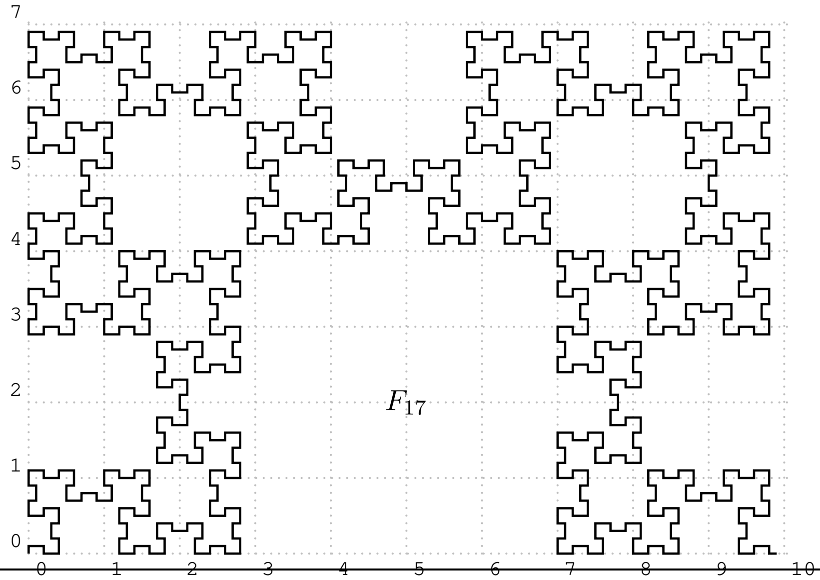




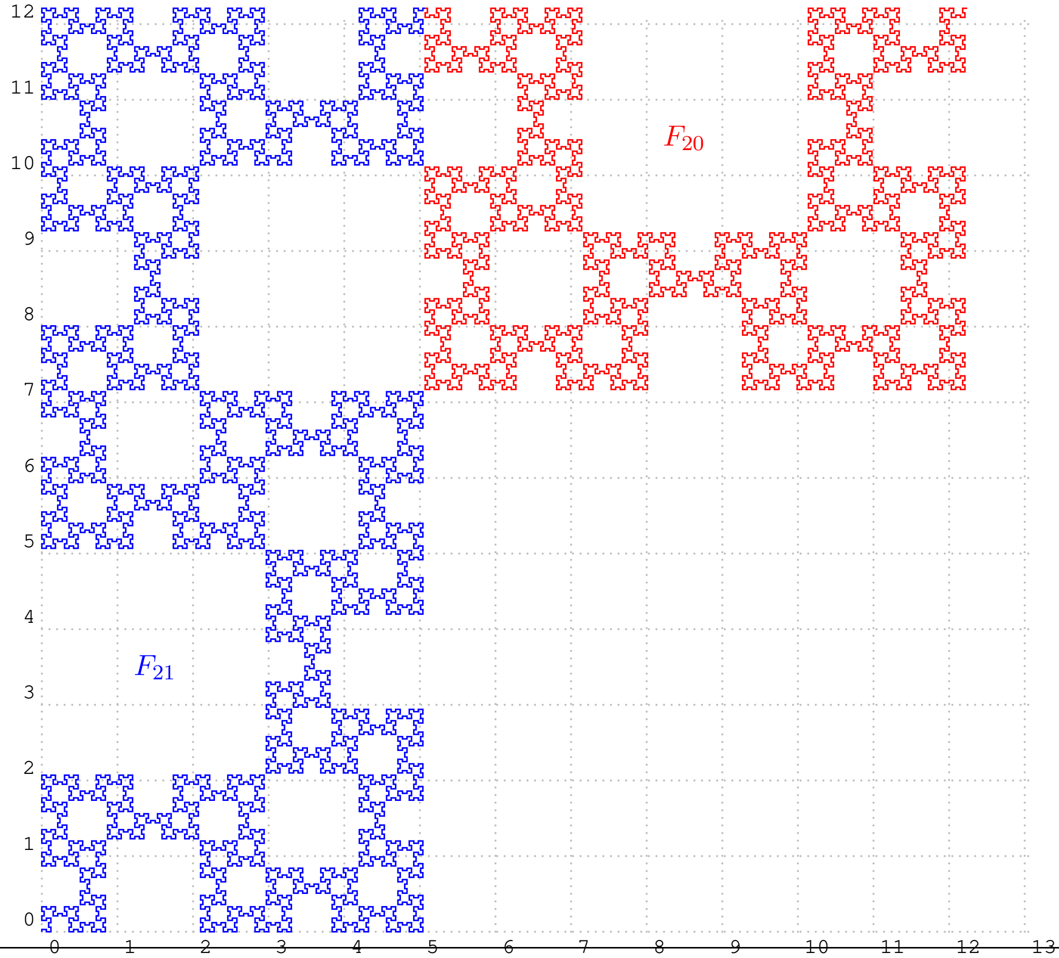

```
\begin{pspicture}[showgrid](0,0)(8,5)
\psFibonacci[unit=0.05,linecolor=blue,n=17,linewidth=0.02cm,juxtaposition]
\rput(2.5,4){\blue$F_{17}$}
\rput(6,4){\red$F_{16}$}
\end{pspicture}
\hfill
\begin{pspicture}[showgrid](0,0)(8,5)
\psFibonacci[unit=0.05,n=18,linewidth=0.02cm]
\rput(4,4){$F_{18}$}
\end{pspicture}
```



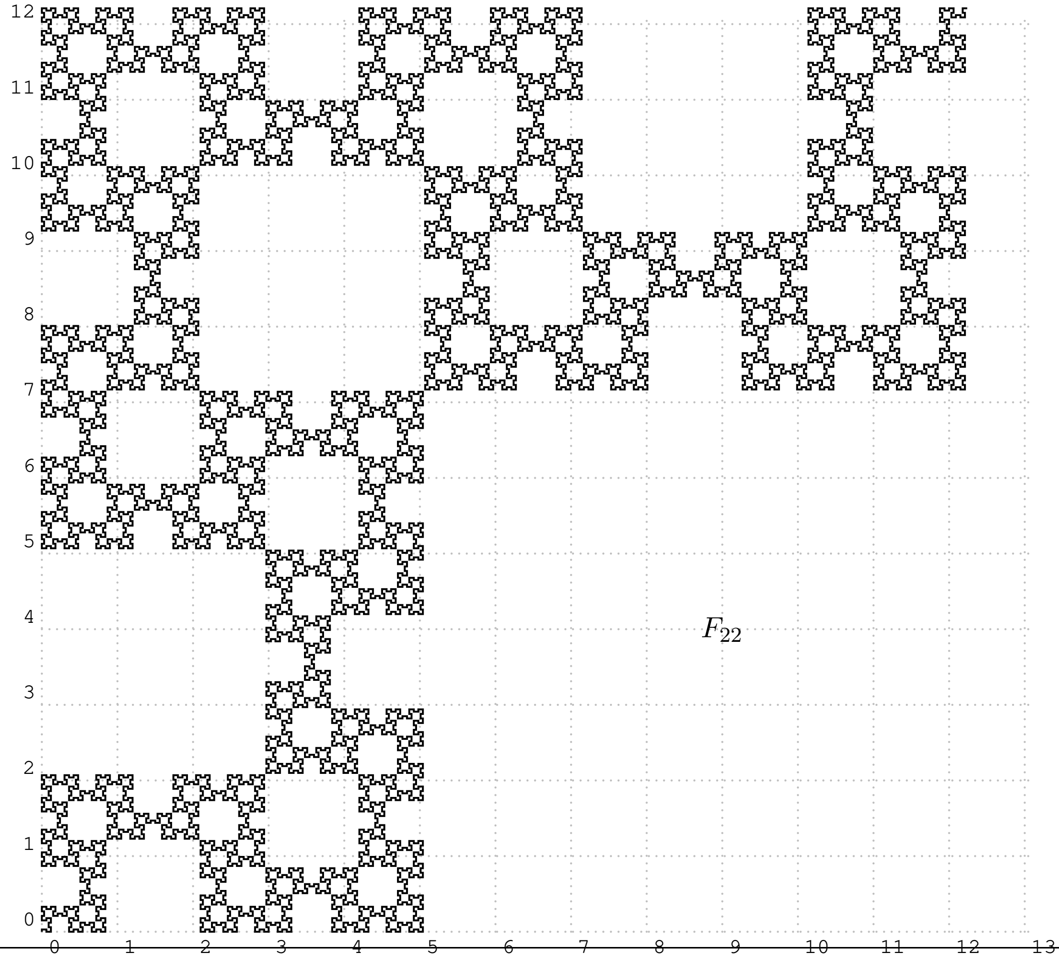

```
\begin{pspicture}[showgrid](0,0)(10,7)
\psFibonacci[unit=0.1,linecolor=blue,n=16,linewidth=0.03cm,juxtaposition]
\rput(5,2){\blue$F_{16}$}
\rput(9,2){\red$F_{15}$}
\end{pspicture}
```



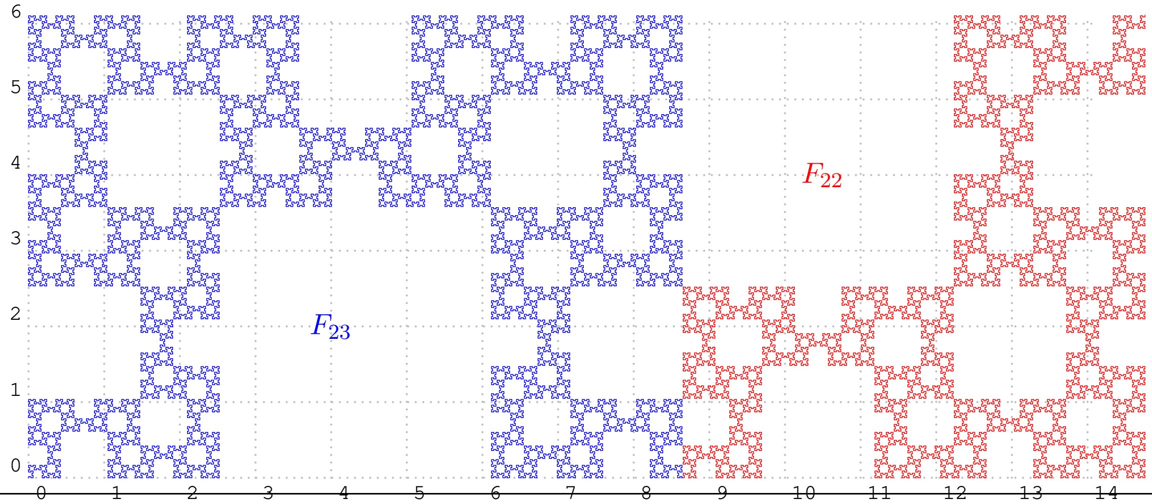

```
\begin{pspicture}[showgrid](0,0)(10,7)
\psFibonacci[unit=0.1,n=17,linewidth=0.03cm]
\rput(5,2){$F_{17}$}
\end{pspicture}
```



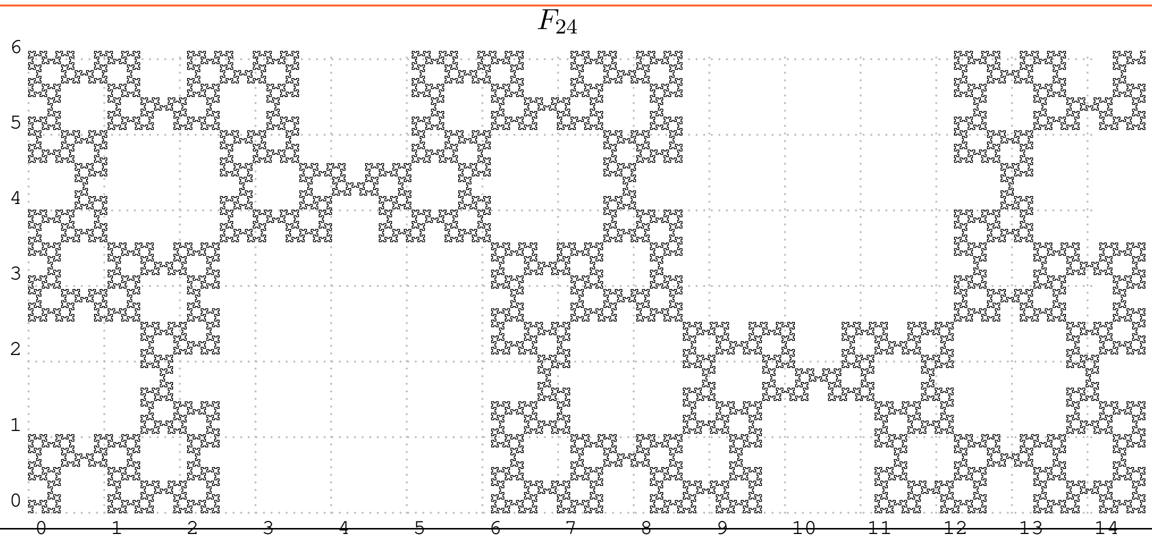

```
\begin{pspicture}[showgrid](0,0)(13,12)
\psFibonacci[unit=0.03,linecolor=blue,n=21,linewidth=0.02cm,juxtaposition]
\rput(1.5,3.5){\blue$F_{21}$}
\rput(8.5,10.5){\red$F_{20}$}
\end{pspicture}
```




```
\begin{pspicture}[showgrid](0,0)(13,12)
\psFibonacci[unit=0.03,n=22,linewidth=0.025cm]
\rput(9,4){$F_{22}$}
\end{pspicture}
```



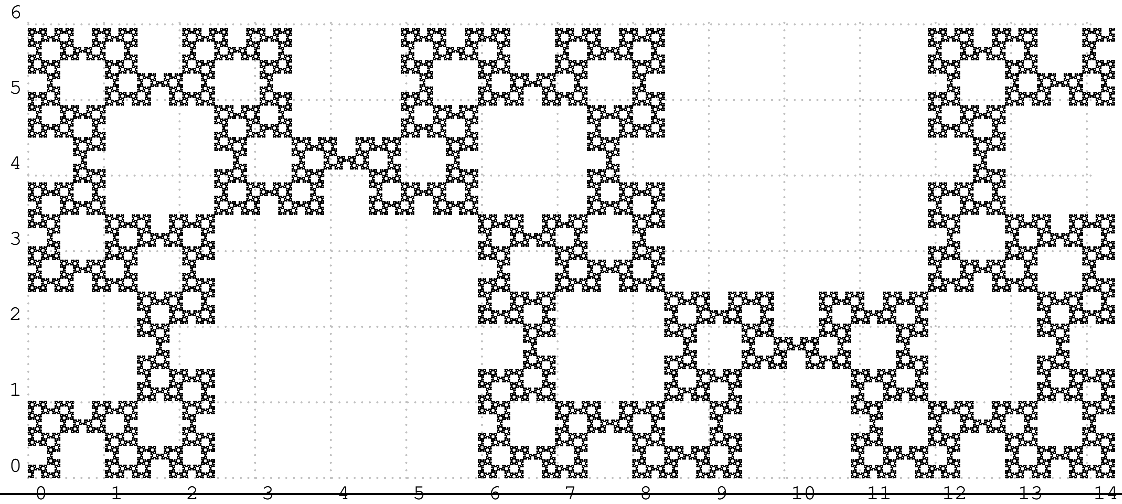

```
\begin{pspicture}[showgrid](0,0)(14,6)
\psFibonacci[unit=0.015,linecolor=blue,n=23,linewidth=0.01cm,juxtaposition]
\rput(4,2){\blue$F_{23}$}
\rput(10.5,4){\red$F_{22}$}
\end{pspicture}
```




```
\begin{pspicture}[showgrid](0,0)(14,6)
\psFibonacci[unit=0.015,n=24,linewidth=0.01cm]
\rput(7,6.5){$F_{24}$}
\end{pspicture}
```

