Plane-filling curves on all uniform grids

Jörg Arndt, <arndt@jjj.de>
Technische Hochschule Nürnberg

Mons, Belgium, Friday, August 25, 2017

Abstract

We describe a search for plane-filling curves traversing all edges of a grid once. The curves are given by Lindenmayer systems with only one non-constant letter. All such curves for small orders on three grids have been found. For all uniform grids we show how curves traversing all points once can be obtained from the curves found. Curves traversing all edges once are described for the four uniform grids where they exist.

Contents

1 Introduction

1.1 Self-avoiding edge-covering curves on a grid

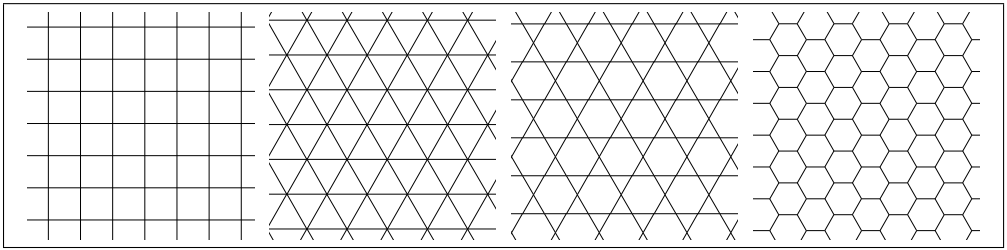

Figure 1.1-A: From left to right: square grid, triangular grid, tri-hexagonal grid, and hexagonal (honeycomb) grid.

A curve is *self-avoiding* if it neither crosses itself nor has an edge that is traversed twice. *self-avoiding*

It is *edge-covering* if it traverses all edges of some grid. *edge-covering*

For edge-covering curves to exist, the grid must have an even number of incident edges at each point. Otherwise a dead end is produced after the point is traversed sufficiently often. This rules out the hexagonal (honeycomb) grid as every point has incidence 3.

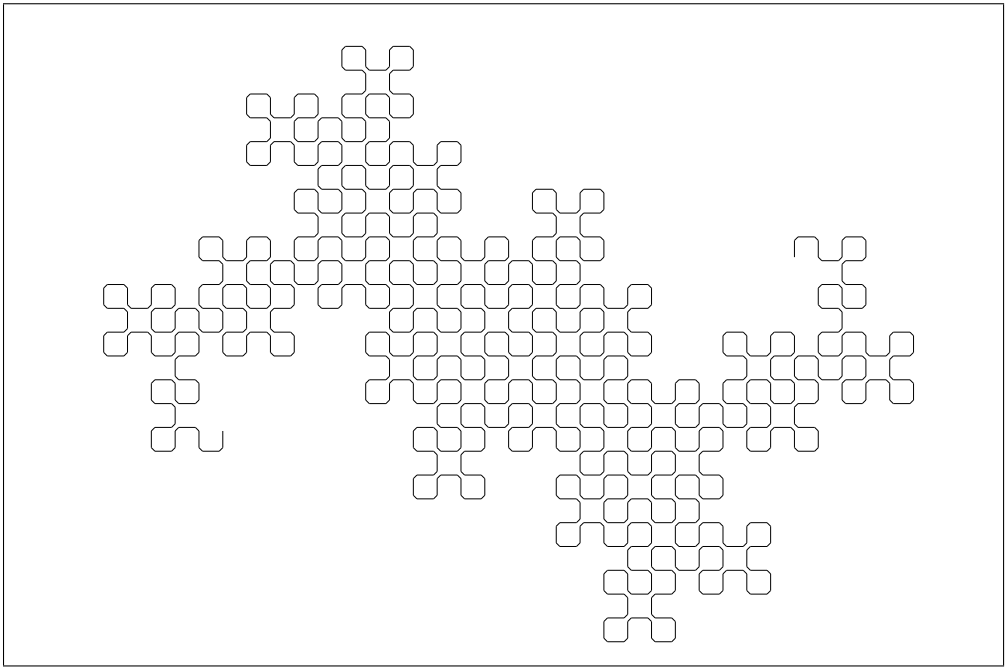

Figure 1.1-B: The R5-dragon, a curve on the square grid (`R5-1`).

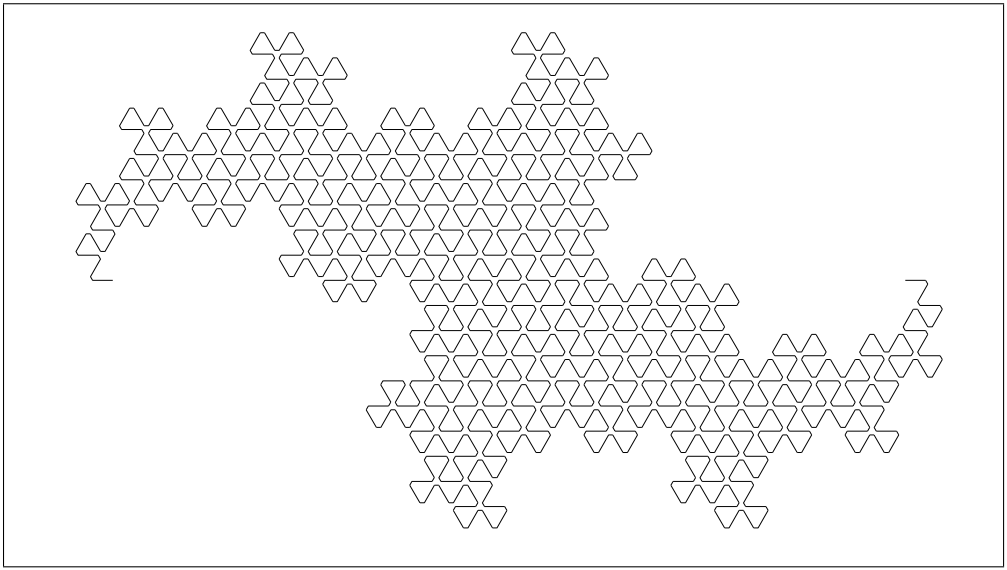

Figure 1.1-C: The terdragon, a curve on the triangular grid (`R3-1`).

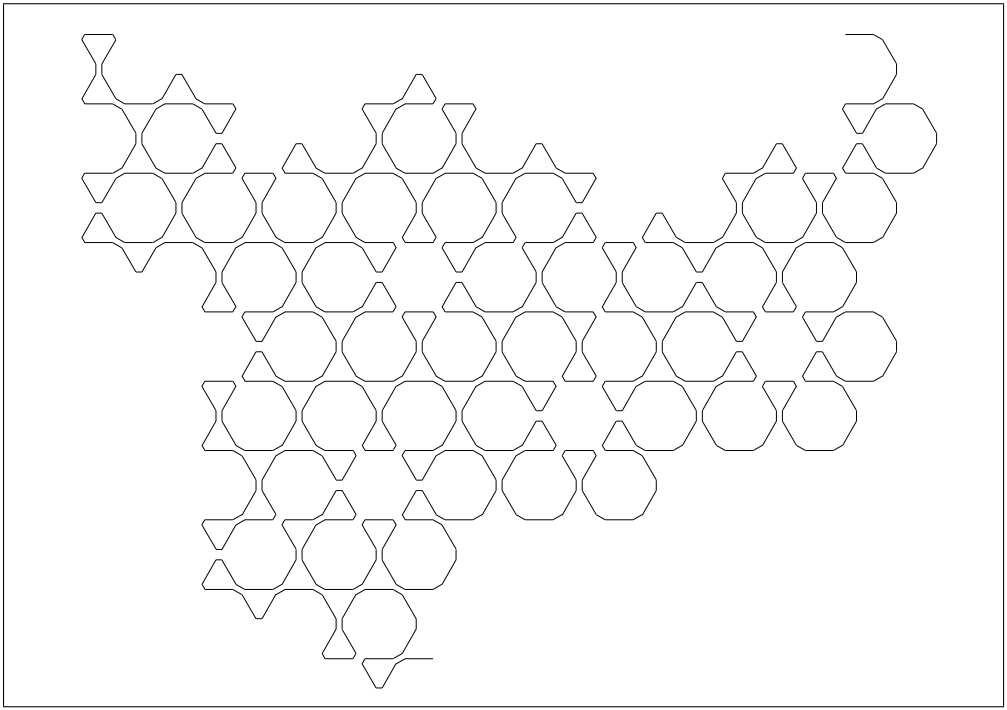

Figure 1.1-D: Ventrella's curve, a curve on the tri-hexagonal grid (`R7-1`).

1.2 Description via simple Lindenmayer-systems

A *Lindenmayer-system* is a triple (Ω, A, P) where Ω is an alphabet, A a word over Ω (called the *axiom*), and P a set of maps from letters $\in \Omega$ to words over Ω that contains one map for each letter.

Lindenmayer-system

axiom

The word that a letter is mapped to is called the *production* of the letter. If the map for a letter is the identity, we call the letter a *constant* of the L-system.

production

constant

We specify curves by L-systems interpreted as a sequence of (unit-length) edges and turns. Letters are interpreted as "draw a unit stroke in the current direction", + and − as turns by $\pm$ a fixed angle ϕ (set to either $60°$, $90°$, or $120°$). We will also use the constant letter 0 for turns by $0°$ (non-turns).

We call an L-system *simple* if it has just one non-constant letter. Only curves with simple L-systems are considered for the search to keep the search space manageable. *simple*

For simple L-systems we always use `F` for the only non-constant letter. The axiom (`F`) and the maps for the constant letters (+ $\mapsto$ +, - $\mapsto$ -) will be omitted.

The *order* R of a curve is the number of `F`s in the production of `F` by the (simple) L-system. *order*

For example,
the terdragon has order $R = 3$ (`F` $\mapsto$ `F+F-F`)
and the R5-dragon has order $R = 5$ (`F` $\mapsto$ `F+F+F-F-F`).

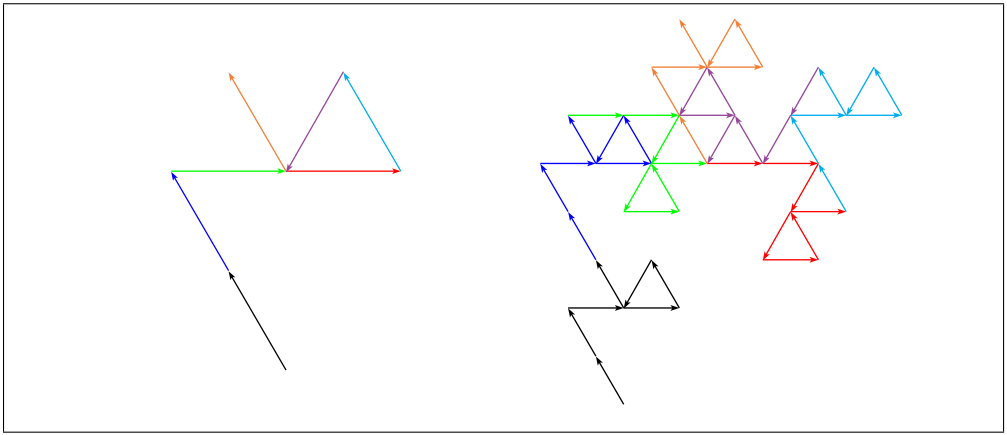

Figure 1.2-A: First iterate (motif) and second iterate of a curve of order 7 (`R7-1`). The L-system is `F` $\mapsto$ `F0F+F0F-F-F+F`.

We call the curve corresponding to the nth iterate of an L-system the nth *iterate of the curve*.

iterate of the curve

We call the first iterate the *motif* of the curve.

motif

Iterate 0 corresponds to a single edge and iterate n is obtained from iterate $n-1$ by replacing every edge by the motive.

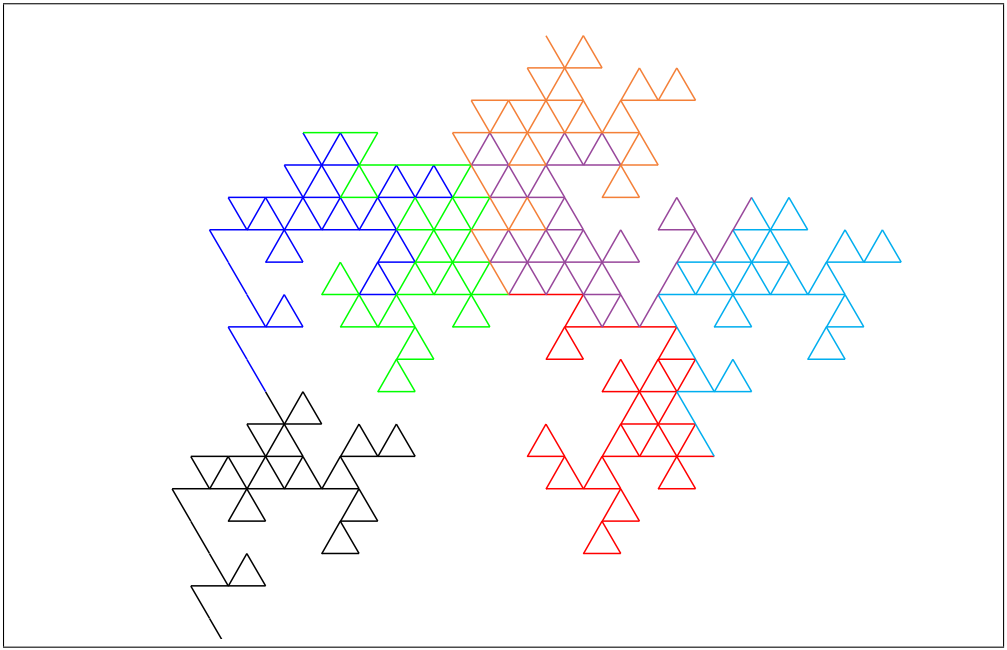

Figure 1.2-B: Third iterate of the curve `R7-1`.

