КУРСОВ ПРОЕКТ

за избираема дисциплина „Фрактали“ на тема

„Нютонов фрактал“

Преподавател: доц. д-р Милко Такев

Изготвил: Валентин Павлинов Крумов, Софтуерно инженерство, III курс, ФН: 62539

летен семестър 2022/2023

гр.София

Съдържание:

Инструкции за използване на програмата:

1. Зареждаме диска на компютър
2. Отваряме файла 62539.html
3. В самата програма има допълнителни инструкции и описание

1. Фрактал – Дефиниция

Фракталите са сложни математически обекти, които демонстрират свойството на **самоподобие** на различни мащаби. Те се състоят от повтарящи се шаблони, които се наблюдават в различни нива на детайли и размери. Терминът фрактал е въведен през 1975г. от **Беноа Манделброт**. Фракталите имат широко приложение и се срещат както в природата, така и в изкуството и науката. Един от най-известните фрактали е **"Множеството на Манделброт".**

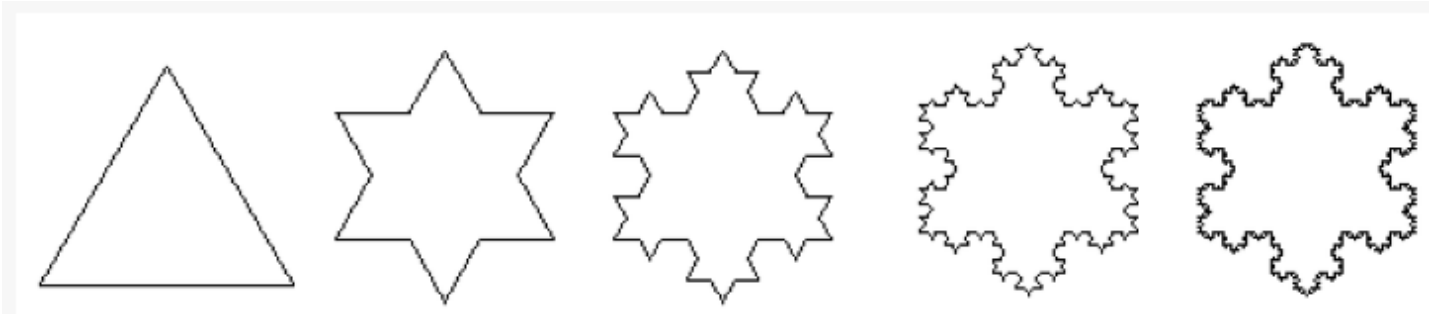
В природата, при растенията можем да видим фрактални структури в разклоненията на дърветата, възлите на стъблата, а също и в изграждането на листата и цветовете. Облаците в атмосферата също се оформят по фрактални принципи, като образуват структури със самоподобни форми. Фракталите се наблюдават и в други природни явления като речни системи, геологични формации и океански вълни. Те помагат за обясняване на сложни процеси и формирането на природната среда. Например, речните системи се разклоняват по фрактални принципи, което улеснява разпределението на вода и хранителни вещества в екосистемите.

Фракталите имат и практически приложения в научни и технически области. В компютърната графика се използват фрактални алгоритми за създаване на сложни и реалистични изображения, като пейзажи, текстури и симулации на природни явления. Фракталната компресия на данни позволява ефективно съхранение и предаване на информация чрез използване на самоподобни структури.

Фракталите намират приложение и в финансовата област. Те се използват за моделиране на финансови времеви редове и