9.2 Curves with a big number of iterations

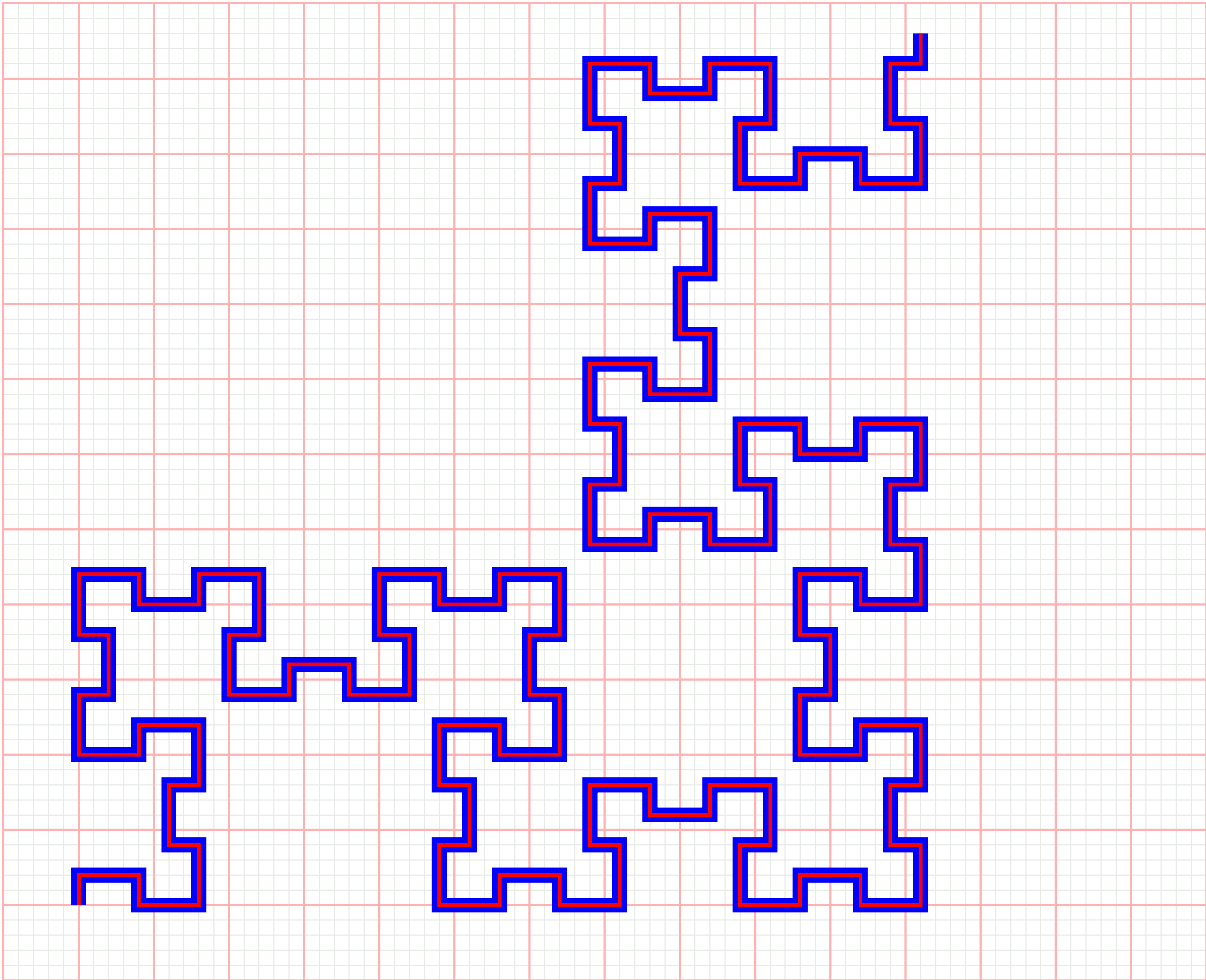
With n=30 it takes a long time and the number is not readable.



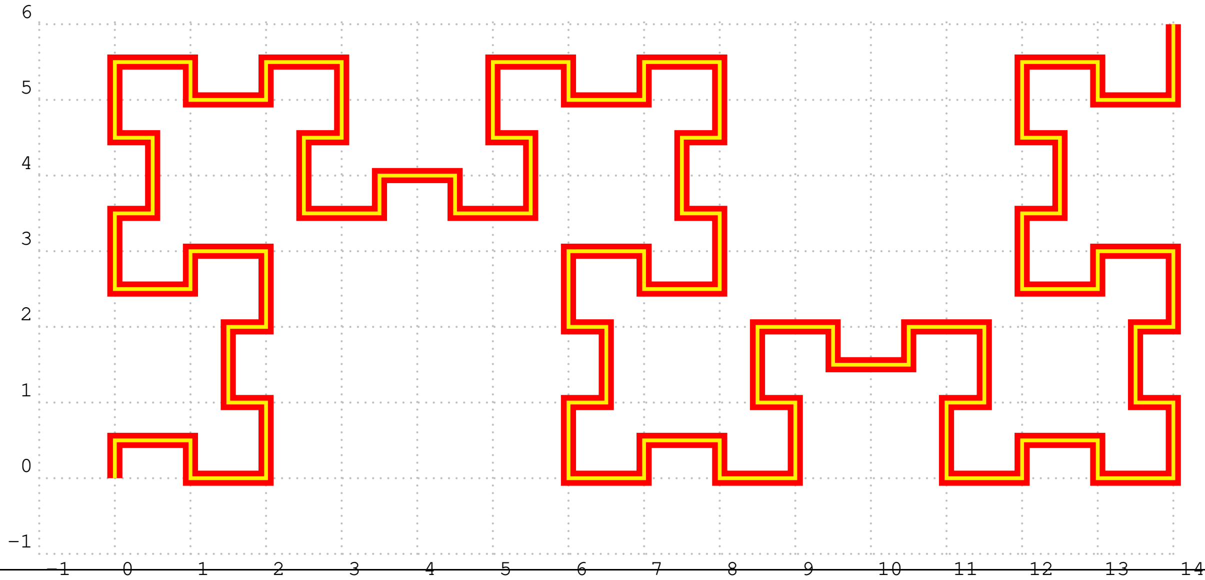

```
\begin{pspicture}[showgrid](0,0)(14,6)
\psFibonacci[unit=0.0025,n=30,linewidth=0.001cm]
\end{pspicture}
```

9.3 Double color curves

Superposition of two curves are possible by choosing diffrent color and line thickness.



```
\begin{pspicture}(-1,-1)(15,12)
\psgrid[style=gridstyleA]
\psFibonacci[unit=0.4,linecolor=blue,n=13,linewidth=0.2cm]%
\psFibonacci[unit=0.4,linecolor=red,n=13,linewidth=0.05cm]%
\end{pspicture}
```




```
\begin{pspicture}[showgrid](-1,-1)(14,6)
\psFibonacci[unit=0.5,linecolor=red,n=12,linewidth=0.2cm]%
\psFibonacci[unit=0.5,linecolor=yellow,n=12,linewidth=0.05cm]%
\end{pspicture}
```

10 “Dense Fibonacci Word” and the command \psNewFibonacci

In the chapter “The dense Fibonacci word: a whole family of curves”, Alexis Monnerot-Dumaine wrote:

> The odd-even design rule is not easy to manage and we can change to a more practical rule. As Jean-Paul Allouche suggested, we can create a word of 3 letters with {0; 1; 2} that can draw the Fibonacci fractal with the simplest drawing rules following:
>
> - 0, draw a segment in line with the previous one
> - 1, draw a segment by turning to the right
> - 2, draw a segment by turning to the left

By replacing in the Fibonacci word 00→0, 01→1 and 10→2. Alexis Monnerot-Dumaine defines the “Dense Fibonacci Word” (DFW). From the DFW, we get a whole family of curves by doing, for example, substitutions following:

- $\mu_1 : 1 \longrightarrow 10\ ;\ 0 \longrightarrow 12\ ;\ 2 \longrightarrow 02$
- $\mu_2 : 1 \longrightarrow 010\ ;\ 0 \longrightarrow 0102\ ;\ 2 \longrightarrow 002$
- $\mu_3 : 1 \longrightarrow 02\ ;\ 0 \longrightarrow 21\ ;\ 2 \longrightarrow 10$
- $\mu_4 : 1 \longrightarrow 02\ ;\ 0 \longrightarrow 00\ ;\ 2 \longrightarrow 10$

We will find all these families of curves with explanations and references in the article Alexis Monnerot-Dumaine. These are just brief explanations for using the commands PSTricks to draw these families of curves. In their article "Properties and Gener- Fractal Exploring Fractal Curves “alizations of the Fibonacci” [4] illustrate this family of curves with Mathematica by designating them under the name of New-Fibonacci. This name seems to me sensible the PSTricks command will be called \psNewFibonacci.

10.1 “Dense Fibonacci Word”

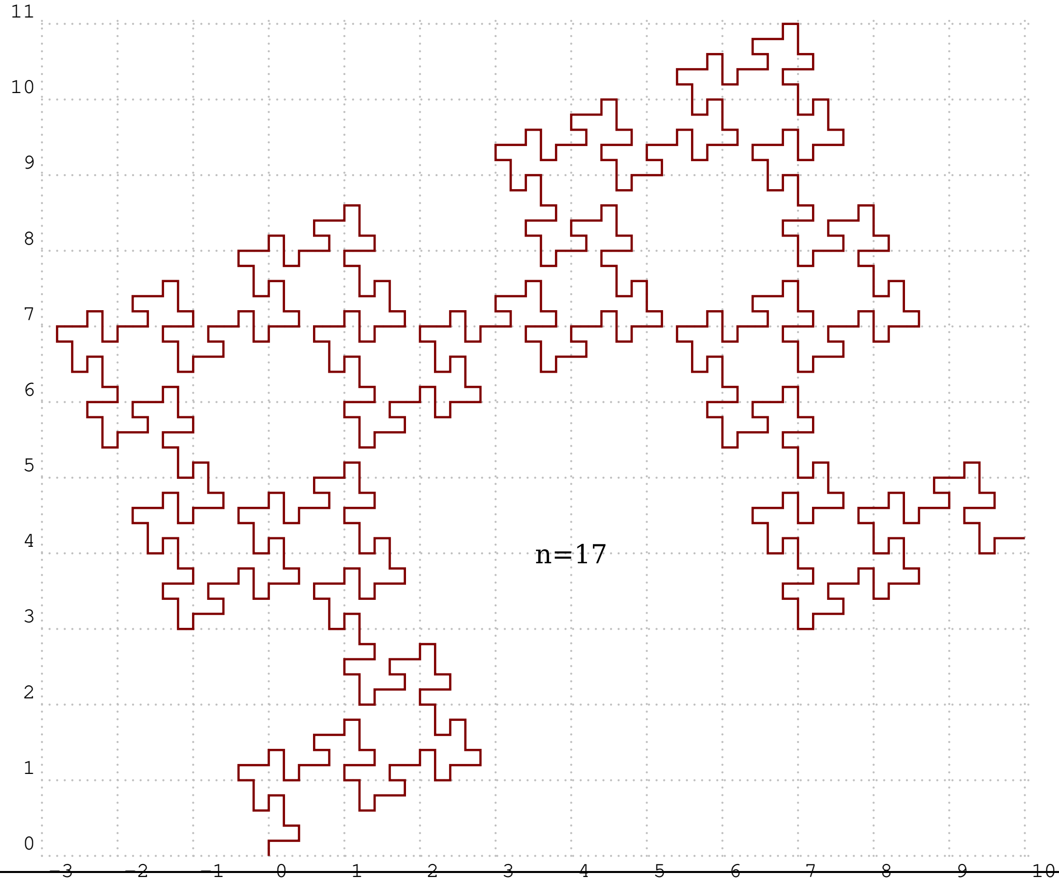
FW=0100101001001010010100100101001001010010100100101001010

DFW=1022102211021102110221022 11

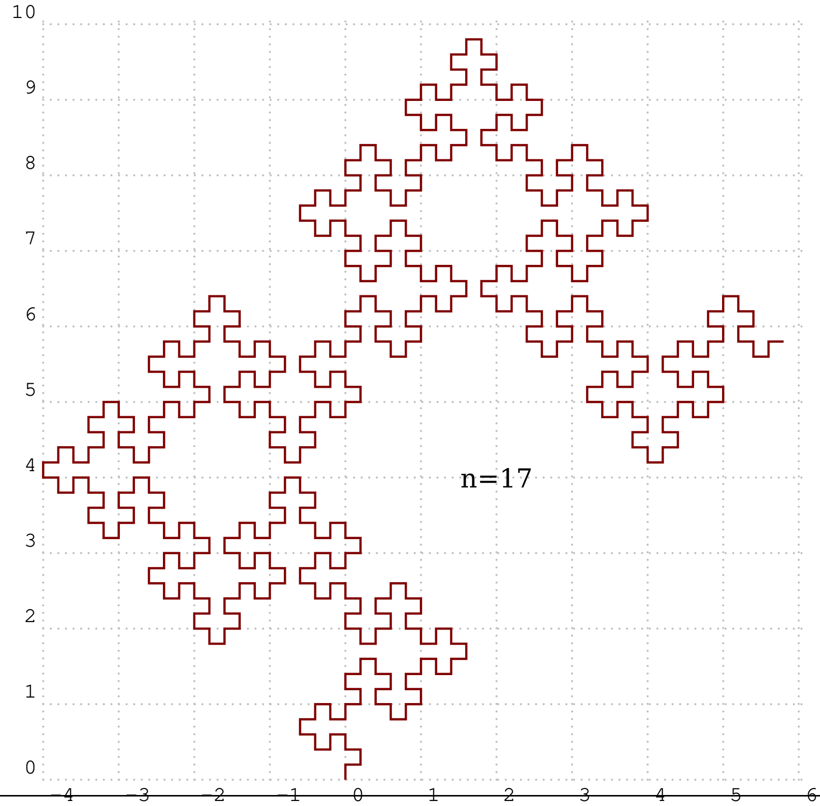
```
\begin{pspicture}[showgrid=false](-1,-0.2)(10,2)
\uput[r](-0.5,1){FW=}%
\psFibonacciWord[n=10,fontscale=12](0.5,0.9)
\uput[r](-0.75,0){DFW=}%
\psFibonacciWord[n=10,DFW,fontscale=12](0.5,-0.1)
\end{pspicture}
```