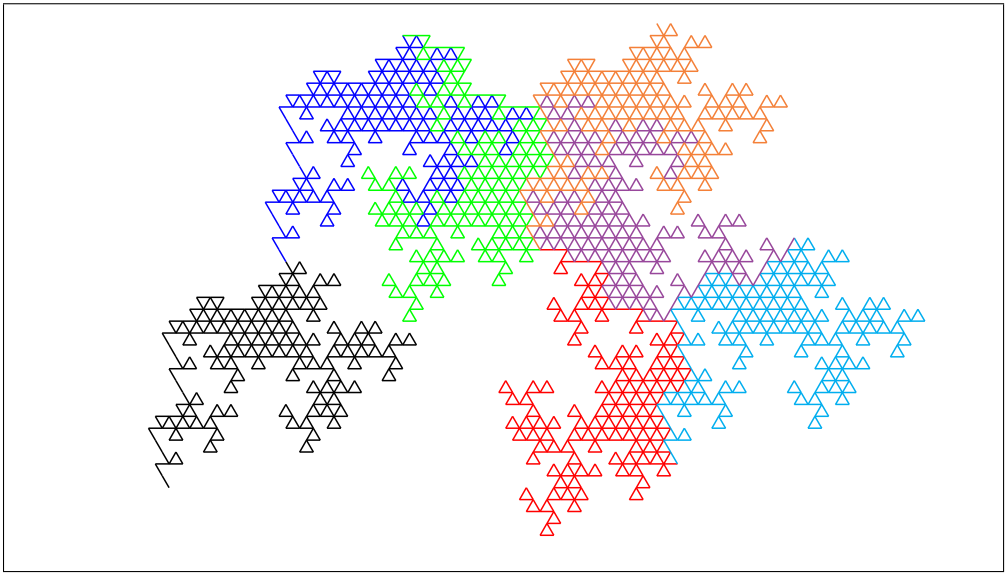

Figure 1.2-C: Fourth iterate of the curve `R7-1`.

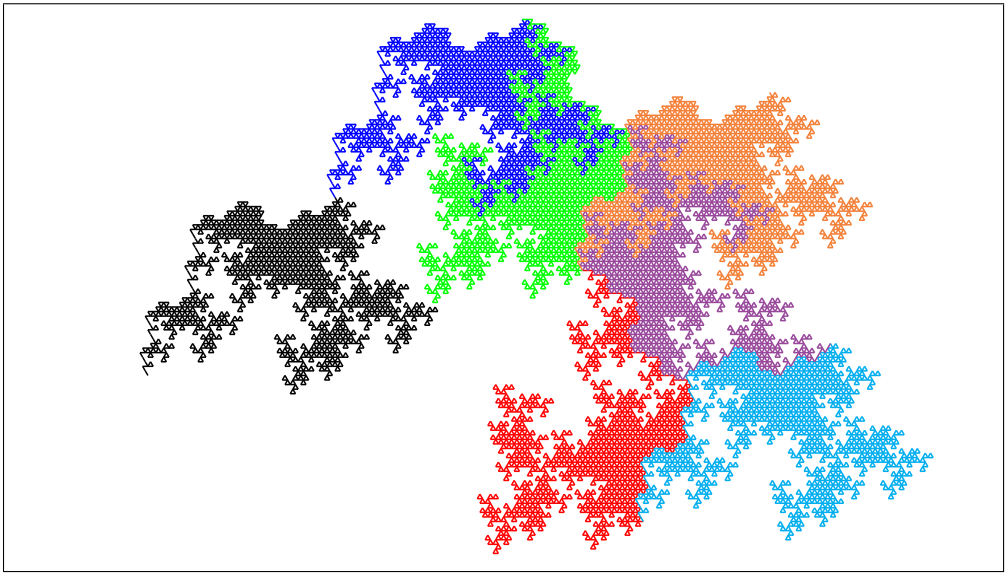

Figure 1.2-D: Fifth iterate of the curve R7−1.

2 The search

2.1 Conditions for a curve to be self-avoiding and edge-covering

Let C_n be the nth iterate of the curve corresponding to the L-system and C be the set of all iterates C_n of the curve $n \geq 0$.

We say C is self-avoiding or edge-covering if every curve in C has the respective property.

2.1.1 Sufficient conditions

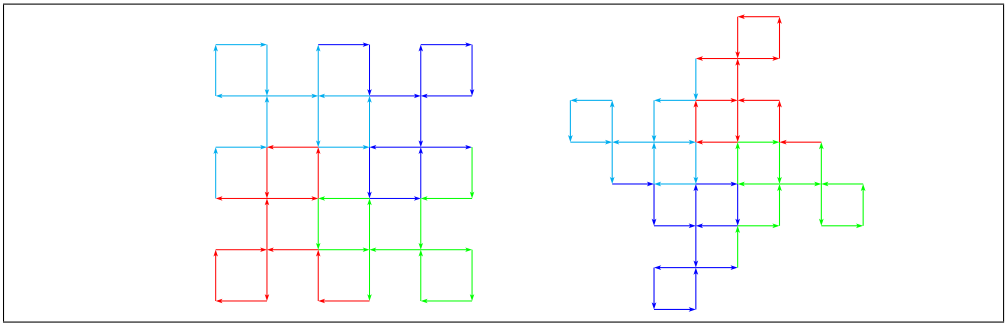

Figure 2.1-A: Tiles Θ_{+1} (left) and Θ_{-1} (right) for the curve of order 13 on the square grid with L-system `F` $\mapsto$ `F+F-F-F+F+F+F-F+F-F-F-F+F` (`R13-3`).

We need the concept of a *tile* for the following facts. *tile*

For the square grid, let Θ_{+1} be the (closed) curve corresponding to the first iterate of the map of the L-system with axiom `F+F+F+F` and Θ_{-1} for the axiom `F-F-F-F`.

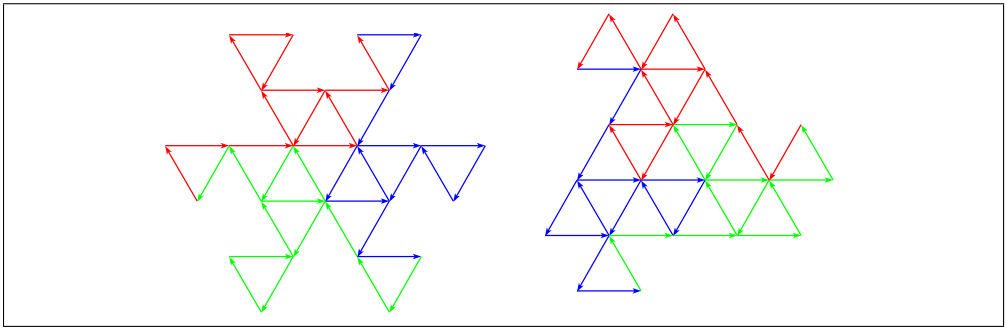

Figure 2.1-B: Tiles Θ_{+1} (left) and Θ_{-1} (right) for the curve of order 13 on the triangular grid with L-system `F` $\mapsto$ `F+F0F0F-F-F+F0F+F+F-F0F-F` (R13-15).

For the triangular grid, the respective axioms are `F+F+F` and `F-F-F`, and the turns are by $\phi = 120°$.

The tiles of edge-covering curves do indeed tile the grid: infinitely many disjoint translations of them do cover all edges of the grid.

2.1.2 Sufficient conditions

Fact 1 (Tiles-SA)**.** *C is self-avoiding if and only if both tiles Θ_{+1} and Θ_{-1} are self-avoiding.*

We call a tile edge-covering if all edges in its interior are traversed once.
Fact 2 (Tiles-Fill)**.** *C is edge-covering if and only if both tiles Θ_{+1} and Θ_{-1} are edge-covering.*

Proof by authority (Michel Dekking).

2.1.3 Necessary conditions

The motif must obviously be self-avoiding:

Fact 3 (Obv)**.** *For C to be self-avoiding, C_1 must be self-avoiding.*

Fact 4 (Turn)**.** *For C to be self-avoiding and edge-covering, the net rotation of the curve C_1 must be zero.*

That is, the number of + and − in X must be equal.

Fact 5 (Dist)**.** *For C to be self-avoiding and edge-covering, the squared distance between the start and the endpoint of the curve must be equal to R.*

For the square grid, the possible orders are the odd numbers of the form $x^2 + y^2$ (*Gaussian integers*).

Gaussian integers

For the triangular grid the possible orders are of the form $x^2 + x\,y + y^2$ (equivalently, numbers of the form $3\,x^2 + y^2$; *Eisenstein integers*).

Eisenstein integers

2.2 The shape of a curve

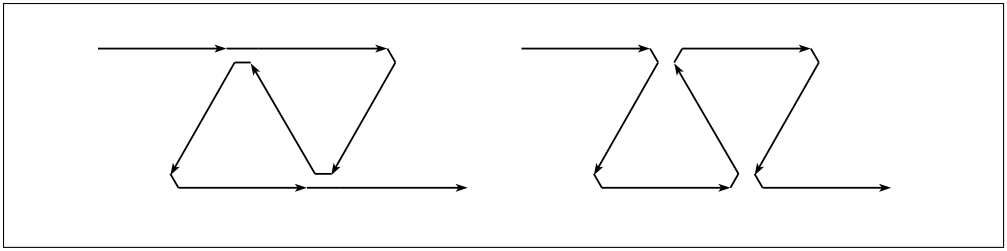

Figure 2.2-A: Two different curves of order 7 with same shape (`R7-2` and `R7-5`).

The *shape* of a curve is the set of grid points traversed in the first iterate. Different curves can have the same shape. The L-systems are respectively `F` $\mapsto$ `F0F+F+F-F-F0F` and `F` $\mapsto$ `F+F-F-F+F+F-F`. *shape*

We consider two curves to be of the same shape whenever any transformation of the symmetry of the underlying grid (rotations and flips) maps one shape into the other.

If two curves have the same shape, we call them *similar*. This is an equivalence relation. *similar*

2.3 Structure of the program for searching

The program consists of the following parts:

generation of the L-systems,

testing of the corresponding curves,

detection of similarity to shapes seen so far,

and detection of symmetries.

Getting it fast was *not* easy.

2.4 Format of the files specifying the curves

```
F F+F+F-F+F-F-F-F+F-F+F+F+F-F+F-F-F  R17-1  # # symm-dr
F F+F+F-F-F+F+F+F-F+F-F-F-F+F+F-F-F  R17-2  # # symm-dr
F F+F+F-F-F+F+F+F-F-F+F+F-F-F-F+F-F  R17-3  #
F F+F+F-F-F-F+F+F+F-F+F+F-F-F-F+F-F  R17-4  # # symm-r ## same = 1 P R
F F+F-F+F+F+F-F-F+F+F-F-F-F+F+F-F-F  R17-5  # ## same = 3 R X
F F+F-F+F+F+F-F-F+F-F+F+F-F-F-F+F-F  R17-6  # # symm-dr
F F+F-F+F+F+F-F-F+F-F-F-F+F+F+F-F-F  R17-7  # # symm-r ## same = 1 P R
F F+F-F+F+F+F-F-F-F+F+F+F-F-F-F+F-F  R17-8  # # symm-dr ## same = 1 P R
F F+F-F+F+F-F+F+F+F-F-F-F+F-F-F+F-F  R17-9  # # symm-dr ## same = 1 P R
F F+F-F+F+F-F+F+F-F-F-F+F+F-F-F-F+F  R17-10  #
F F+F-F+F+F-F+F-F+F+F-F-F-F+F-F-F+F  R17-11  #
F F+F-F-F+F-F-F-F+F+F-F+F-F+F+F-F+F  R17-12  # ## same = 11 Z T
F F+F-F-F-F+F+F-F-F-F+F+F-F+F+F-F+F  R17-13  # ## same = 10 Z T
```

Descriptions of the curves of order 17 on the square grid.

2.5 Numbers of shapes found

Triangular grid, sequence A234434 in the OEIS (http://oeis.org/):
3:1, 4:1, 7:3, 9:5, 12:10, 13:15, 16:17, 19:71, 21:212, 25:184, 27:543, 28:842, 31:1848.

Square grid, sequence A265685:
5:1, 9:1, 13:4, 17:6, 25:33, 29:39, 37:164, 41:335, 49:603, 53:2467.

Tri-hexagonal grid, sequence A265686:
7:1, 13:3, 19:7, 25:10, 31:63, 37:157, 43:456, 49:1830, 61:8538.

The number of curves is much greater than the number of shapes. For example, for order $R = 53$ on the square grid there are 2467 shapes and 401738 curves, so about 162 curves share one shape on average.

3 Properties of curves and tiles

3.1 Self-similarity, symmetries, and tiling property

All curves are self-similar by construction: every curve of order R can be decomposed into R disjoint rotated copies of itself.

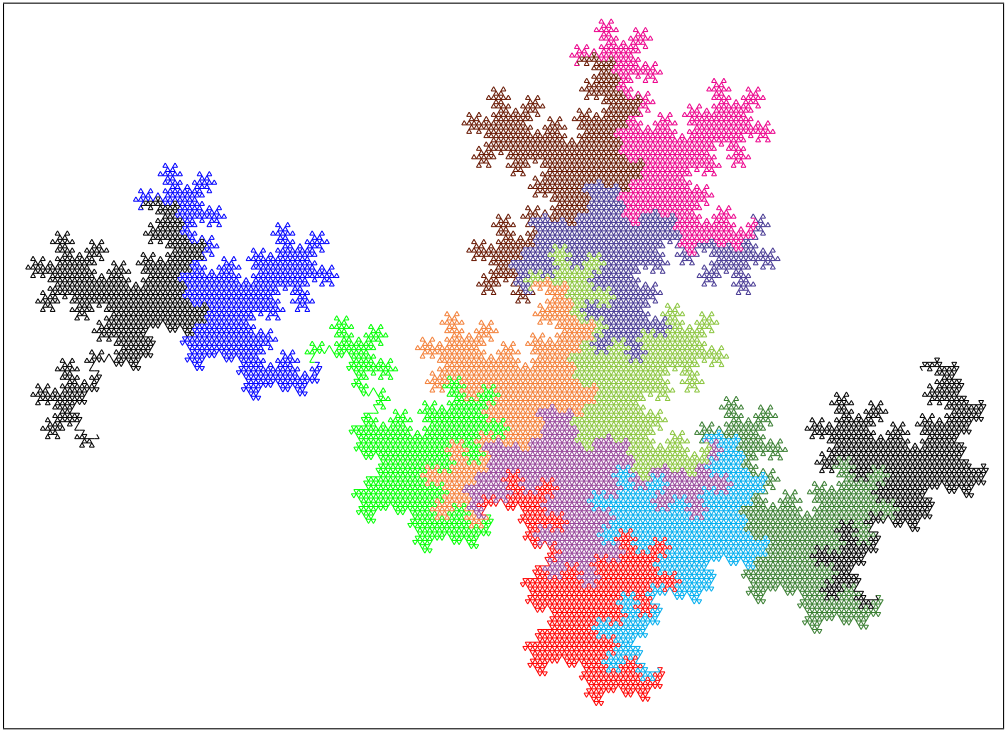

Figure 3.1-A: Self-similarity of the order-13 curve R13−15 on the triangular grid.

