Софийски университет “Св. Климент Охридски”
Факултет по математика и информатика

# Курсов проект по Фрактали летен семестър 2020/2021

Проект - Рекурсивни дървета

Изработил: Веселин Славов Тодоров
Специалност: Информационни системи
Курс: 2
Факултетен номер: 71923

Гр. София
Май 2021

# Съдържание:

# 1. Какво е фрактал?

Фракталът е обект с доста сложна форма, получен в резултат на прост итерационен цикъл. Итерационността и рекурсивността определят такива свойства на фракталите, както самоподобие - отделните части приличат на целия фрактал.

Фракталите са геометрични обекти с дробна размерност. Например, размерността на линия е 1, на площта – 2, на обема – 3. При фрактала това значение на размерността може да бъде между 1 и 2 или между 2 и 3. В математиката съществува специална сложна формула за изчисление на размерността на фракталите. Разклонената система от тръбички на трахеите, листата на дърветата, вените на ръцете, реките – това са фрактали. Както е казано по-горе, фракталът е геометрична фигура, определена част, от която се повторя отново и отново, изменяйки се по размери. Фракталите са подобни сами на себе си на всички нива (т.е. във всеки мащаб). Съществуват много различни типове фрактали. По принцип, може да се каже, че всичко, което съществува в реалния свят е фрактал, било то облак или молекула кислород.

Но ако трябва да дадем определение за фрактал, най-подходящо е да използваме определението на Манделброт за Фрактал:

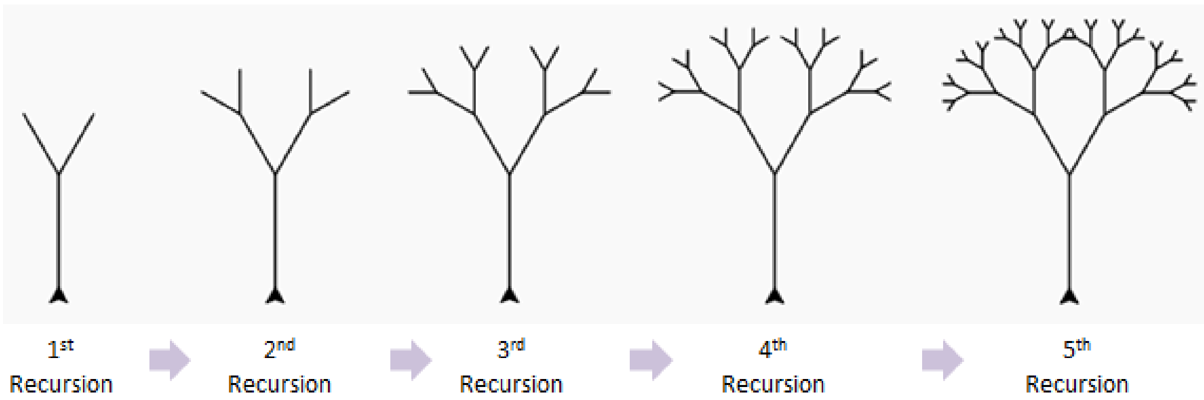
***“Фракталът е структура състояща се от части, които в някакъв смисъл са подобни на цялото.“***

# 2. Рекурсивни дървета

Рекурсивното дърво е равнинен фрактал изработен от линии. Базира се като комбинация на два фрактала. Основно се базира на дървото на Питагор, което създава същия фрактал, но вместо квадрати с изменящ се ъгъл и правоъгълни триъгълници, които се образуват от допирането на 3 квадрати, се използват линии по подобие на Снежинката на Кох.

## Изграждане:

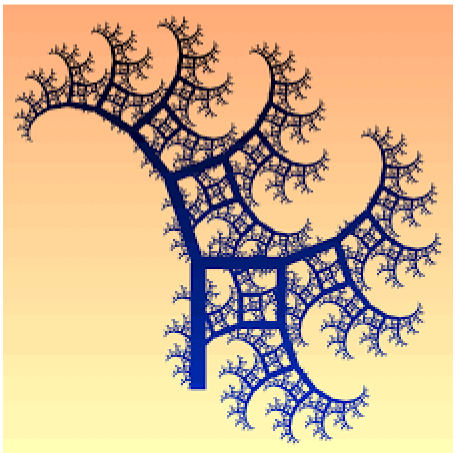
Изграждането на Питагоровото дърво започва с права линия. Върху него се построяват 2 линии, намалени като дължина в отношение 2:3/2, сключващи ъгъл, варираш между 20 и 90 градуса. След това същата процедура се прилага рекурсивно към двете по-къси линии до безкрайност, както е показано на картинката:



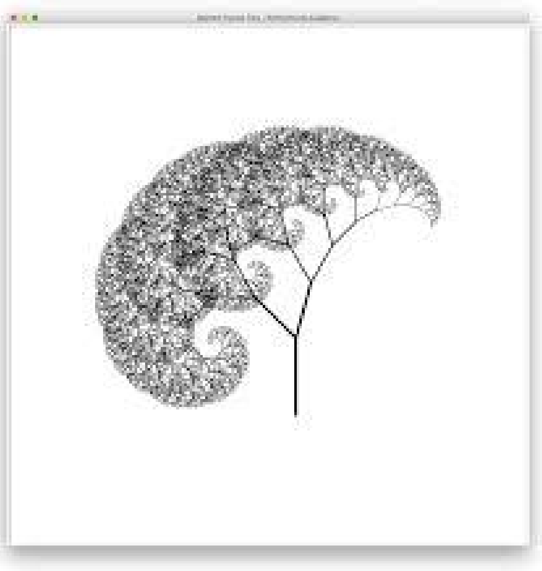

## Вариране на ъгъла:

Интересен набор от вариации може да бъде изграден чрез промяна на основния ъгъл (45 градуса по стандарт). По-специално, когато основният полуъгъл е зададен на (30 °), лесно се вижда, че размерът на линиите остава постоянен.

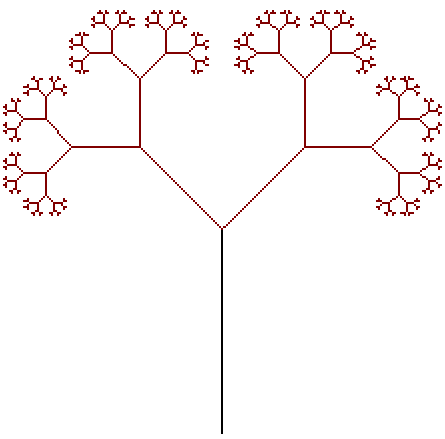
# 3. Варианти



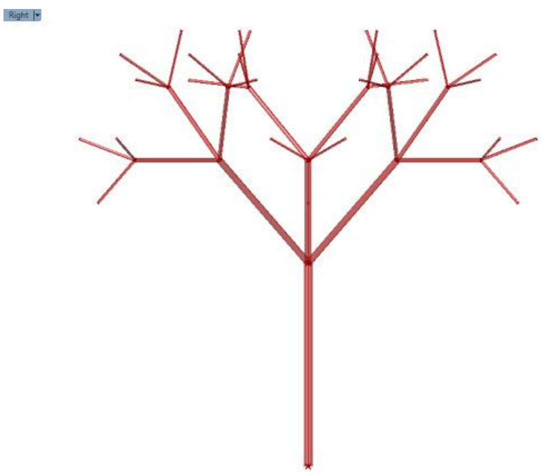
Рекурсивно дърво с ъгъл 100 градуса. Вижда се как ъгъла не образува огледален образ.



Рекурсивно дърво с ъгъл 60 градуса. Вижда се как ъгъла не образува огледален образ.



Рекурсивно дърво с ъгъл 90 градуса. Вижда се как ъгъла образува огледален образ.