10.2 Fractal of “*Dense Fibonacci Word*”

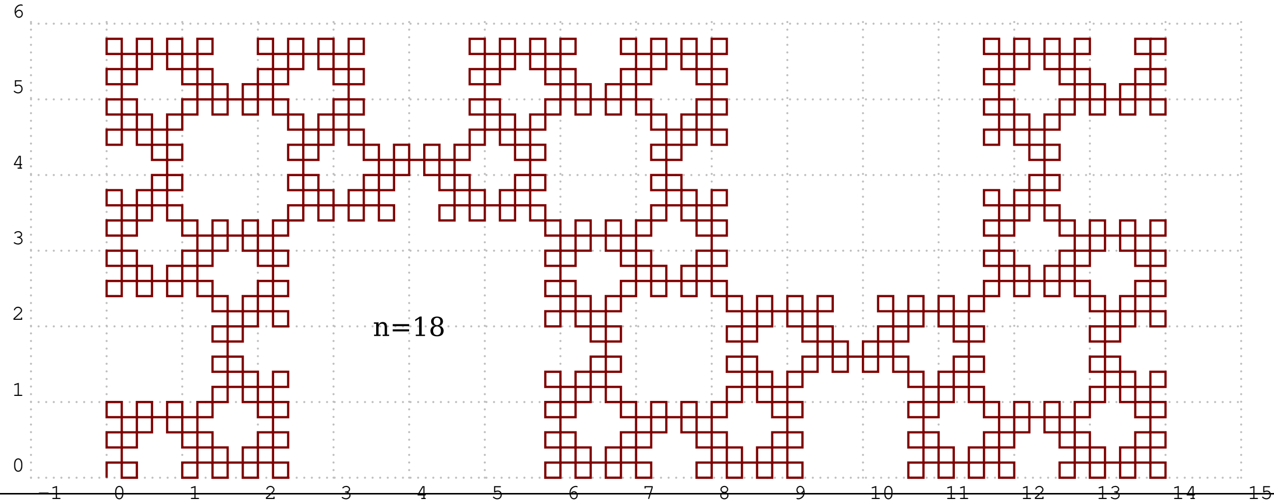
The curve can be created with `\psNewFibonacci`



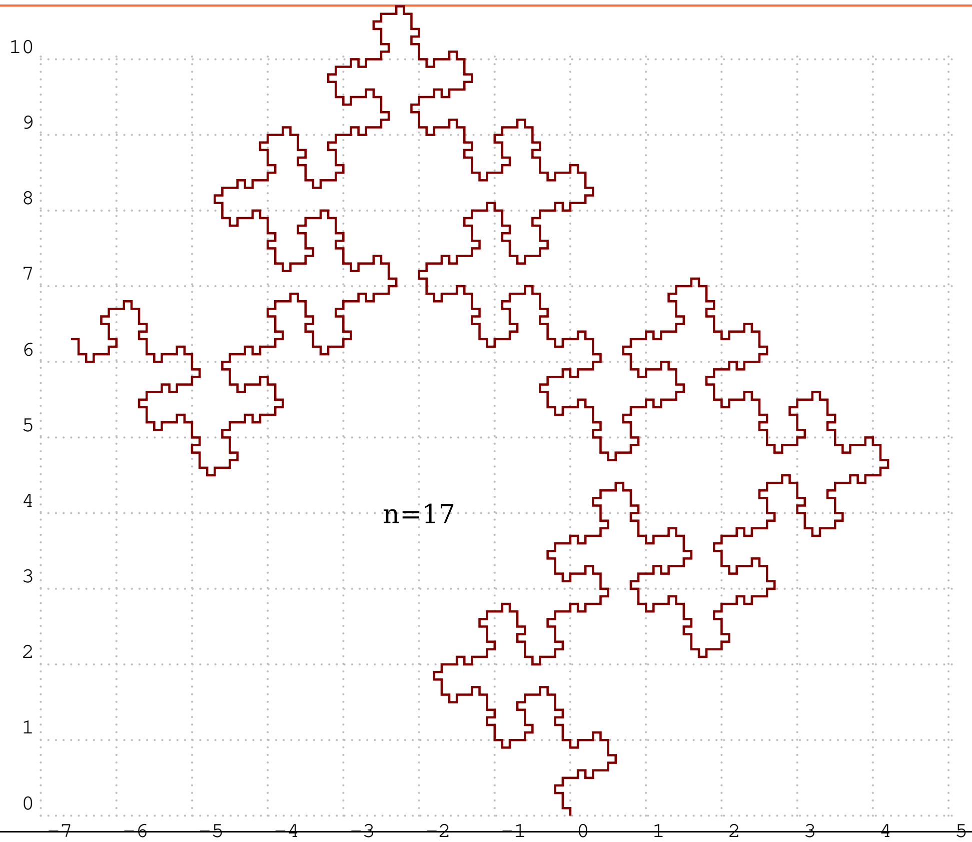

```
\begin{pspicture}[showgrid](-3,0)(10,11)
\psNewFibonacci[unit=0.2,linecolor={[rgb]{0.5 0 0}},n=17,linewidth=0.03cm]
\rput(4,4){n=17}
\end{pspicture}
```



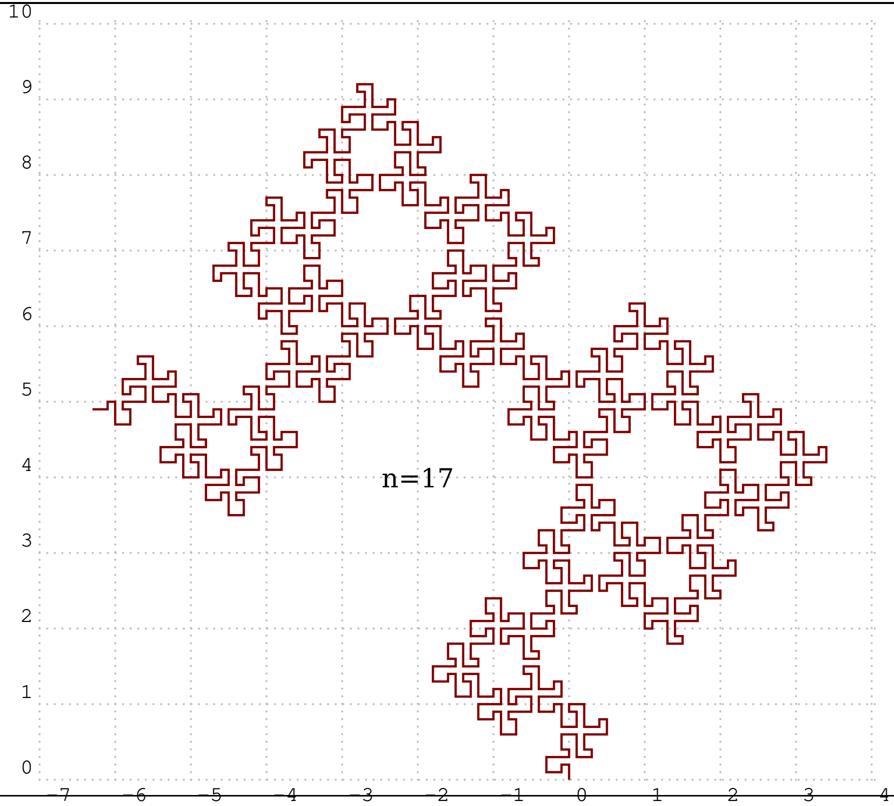

```
\begin{pspicture}[showgrid](-4,0)(6,10)
\psNewFibonacci[unit=0.2,linecolor={[rgb]{0.5 0 0}},n=17,linewidth=0.03cm,morphism=() (1) (2)]
\rput(2,4){n=17}
\end{pspicture}
```



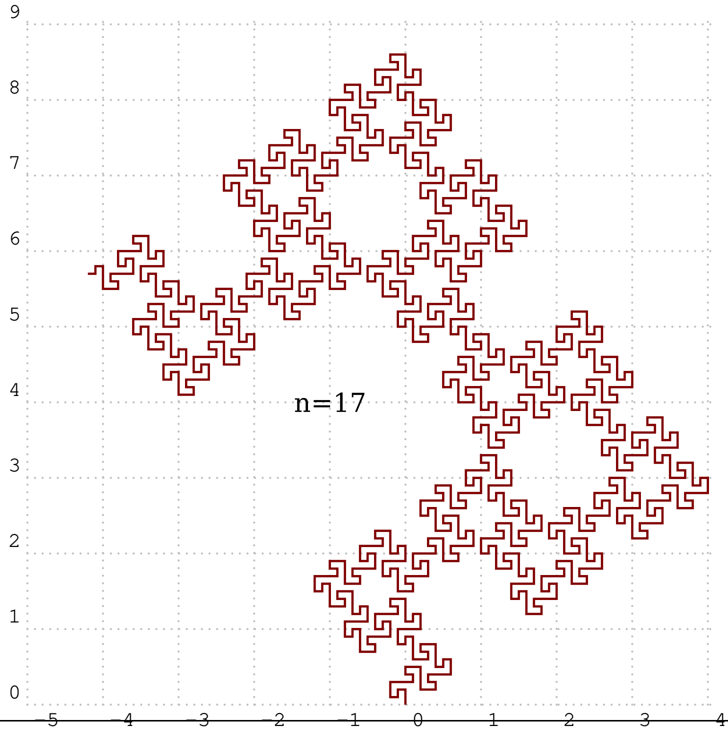

```
\begin{pspicture}[showgrid](-1,0)(15,6)
\psNewFibonacci[unit=0.2,linecolor={[rgb]{0.5 0 0}},n=18,linewidth=0.03cm,morphism=(12) (1) (2)]
\rput(4,2){n=18}
\end{pspicture}
```



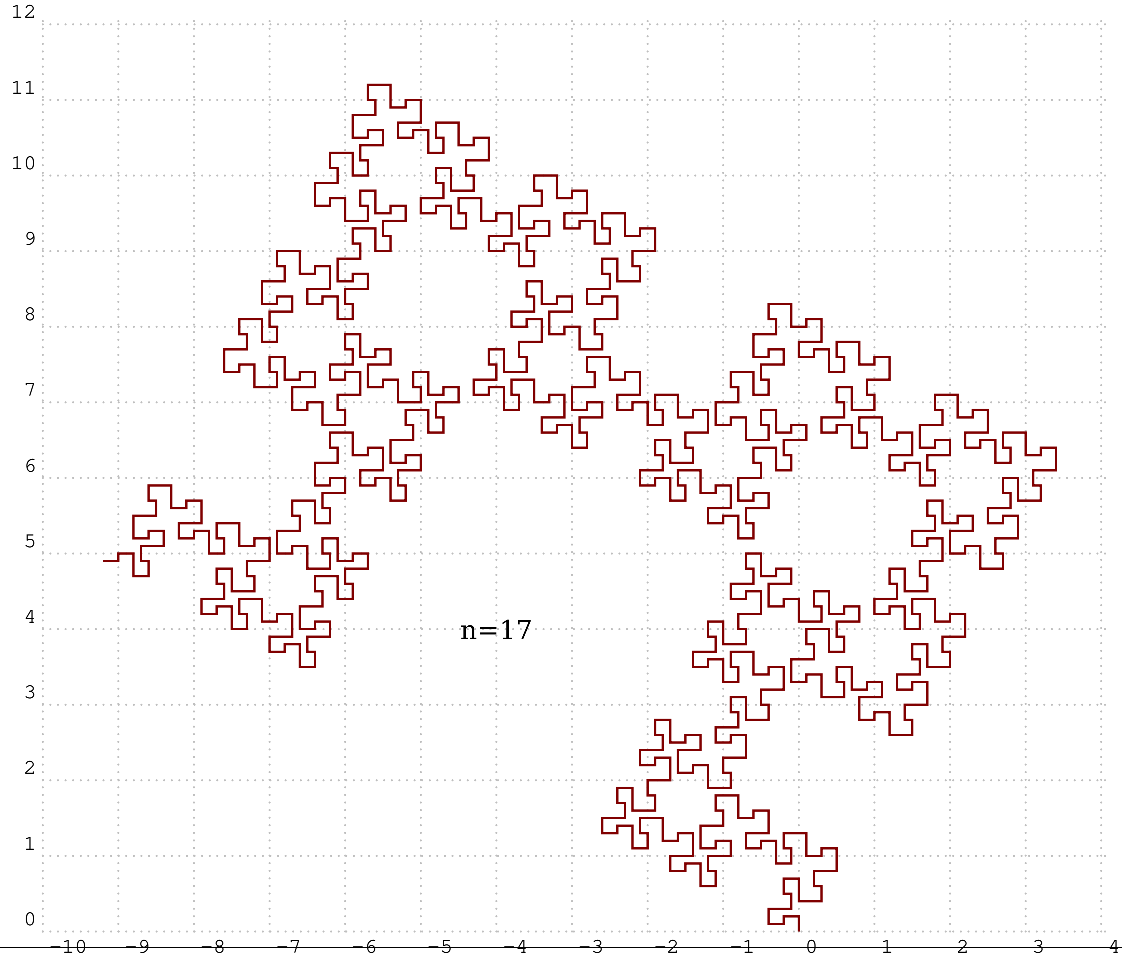

```
\begin{pspicture}[showgrid](-7,0)(5,10)
\psNewFibonacci[unit=0.1,linecolor={[rgb]{0.5 0 0}},n=17,linewidth=0.03cm,morphism=(102) (2) (1)]
\rput(-2,4){n=17}
\end{pspicture}
```



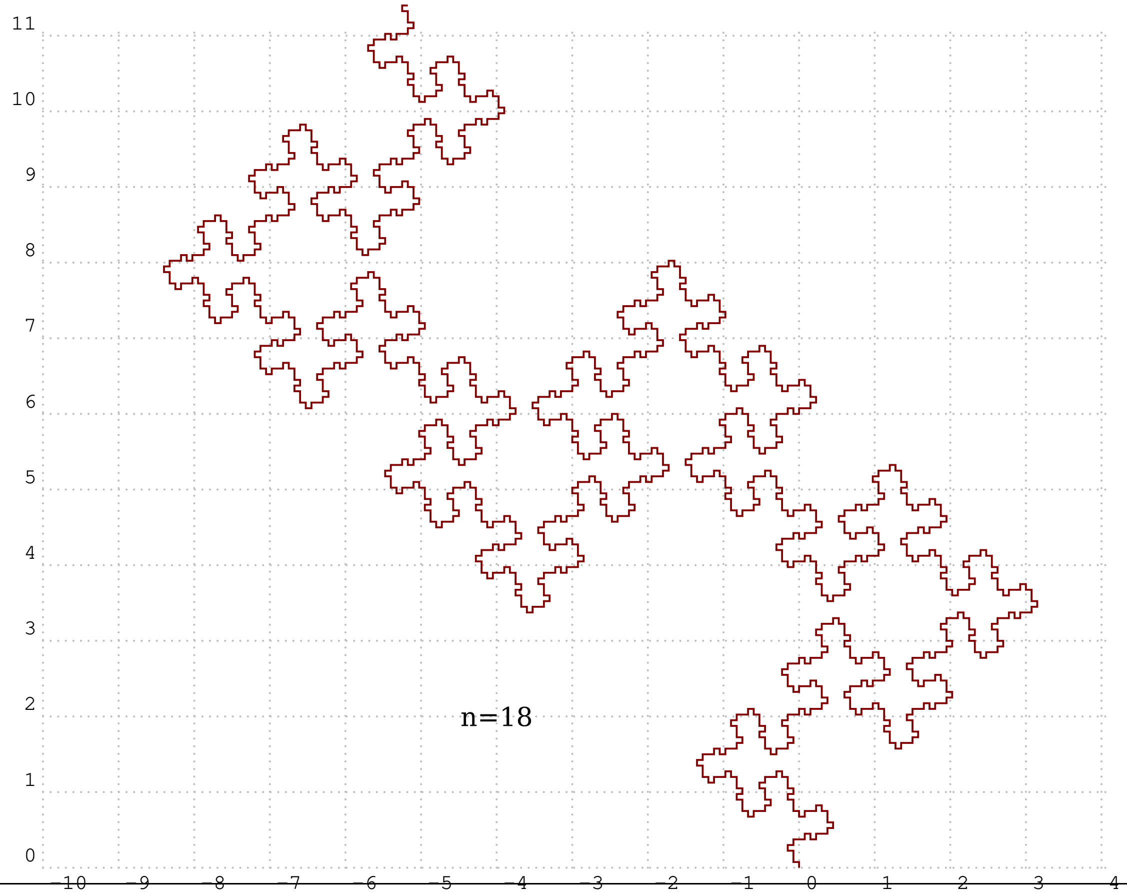

```
\begin{pspicture}[showgrid](-7,0)(4,10)
\psNewFibonacci[unit=0.1,linecolor={[rgb]{0.5 0 0}},n=17,linewidth=0.03cm,morphism=(210) (02) (10)]
\rput(-2,4){n=17}
\end{pspicture}
```




```
\begin{pspicture}[showgrid](-5,0)(4,9)
\psNewFibonacci[unit=0.1,linecolor={[rgb]{0.5 0 0}},n=17,linewidth=0.03cm,morphism=(21) (02) (10)]
\rput(-1,4){n=17}
\end{pspicture}
```




```
\begin{pspicture}[showgrid](-10,0)(4,12)
\psNewFibonacci[unit=0.1,linecolor={[rgb]{0.5 0 0}},n=17,linewidth=0.03cm,morphism=(210) (020) (10)]
\rput(-4,4){n=17}
\end{pspicture}
```




```
\begin{pspicture}[showgrid](-10,0)(4,11)
\psNewFibonacci[unit=0.075,linecolor={[rgb]{0.5 0 0}},n=18,linewidth=0.025cm,morphism=(102) (2) (1)]
\rput(-4,2){n=18}
\end{pspicture}
```