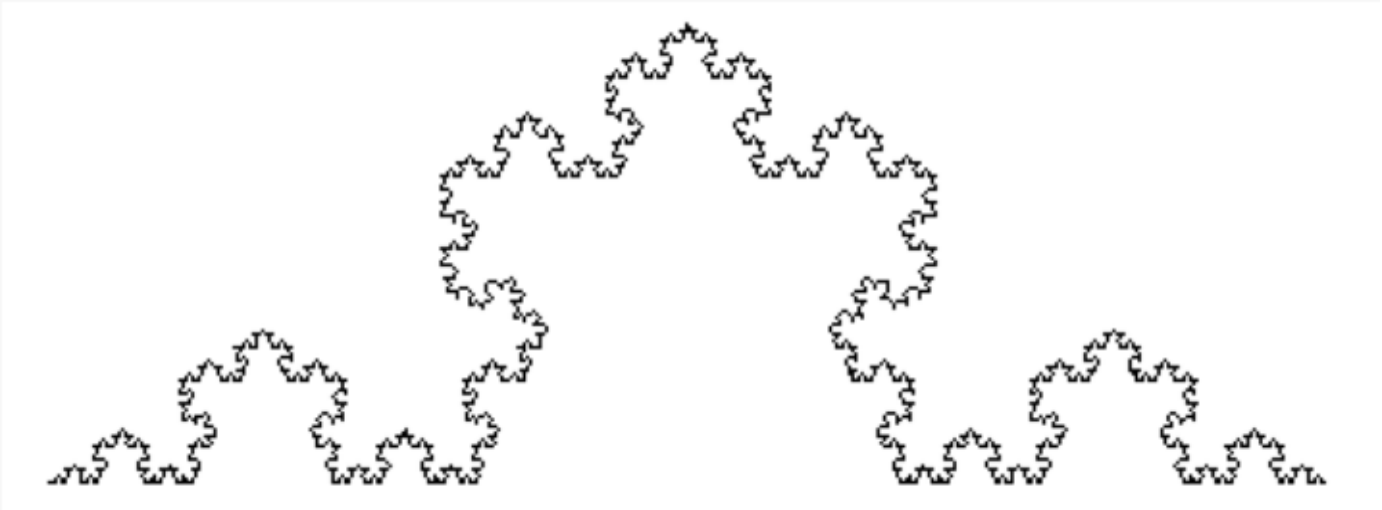
предсказване на финансови пазари. Фракталната анализа помага в идентифицирането на шаблони и трендове в финансовите данни, което може да подпомогне при вземането на инвестиционни решения.

2. Визуални примери



пример 1: Снежинка на Кох

Тук в началото имаме равностранен триъгълник. След това всяка една от страните му се разделя. Средната една трета от всяка страна на



триъгълника се маха. Тази премахната отсечка се използва като основа на друг равностранен триъгълник, като се добавят другите му две страни. На фигурата по-горе са изобразени първите няколко итерации.
Продължавайки този процес накрая се получава фрактала „Снежинка на Кох“. Ето и приближена снимка на единия край след доста итерации:

пример 2: Триъгълник на Серпински

Фракталът наречен „Триъгълник на Серпински“ представлява триъгълник, съставен от по-малки копия на себе си. Започвайки с пълен равностранен триъгълник, свързваме средата на всяка страна и премахваме средния триъгълник, който се получава. При следващата итерация повтаряме този процес за върху останалите три пълни триъгълника и така нататък.

3. Нютонов фрактал

Нютоновият фрактал е математически обект, който се основава на метода на Нютон за намиране на корени на уравнения. Той представлява графично представяне на итеративния процес за намиране на корените на комплексно аналитични функции.

За да създадем Нютонов фрактал, избираме комплексна функция, например полиномиална функция от вида **f(z) = z^n - c**, където **n** е положително цяло число, а c е комплексна константа. Започваме с определено начално предположение за корена на уравнението и използваме итеративен процес, за да намерим все по-точното приближение към корена.

Итеративният процес се основава на следната формула:

z_new = z_old - (f(z_old) / f'(z_old))

където **z_old** е предишното приближение към корена, **f(z_old**) е стойността на функцията в тази точка, **f'(z_old)** е първата производна на функцията в тази точка, а **z_new** е новото приближение към корена. Този процес се повтаря множество пъти, докато достигнем желана точност или зададен брой итерации.

В графичното представяне на Нютоновия фрактал, цветовете на пикселите или точките в равнината се определят от броя итерации, необходими за сходимост към корена или отблъскване от него. Това води до сложни и красиви геометрични структури, които се повтарят в различни мащаби и детайли.

