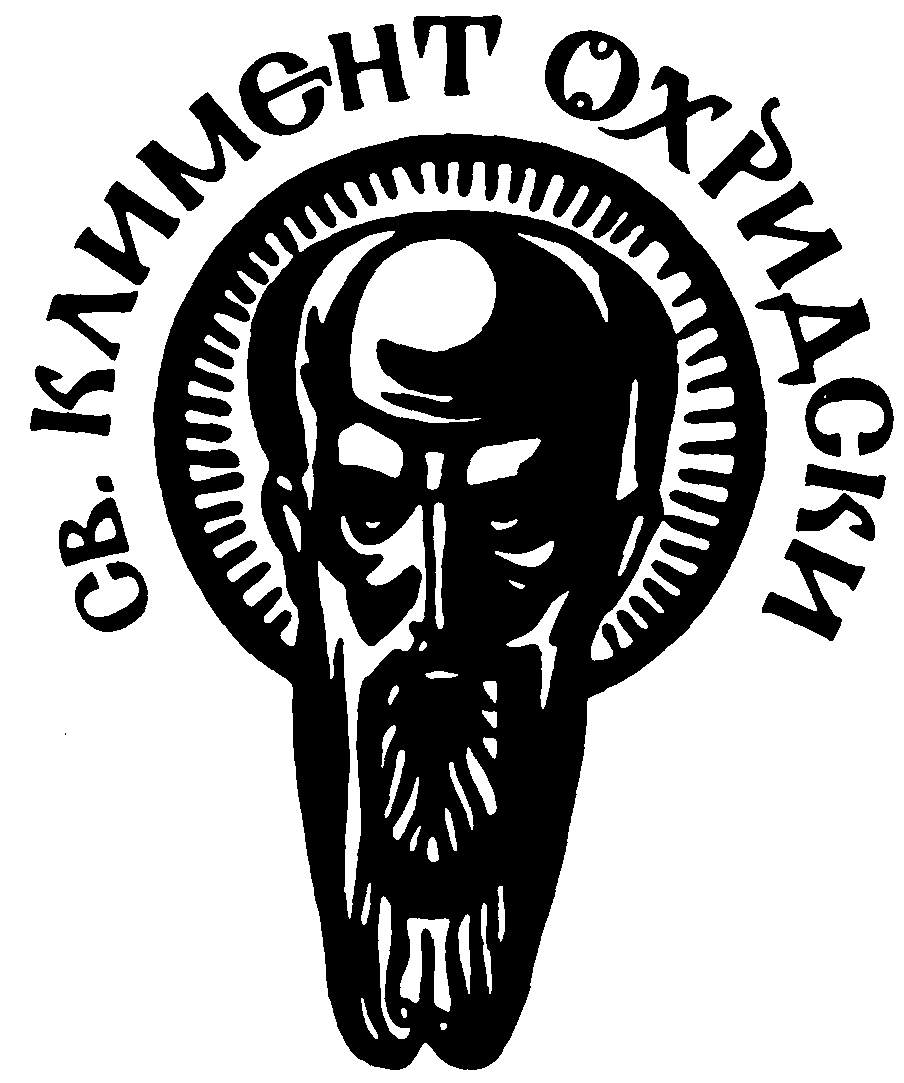
Софийски университет св.Климет Охридски

ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА



Проект по Фрактали

на тема

Снежинка на Кох

(Koch snowflake)