11 The command \psiFibonacci

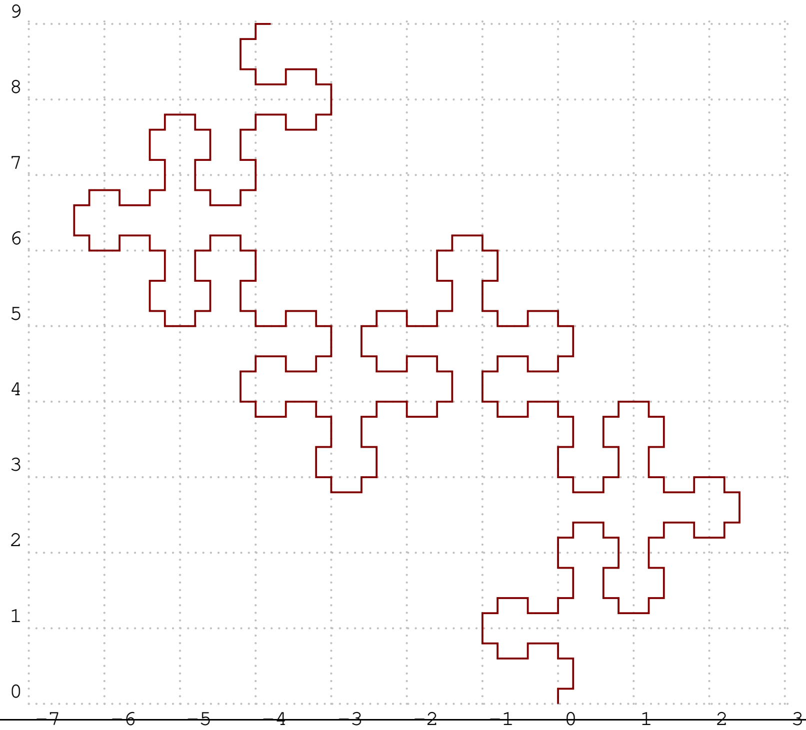
11.1 How it works

Briefly, (read the article [4] for more details) more “I-Fibonacci Word” depends on the parameter i and the number of iterations n with the following rules, according to the authors’ notations. More informations and the original code are available from https://pstricks.blogspot.com/2017/09/fractale-du-mot-de-fibonacci.html.

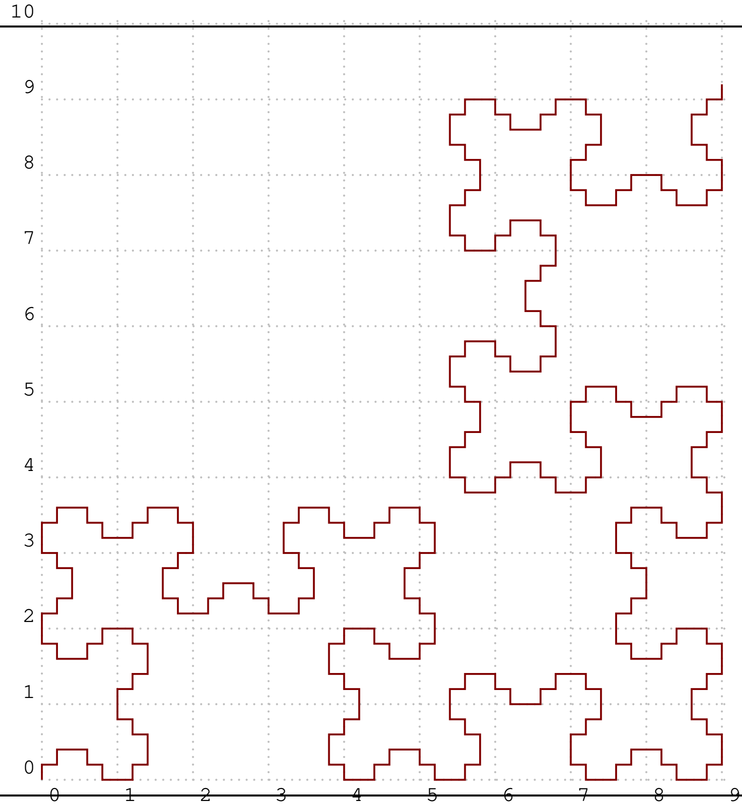
- $f_0^{[i]}$ =0
- $f_1^{[i]} = 0^{i-1}1$: this notation means that it is necessary to put (i-1) 0 before the 1
- $f_n^{[i]} = f_{n-1}^{[i]} f_{n-2}^{[i]}$ pour $n \geq 2$ et $i \geq 1$.

The construction of the associated fractal curves follows the “even-odd” rule as for the fractal of the word Fibonacci.

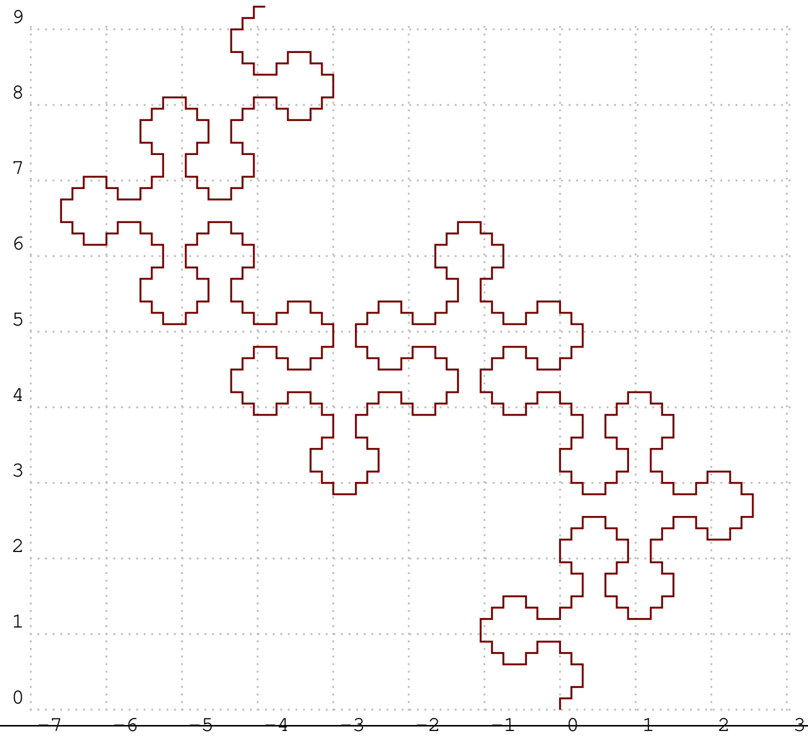
11.2 Examples



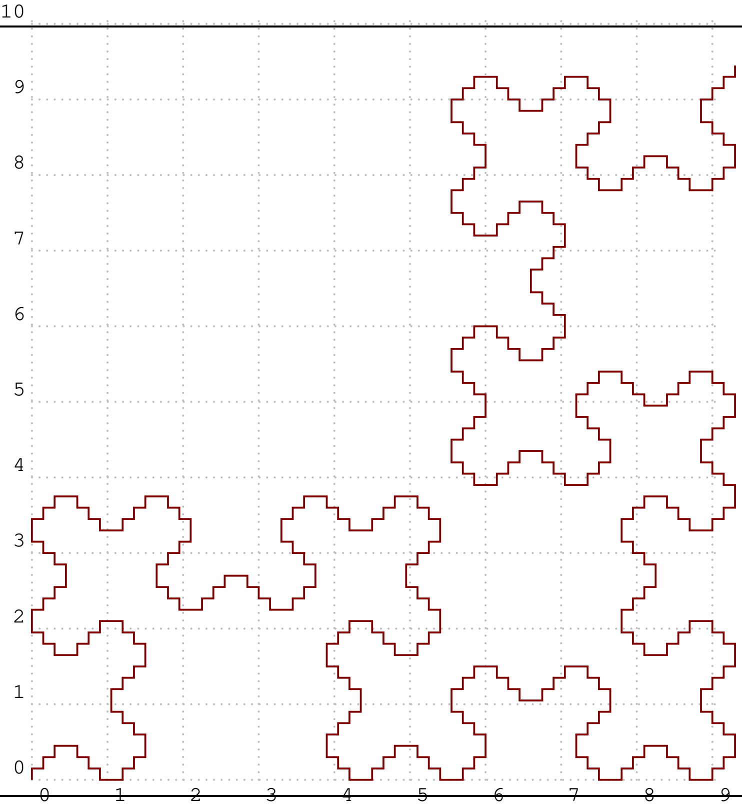

```
\begin{pspicture}[showgrid](-7,0)(3,9)
\psiFibonacci[unit=0.2,linecolor={[rgb]{0.5 0 0}},n=10,linewidth=0.025cm,i=3]
\end{pspicture}
```



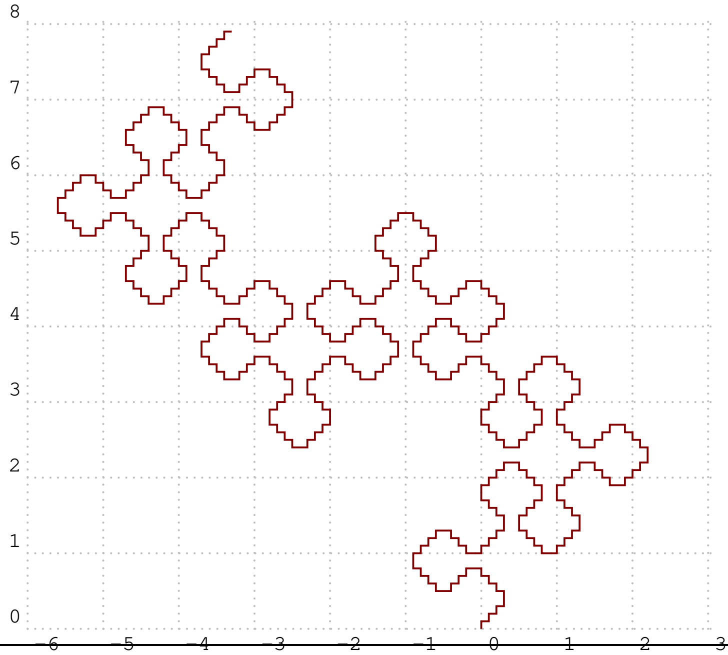

```
\begin{pspicture}[showgrid](0,0)(9,9.5)
\psiFibonacci[unit=0.2,linecolor={[rgb]{0.5 0 0}},n=10,linewidth=0.025cm,i=4]
\end{pspicture}
```



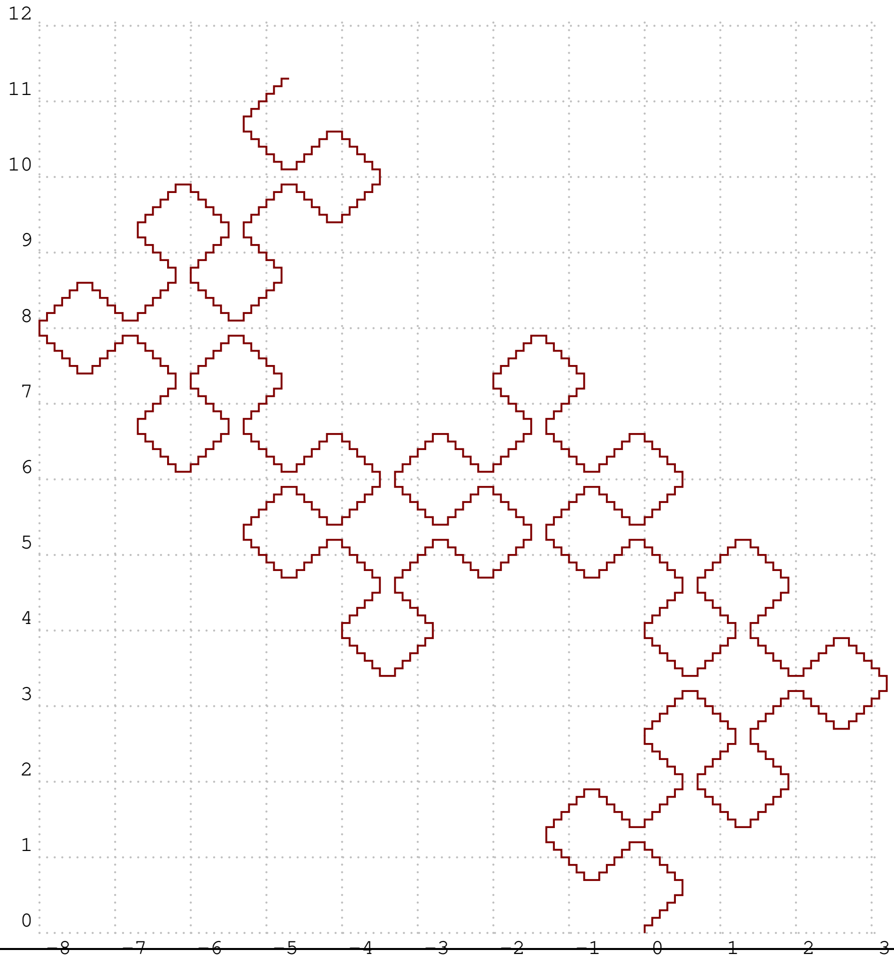

```
\begin{pspicture}[showgrid](-7,0)(3,9)
\psiFibonacci[unit=0.15,linecolor={[rgb]{0.5 0 0}},n=10,linewidth=0.025cm,i=5]
\end{pspicture}
```