Нютоновият фрактал е изключително интересен, тъй като демонстрира красотата и сложността на математиката и комплексните числа. Ето някои допълнителни факти за Нютоновия фрактал:

1. Структура: Нютоновият фрактал има характерна структура, която се състои от области с различни цветове или нюанси. Всяка област представлява сходимост към различен корен на уравнението. Фракталът може да има сложни извивки, острови, спирални структури и други геометрични форми.

2. Корени: Числото на корените на уравнението определя броя области в Нютоновия фрактал. Например, ако имаме полином от вида z^3 - c, където c е комплексна константа, ще имаме три корена и съответно три области във фрактала.

3. Цветове: Цветовете, използвани в графичното представяне на Нютоновия фрактал, се определят от броя итерации за сходимост или отблъскване от корена. Обикновено се използват различни палитри или градиенти, които придават визуална привлекателност и дълбочина на фрактала.

4. Детайли и мащаб: Нютоновите фрактали са самоподобни в мащаба. Това означава, че при увеличаване или намаляване на

мащаба, подобни структури се повтарят във все по-фини детайли. Това прави фрактала уникален и красив при разглеждане на различни нива на увеличение.

5. Приложения: Нютоновите фрактали имат приложения в математиката, особено в областта на комплексния анализ и числените методи за намиране на корени на уравнения. Те се използват и за изследване на динамични системи и за създаване на визуални ефекти и изкуство.