Изготвила: Любка Димитрова Ангелинина

Специалност: Софтуерно инженерство

Курс: Втори

Ф.Н.:62342

Малко повече за фракталите

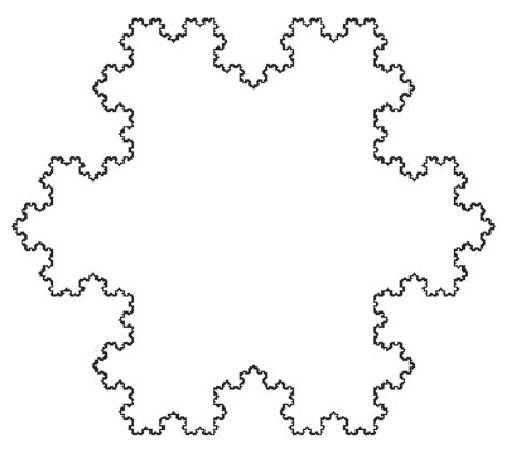
Фракталите са фигури с безкрайно многото детайли. Наименованието фрактал произлиза от латинската 'fractus' – разбит и е въведен от Манделброд. През 1982 година Манделброт издава книгата „Фрактална геометрия на природата“, в която авторът събрал и систематизирал практически цялата налична до този момент информация за фракталите и в лек и достъпен стил я е изложил. Когато ги гледаме от по-близо те не стават по-прости, а се запазват толкова сложни, колкото и без увеличение. Свойството частите на един обект да са подобни на целият обект се нарича самоподобие. Самоподобието е характерно за всички математически фрактали, а една от основните им характеристики е размерността.

Фракталите са средства за описване на обекти като модели на планински вериги, грапави брегови линии, кръвоносни системи на много капиляри и съдове, корони на дървета и т.н. Изображенията на такива обекти могат да бъдат представени с помощта на фрактална графика. В наши дни теорията на фракталите намира широко приложение в различни области от човешката дейност. Освен чисто научния обект за изследване, фракталите се използват в теорията на информацията за свиване на географски данни и при фракталната живопис.

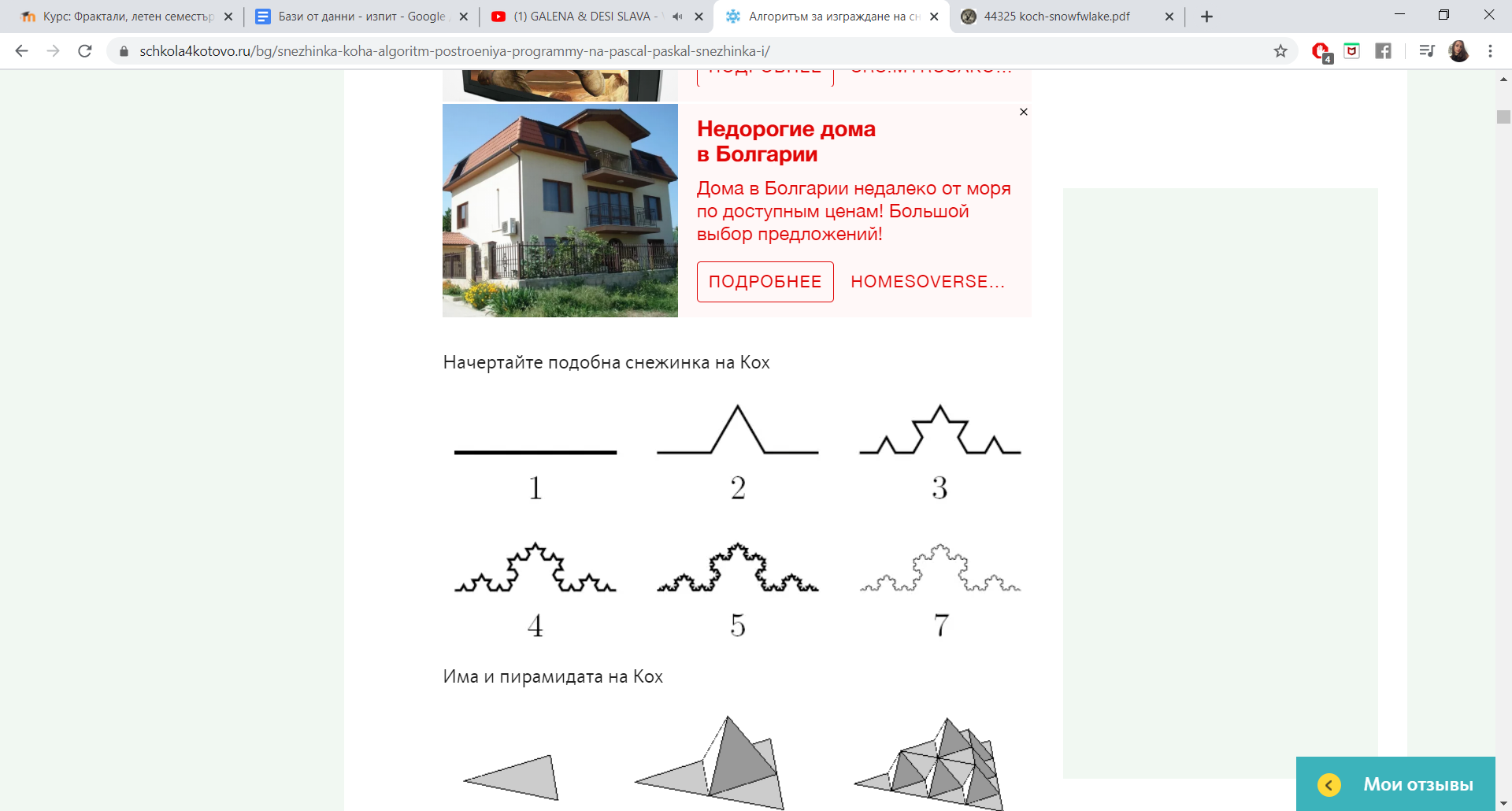
*Естествен природен фрактал е снежинката*