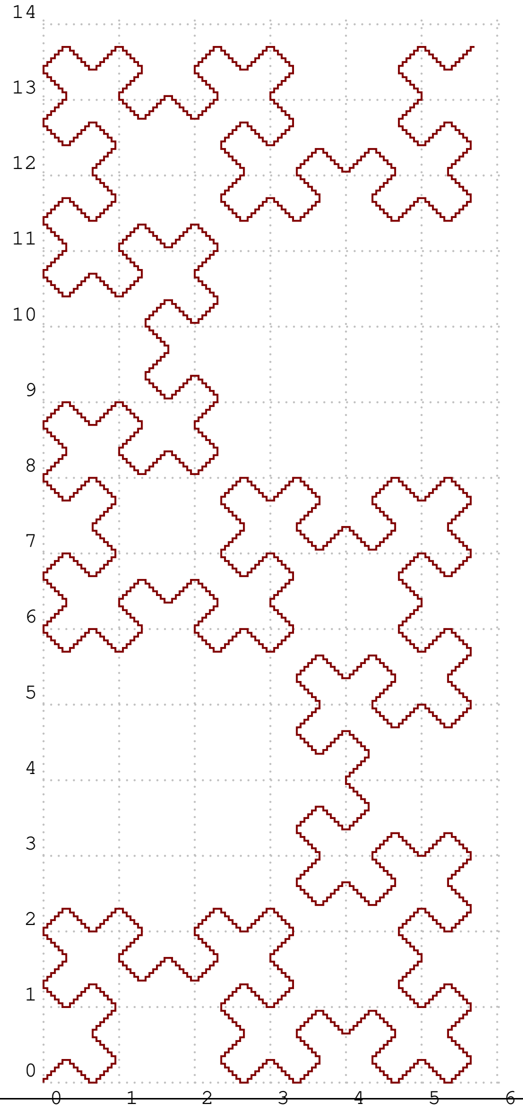

```
\begin{pspicture}[showgrid](0,0)(9,9.5)
\psiFibonacci[unit=0.15,linecolor={[rgb]{0.5 0 0}},n=10,linewidth=0.025cm,i=6]
\end{pspicture}
```




```
\begin{pspicture}[showgrid](-6,0)(3,8)
\psiFibonacci[unit=0.1,linecolor={[rgb]{0.5 0 0}},n=10,linewidth=0.025cm,i=7]
\end{pspicture}
```



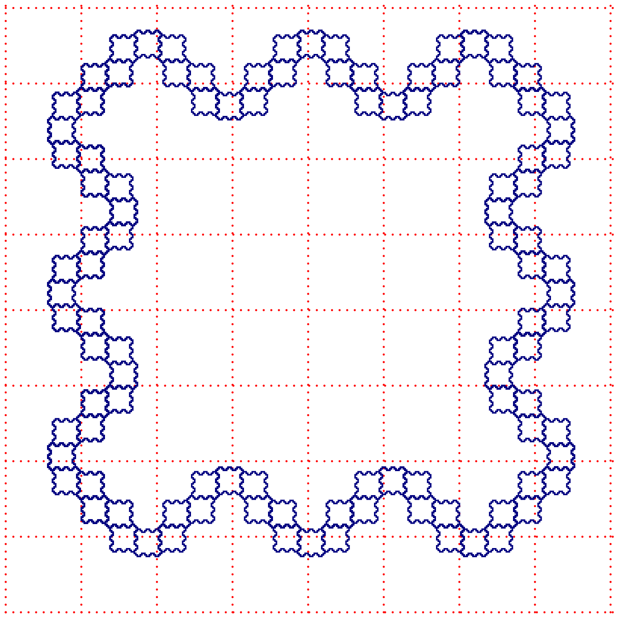

```
\begin{pspicture}[showgrid](-8,0)(3,12)
\psiFibonacci[unit=0.1,linecolor={[rgb]{0.5 0 0}},n=10,linewidth=0.025cm,i=11]
\end{pspicture}
```



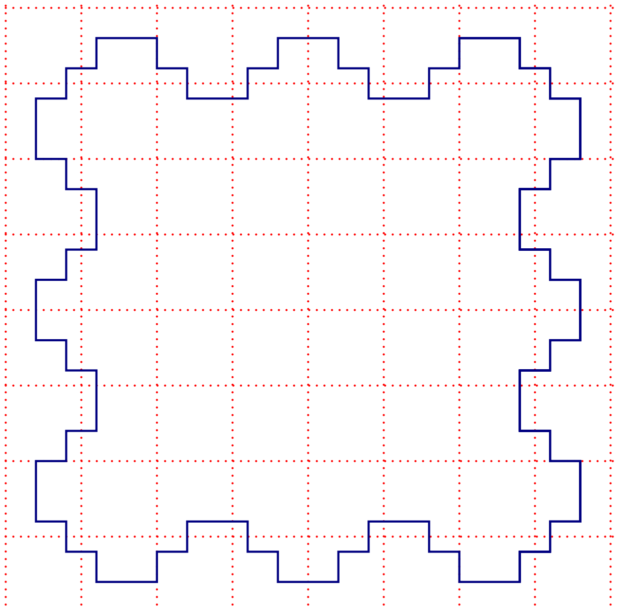

```
\begin{pspicture}[showgrid](0,0)(6,14)
\psiFibonacci[unit=0.05,linecolor={[rgb]{0.5 0 0}},n=12,linewidth=0.025cm,i=12]
\end{pspicture}
```

12 The command \pskFibonacci

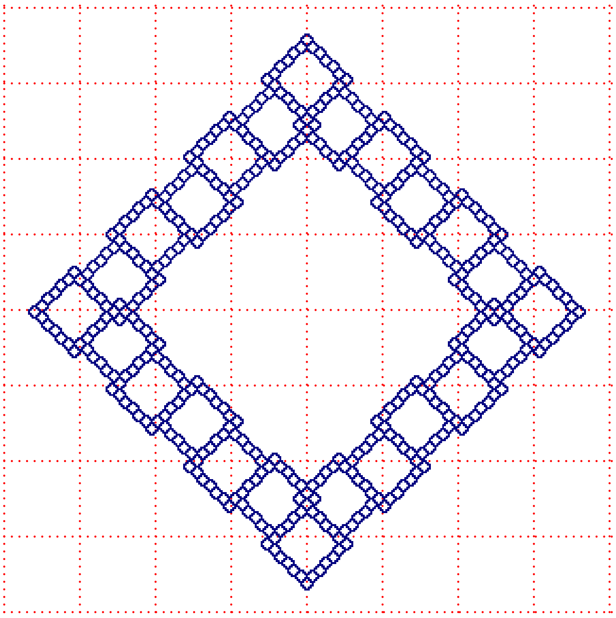
The 2 mandatory parameters are n and k. The following coordinates are optional but put, possibly, to center the curve at the origin of the mark.



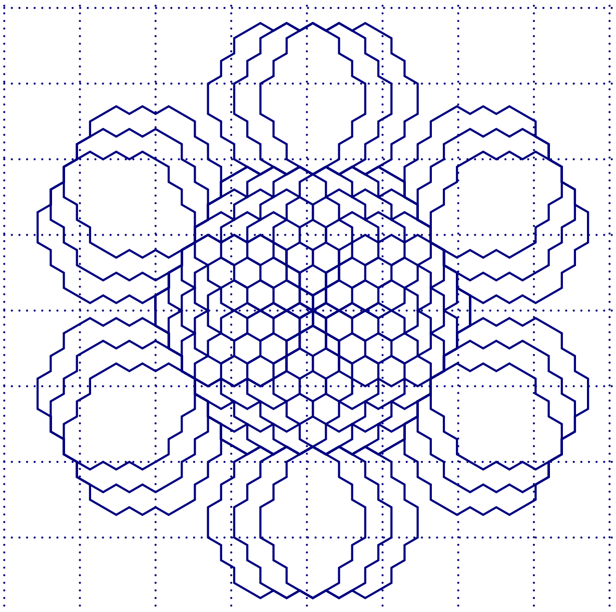
```
\begin{pspicture}(-4,-4)(4,4)
\psgrid[gridlabels=0pt,subgriddiv=0,gridcolor=red,griddots=10]
\pskFibonacci[unit=0.02,linecolor={[rgb]{0 0 0.5}},linewidth=0.02cm,n=6,k=5](-2.3,-3.2)
\end{pspicture}
```



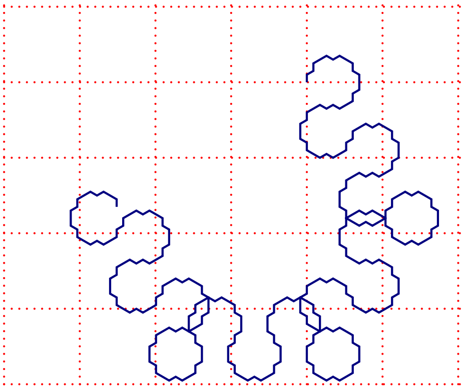
```
\begin{pspicture}(-4,-4)(4,4)
\psgrid[gridlabels=0pt,subgriddiv=0,gridcolor=red,griddots=10]
\pskFibonacci[unit=0.4,linecolor={[rgb]{0 0 0.5}},n=3,k=5](2.8,-3.6)
\end{pspicture}
```



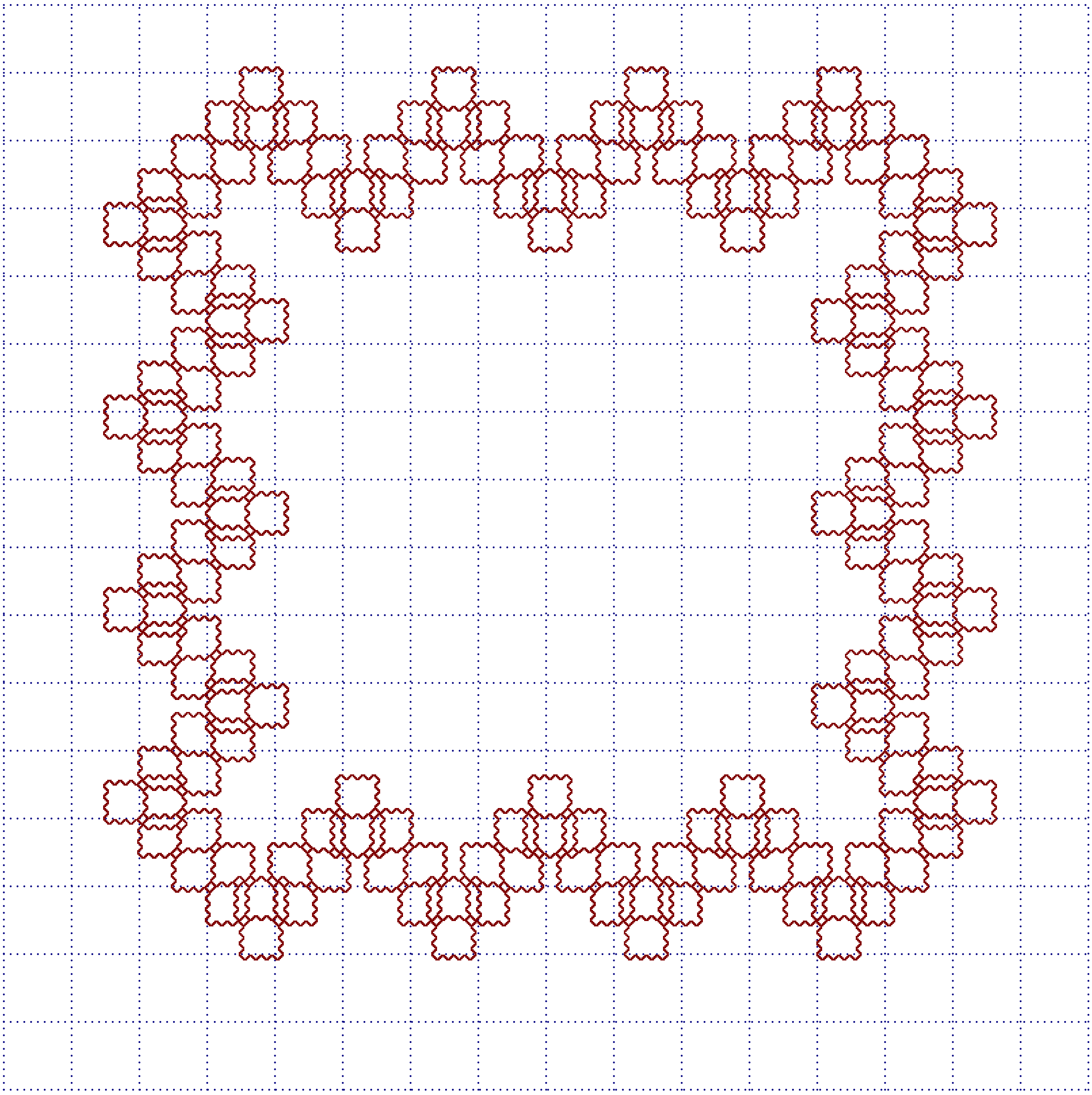
```
\begin{pspicture}(-4,-4)(4,4)
\psgrid[gridlabels=0pt,subgriddiv=0,gridcolor=red,griddots=10]
\pskFibonacci[unit=0.025,linecolor={[rgb]{0 0 0.5}},linewidth=0.02cm,n=6,k=6](3,0.5)
\end{pspicture}
```



```
\begin{pspicture}(-4,-4)(4,4)
\psgrid[gridlabels=0pt,subgriddiv=0,gridcolor={[rgb]{0 0 0.5}},griddots=10]
\pskFibonacci[unit=0.2,linecolor={[rgb]{0 0 0.5}},n=4,k=6,angle=60](-2,0)
\end{pspicture}
```



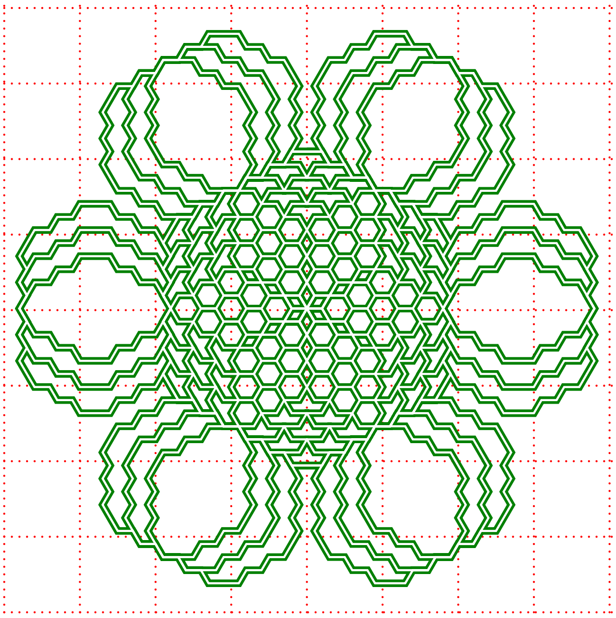
```
\begin{pspicture}(-4,-4)(2,1)
\psgrid[gridlabels=0pt,subgriddiv=0,gridcolor=red,griddots=10]
\pskFibonacci[unit=0.1,linecolor={[rgb]{0 0 0.5}},n=4,k=4,angle=60](0,0)
\end{pspicture}
```



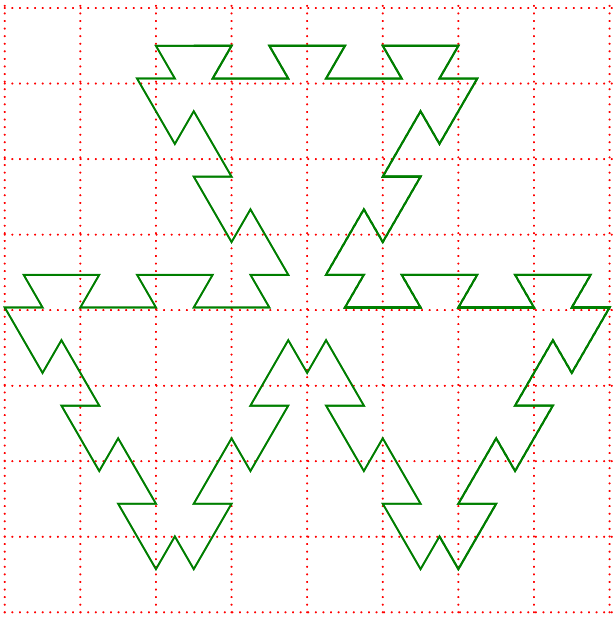
```
\begin{pspicture}(-8,-8)(8,8)
\psgrid[gridlabels=0pt,subgriddiv=0,gridcolor={[rgb]{0 0 0.5}},griddots=10]
\pskFibonacci[unit=0.02,linecolor={[rgb]{0.5 0 0}},linewidth=0.02cm,n=6,k=7](6,-4)
\end{pspicture}
```

13 The command \psBiperiodicFibonacci

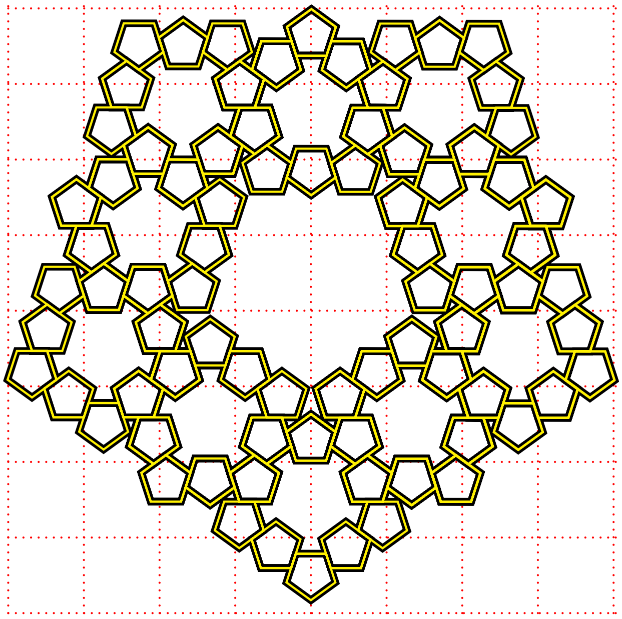
The 3 mandatory parameters are n, a and b. As for the previous command, the coordinates following are optional but allow, eventually, to center the curve at the origin of the mark.



