Софийски университет „Св. Климент Охридски“

Факултет по математика и информатика
Специалност: “Софтуерно инженерство”
Курс: 3, Група: 2
Дисциплина: “Фрактали”

КУРСОВ ПРОЕКТ

Тема: Множество на Манделброт

Изготвил:
Надежда Койкова, ФН: 8MI0600112

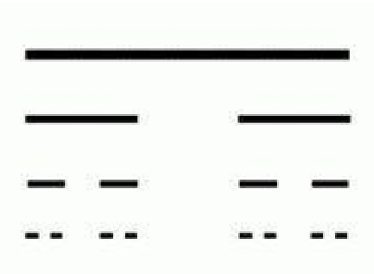
София
Летен семестър 2023/2024

Съдържание

1. История на фракталите

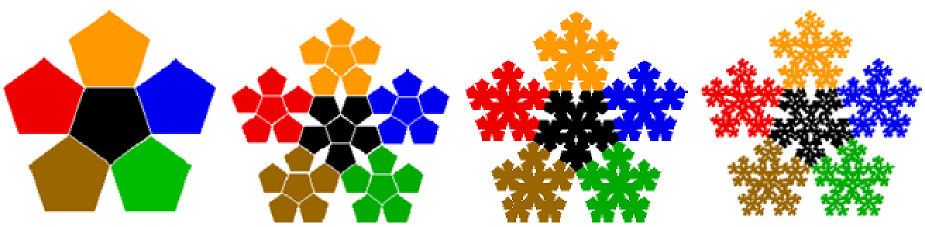
Макар че понятието „фрактал“ се появява сравнително отскоро, първите идеи на фракталната геометрия, са възникнали още през 19 век.

През 1883 г. Георг Кантор, един от основателите на теорията на множествата, описва как с помощта на проста, повтаряща се процедура, превръща линия в несвързани точки - фрактал, наречен прах на Кантор или гребен на Кантор. Образува се като последователно се премахва средната третина на една отсечка. Повторението на тази операция до безкрайност води до образуването на т.нар. канторови прашинки, сумата от дължините на които е равна на 0.



Фигура 1. Прах на Кантор

Може да се каже, че първия фрактал е описан в „Ръководство на живописеца“ (1525г) от Албрехт Дюрер. Тази творба представя т.нар. "петоъгълник на Дюрер", където пет петоъгълника са подредени около един централен. Получената фигура, състояща се от шест петоъгълника, наподобява петоъгълник, от който са изрязани пет триъгълни клина. Чрез повтаряне на този процес на изрязване на клинове, се генерират безброй следващи итерации, оформяйки фрактална структура.



Фигура 2. Петоъгълници на Дюрер – стъпки 0, 1, 2 и 3

Изучаването на фракталите в края на XIX и XX век е било по-скоро епизодично, а не систематично занимание. Причината е, че тогавашните математици са предпочитали да се фокусират върху обекти, които могат да бъдат описани с класическите методи и теории. Но дори и да се захванат с тази тема, те не са могли да съставят изображения на този математически модел, защото са били необходими огромно количество изчисления, които не било възможно да се извършат ръчно. С възхода на компютърната техника става възможно да се видят фракталите в целият им блясък и многообразие. Първият, който използва компютър за тази цел е математикът от компанията на IBM Беноа Манделброт.

Изследвайки на пръв поглед случайни колебания в цените, Манделброт открил, че те следват скрит математически ред във времето, който не се описва със стандартни криви. Колебанията на цените изглеждали произволни, но Манделброт открил симетрия между дългосрочните и краткосрочните колебания в цените. Така Беноа Манделброт полага основите на своя рекурсивен (фрактален) метод.

През 1975 г. Манделброт въвежда термина „фрактал“ (от латинската дума “fractus”, което означава "счупен"), за да опише математически обекти, които показват самоподобие и дробна размерност.