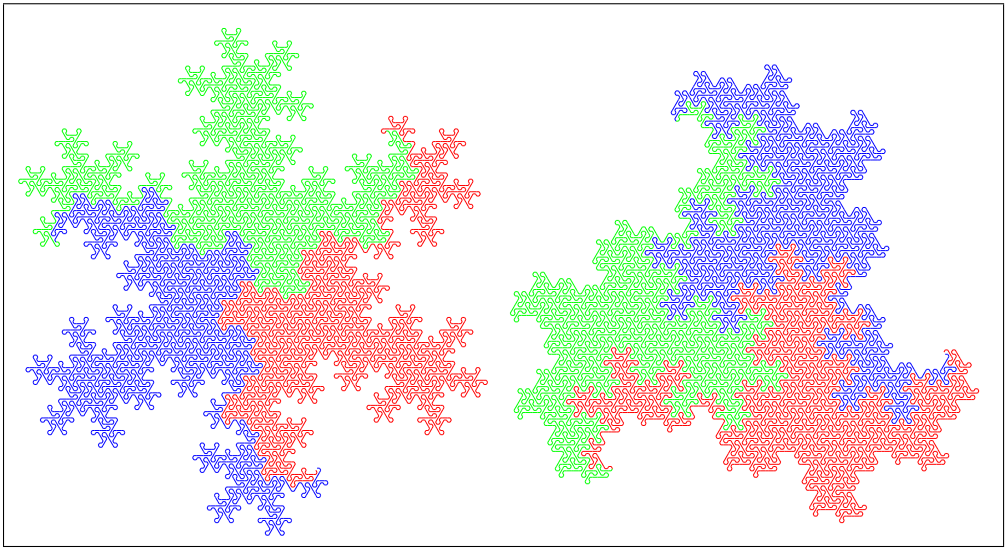

Figure 3.1-B: The two tiles Θ_{+3} and Θ_{-3} of the curve in the previous figure (R13-15).

Let Θ_{+k} and Θ_{-k} be the tiles for the kth iterate of a curve, and $\Theta_{+\infty}$ and $\Theta_{-\infty}$ the limiting shape of the tiles.

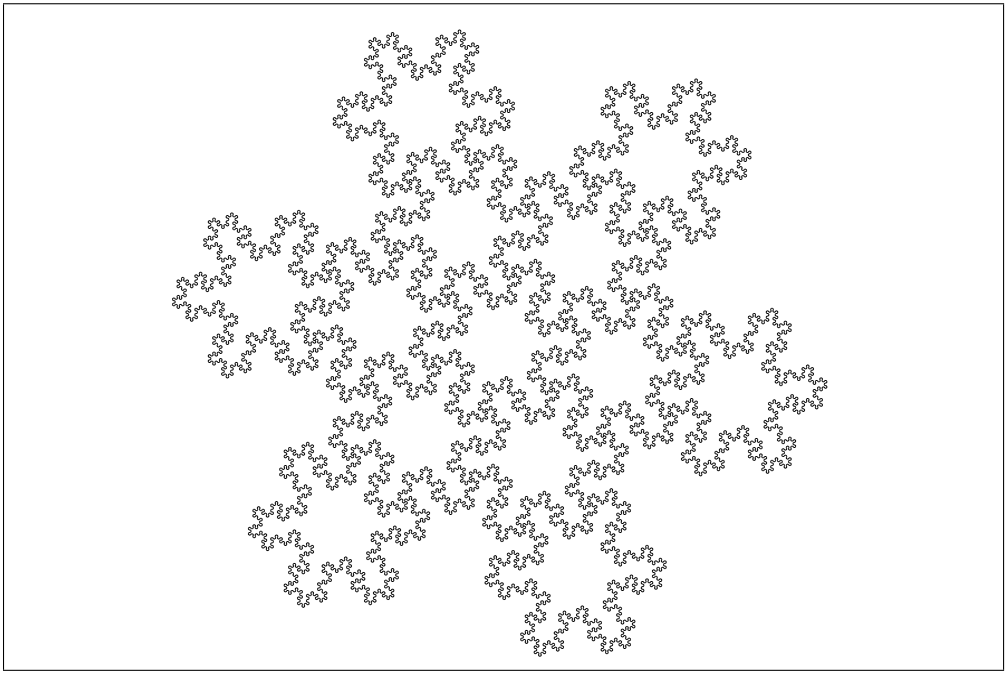

Figure 3.1-C: The shape $\Theta_{+\infty}$ of the tile at the left in the previous figure, decomposed into 13 smaller rotated copies of itself.

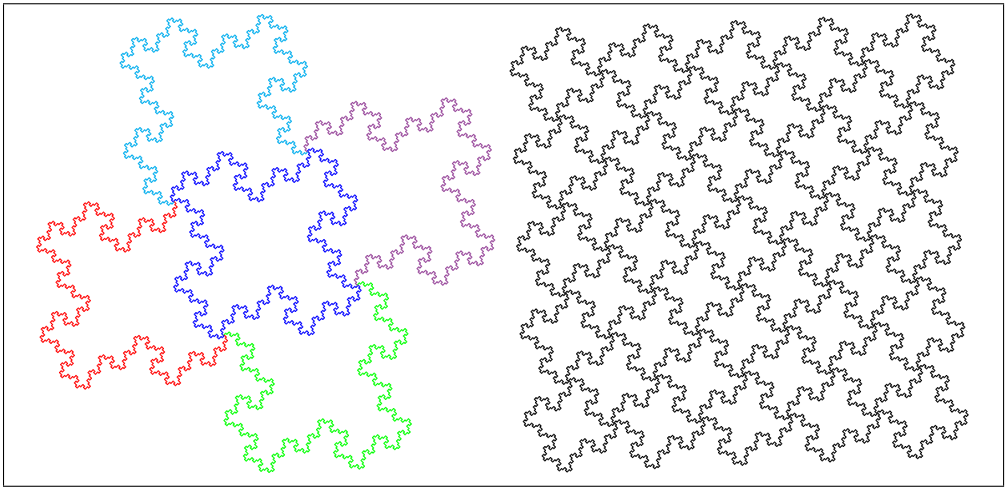

Figure 3.1-D: Decomposition of the tile $\Theta_{+\infty}$ of the R5-dragon ($\mathtt{R5{-}1}$) into rotated copies of itself (left) and the tiling of the plane by such tiles (right).

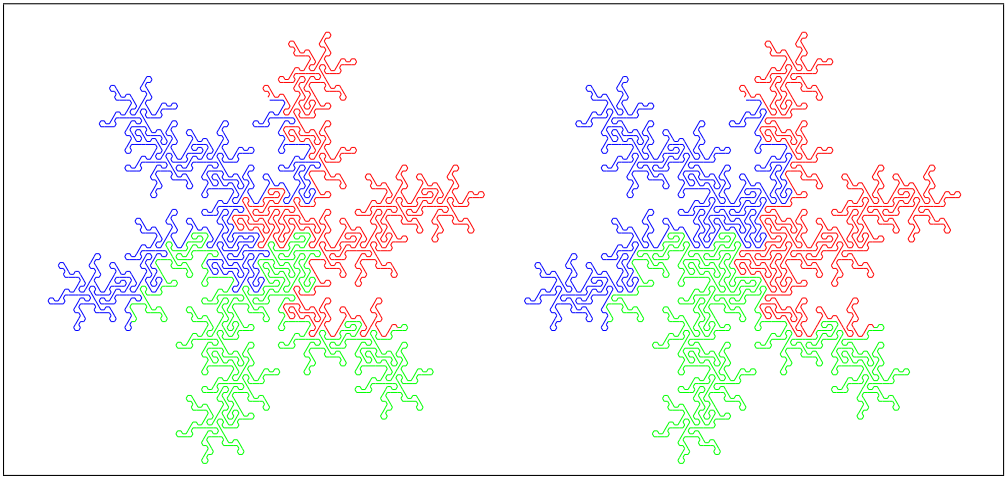

Figure 3.1-E: Two curves with different shapes but same shape of the tiles Θ_{-k} for all k, shown are the tiles Θ_{-2} (R25−247 and R25−248).

3.2 Tiles and complex numeration systems

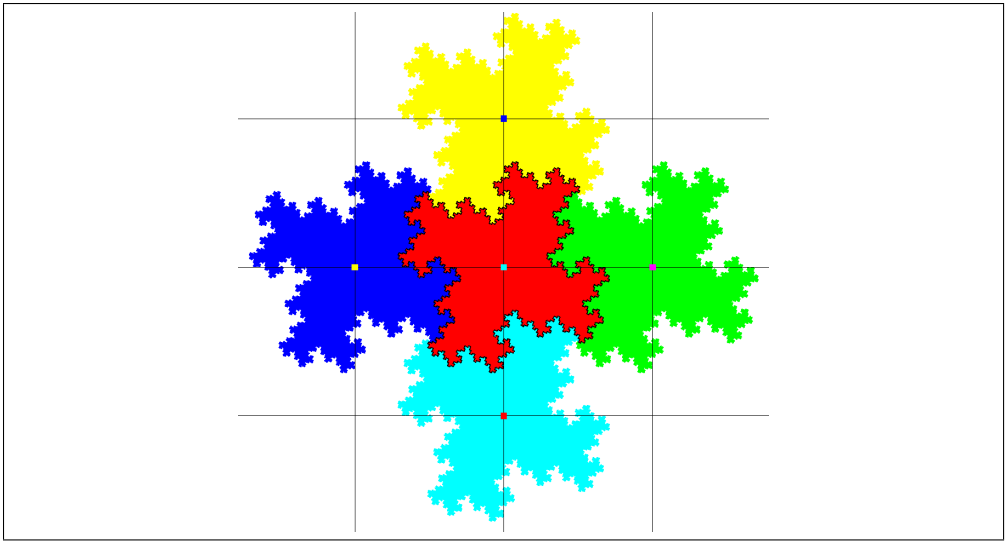

Figure 3.2-A: The fundamental region for the complex numeration system with base $2+i$ and digit set $\{0, +1, -1, +i, -i\}$ where $i = \sqrt{-1}$. The set is scaled up by the factor $\sqrt{5}$ to let the digits (small squares) lie in the five subsets.

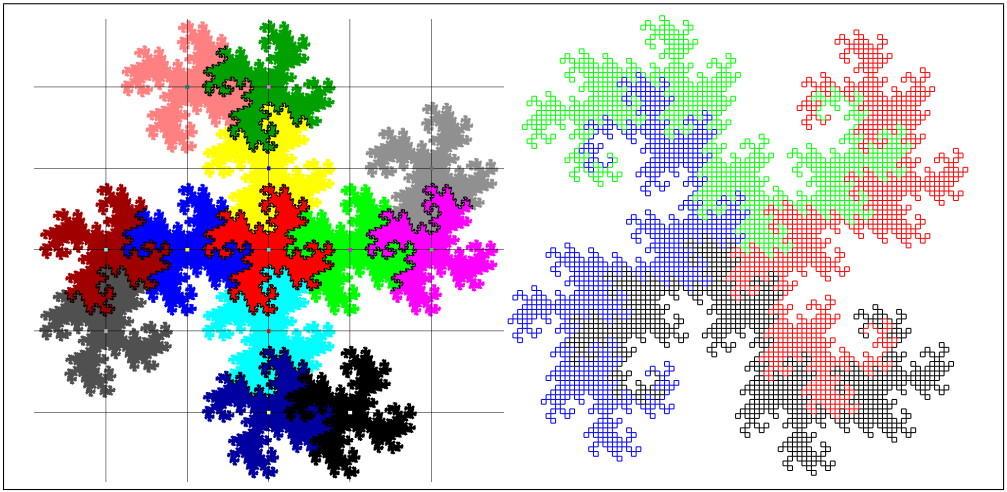

Figure 3.2-B: Fundamental region of a numeration system (left, scaled up by $\sqrt{13}$) and the tile Θ_{+3} of the curve `R13-1` on the square grid (right).

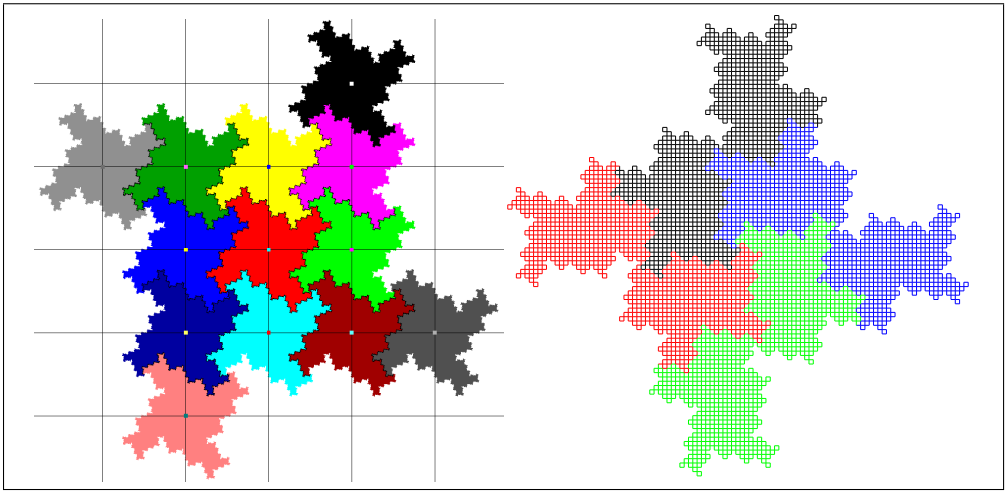

Figure 3.2-C: Fundamental region of a numeration system (left, scaled up by $\sqrt{13}$) and the tile Θ_{+3} of the curve R13–4 on the square grid (right).