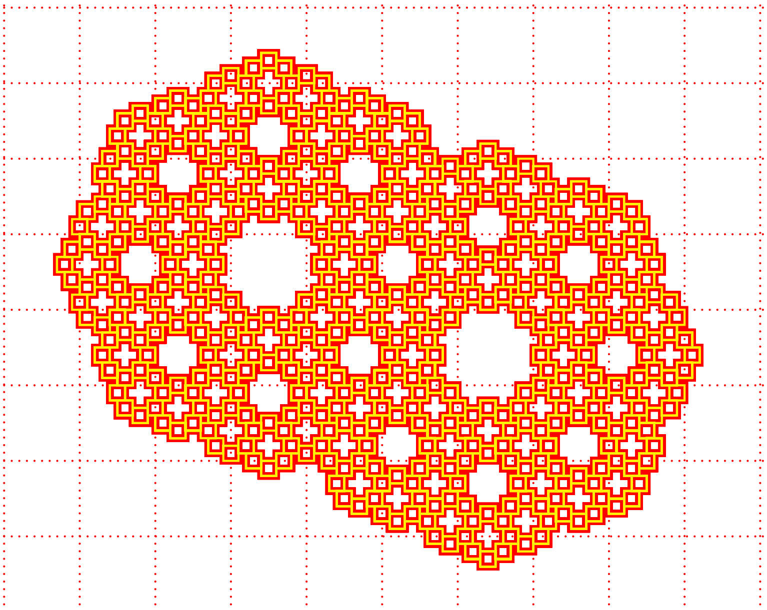
```
\begin{pspicture}[showgrid=false](-4,-4)(4,4)
\psgrid[gridlabels=0pt,subgriddiv=0,gridcolor=red,griddots=10]
\psBiperiodicFibonacci[unit=0.2,linecolor={[rgb]{0 0.5 0}},linewidth=0.1cm,n=5,a=6,b=6,angle
    =60](0,2.1)
\psBiperiodicFibonacci[unit=0.2,linecolor=white,n=5,a=6,b=6,angle=60](0,2.1)
\end{pspicture}
```



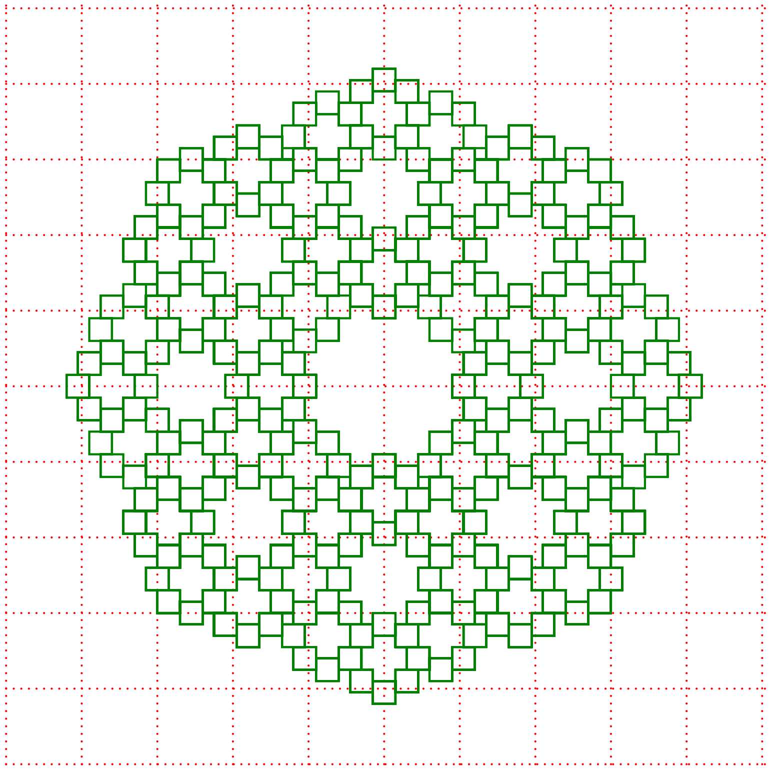
```
\begin{pspicture}[showgrid=false](-4,-4)(4,4)
\psgrid[gridlabels=0pt,subgriddiv=0,gridcolor=red,griddots=10]
\psBiperiodicFibonacci[unit=0.5,linecolor={[rgb]{0 0.5 0}},n=5,a=3,b=4,angle=120](-1.5,3.5)
\end{pspicture}
```



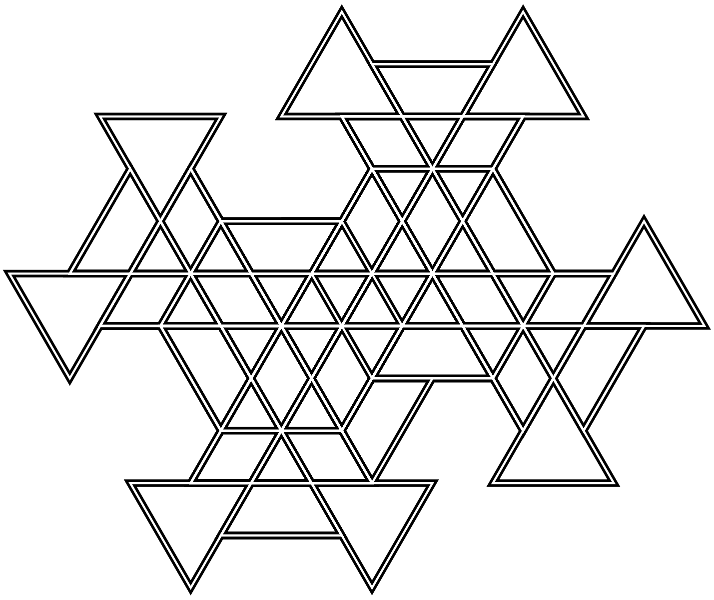
```
\begin{pspicture}[showgrid=false](-4,-4)(4,4)
\psgrid[gridlabels=0pt,subgriddiv=0,gridcolor=red,griddots=10]
\psBiperiodicFibonacci[unit=0.2,linecolor=black,linewidth=0.1cm,n=7,a=2,b=6,angle=72](2.62,2)
\psBiperiodicFibonacci[unit=0.2,linecolor=yellow,n=7,a=2,b=6,angle=72](2.62,2)
\end{pspicture}
```



```
\begin{pspicture}[showgrid=false](-5,-4)(5,4)
\psgrid[gridlabels=0pt,subgriddiv=0,gridcolor=red,griddots=10]
\psBiperiodicFibonacci[unit=0.1,linecolor=red,linewidth=0.1cm,n=10,a=2,b=5](3.5,-1.5)
\psBiperiodicFibonacci[unit=0.1,linecolor=yellow,n=10,a=2,b=5](3.5,-1.5)
\end{pspicture}
```



```
\begin{pspicture}(-5,-5)(5,5)
\psgrid[gridlabels=0pt,subgriddiv=0,gridcolor=red,griddots=10]
\psBiperiodicFibonacci[unit=0.15,linecolor={[rgb]{0 0.5 0}},n=9,a=2,b=5](3.15,-1.35)
\end{pspicture}
```

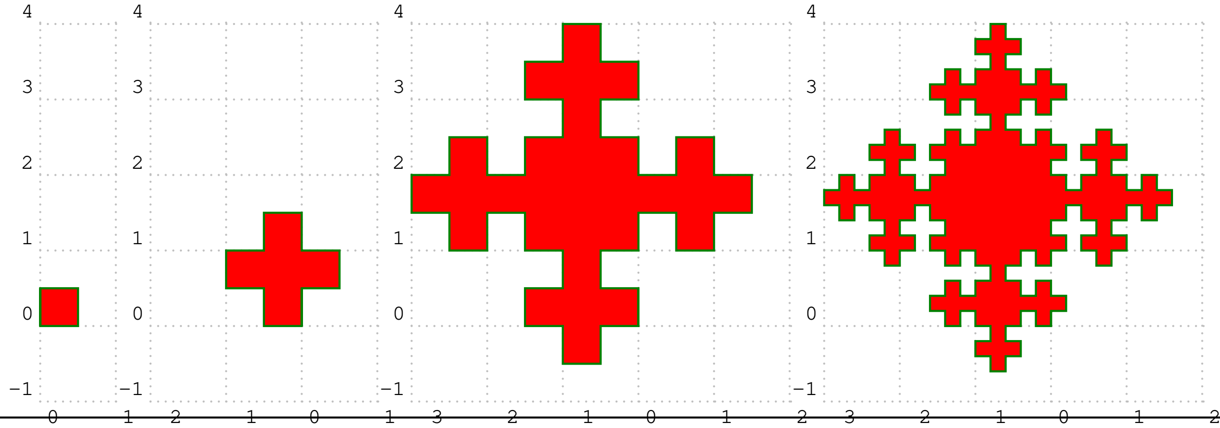


```
\begin{pspicture}(-5,-5)(5,5)
\psBiperiodicFibonacci[unit=0.8,linecolor=black,linewidth=0.1cm,,n=8,a=2,b=3,angle=120](-1,1)
\psBiperiodicFibonacci[unit=0.8,linecolor=white,n=8,a=2,b=3,angle=120](-1,1)
\end{pspicture}
```

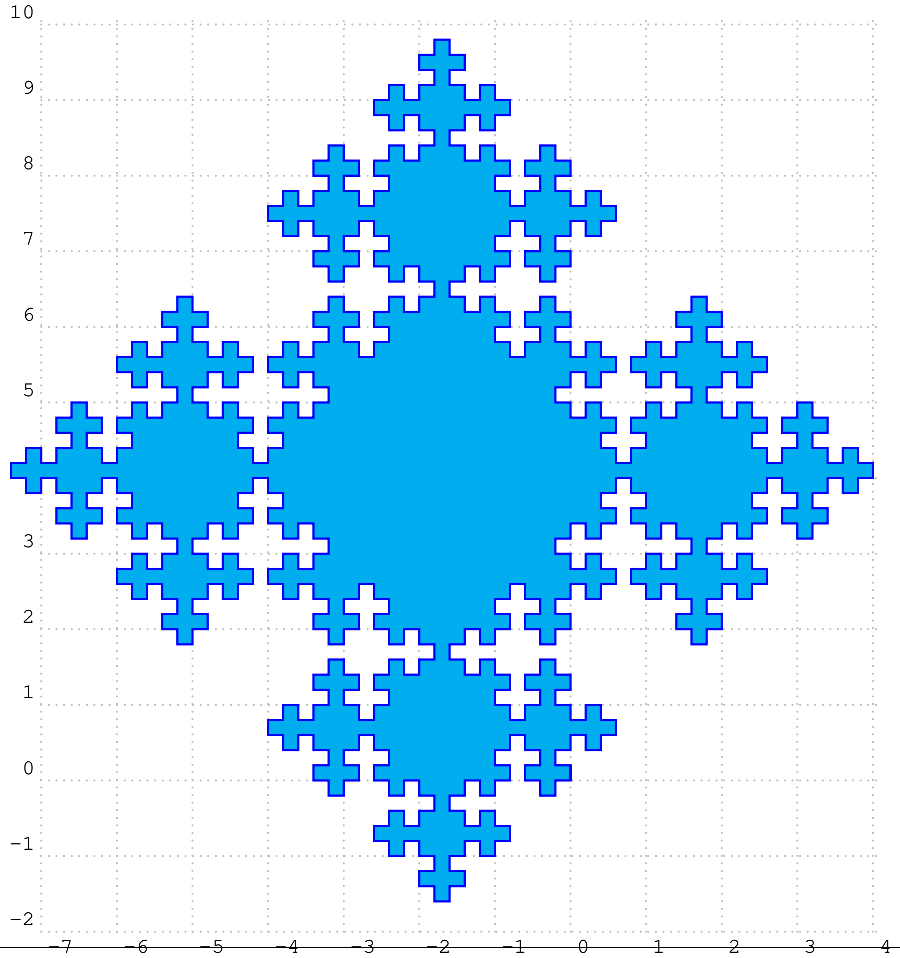
14 The command \psFibonacciPolyominoes

The only mandatory parameter is the order of the tile: n. Coordinates are optional, but they will be used for paving the plan.