Снежинка на Кох



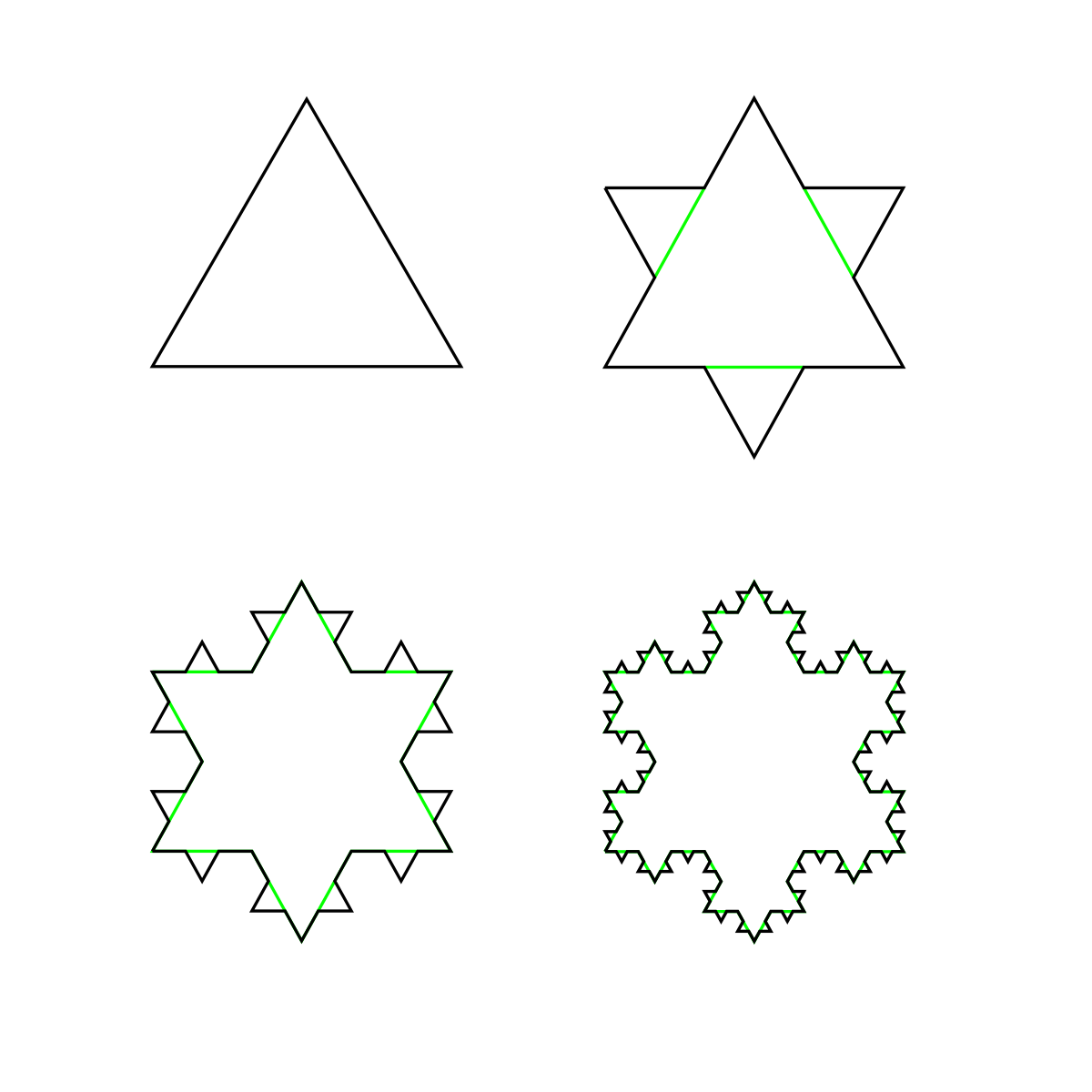
През 1904 г. Хелге фон Кох измислил непрекъсната крива, която никъде няма допирателна, освен това тя лесно се рисува. Всъщност кривата е измислена като пример за непрекъсната линия, към която е невъзможно да се начертае допирателна линия във всяка точка. Кривата на Кох е проста в дизайна. Оказало се, че тя притежава свойствата на фрактала. Един от вариантите на тази крива носи името „снежинка на Кох“. Снежинката на Кох е един от първите фрактали, изследвани от учените. От три екземпляра на кривата на Кох се получава снежинка.