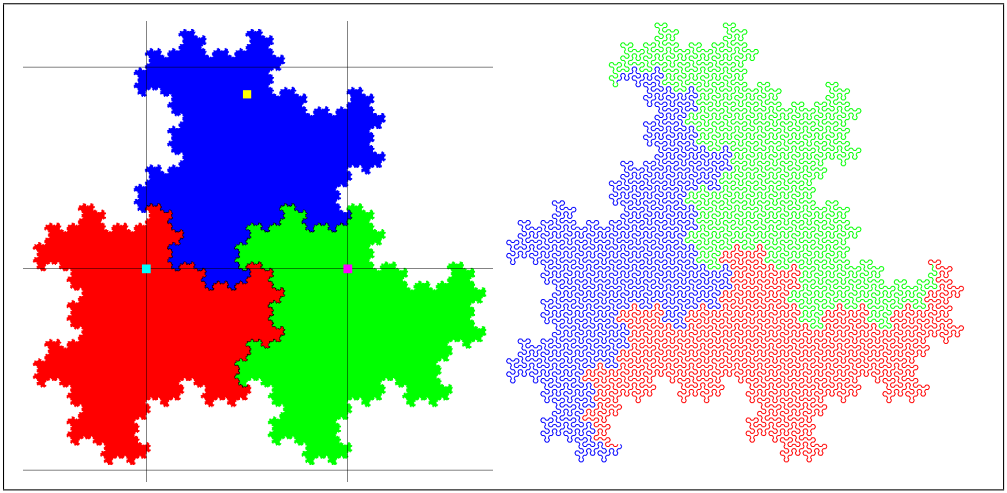

Figure 3.2-D: The fundamental region (left) for the complex numeration system with base $\sqrt{3}\,i$ and digit set $\{0, +1, \omega_6\}$ where $\omega_6 = \exp(+2\pi i/6) = (1 + i\sqrt{3})/2$ (the set is scaled up by the factor $\sqrt{3}$ to let the digits (small squares) lie in the three subsets) and the tile Θ_{+7} for the terdragon (right).

3.3 Curves and tiles on the tri-hexagonal grid

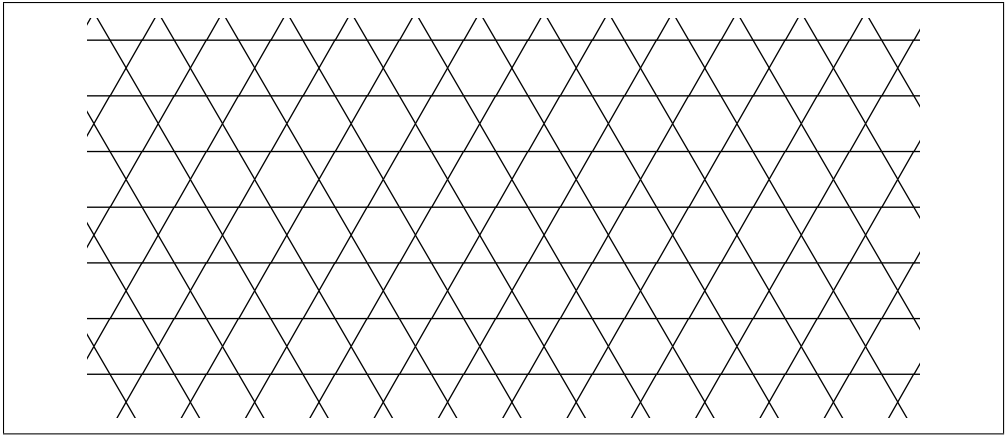

Figure 3.3-A: The tri-hexagonal grid.

Curves on the tri-hexagonal grid exist only for orders $R = 6k + 1$ where R is of the form $x^2 + x\,y + x^2$.

The curves never have any non-trivial symmetry.

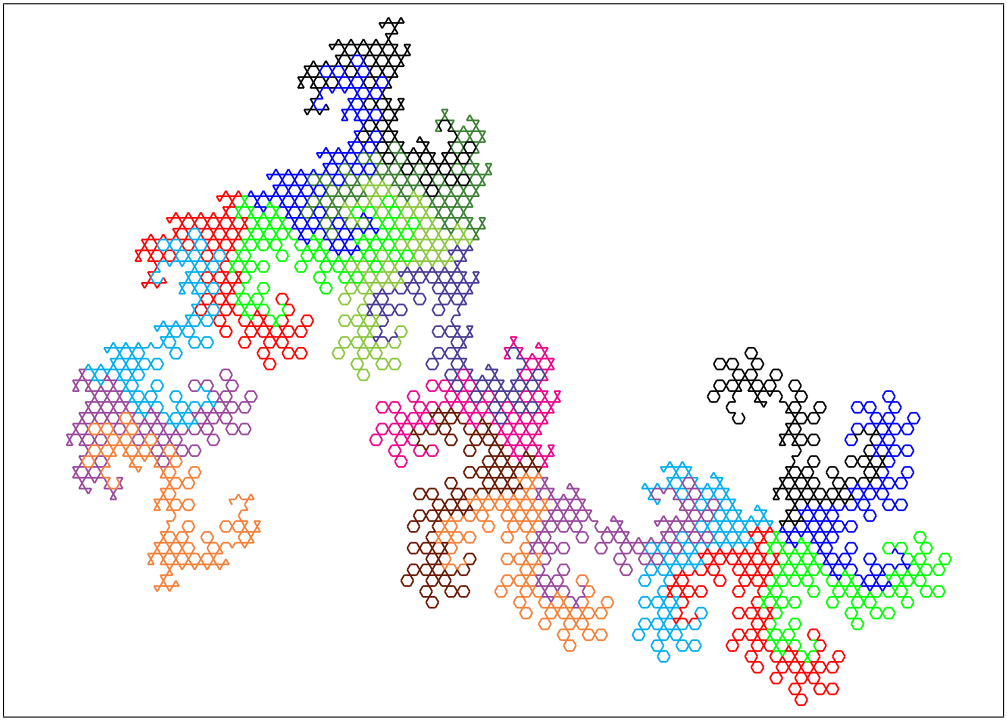

Figure 3.3-B: Third iterate of a curve of order 19 on the tri-hexagonal grid (`R19-1`). The coloration emphasizes the self-similarity.

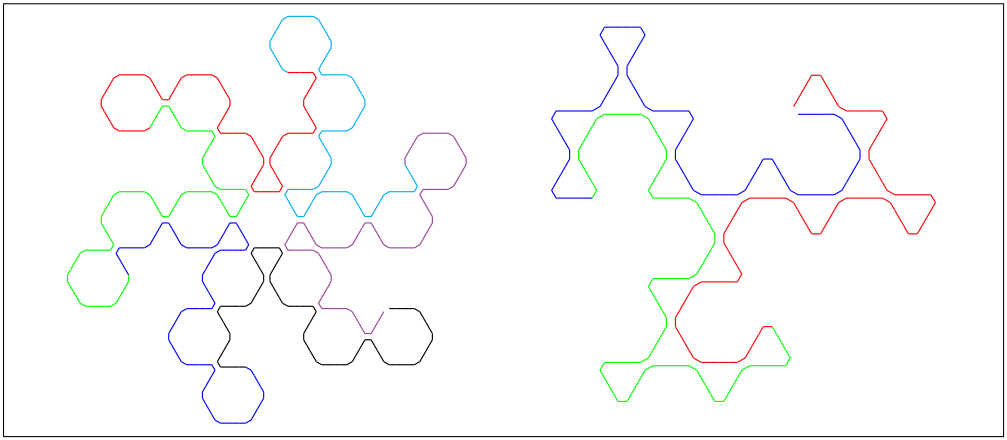

Figure 3.3-C: Tiles Θ_{+1} and Θ_{-1} for the curve in Figure 3.3-B (R19−1).

The center of the left tile is a hexagon, it consists of 6 curves.

The center of the right tile is a triangle, it consists of 3 curves.

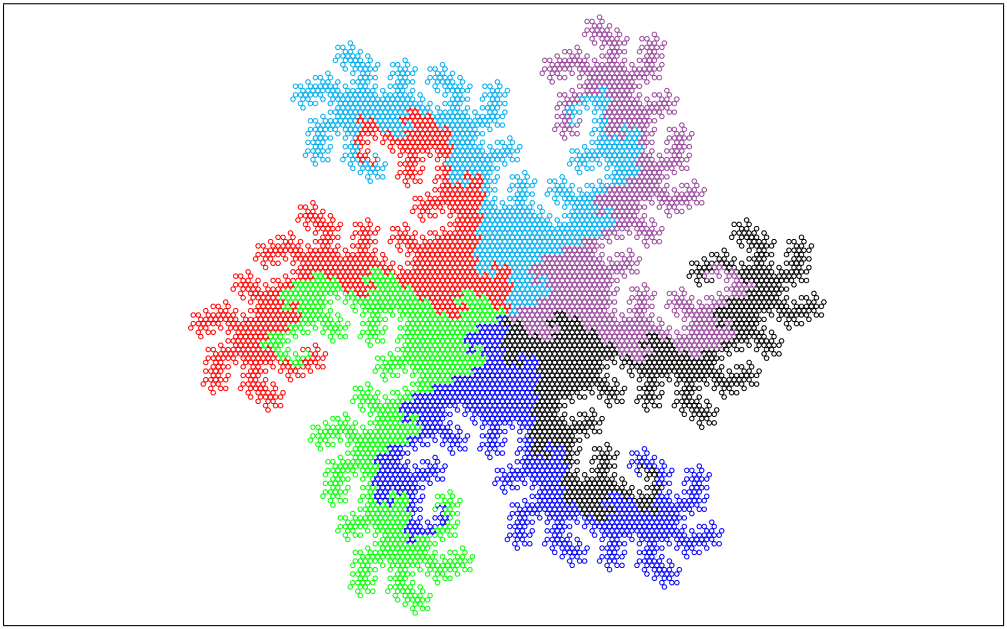

Figure 3.3-D: Tile Θ_{+3} for the curve in Figure 3.3-B ($\mathbb{R}19-1$).

Again tile of a lattice tiling.

These tiles are the same found for the triangular grid which have a 6-fold rotational symmetry in the limit.

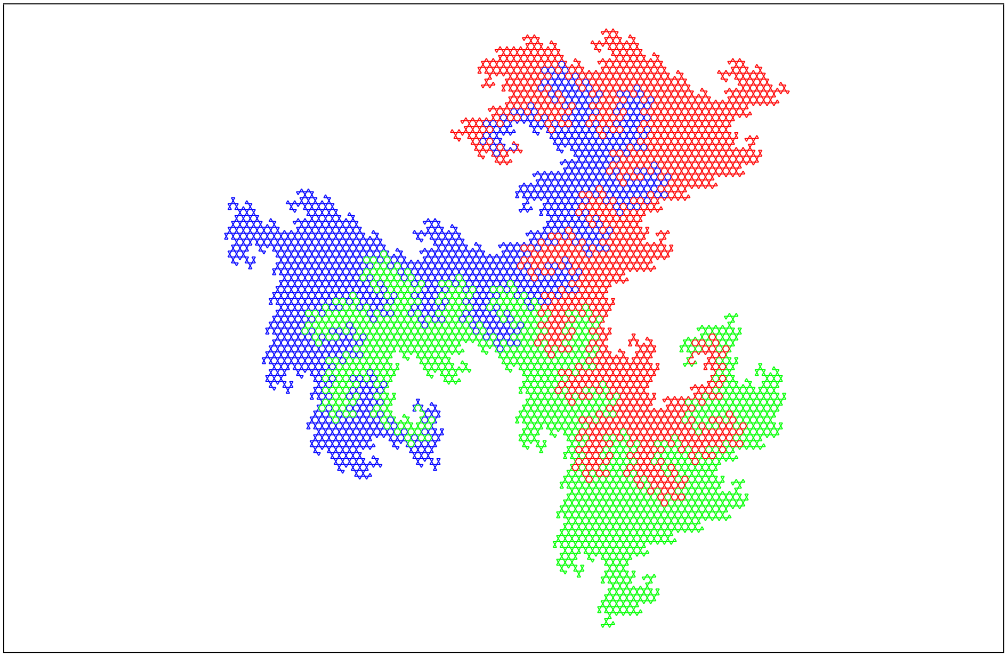

Figure 3.3-E: Tile Θ_{-3} for the curve in Figure 3.3-B ($\mathtt{R19{-}1}$).

Tiling of the plane needs half of the tiles rotated by 180 degrees.

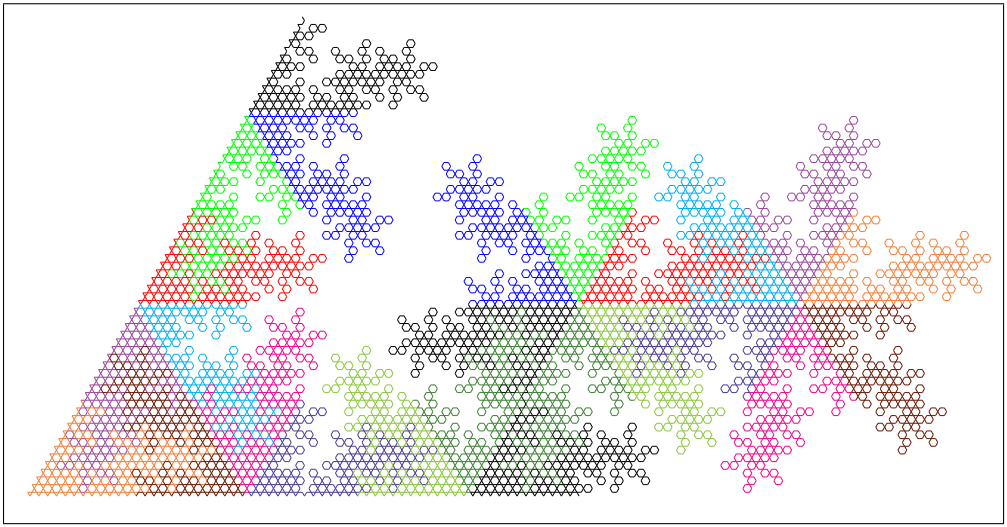

Figure 3.3-F: Self-similarity of a curve of order 25 (R25−11).