14.1 The order 0, 1, 2, 3, and 4



```
\psset{linecolor={[rgb]{0 0.5 0}},fillstyle=solid,fillcolor=red}
\begin{pspicture}[showgrid](0,-1)(1,4)
\psFibonacciPolyominoes[n=0,unit=0.5]
\end{pspicture}
\quad
\begin{pspicture}[showgrid](-2,-1)(1,4)
\psFibonacciPolyominoes[n=1,unit=0.5]
\end{pspicture}
\quad
\begin{pspicture}[showgrid](-3,-1)(2,4)
\psFibonacciPolyominoes[unit=0.5,n=2]
\end{pspicture}
\quad
\begin{pspicture}[showgrid](-3,-1)(2,4)
\psFibonacciPolyominoes[unit=0.2cm,n=3]
\end{pspicture}
```




```
\begin{pspicture}[showgrid](-7,-2)(4,10)
\psFibonacciPolyominoes[unit=0.2,n=4,fillcolor=cyan,linecolor=blue,fillstyle=solid]
\end{pspicture}
```

15 The command \psFibonacciWord

$F_1 = 1$

$F_2 = 0$

$F_3 =$ 01

$F_4 =$ 010

$F_5 =$ 01001

$F_6 =$ 01001010

$F_7 =$ 0100101001001

$F_8 =$ 010010100100101001010

$F_9 =$ 0100101001001010010100100101001001

$F_{10} =$ 0100101001001010010100100101001001010010100100101001010

```
\begin{pspicture}(-1,0)(10,5)
\rput(0.15,5){\small$F_{1}=1$}
\rput(0.15,4.5){\small$F_{2}=0$}
\multido{\i=3+1,\I=3+1,\n=4.0+-0.5}{8}{%
  \psFibonacciWord[n=\i](0.5,\n)
  \rput(0,\n){$F_{\I}=$}}
\end{pspicture}
```

$F_5^{[1]}$ =1011010110110

$F_5^{[2]}$ =010010100100101001010

$F_5^{[3]}$ =00100010010001000100100010010

$F_5^{[4]}$ =0001000010001000010000100010000100010

$F_5^{[5]}$ =0000100000010000100000010000001000010000001000010

$F_5^{[6]}$ =00000100000001000000100000001000000010000001000000010000010

```
\begin{pspicture}(-1,0)(10,6)
\psset{n=5}
\multido{\i=1+1,\I=1+1,\n=3.5+-0.5}{6}{%
\psFibonacciWord[i=\i,iFibonacci](0.5,\n\space 0.1 sub)
\rput(0,\n){$F_{5}^{[\i]}=$}
}
\end{pspicture}
```

16 The Hilbert fractal

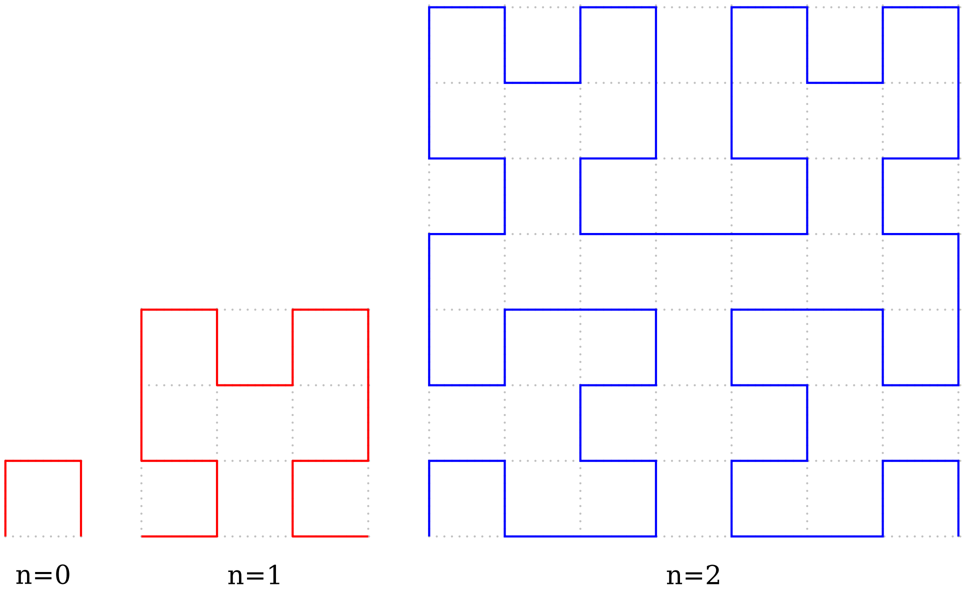
For more informations see https://pstricks.blogspot.com/2018/08/lattracteur-de-henon-mise-jour.html.

```
\psHilbert [Options]
```

- `n=4` : Number of iterations;
- `N=all` : number of points to place, by default all. This option allows you to create an animation in placing and connecting the successive points to the indicated number.
- `dotcolor=red` : color of the points, the size of the points is fixed with the option `dotsize` of PSTricks;
- `showpoints=false` : boolean of PSTricks to display the points.

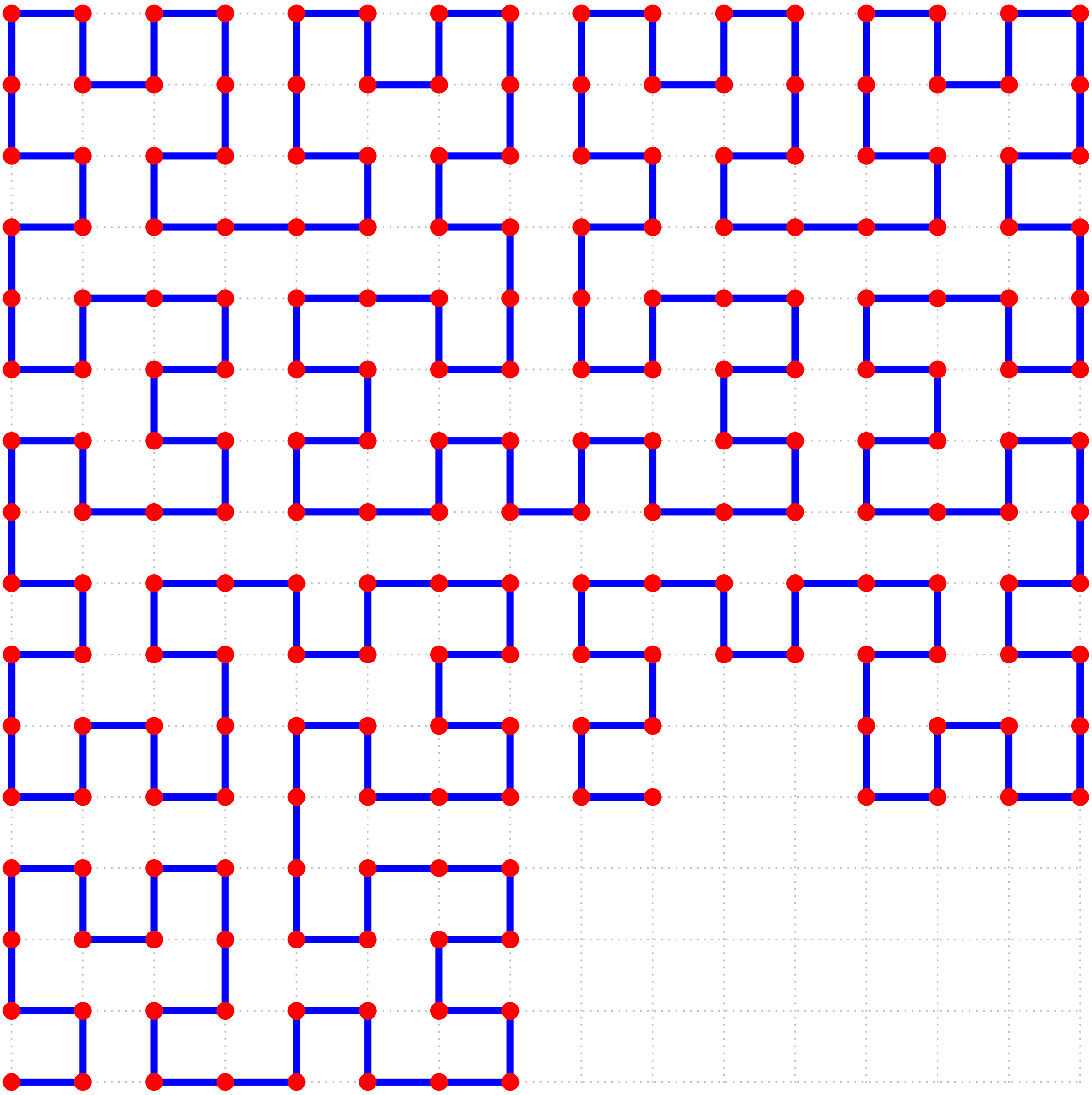
The color and the thickness of the lines are adjusted with the usual PSTricks options: `linecolor` and `linewidth`. The scale of the drawing is defined with the unit option. We can use the option `linestyle=none` , as well as `fillstyle=solid` , `fillcolor` but, in the latter case only, if all the points are kept (`N=all`).

16.1 Examples

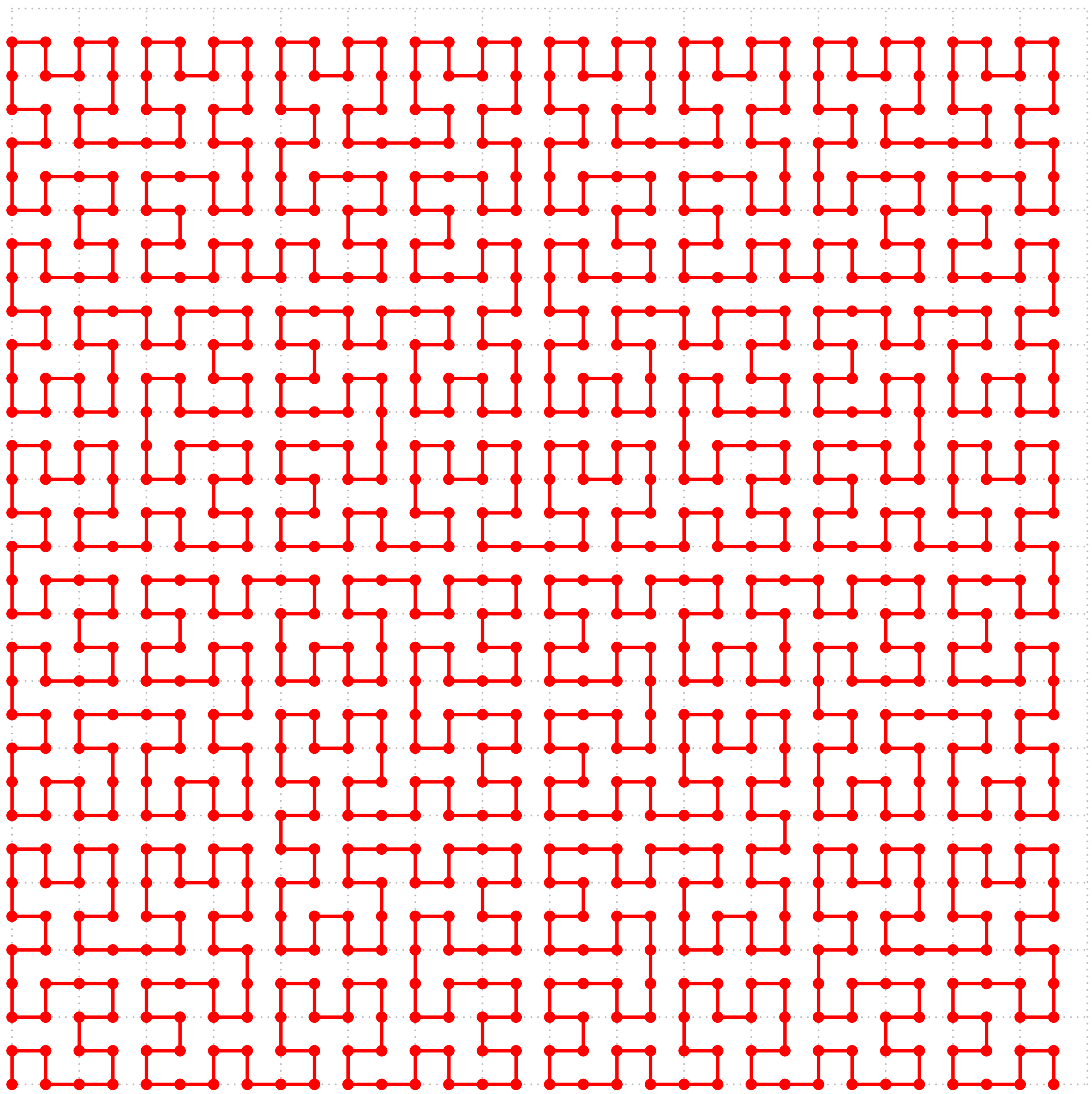



```
\begin{pspicture}(0,-1)(1,1)
\psgrid[subgriddiv=0,gridcolor=lightgray,griddots=10,gridlabels=0pt](1,1)
\psHilbert[linecolor=red,n=0,linejoin=1,fillstyle=solid,fillcolor=blue]
\rput(0.5,-0.5){n=0}
\end{pspicture}
\qquad
\begin{pspicture}(0,-1)(3,3)
\psgrid[subgriddiv=0,gridcolor=lightgray,griddots=10,gridlabels=0pt](3,3)
\psHilbert[linecolor=red,n=1,linejoin=1,fillstyle=solid,fillcolor=blue]
\rput(1.5,-0.5){n=1}
\end{pspicture}
\qquad
\begin{pspicture}(0,-1)(7,7)
\psgrid[subgriddiv=0,gridcolor=lightgray,griddots=10,gridlabels=0pt](7,7)
\psHilbert[linecolor=blue,n=2,showpoints=false,dotsize=0.2,fillstyle=solid,fillcolor=red]
\rput(3.5,-0.5){n=2}
\end{pspicture}
```

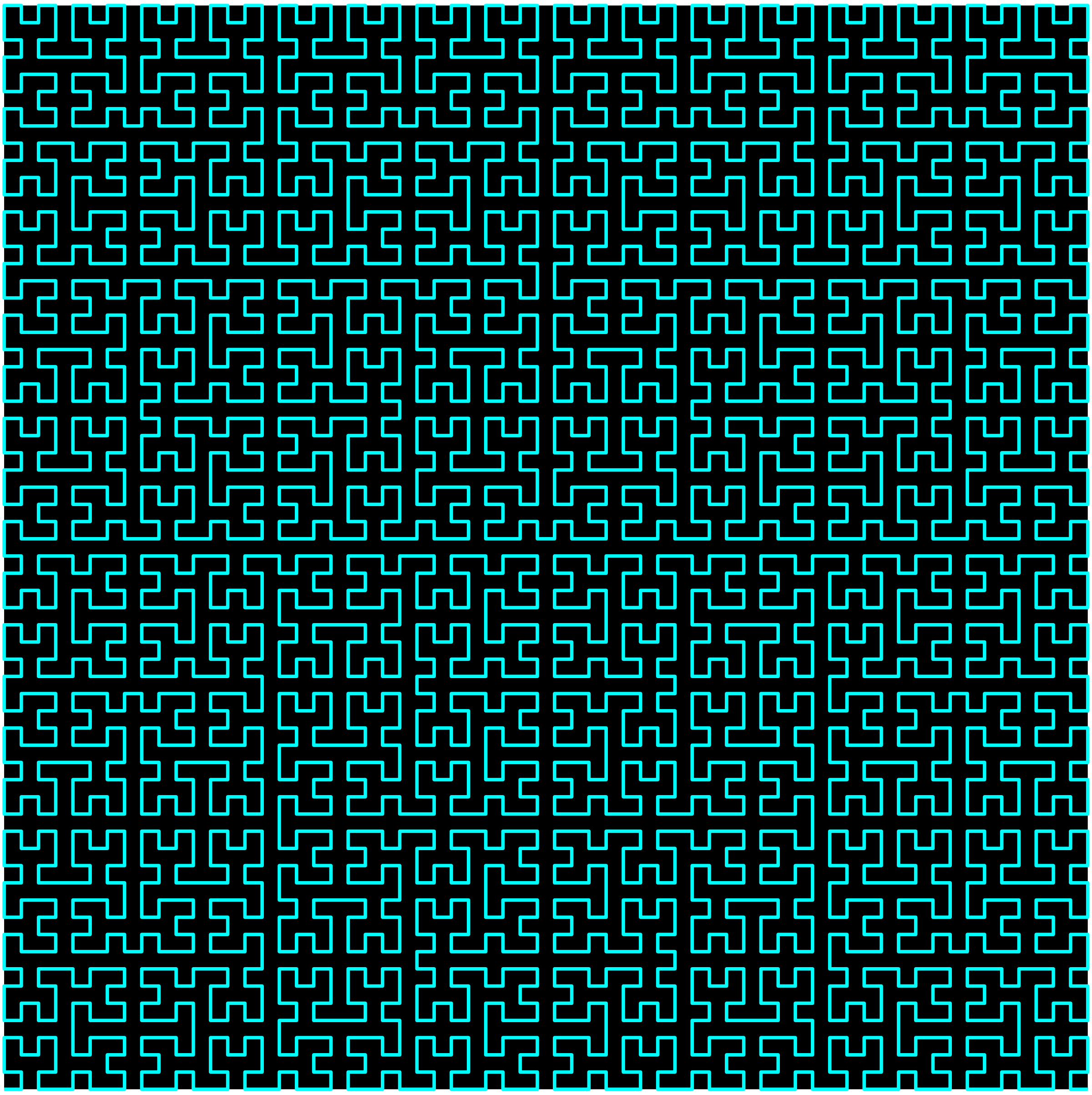
Draw the first 220 of 256 points (n=3):



```
\begin{pspicture}(0,0)(15,16)
% 4^(n+1)=4^4=256
\psgrid[subgriddiv=0,gridcolor=lightgray,griddots=10,gridlabels=0pt](15,15)
\psHilbert[unit=1,linecolor=blue,linewidth=0.1,n=3,showpoints=true,dotsize=0.25,N=220]
\end{pspicture}
```