2. Фракталите - скритите съкровища на математическия свят

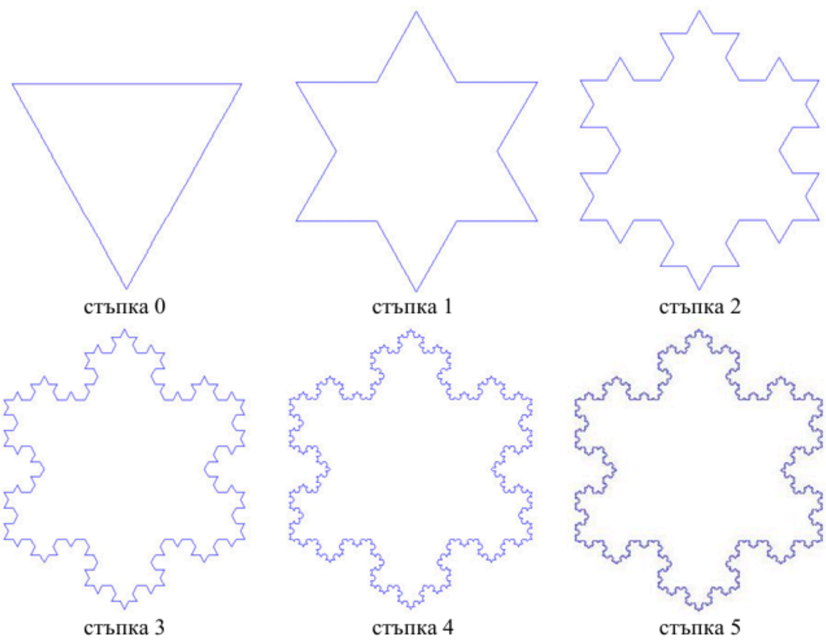
Едно от определенията за фрактал е геометрична фигура, състояща се от части, всяка, от които представлява по-малко, поне приблизително копие на цялото. Фракталът е такъв обект, за който няма значение под какъв мащаб го разглеждаме - структурата му остава същата. В основата на това явление е много проста идея: безкрайната красота и разнообразие на множество форми може да бъдат получени от сравнително прост модел само с две операции - копиране и мащабиране.

3. Видове фрактали

a. Геометрични фрактали

Наричат ги още линейни или класически фрактали, защото математиците са стигнали първо до тях. Тези фрактали са лесни за построяване и буквално всеки може да начертае такъв фрактал върху кариран лист. Тези фрактали се образуват с итерации.

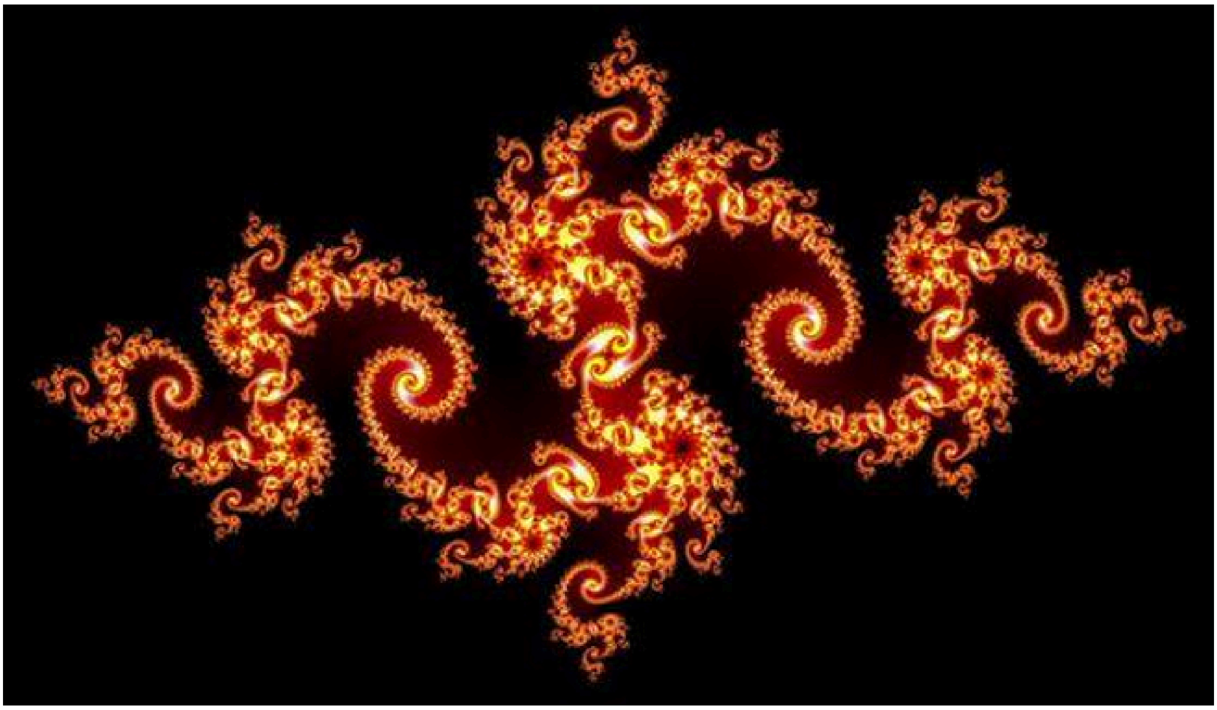
В двумерния случай геометричните фрактали се получават с помощта на някаква начупена линия или плоска геометрична фигура, наречена основа или генератор. За всяка стъпка от алгоритъма, всяка от отсечките, съставящи начупената линия, се заменя с генератора, в съответния мащаб. В резултат на безкрайни повторения (итерации) на тази процедура, се получава геометричния фрактал.