Curves with one tile a triangle exists for orders that are squares.

4 Plane-filling curves on all uniform grids

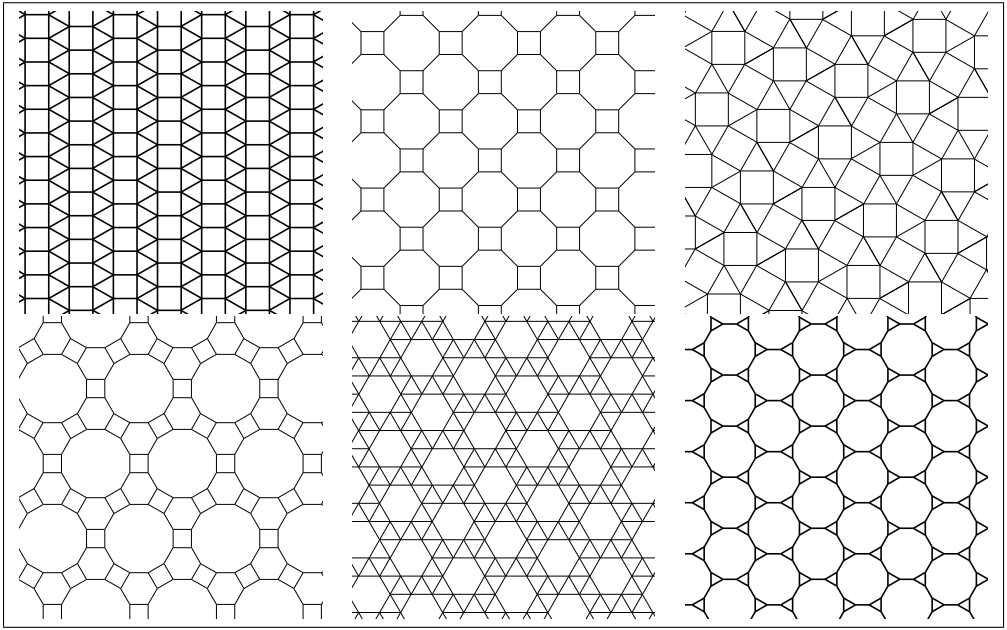

Figure 4.0-A: The grids $(3.3.3.4.4) = (3^3.4^2)$, $(4.8.8)$, $(3.3.4.3.4)$ (top), and $(4.6.12)$, $(3.3.3.3.6) = (3^4.6)$, $(3.12.12) = (3.12^2)$ (bottom).

The curves corresponding to simple L-systems are edge-covering (traverse each edge of the underlying grid once) and necessarily traverse each point more than once (hence are not *point-covering*).

point-covering

Here we give methods to convert these curves into point-covering curves on all uniform grids and to edge-covering curves on two uniform grids.

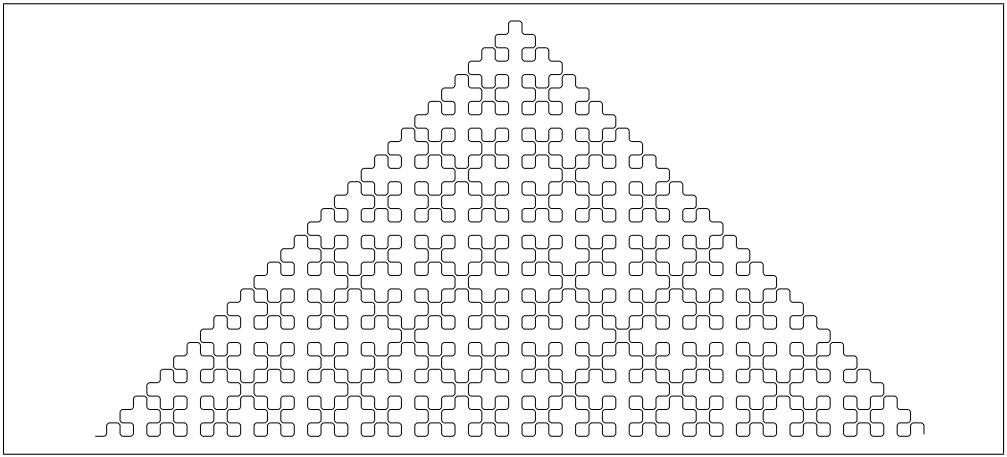

Figure 4.0-B: A plane-filling curve on the square grid that is neither EC nor PC: all points are traversed, but some twice and not all edges are traversed.

4.1 Conversions to point-covering curves

4.1.1 Triangular grid: curves for (6^3), $(3.6.3.6)$, $(3^3.4^2)$, (3^6), $(3.12.12)$, $(3.4.6.4)$, $(4.6.12)$, $(3^4.6)$, and (4^4)

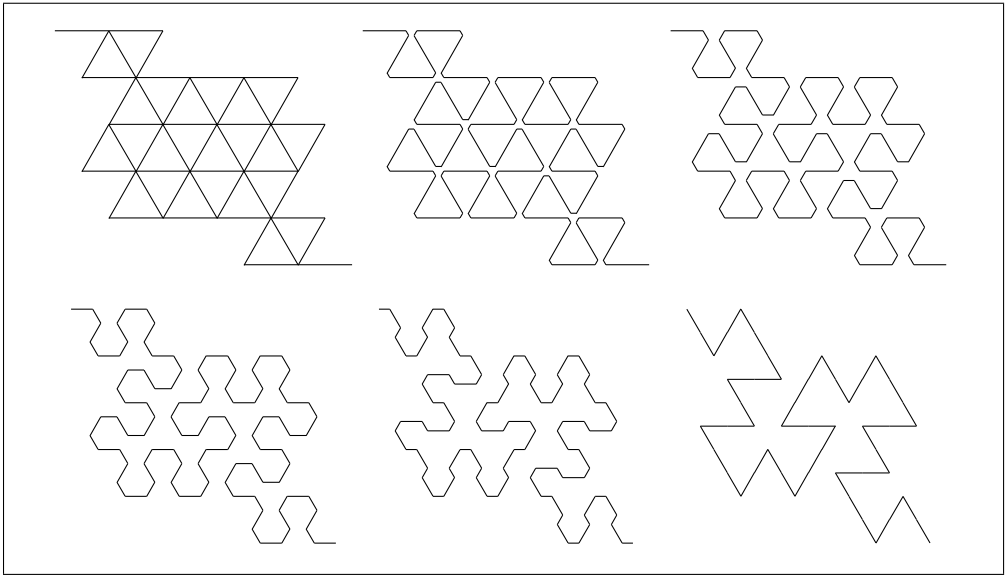

Figure 4.1-A: Renderings of the curve `R7-5` with map `F` $\mapsto$ `F+F-F-F+F+F-F` on the triangular grid for rounding parameter $e \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$.

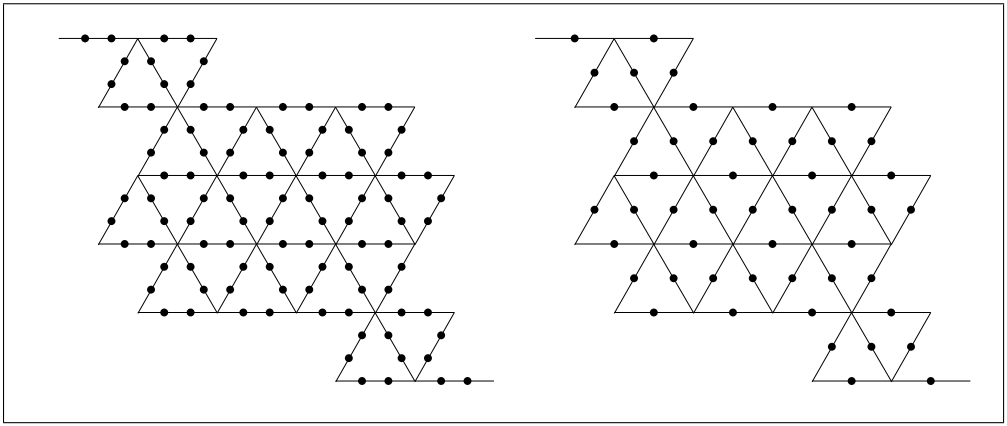

Figure 4.1-B: The turning points with rounded corners for $e = 1/3$ (left) and $e = 1/2$ (right) respectively give the hexagonal and tri-hexagonal grid.

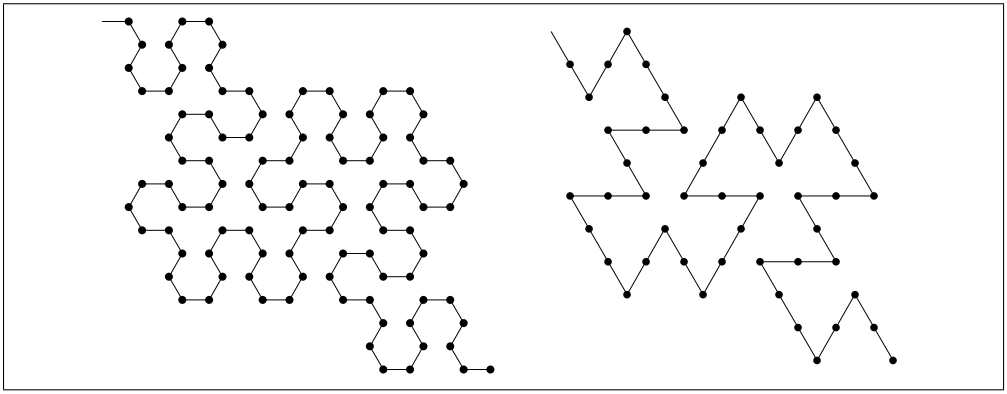

Figure 4.1-C: The curves obtained by post-processing corresponding to $e = 1/3$ (left), a (6^3)-PC curve, and $e = 1/2$, a $(3.6.3.6)$-PC curve (right).

The words for these curves can be computed using a post-processing step on the final iterate of the L-system.

For the (6^3)-PC curve, replace all `+F` by `+F+F`, all `−F` by `−F−F`, and use turns by $60°$.

For the $(3.6.3.6)$-PC curve, drop all `F`, replace all `+` by `+F+`, all `−` by `−F−`, and use turns by $60°$.

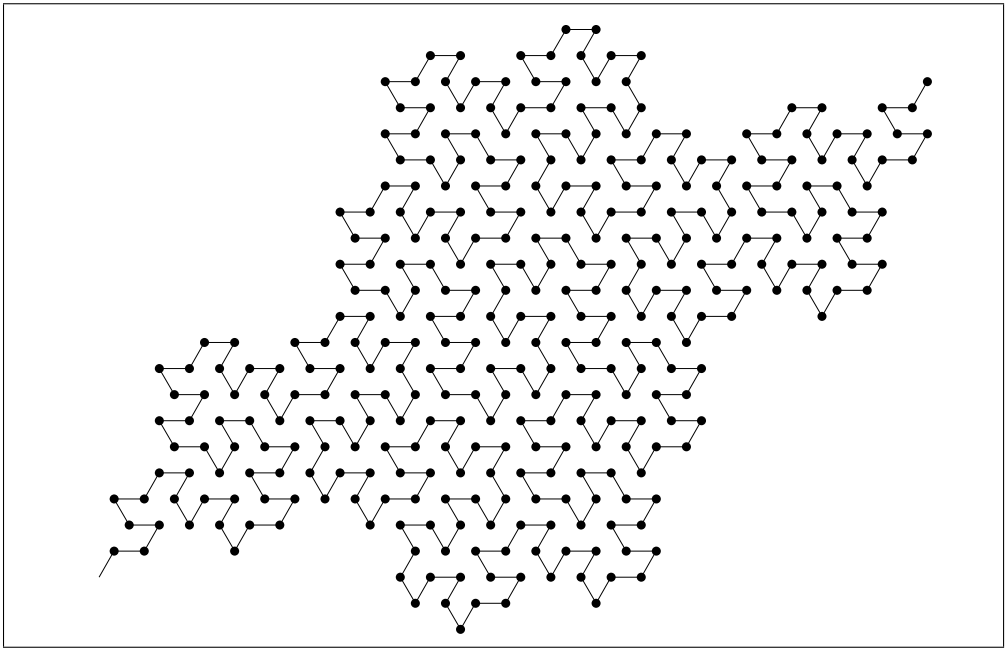

Figure 4.1-D: The (3^6)-PC curve from the third iterate of `R7-5`.

Curves that are (3^6)-PC can be obtained by post-processing as well: drop all `F`, replace all `+` by `++F`, all `-` by `-F-`, and use turns by $60°$.

