

L-Systems in R

Alexander Ptok

29. August 2024

L-Systems

```

lsystem <- function(alphabet, axiom, productions) {

  derivation <- function(axiom, new_word) {
    a <- substring(axiom,1,1)
    if (a == "") {
      new_word
    } else derivation(substring(axiom,2),
                      paste(new_word,
                            productions[a],
                            sep=""))
  }

  derive_n <- function(axiom, n) {
    if (n == 0) {
      axiom
    } else derive_n(derivation(axiom, ""),
                    n - 1)
  }

  function(n) {
    derive_n(axiom, n)
  }
}

lsystem_iter <- function(alphabet, axiom, productions) {
  function(n) {
    new_word <- ""
    while (n > 0) {
      for (char in as.list(strsplit(axiom,""))[[1]]) {
        new_word <- paste(new_word,
                          productions[char],
                          sep="")
      }
      n <- n - 1
      axiom <- new_word
      new_word <- ""
    }
    axiom
  }
}

```

The Turtle Interpreter

```

turtle <- function(x, y, alpha, stepsize, delta) {

  forward <- function() {
    x <- x + stepsize * cos(alpha)
    y <- y + stepsize * sin(alpha)
  }

  forward_draw <- function() {
    linesxy$x1 <- c(x, linesxy$x1)
    linesxy$y1 <- c(y, linesxy$y1)
    forward()
    linesxy$x2 <- c(x, linesxy$x2)
    linesxy$y2 <- c(y, linesxy$y2)
  }

  turn_right <- function() {
    alpha <- alpha - delta
  }
}

```

```
}

turn_left <- function() {
  alpha <- alpha + delta
}

linesxy <- list(x1=c(),x2=c(),y1=c(),y2=c())

function_table <-
  list("F" = forward_draw,
        "f" = forward,
        "-" = turn_right,
        "+" = turn_left)

rec_over_nu <- function(nu) {
  if (nu == "") {
    linesxy
  } else {
    function_table[[substring(nu,1,1)]]()
    rec_over_nu(substring(nu,2))
  }
}

iter_over_nu <- function(nu) {
  for (i in 1:nchar(nu)) {
    function_table[[substring(nu,i,i)]]()
  }
  linesxy
}

function(nu) {
  # rec_over_nu(nu)
  iter_over_nu(nu)
}

}

plotting_turtle <- function(x, y, alpha, stepsize, delta) {

  forward <- function() {
    x <- x + stepsize * cos(alpha)
    y <- y + stepsize * sin(alpha)
  }

  forward_draw <- function() {
    x1 <- x
    y1 <- y
    x <- x + stepsize * cos(alpha)
    y <- y + stepsize * sin(alpha)
    lines(c(x1,x),c(y1,y))
  }

  turn_right <- function() {
    alpha <- alpha - delta
  }

  turn_left <- function() {
    alpha <- alpha + delta
  }

  function_table <-
    list("F" = forward_draw,
          "f" = forward,
          "-" = turn_right,
          "+" = turn_left)
```

```

iter_over_nu <- function(nu) {

  for (i in 1:nchar(nu)) {
    function_table[[substring(nu,i,i)]]()
  }

  function(nu) {

    iter_over_nu(nu)

  }
}

draw_turtle <- function(lines) {
  if (length(lines$x1) == 0) {
    TRUE
  } else {
    lines(x=c(lines$x1[1], lines$x2[1]), y=c(lines$y1[1], lines$y2[1]))
    draw_lines(list(x1=lines$x1[-1],
                    x2=lines$x2[-1],
                    y1=lines$y1[-1],
                    y2=lines$y2[-1]))
  }
}

draw_turtle_iter <- function(lines) {
  for (i in 1:length(lines$x1)) {
    lines(x=c(lines$x1[i], lines$x2[i]), y=c(lines$y1[i], lines$y2[i]))
  }
}

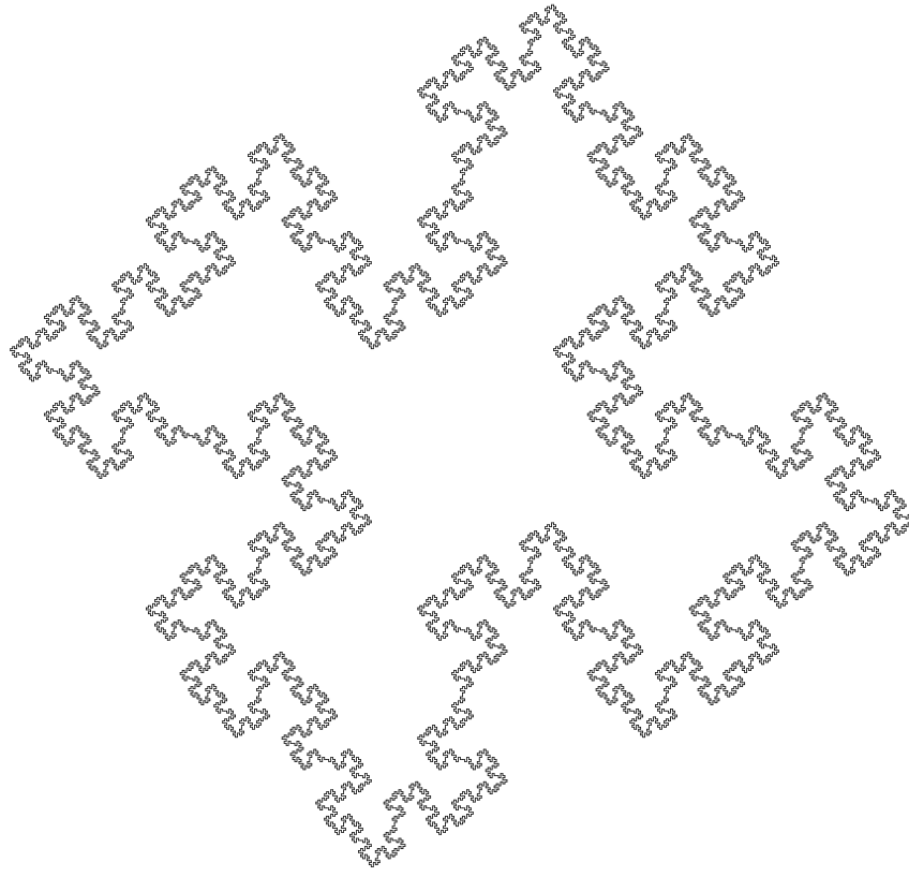
```