Създаване на Кохова снежинка

Изграждането е много прост итеративен процес:

1. Започваме с равностранен триъгълник, който всъщност е нулева итерация на снежинката на Кох.
2. Намираме централната точка на всеки ръб на текущата снежинка.
3. В центъра на всеки ръб добавяме равностранен триъгълник, стърчащ навън със страна, равна на 1/3 от дължината на текущия ръб.
4. Определяме следващата итерация да бъде извън външната страна на предишната снежинка и всички добавени триъгълници.
5. Повторете стъпките по-горе толкова пъти, колкото е необходимо.



# Доказано е, че има редица интересни свойства. Например, дължината на периметъра е равна на безкрайността, което обаче не пречи да покрие ограничена площ, чиято стойност е 8/5 от площта на основния триъгълник. Поради този факт, някои приложени техники и параметри на равнинни фигури, като например граничния индекс (съотношението на периметъра към корена на района), не са приложими при работа със снежинката на Кох. Периметър и лице на снежинката на Кох

1. Периметър:

След всяка итерация броят на страните на снежинката на Кох се увеличава със степен на 4, така че броят на страните след n итерации е:

N_{n} = N_{n-1} \cdot 4 = 3 \cdot 4^{n}\, .

Нека равностранният триъгълник има страна с дължина s. Тогава дължината на всяка страна на снежиката след n итерации ще е:

S_{n} = \frac{S_{n-1}}{3} = \frac{s}{3^{n}}\, .

И така за периметъра получаваме, че след n итерации е:

 P_{n} = N_{n} \cdot S_{n} = 3 \cdot s \cdot {\left(\frac{4}{3}\right)}^n\, .

Общо взето след всяка итерация периметърът нараства-той расте до безкрайност, въпреки че затворената площ остава крайна.

1. Лице

След всяка итерация се добавя нов триъгълник на всяка страна от предишната итерация. Следователно броят на добавените нови тръгълници след n итерации е:

T_{n} = N_{n-1} = 3 \cdot 4^{n-1} = \frac{3}{4} \cdot 4^n\, .