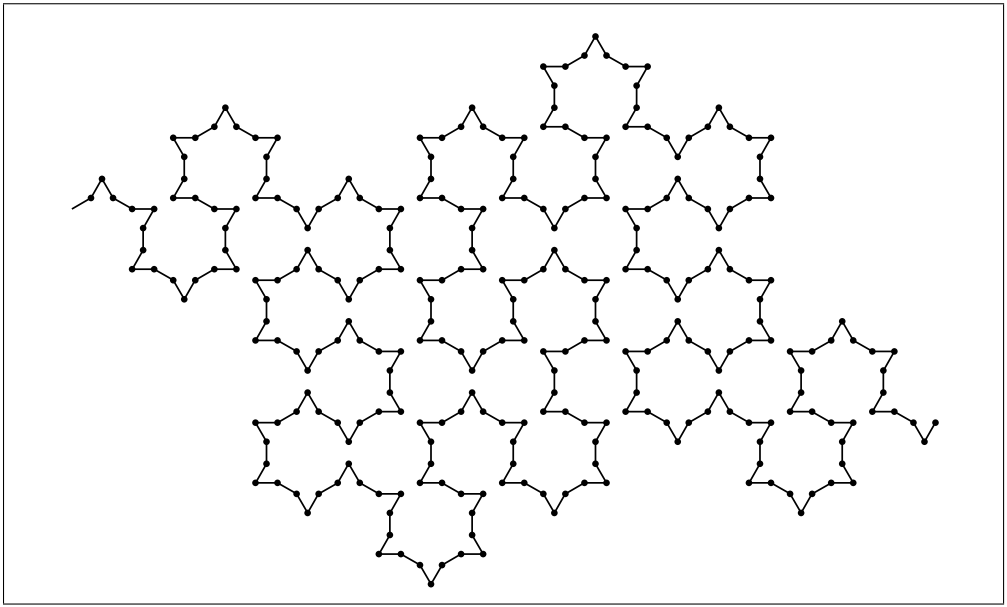

Figure 4.1-E: The $(3.12.12)$-PC curve from the second iterate of `R7-5`.

Drop the initial F, then replace all `+F` by `X` and all `-F` by `Y`, then replace all `X` by `F-F++++F-F-F++++F-` and all `Y` by `F+F----F+F+F----F+`, use turns by $30°$.

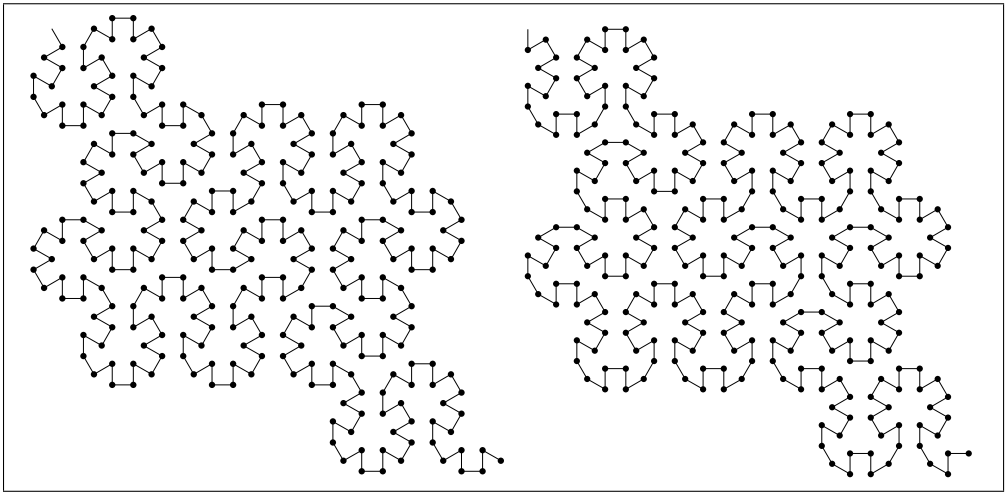

Figure 4.1-F: Two $(3.4.6.4)$-PC curves from the second iterate of `R7-5`.

For the curve on the left, drop all `F`, replace all `+` by `p` and all `-` by `m`,

then replace all `p` by `++F+++F----F+++F+++F----F+`
and all `m` by `--F---F++++F---F---F++++F-`, use turns by $30°$.

For the curve on the right, drop all `F`, replace all `+` by `p` and all `-` by `m`,

then replace all `p` by `+++F----F+++F+++F----F+++F`
and all `m` by `+++F----F-F-F----F+++F`, again use turns by $30°$.

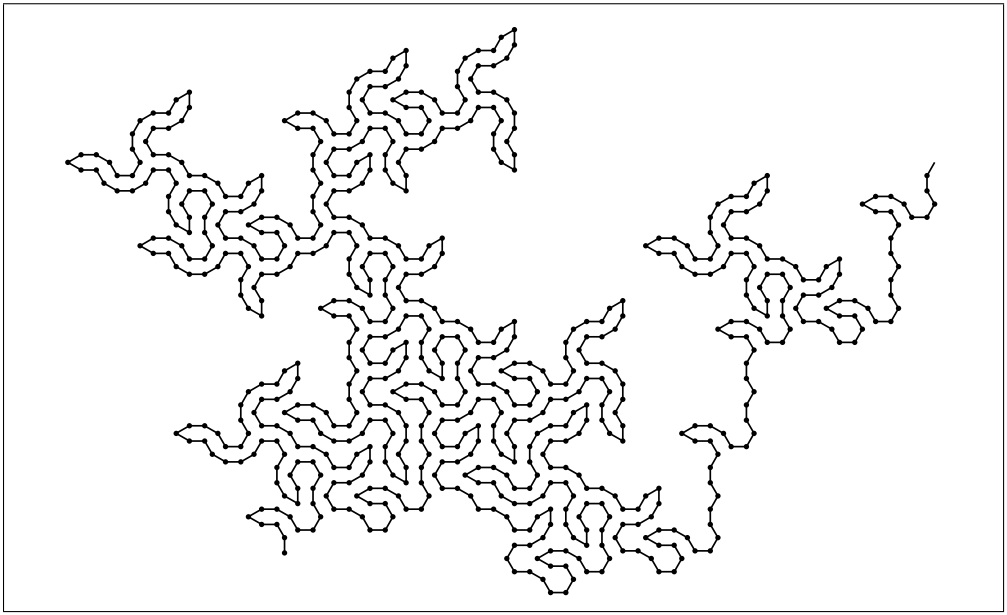

Figure 4.1-G: $(3.4.6.4)$-PC curve from the third iterate of the balanced curve `R7-1`.

For balanced curves, $(3.4.6.4)$-PC curves are obtained by replacing all + by `p` and all - by `m`, then all `Fp` by `+F++F+`, all `Fm` by `--F--`, all `F0` by `+F-F-F+`, use turns by $30°$.

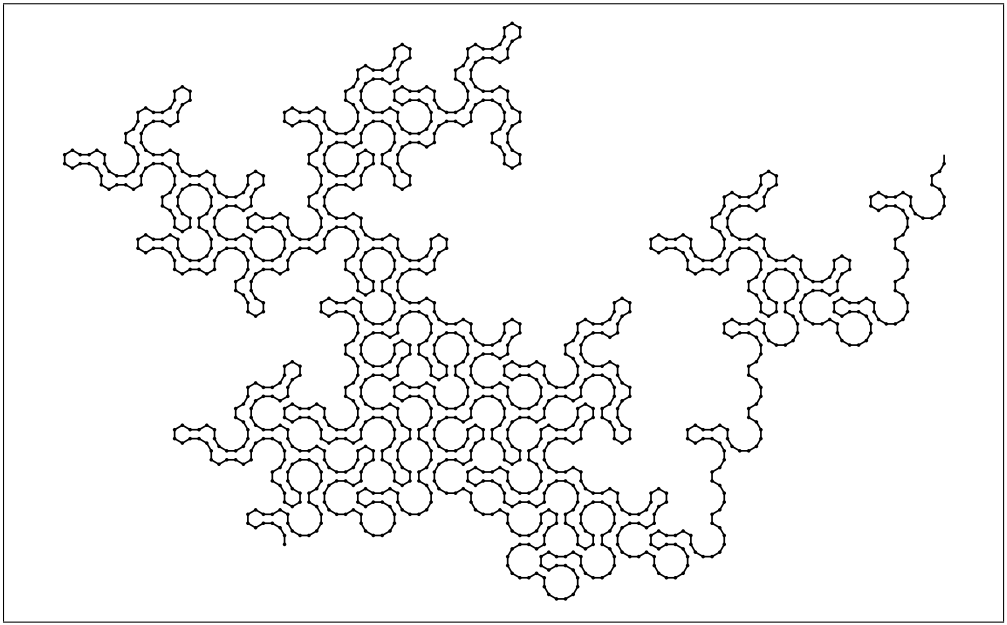

Figure 4.1-H: $(4.6.12)$-PC curve from the third iterate of the balanced curve `R7-1`.

For balanced curves, $(4.6.12)$-PC curves are obtained by replacing all `F+` by `F+F+F+F+`, all `F-` by `F--F--`, and all `F0` by `F+F+F--F--F+F+`, again using turns by $30°$.

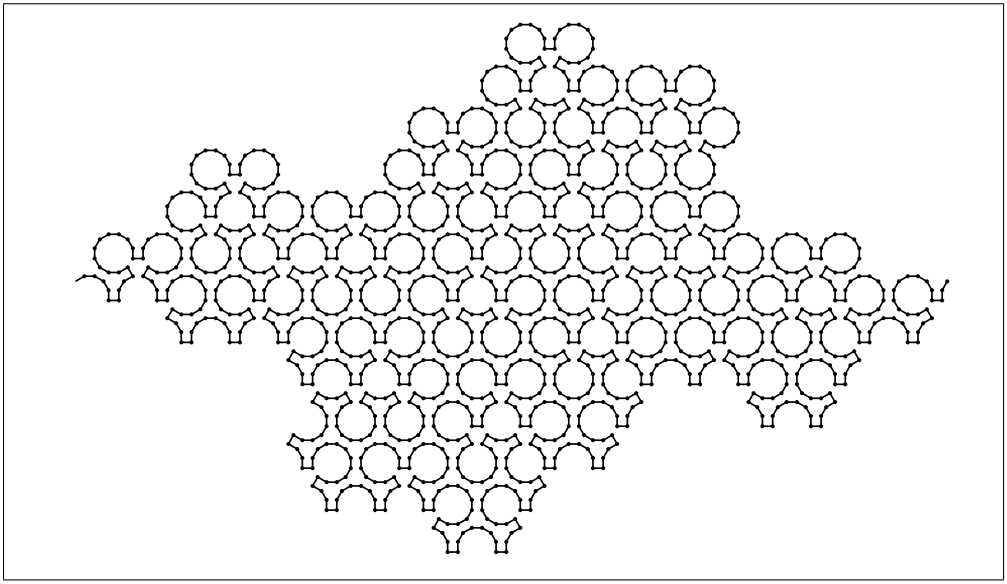

Figure 4.1-I: The $(4.6.12)$-PC curve from the third iterate of `R7-5`.

For curves that are $(4.6.12)$-PC, drop all `F`, replace all + by `p` and all − by `m`, then replace all `p` by `+F+F+F+F` and all `m` by `+F---F---F+F`. Use turns by $30°$.

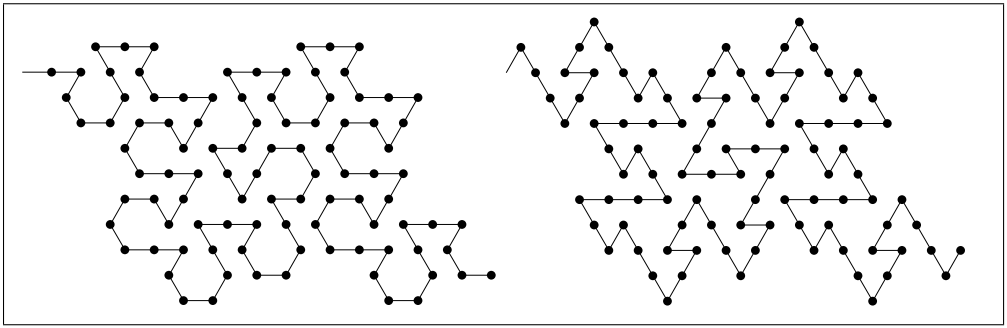

Figure 4.1-J: Two $(3^4.6)$-PC curves from the second iterate of `R7-5`.

Curves that are $(3^4.6)$-PC are obtained by replacing all + by `F++`, all − by `-F-`, and using turns by $60°$.

Replacing all + by `+T+` and all − by `-T-`, then all `F` by `-T+`, drawing edges for `T` using turns by $60°$ gives the curve shown on the right.

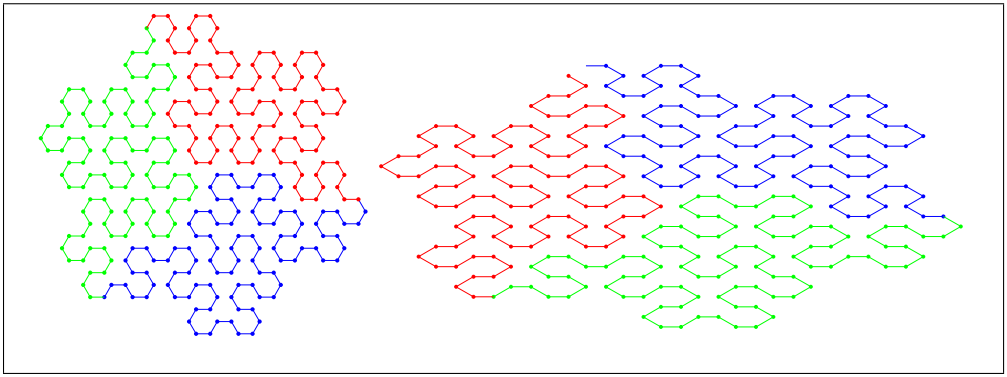

Figure 4.1-K: The second iterate of the tile of R7−5 (Θ_{+2}, left) and the $(3^3.4^2)$-PC curve derived from it by redirecting non-horizontal edges (right).

The $(3^3.4^2)$-PC curve is obtained from the curve at the left by adjusting the directions of the edges that are not horizontal.