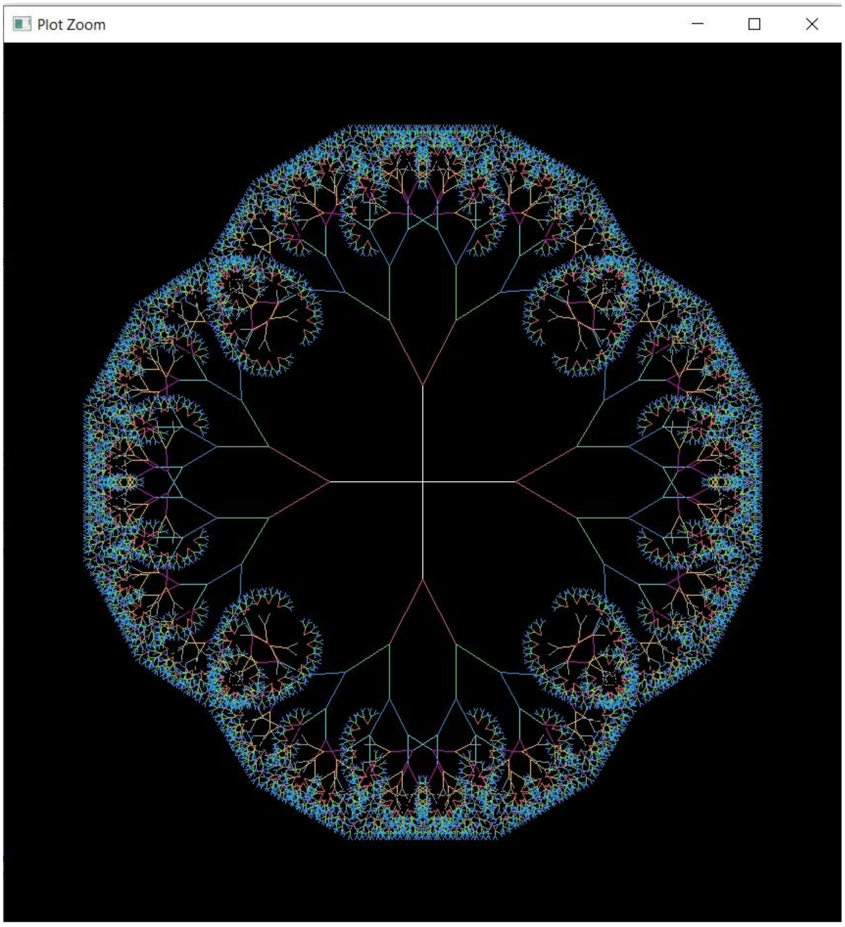

Рекурсивно дърво с ъгъл 90 градуса и намаляване на ъгъла при всяка итерация.

# 4. Програмна реализация

```
# Fractal project - Recursive trees. Creator: Veselin Todorov 2021
rm(list = ls())
#background creation
background <- function(xLimits, yLimits, color)
{
  par(mar=rep(1,4), bg=color)
  plot(1, type="n", bty="n", col = "black", bg = color, xlab= "", ylab = "",xlim = xLimits, ylim = yLimits) #s
}

#function to draw a line
drawLine <- function(lineCoordinates, color = "green", linewidth = 1)
{
  segments(lineCoordinates[1], lineCoordinates[2], lineCoordinates[3], lineCoordinates[4], color, linewidth) #
}

#newline function
newLine <- function(line, angle, reduce=1) {

  x0 <- line[1]
  y0 <- line[2]
  x1 <- line[3]
  y1 <- line[4]

  dx <- unname(x1-x0)                    # how much x have changed
  dy <- unname(y1-y0)                    # how much y have changed
  l <- sqrt(dx^2 + dy^2)                 # length of the line

  angleBetween <- atan(dy/dx) * (180 / pi)          # angle between line and origin
  rad <- (angle + angleBetween) * (pi / 180)          # (angle + new angle) in radians

  coeff <- sign(angleBetween) * sign(dy)            # coefficient of direction
  if(coeff == 0) coeff <- -1

  x2 <- x0 + coeff*l*cos(rad)*reduce + dx  # new x location
  y2 <- y0 + coeff*l*sin(rad)*reduce + dy  # new y location
  return(c(x1,y1,x2,y2))

}

iterate <- function(object, ifun, ...) {
  linesList <- vector("list",0)
  for(i in 1:nrow(object)) {
    old_line <- matrix(object[i,], nrow=1)
    new_line <- ifun(old_line, ...)
    linesList[[length(linesList)+1]] <- new_line
  }
  new_object <- do.call(rbind, linesList)
  return(new_object)
}

#function that draws object
drawObject <- function(object, col="white", lwd=1)
{
  invisible(apply(object, 1, drawLine, col=col, linewidth=lwd))
}
```

```r
# iterator function: recursive tree
tree <- function(line0, angle=30, reduce=.7, randomness=0) {

  # angles and randomness
  angle1 <- angle+rnorm(1,0,randomness)  # left branch
  angle2 <- -angle+rnorm(1,0,randomness) # right branch

  # new branches
  line1 <- newLine(line0, angle=angle1, reduce=reduce)
  line2 <- newLine(line0, angle=angle2, reduce=reduce)

  # store in matrix and return
  mat <- matrix(c(line1,line2), byrow=T, ncol=4)
  return(mat)

}

# example: recursive tree (after ten iterations)
Z <- c(0,0) #point of center
A <- c(1e-9,5) #direction A
B <- c(5,-1e-9) #direction B
background(xLimits = c(-20,20), yLimits = c(-20, 20), color = "black")
fractal <- matrix(c(Z,A,Z,B,Z,-A,Z,-B), nrow=4, byrow=T) #first fractal
drawObject(fractal) #draw E1
for(i in 1:11) {
  fractal <- iterate(fractal, ifun=tree, angle=29, reduce=.75) #change l
  drawObject(fractal, col=i+1) #draw E2 ... E12
}
```

## Резултат: (4 рекурсивни дървета с общо начало)



## 5. Източници


Fractal tree - Rosetta Code

RPubs - Fractals with R