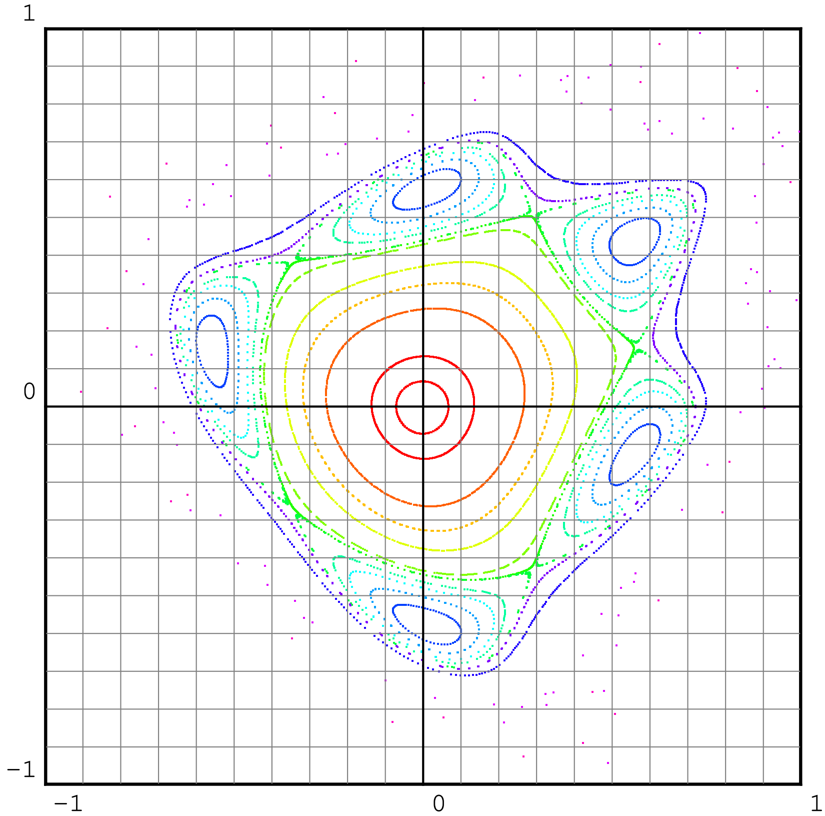


```
\begin{pspicture}(0,0)(16,16)
\psgrid[subgriddiv=0,gridcolor=lightgray,griddots=10,gridlabels=0pt](16,16)
\psHilbert[unit=0.5,linecolor=red,linewidth=0.1,showpoints]
\end{pspicture}
```



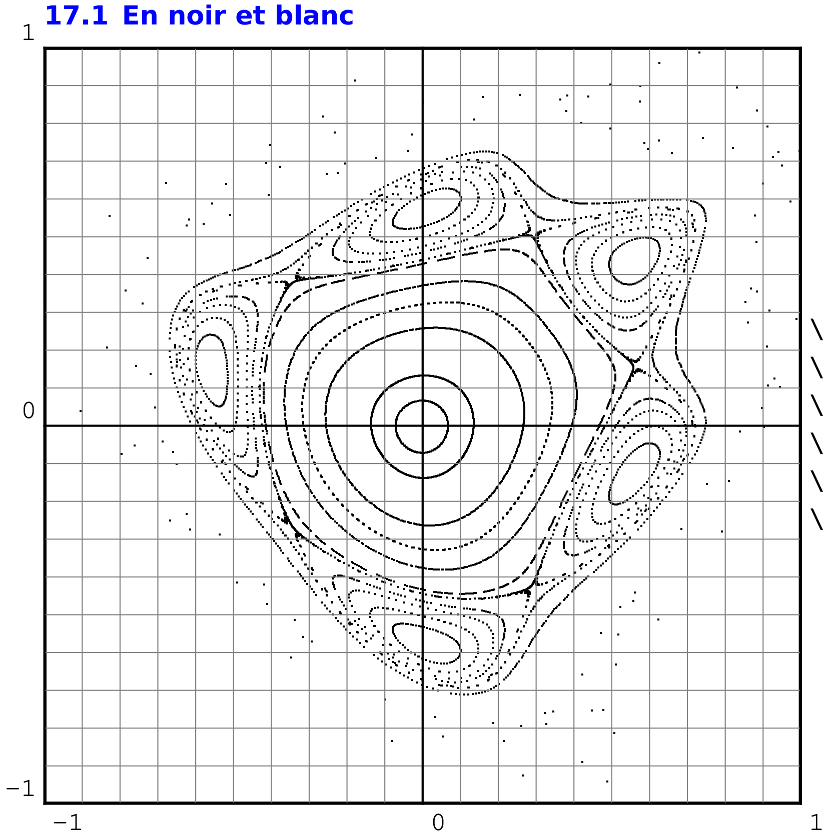
```
\begin{pspicture}(0,0)(16,16)
\psframe*(0,0)(15.75,15.75)
\psHilbert[unit=0.25,linecolor=-red,n=5,linewidth=0.2,linejoin=1,fillstyle=solid,fillcolor=-blue]
\end{pspicture}
```

17 The Henon Sttractor




```
\begin{pspicture}(-5,-6)(5,6)
\psclip{\psframe(-5,-5)(5,5)}
\psHenon
\endpsclip
\psgrid[unit=5,subgriddiv=10](-1,-1)(1,1)
\end{pspicture}
```

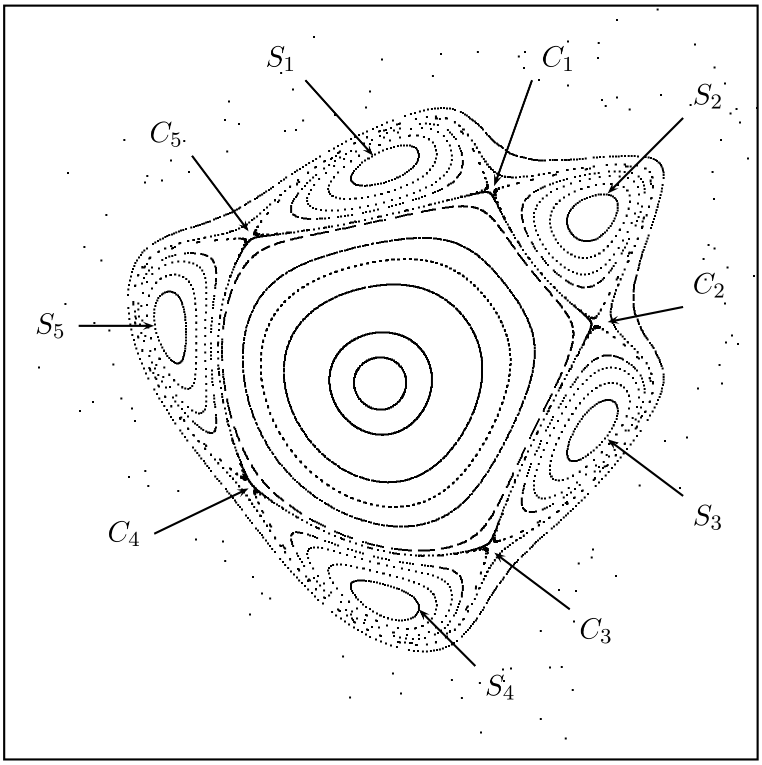
17.1 En noir et blanc



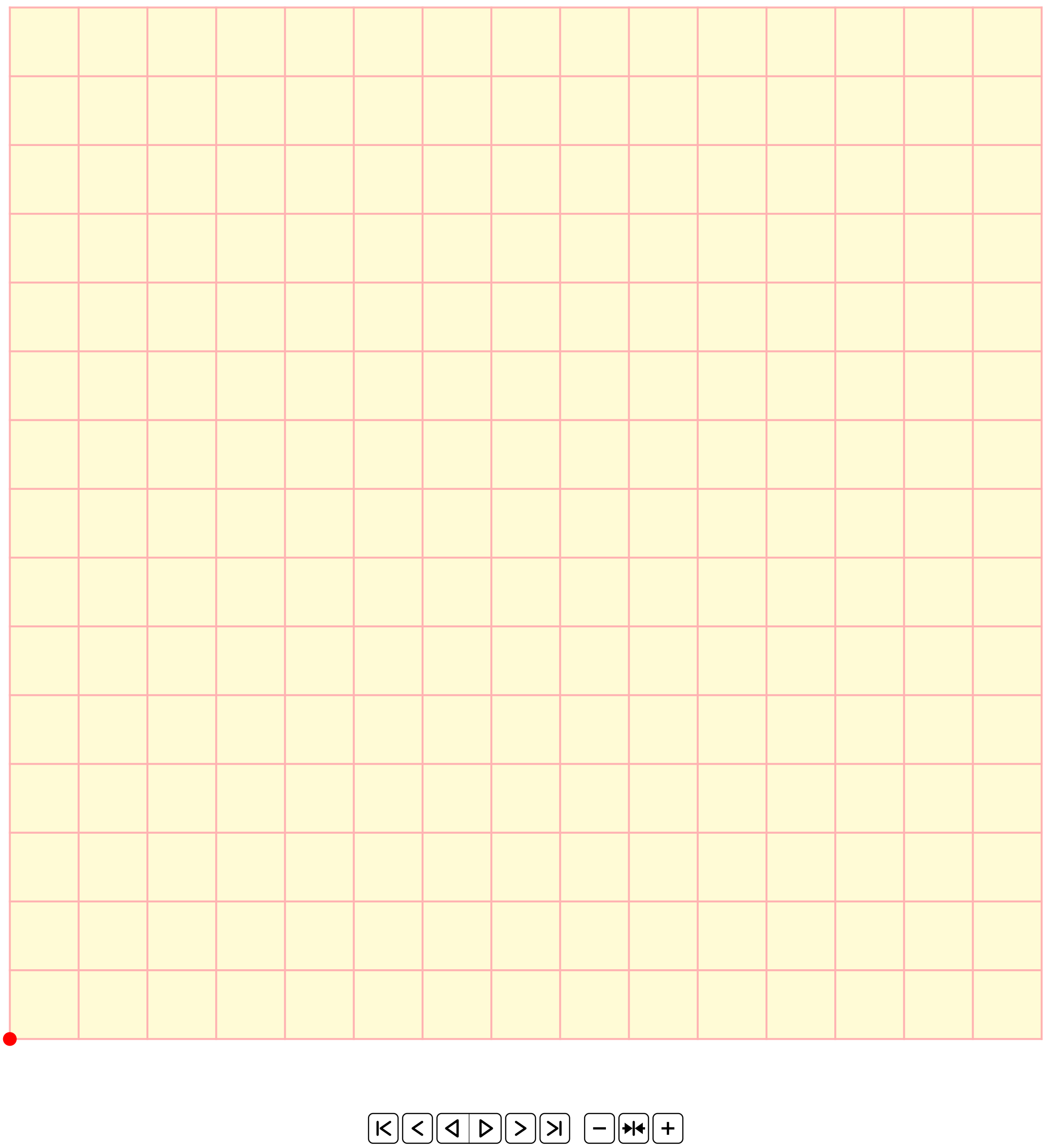

```
\begin{pspicture}(-5,-5)(5,5)
\psclip{\psframe(-5,-5)(5,5)}
\psHenon[pixelscolor=false]
\endpsclip
\psgrid[unit=5,subgriddiv=10](-1,-1)(1,1)
\end{pspicture}
```

18 Animation pour illustrer le principe de l'attracteur de Hénon

En fonction du tableau des points initiaux, la figure obtenue dépend de l'angle a. Les commentaires de Robert Dony correspondent à la premiére figure pour $a = 1.32837$ rd, les voici :



18.1 Animation



```
\begin{animateinline}[controls,palindrome,
                     begin={\begin{pspicture}(-1,-1)(16,16)},
                     end={\end{pspicture}}]{5}% 5 image/s
\multiframe{256}{i=1+1}{%
\psframe*[linecolor=yellow!20](0,0)(15,15)
\psgrid[subgriddiv=1,gridcolor=red!30,gridlabels=0pt](0,0)(15,15)
\psHilbert[linecolor=blue,linewidth=0.05,n=3,showpoints,dotsize=0.2,N=\i]}
\end{animateinline}
```

19 List of all optional arguments for pst-fractal

Key	Type	Default
xWidth	ordinary	1cm
yWidth	ordinary	1cm
type	ordinary	Julia
baseColor	ordinary	white
cx	ordinary	0
cy	ordinary	0
dIter	ordinary	1
maxIter	ordinary	255
maxRadius	ordinary	100
plotpoints	ordinary	2000
angle	ordinary	0
c	ordinary	5
minWidth	ordinary	1pt
scale	ordinary	1
Radius	ordinary	5cm
Color	boolean	true
n	ordinary	[none]
dotcolor	ordinary	[none]
N	ordinary	[none]
i	ordinary	[none]
morphism	ordinary	[none]
juxtaposition	boolean	true
colorF	ordinary	[none]
k	ordinary	[none]
a	ordinary	[none]
b	ordinary	[none]
DFW	boolean	true
iFibonacci	boolean	true
NbrIter	ordinary	[none]
tabPts	ordinary	[none]
angleH	ordinary	[none]
zoom	ordinary	[none]
pixelscolor	boolean	true

References

[1] Denis Girou. “Présentation de PSTricks”. In: *Cahier GUTenberg* 16 (Apr. 1994), pp. 21–70.

[2] Michel Goosens et al. *The LaTeX Graphics Companion*. 2nd ed. Reading, Mass.: Addison-Wesley Publishing Company, 2007.

[3] Nikolai G. Kollock. *PostScript richtig eingesetzt: vom Konzept zum praktischen Einsatz*. Vaterstetten: IWT, 1989.

[4] José L. Ramírez and Gustavo N. Rubiano. *Properties and Generalizations of the Fibonacci Word Fractal Exploring Fractal Curves*. URL: http://www.mathematica-journal.com/2014/02/properties-and-generalizations-of-the-fibonacci-word-fractal/.

[5] Herbert Voß. “Die mathematischen Funktionen von PostScript”. In: *Die TEXnische Komödie* 1/02 (Mar. 2002).

[6] Herbert Voß. *`pst-tools` – Helper functions*. 2012. URL: /graphics/pstricks/contrib/pst-tools.

[7] Herbert Voß. *PSTricks – Grafik für TEX und LATEX*. 7th ed. Heidelberg and Berlin, 2016.

[8] Herbert Voß. *PSTricks – Graphics for TEX and LATEX*. Cambridge: UIT, 2011.

[9] Herbert Voß. *LATEX quick reference*. Cambridge: UIT, 2012.

[10] Eric Weisstein. *Wolfram MathWorld*. 2007. URL: http://mathworld.wolfram.com.

[11] Timothy van Zandt. *PSTricks - `PostScript` macros for generic TEX*. 1993. URL: http://www.tug.org/application/PSTricks.

[12] Timothy van Zandt and Denis Girou. “Inside PSTricks”. In: *TUGboat* 15 (Sept. 1994), pp. 239–246.

Index