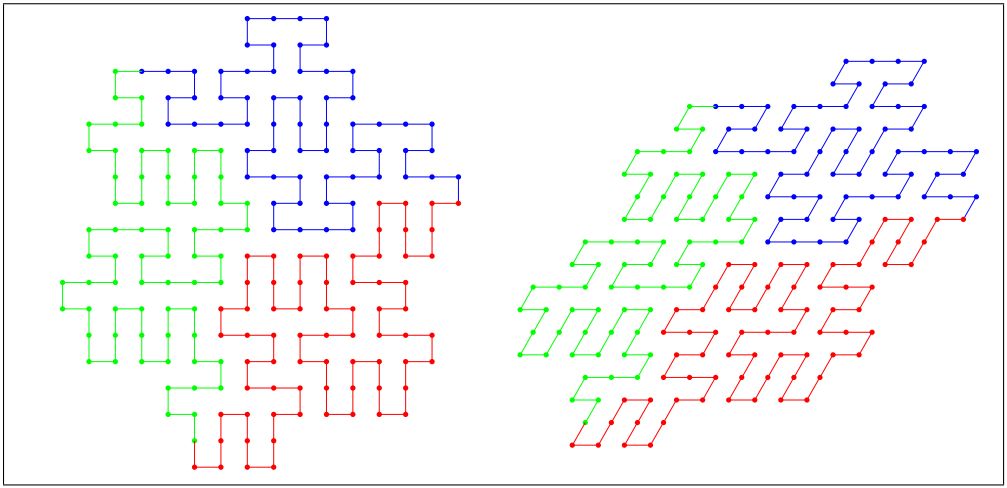

Figure 4.1-L: Rendering of the tile Θ_{+2} as (4^4)-PC (left) and (3^6)-PC curves (right).

(4^4)-PC curves can be obtained by changing all non-horizontal edges in a (6^3)-PC curve, [... details in the paper].

Turning all vertical edges in the (4^4)-PC curve gives the (3^6)-PC curve at the right.

4.1.2 Square grid: curves for $(4.8.8)$, (4^4), $(3.3.4.3.4)$, and (3^6)

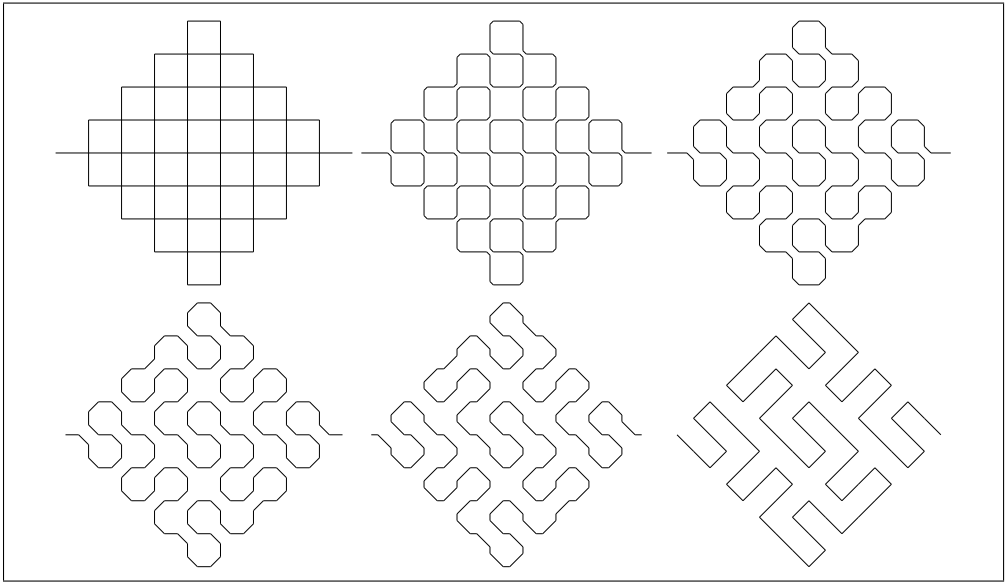

Figure 4.1-M: Renderings of the curve `R9-1` with map `F` $\mapsto$ `F+F-F-F-F+F+F+F-F` on the square grid for rounding parameter $e \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$.

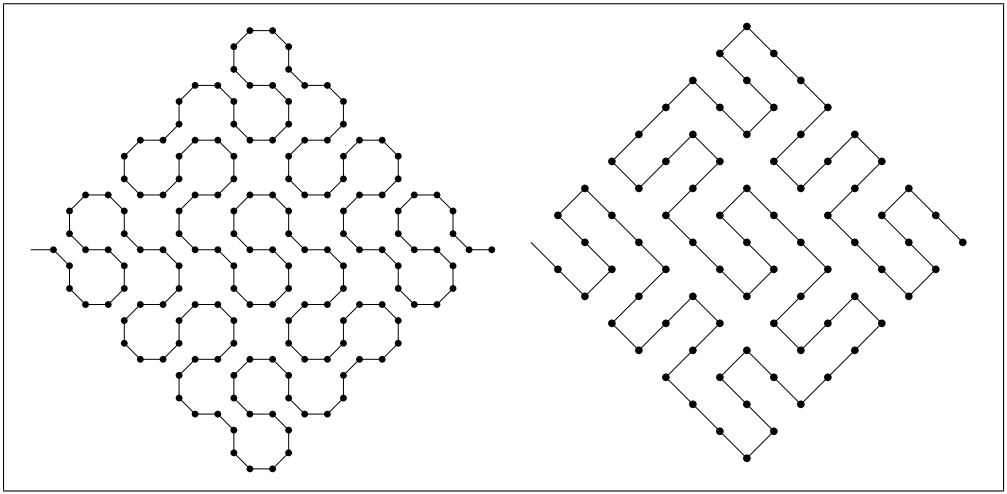

Figure 4.1-N: Points traversed by the curves corresponding to $e = 1/(2+\sqrt{2}) \approx 0.292893\ldots$, a $(4.8.8)$-PC curve (left), and $e = 1/2$, a (4^4)-PC curve (right).

The curves can again be obtained by post-processing steps.

For $e = 1/(2+\sqrt{2}) \approx 1/3$ replace all `+F` by `+F+F`, all `-F` by `-F-F`, and use turns by $45°$. For $e = 1/2$ drop all `F`, replace all `+` by `+F+`, all `-` by `-F-`, and use turns by $90°$.

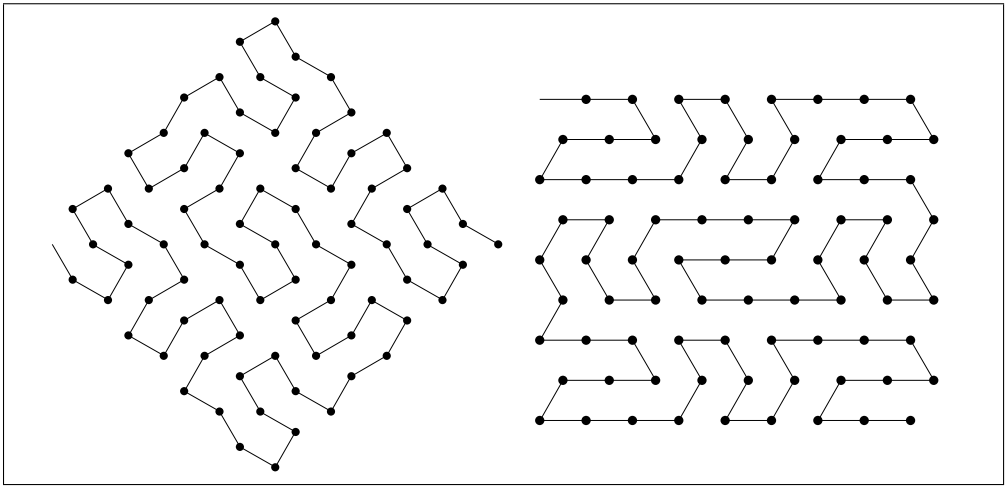

Figure 4.1-O: The $(3.3.4.3.4)$-PC and the (3^6)-PC curves from the second iterate.

For the $(3.3.4.3.4)$-PC curve shown on the left, drop all F, replace all + by ++F+, all − by −−F−, and use turns by $30°$.

The (3^6)-PC curve shown on the right results from rendering the (4^4)-PC curve (right of Figure 4.1-N) such that one set of parallel edges (horizontal in the image) is kept and the other (vertical) edges are alternatingly turned by $\pm 30°$ against their original direction.

4.1.3 Tri-hexagonal grid: curves for $(4.6.12)$, $(3.4.6.4)$, $(3^4.6)$, and (4^4)

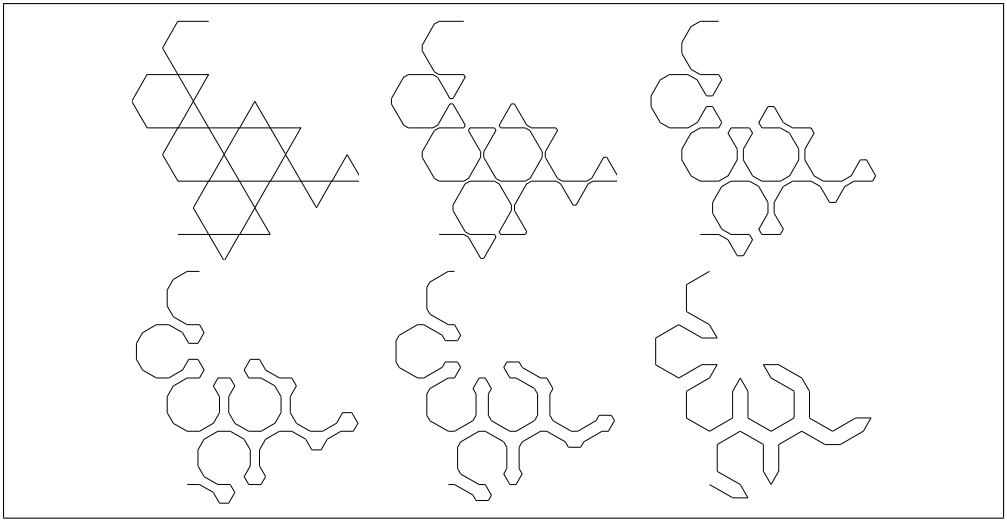

Figure 4.1-P: Renderings of the second iterate of Ventrella's curve (`R7-1`) for rounding parameter $e \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$.

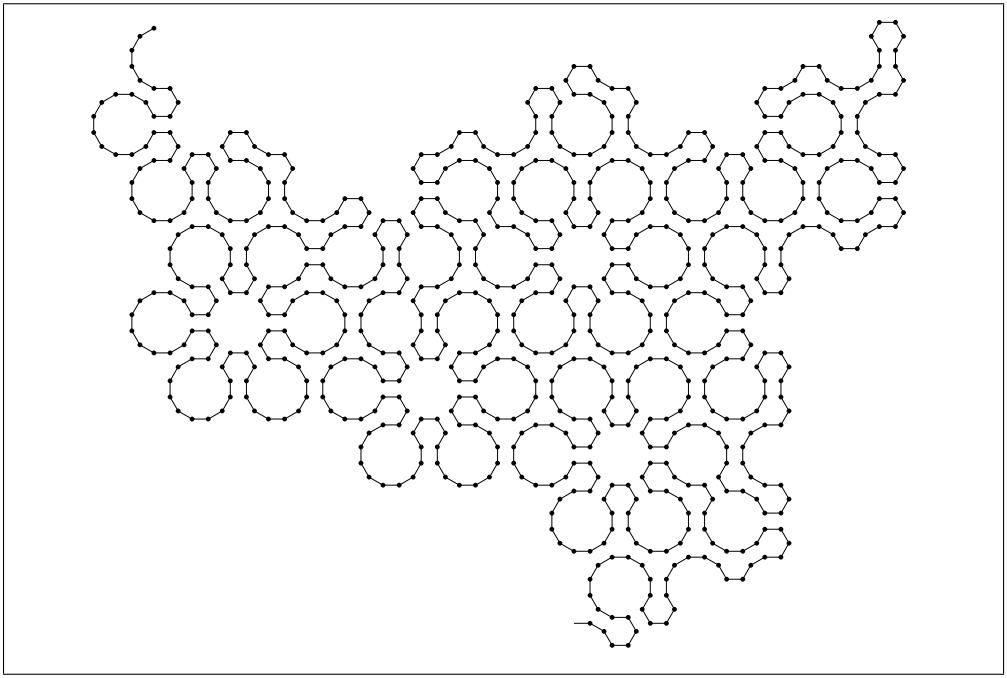

Figure 4.1-Q: A $(4.6.12)$-PC curve from the third iterate.

Drop all F, replace all + by F+F+ and all -- by F--F--, use turns by $30°$.

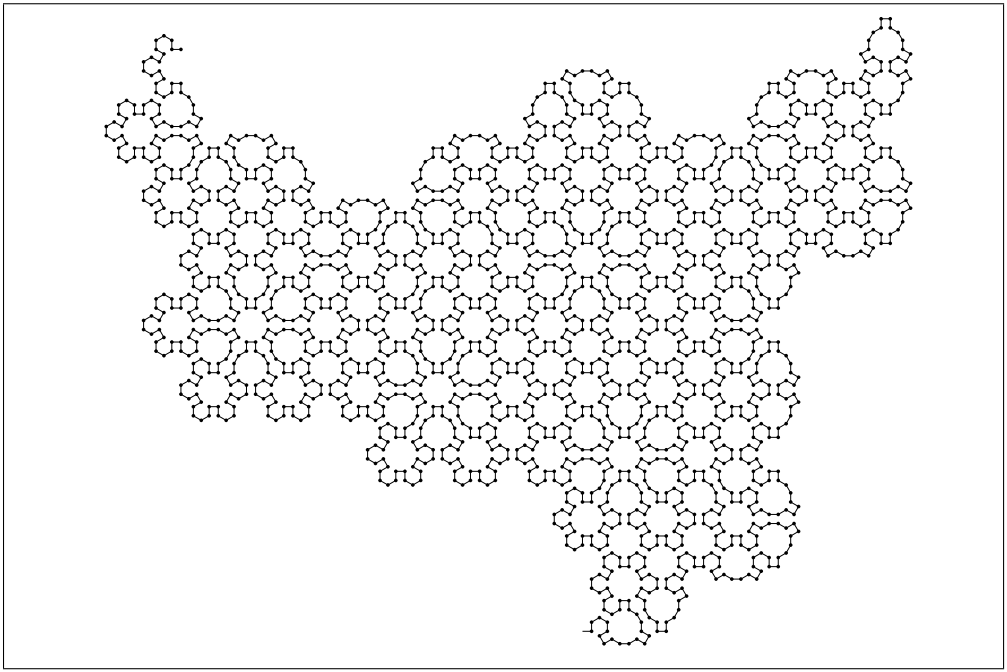

Figure 4.1-R: Another $(4.6.12)$-PC curve from the third iterate.