Примерен код за изчертаване на снежинка на Кох до 7 итерации:

style.css

body{

background-color: #fff;

width: 70%;

margin: 15% auto;

font-family: 'Abhaya Libre', serif;

}

h1 {

font-family: 'Amita', cursive;

text-align: left;

font-size:48px;

color: #606D72;

}

h2 {

font-family: 'Amita', cursive;

color: #606D72

}

button {

font-size: 20px;

}

#main {

position:relative;

}

#sidepanel {

position:absolute;

right:0;

top:50;

padding-bottom:10%;

}

#container {

width:60%;

position:relative;

left:0;

top:0;

}

#imageView {

position:absolute;

top:50;

left:0;

}

#Snowflake {

font-size: 24px;

}

*Функциите за изчертаване:*

Index.js

var canvas=null;

var context=null;

window.onload = init()

function init()

{

W = 600

H = 600

var arg1 = document.getElementById("iter").value;

var flen = document.getElementById("size").value;

//var nlen = flen\*13.4;

colorarg1 = document.getElementById("scolor").value;

//colorarg2 = document.getElementById("bcolor").value;

canvas = document.getElementById("imageView");

context = canvas.getContext('2d');

canvas.width=W;

canvas.height=H;

container.appendChild(canvas);

context = canvas.getContext("2d");

context.beginPath();

context.stroke();

context.closePath();

fractal([5\*flen,15\*flen], [50\*flen,15\*flen],arg1);

fractal([27\*flen,49\*flen], [5\*flen,15\*flen],arg1);

fractal([50\*flen,15\*flen],[27\*flen,49\*flen],arg1);

};

function fractal(A, B, depth){

if (depth < 0){

return null;

}

var C = divide(add(multiply(A, 2), B), 3);

var D = divide(add(multiply(B, 2), A), 3);

var F = divide(add(A, B), 2);

var V1 = divide(minus(F, A), length(F, A));

var V2 = [V1[1], -V1[0]];

var E = add(multiply(V2, Math.sqrt(3)/6 \* length(B, A)), F);

DrawLine(A, B, colorarg1);

if (depth !=0){

for (var i=0;i<10;i++)

DrawLine(C, D, '#FFFFFF');

};

fractal(A, C, depth-1);

fractal(C, E, depth-1);

fractal(E, D, depth-1);

fractal(D, B, depth-1);

};

function multiply(v, num){

return [v[0]\*num, v[1]\*num];

};

function divide(v, num){

return [v[0]/num, v[1]/num];

};

function add(a, b){

return [a[0]+b[0], a[1]+b[1]];

};

function minus(a, b){

return [a[0]-b[0], a[1]-b[1]];

};

function length(a, b){

return Math.sqrt(Math.pow(a[0] - b[0],2) +

Math.pow(a[1] - b[1],2));

};

function DrawLine(a, b, c){

context.beginPath();

context.strokeStyle = c;

context.lineWidth=3;

context.moveTo(a[0], a[1]);

context.lineTo(b[0], b[1]);

context.stroke();

context.closePath();

};

function ResetFlake() {

context.clearRect(0,0, canvas.width, canvas.height);

};

Index1.html

<!DOCTYPE html>

<html lang="en" >

<head>

<meta charset="UTF-8">

<title>Koch Snowflake</title>

<link href="https://fonts.googleapis.com/css?family=Amita" rel="stylesheet">

<link href="https://fonts.googleapis.com/css?family=Abhaya+Libre" rel="stylesheet">

<link rel='stylesheet prefetch' href='https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/3.3.7/css/bootstrap.min.css'>

<link rel="stylesheet" href="css/style.css">

</head>

<body>

<head>

<title>Koch Snowflake Fractal</title>

</head>

<body>

<div id="main"><h1>Снежинка на Кох</h1>

<div id="sidepanel">

<form id="Snowflake">

Цвят:<br>

<input type="text" id="scolor" name="scolor"

value="#000000"><br><br>

Брой итерации:<br>

<input type="number" id="iter" name="iter" value="4" min="0" max="7"><br><br>

Размер:<br>

<input type="number" id="size" name="size" value="7" min="7" max="12"><br><br>

<button type="button" onclick="init(); return false;">Генерирай Снежинка на Кох</button>

<button type="button" onclick="ResetFlake(); return false;">Нулирай</value>

</form>

</div>

<div id="container">

<canvas id="imageView" ></canvas>

</div> </div>

</body>

<script src="js/index.js"></script>

</body>

</html>