4. Програма за нютонов фрактал, написана на HTML5 и JS

```
<!DOCTYPE html>
<html>

<head>
  <title>Нютонов фрактал</title>
  <style>
    #display {
      border: 1px solid black;
    }

    .start-button {
      background-color: #e1ecf4;
      border-radius: 3px;
      border: 1px solid #7aa7c7;
      box-shadow: rgba(255, 255, 255, .7) 0 1px 0 0 inset;
      box-sizing: border-box;
      color: #39739d;
      cursor: pointer;
      display: inline-block;
      font-family: -apple-system, system-ui, "Segoe UI", "Liberation Sans",
sans-serif;
      font-size: 13px;
      font-weight: 400;
      line-height: 1.15385;
      margin: 0;
      outline: none;
      padding: 8px .8em;
      position: relative;
      text-align: center;
      text-decoration: none;
```

```
    user-select: none;
    -webkit-user-select: none;
    touch-action: manipulation;
    vertical-align: baseline;
    white-space: nowrap;
  }

  .start-button:hover,
  .start-button:focus {
    background-color: #b3d3ea;
    color: #2c5777;
  }

  .start-button:focus {
    box-shadow: 0 0 0 4px rgba(0, 149, 255, .15);
  }

  .start-button:active {
    background-color: #a0c7e4;
    box-shadow: none;
    color: #2c5777;
  }

  .stop-button {
    align-items: center;
    background-color: #FFE7E7;
    background-position: 0 0;
    border: 1px solid #FEE0E0;
    border-radius: 11px;
    box-sizing: border-box;
    color: #D33A2C;
    cursor: pointer;
    display: flex;
    font-size: 1rem;
    font-weight: 700;
    line-height: 33.4929px;
    list-style: outside
url(https://www.smashingmagazine.com/images/bullet.svg) none;
    padding: 2px 12px;
    text-align: left;
    text-decoration: none;
    text-shadow: none;
    text-underline-offset: 1px;
    transition: border .2s ease-in-out, box-shadow .2s ease-in-out;
    user-select: none;
    -webkit-user-select: none;
    touch-action: manipulation;
    white-space: nowrap;
    word-break: break-word;
```

```
}

.stop-button:active,
.stop-button:hover,
.stop-button:focus {
  outline: 0;
}


.stop-button:active {
  background-color: #D33A2C;
  box-shadow: rgba(0, 0, 0, 0.12) 0 1px 3px 0 inset;
  color: #FFFFFF;
}

.stop-button:hover {
  background-color: #FFE3E3;
  border-color: #FAA4A4;
}

.stop-button:active:hover,
.stop-button:focus:hover,
.stop-button:focus {
  background-color: #D33A2C;
  box-shadow: rgba(0, 0, 0, 0.12) 0 1px 3px 0 inset;
  color: #FFFFFF;
}

html {
  text-align: center;
}

body {
  background-color: #dfded9;
  margin-bottom: 50px;
}

canvas {
  background-color: white;
}

h3 {
  font-weight: 400;
}

h4 {
  text-align: start;
  font-weight: 400;
  margin-left: auto;
```

```
      margin-right: auto;
      width: 50%;
      font-size: 18px;
    }

    .controls {
      margin-top: 15px;
      display: flex;
      align-items: center;
      justify-content: center;
      gap: 10px;
    }

    .stop-animation {
      background-color: #FF4742;
    }
  </style>
</head>

<body>
  <h1>
    Нютонов фрактал
  </h1>
  <h3>
    Курсов проект за избираема дисциплина <b>"Фрактали"</b>, летен семестър 2021/2022
    <br>
    Изготвен от <b>Валентин Крумов, ФН: 62539</b>
  </h3>
  <h4>
    Нютоновият фрактал е математически обект, който се основава на метода на Нютон за намиране на корени на уравнения.
    Той представлява графично представяне на итеративния процес за намиране на корените на комплексно аналитични
    функции.
    <br>
    <br>
    При визуализацията на Нютонов фрактал се използват два основни метода - методът на коритото <b>(basin method)</b> и методът
    на ъглите <b>(angle method)</b>.
    <br>
    <br>
    Методът на коритото се базира на идеята, че ако комплексната функция има корен в определена точка, то при достатъчно
    голям брой итерации точката, започвайки от дадена начална точка, ще се приближи до този корен. Визуализацията се
    извършва, като се изчислява последователност от итерации за всяка точка на комплексната равнина и се определя
```

```
    корена, към който се приближава. В резултат се получава разноцветна карта, където цветът на дадена точка указва към
    кой корен тя се приближава или дали остава в близост до началната точка.
    <br>
    <br>
    Методът на ъглите, от друга страна, използва комплексните аргументи на итерациите, за да определи корените на
    функцията. Всяка итерация се изчислява и се преобразува в комплексно число. След това се изчислява аргументът на
    това комплексно число. Визуализацията се постига чрез отразяване на аргументите в цветови палитра. Като резултат се
    получава изображение, където цветовете указват към кой корен съответства даден ъгъл.
    <br>
    <br>
    В това приложение има възможност да се нарчетае Нютонов фрактал посредством двата метода чрез съответните бутони.
    Има и текстово поле, в което потребителят може да въведе броя итерации на фрактала. За всеки случай има и бутон, който спира анимацията.
  </h4>
  <canvas width="1000" height="500" id="display"></canvas>
  <br>
  <div class="controls">
    <p>Брой итерации:</p>
    <textarea rows="1" , cols="2" , id="iterations">5</textarea>
    <button class="start-button" onclick="generate('basin')">Генерирай с Basin Method (Метод на коритото)</button>
    <button class="start-button" onclick="generate('angle')">Генерирай с Angle Method (Метод на ъглите)</button>