The Koch Curve

```

source("lsystem.r")
source("turtle.r")
source("draw-turtle.r")
dict <- c("F" = "F-F+F+FF-F-F+F", "+" = "+", "-" = "-", "f" = "f")
l <- (lsystem_iter("Ff+-", "F-F-F-F", dict) (4))
t <- turtle(0,0,pi,0.03,pi/2)
lines <- t(l)
png("koch.png", width=1000, height=1000)
plot(range(x=c(lines$x1,lines$x2)),y=range(c(lines$y1,lines$y2)),type="n", ann=FALSE, axes=FALSE)
draw_turtle_iter(lines)
dev.off()

```



More Curves

```
source("lsystem.r")
source("turtle.r")
source("draw-turtle.r")
dict <- c("F" = "FF-F-F-F-F-F+F", "+" = "+", "-" = "-", "f" = "f")
l <- (lsystem_iter("Ff+-", "F-F-F-F", dict) (4))
t <- turtle(0,0,pi,1,pi/2)
lines <- t(l)
png("more-curves-a.png", width=1000, height=1000)
plot(x=range(x=c(lines$x1,lines$x2)),
     y=range(y=c(lines$y1,lines$y2)),
     type="n", ann=FALSE, axes=FALSE)
draw_turtle_iter(lines)
dev.off()
```

```
source("lsystem.r")
source("turtle.r")
source("draw-turtle.r")
```

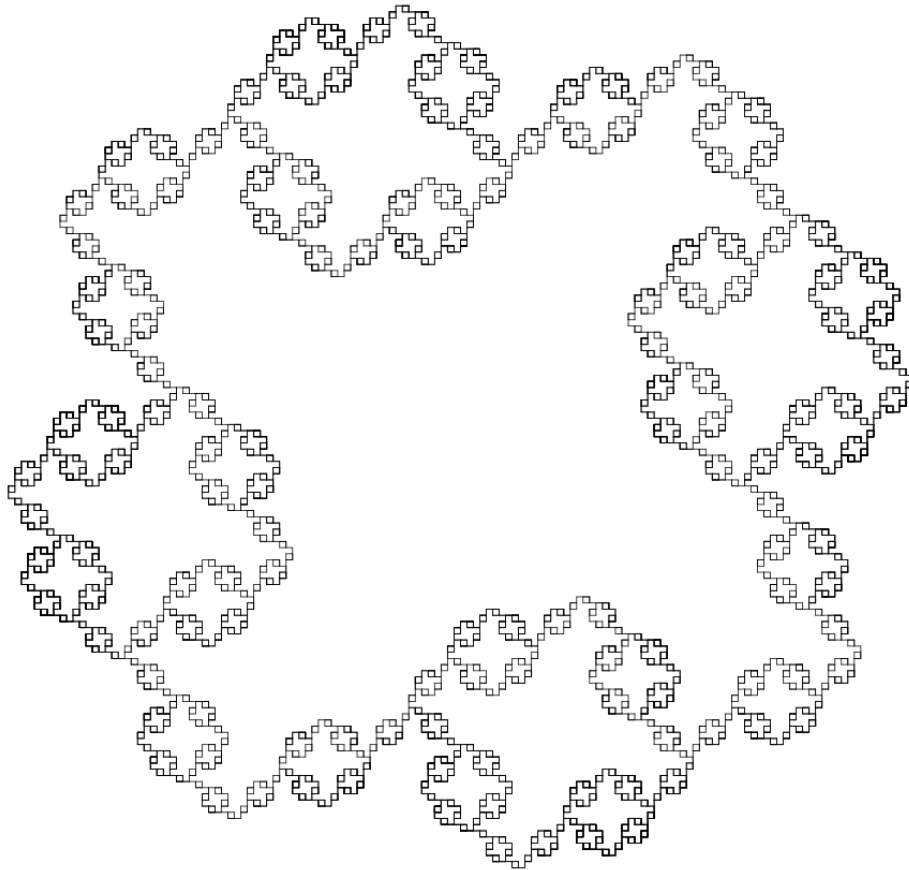


Abbildung 1: More Curves A

```
source("turtle-plot.r")
turtle_plot("sierpinski-gasket.png",
  6,
  pi/3,
  "RL+-",
  "R",
  c("L" = "R+L+R",
    "R" = "L-R-L",
    "+" = "+",
    "-" = "-",
    "f" = "f"))
```