For the curve replace all + by `p` and all `--` by `m`, then replace all `p` by `---F++F++F++F++F---` and all `m` by `---F++F-F-F++F---`, use turns by $30°$.

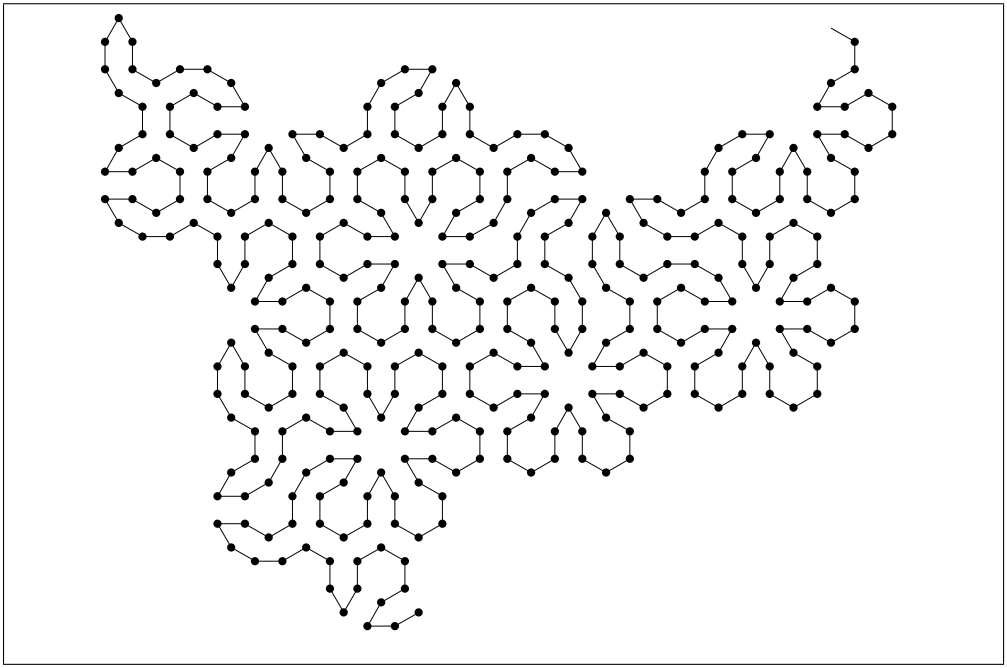

Figure 4.1-S: The $(3.4.6.4)$-PC curve from the third iterate.

Drop all F, replace all + by +F+ and all -- by --F--, and use turns by $30°$.

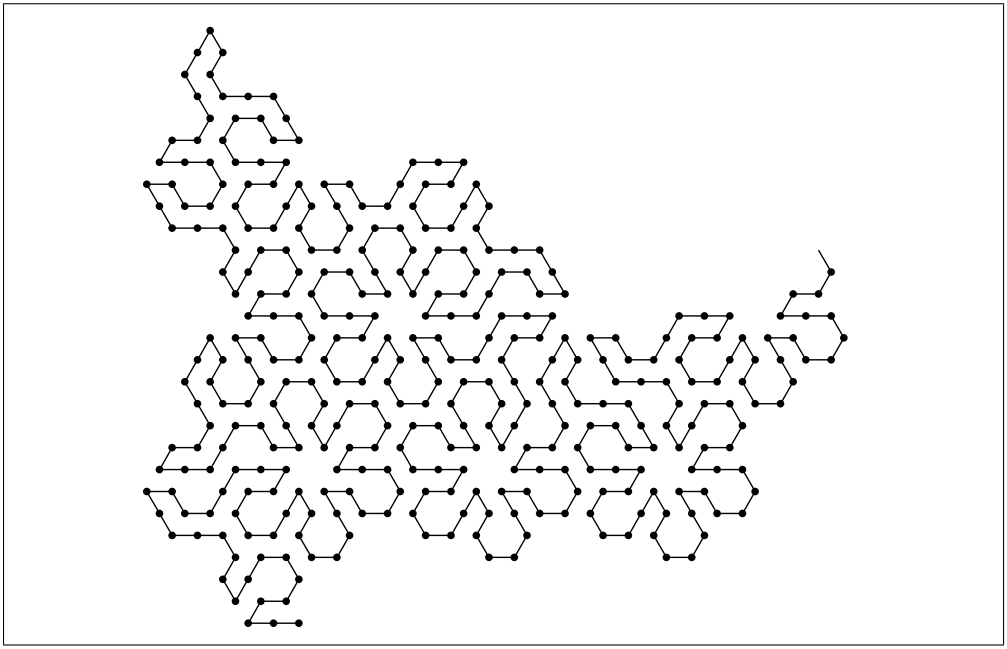

Figure 4.1-T: A $(3^4.6)$-PC curve from the third iterate.

For the $(3^4.6)$-PC curve shown, drop all F, replace all + by +F and all −− by −F−, and use turns by $60°$. Using the replacements −F for + and +F+ for − gives the other enantiomer.

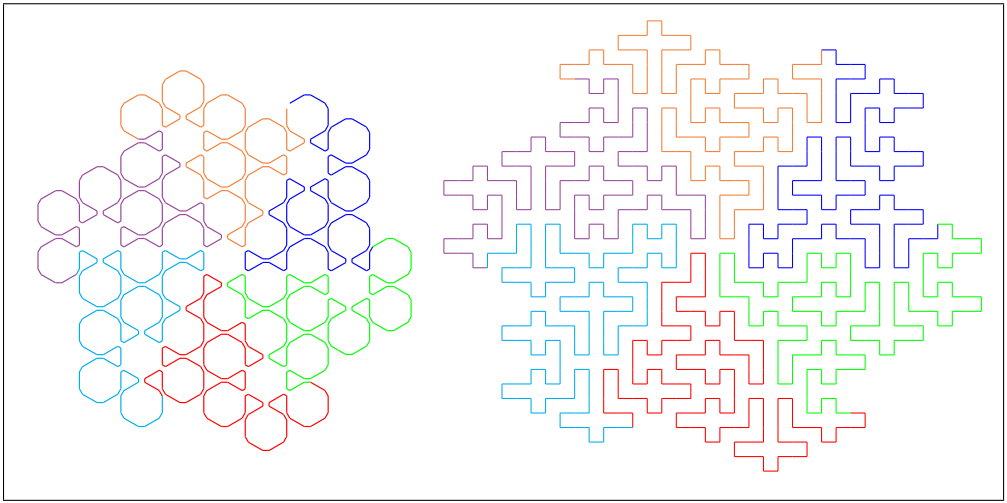

Figure 4.1-U: A (4^4)-PC curve (right) from the tile Θ_{+2} (left).

In the (4^4)-PC curve on the right, all hexagons in the tile shown on the left are replaced by crosses whose horizontal bars are longer than the vertical ones (right).

Somewhat complicated. See the paper.

4.2 Conversions to edge-covering curves

4.2.1 Edge-covering curves on $(3.4.6.4)$ from wiggly (3^6)-EC curves

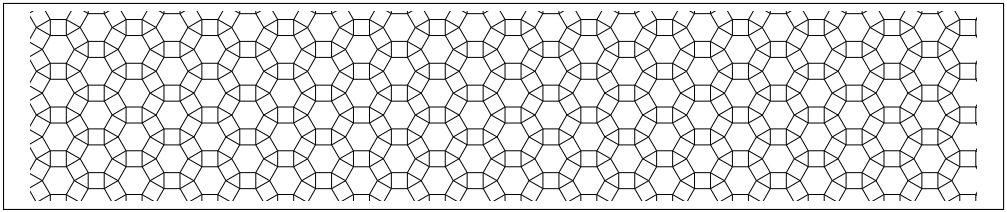

Figure 4.2-A: The $(3.4.6.4)$ grid.

The grid $(3.4.6.4)$ has even incidences on all points, so edge-covering curves do exist.

The grids (3^6), $(3.6.3.6)$, (4.4), and $(3.4.6.4)$ are the only uniform grids where edge-covering curves can exist, so the last gap is closed (HOORAY!).

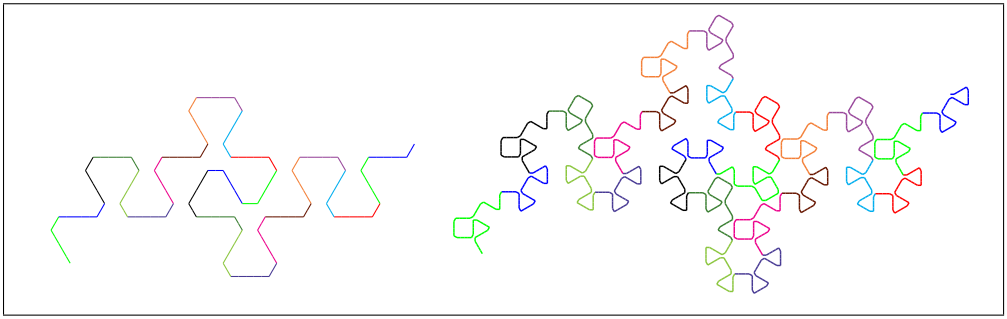

Figure 4.2-B: Third iterate of the terdragon with rounded corners (left) and the corresponding curve on $(3.4.6.4)$ omitting the edges of some hexagons (right).

The initial step gives an incomplete curve.

Drop all `F`, replace all `+` by `p` and `-` by `m`,

then replace all `p` by `---F++++F++++F---F---F++++F++++F---F`
and all `m` by `---F++++F---F++F---F---F---F++++F++++F---F`,

render with turns by $30°$.

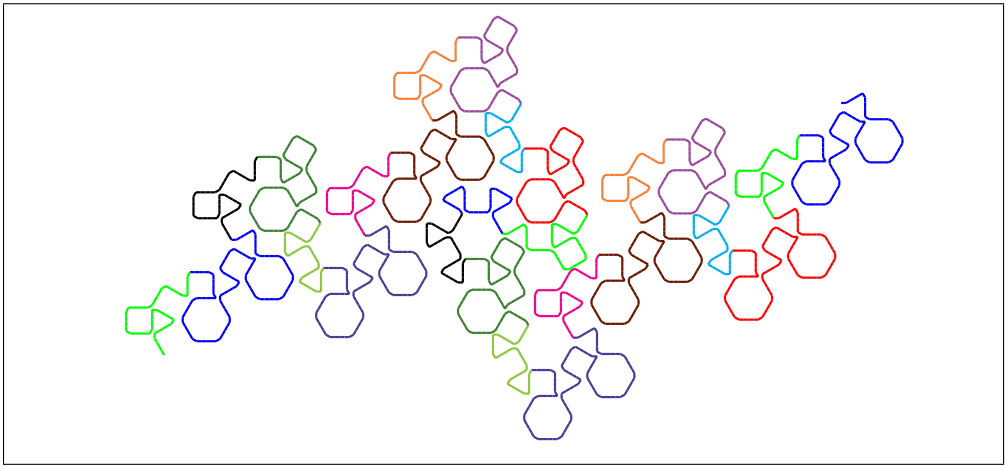

Figure 4.2-C: The completed curve on $(3.4.6.4)$.

To fill the missing hexagons in, we keep track of the directions modulo 3, using representatives 0, 1, and 2.

We add two hexagons (letter `H`) by changing the replacement for `p` to `---F++++FH++++F---F---F++++FH++++F---F` if the direction is 0 and one hexagon by changing the replacement for `m` to `---FH++++F---F++F---F---F---F++++F++++F---F` for direction 2.

Finally, `H` is replaced by `---F++F++F++F++F++F-------`.

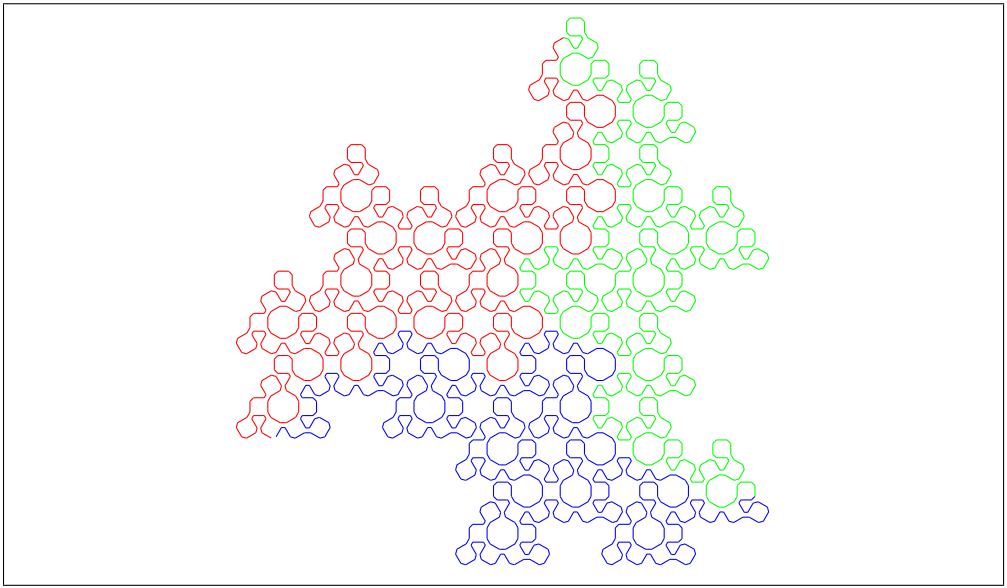

Figure 4.2-D: An edge-covering curve on $(3.4.6.4)$ from the tile Θ_{-3} of the terdragon.

That one was a real bastard to get right. But hey, we are done!

Thanks for listening!

Any Questions?

For every question you do not ask now, a cute little kitten will be run over by a lawnmower. You don't want that.