  </div>
  <div class="controls">
    <button class="stop-button" onclick="if (generating) {stop = true;}">Спри Анимация</button>
  </div>
  <script>
    function complex(real, imag) {
      this.real = real;
      this.imag = imag;

      this.square = function () {
        var a = this.real;
        var b = this.imag;
        return new complex(a * a - b * b, 2 * a * b);
      }

      this.cube = function () {
        var a = this.real;
        var b = this.imag;
```

```
        return new complex(a * a * a - 3 * a * b * b, 3 * a * a * b - b * b *
b);
      }

      this.abs = function () {
        var a = this.real;
        var b = this.imag;
        return (Math.sqrt(a * a + b * b));
      }

      this.neg = function () {
        var a = this.real;
        var b = this.imag;
        return new complex(-1 * a, -1 * b);
      }

      this.angle = function () {
        var a = this.real;
        var b = this.imag;
        if (a >= 0) {
          var theta = Math.atan(b / a);
        } else {
          var theta = Math.atan(b / a) + Math.PI;
        }
        if (arguments[0] == "deg") {
          theta = 180 * theta / Math.PI;
        }
        return theta;
      }
    }

    function add() {
      var a = 0;
      var b = 0;
      for (var i = 0; i < arguments.length; i++) {
        a = a + arguments[i].real;
        b = b + arguments[i].imag;
      }
      return new complex(a, b);
    }

    function multiply() {
      var a = arguments[0].real;
      var b = arguments[0].imag;
      for (var i = 1; i < arguments.length; i++) {
        var a2 = arguments[i].real;
        var b2 = arguments[i].imag;
        var aTemp = a;
        a = a * a2 - b * b2;
```

```
      b = aTemp * b2 + b * a2;
    }
    return new complex(a, b);
  }

  //takes two complex numbers (from class "complex") as parameters and
divides the first by the second
  function divide(comp1, comp2) {
    var a = comp1.real;
    var b = comp1.imag;
    var a2 = comp2.real;
    var b2 = comp2.imag;
    var aNew = (a * a2 + b * b2) / (a2 * a2 + b2 * b2);
    var bNew = (b * a2 - a * b2) / (a2 * a2 + b2 * b2);
    return new complex(aNew, bNew);
  }

  var xMin = -2;
  var xMax = 2;
  var yMin = -1;
  var yMax = 1;

  var position = new complex();
  var root1 = new complex(1, 0);
  var root2 = new complex(-0.5, Math.sqrt(3) / 2);
  var root3 = new complex(-0.5, -1 * Math.sqrt(3) / 2);

  var display = document.getElementById('display');
  var ctx = display.getContext('2d');

  function renderBasin(d1, d2, d3, x, y) {
    if (d1 < d2 && d1 < d3) {
      ctx.fillStyle = "#ff0000";
      ctx.fillRect(x, y, 1, 1);
    } else if (d2 < d3) {
      ctx.fillStyle = "#00ff00";
      ctx.fillRect(x, y, 1, 1);
    } else {
      ctx.fillStyle = "#0000ff";
      ctx.fillRect(x, y, 1, 1);
    }
  }

  function renderAngle(point, x, y) {
    var angle = point.angle("deg");
    ctx.fillStyle = "hsl(" + angle + ", 100%, 50%)";
    ctx.fillRect(x, y, 1, 1);
  }
```

```
var generating = false;
var stop = true;

function generate(method) {
  function loop2() {
    setTimeout(function () {
      if (generating) {
        stop = true;
        loop2();
      } else {
        stop = false;
        generating = true;
        var iterations = document.getElementById('iterations').value;
        var y = 0;
        loop1();
        function loop1() {
          setTimeout(function () {
            for (var x = 0; x < display.width; x++) {
              if (stop) { break; }
              position.real = xMin + x * (xMax - xMin) / display.width;
              position.imag = yMin + y * (yMax - yMin) / display.height;
              for (var i = 0; i < iterations; i++) {
                if (stop) { break; }
                position = add(
                  position,
                  divide(
                    add(
                      new complex(-1, 0),
                      position.cube()
                    ),
                    multiply(
                      new complex(3, 0),
                      position.square()
                    )
                  ).neg()
                )
              }
              var dist1 = add(position, root1.neg()).abs();
              var dist2 = add(position, root2.neg()).abs();
              var dist3 = add(position, root3.neg()).abs();
              if (method == "basin") {
                renderBasin(dist1, dist2, dist3, x, y);
              } else if (method == "angle") {
                renderAngle(position, x, y);
              }
            }
            y++;
            if (y < display.height && !stop) {
              loop1();
```

```
                } else {
                  generating = false;
                }
              }, 0);
            }
          }
        }, 0);
      }
      loop2();
    }
  </script>
</body>

</html>
```

5. Ресурси

- https://bg.wikipedia.org/wiki/%D0%A4%D1%80%D0%B0%D0%BA%D1%82%D0%B0%D0%BB
- https://bolcheknig.ru/bg/spravochnik/primery-fraktalov-v-prirode-raznoobraznyi-mir-fraktalov-fraktaly-muzykalnaya/
- https://hotel-all.ru/bg/novosti/udivitelnyi-mir-fraktalov-udivitelnyi-mir-fraktalov-fraktal/
- https://www.wikiwand.com/bg/%D0%9D%D1%8E%D1%82%D0%BE%D0%BD%D0%BE%D0%B2_%D1%84%D1%80%D0%B0%D0%BA%D1%82%D0%B0%D0%BB