Фигура 3. Снежинка на Кох - итеративно построяване

b. Алгебрични фрактали

Това е другата основна група фрактали. Наричат се алгебрични, защото те са изградени на базата на алгебрични формули. Алгебричните фрактали изглеждат безкрайно сложни в сравнение с геометричните, но могат да бъдат генерирани с много проста формула.



Фигура 3. Множество на Жулиа

c. Случайни фрактали

Природните явления не протичат в стерилна, изолирана среда, а в комплексни условия, съчетание от множество преплетени случайни влияния от външни фактори, затова в природата фракталите не изглеждат така изчистени както илюстрациите досега. За моделирането им се използват стохастични (случайни) фрактали.

Стохастичен фрактал е траекторията на Брауново движение. Такива фрактали са рандомизирани, т.е. при построяването им, за всяка рекурсивна стъпка се въвеждат случайни величини.

Фракталните структури са най-видими при растенията. По същество, самият им растеж е подчинен на итерации. Примерите за фрактали в растителния свят са многобройни, някои очевидни, други не толкова. Повечето растения имат някаква форма на разклонение - основното стъбло се разделя на няколко клона, всеки от тях се разделя на по малки клонки и това продължава до най-малките клончета. Вероятно сте обърнали внимание, че клонче от дървото изглежда подобно на цялото дърво и листче от папрат изглежда почти идентично на цялото растение. Едни от най-популярните фрактали са папратите, получени от британския математик Майкъл Барнсли през 1988 г. Могат да се разглеждат като вариант на дървовидните фрактали.

Фигура 4. Естествена папрат и папрат на Барнсли

4. Множество на Манделброт - откритие

Множеството на Манделброт е едно от най-забележителните и красиви фрактали в математиката, което се откроява със своята сложна структура и естетична привлекателност. Открито и популяризирано от френския математик Беноа Манделброт през 80-те години на 20-ти век, това множество представлява геометричното представяне на комплексна итеративна функция. Беноа Манделброт работи в областта на фракталите и теориите за самоподобие. С помощта на компютърни изчисления, той успява да визуализира сложни форми, които се повтарят на различни мащаби. Множеството на Манделброт е едно от първите и най-известните примери на фрактална геометрия.

5. Множество на Манделброт - определение

Множеството на Манделброт се определя по следния начин:

i. Разглеждаме комплексното уравнение $z_{n+1} = z_n^2 + c$, където z и c са комплексни числа, а n е цял положителен индекс.
ii. Започваме с $z_0 = 0$ и изчисляваме следващите стойности на z чрез рекурентната формула.
iii. Комплексното число c е част от множеството на Манделброт, ако редицата $\{z_n\}$ остава ограничена по модул, тоест не се стреми към безкрайност, когато n расте.

За да разберем по-добре множеството на Манделброт, ще разгледаме един лесен пример с комплексни числа и ще проследим поведението на редицата.

iv. Пример: $c = 0$
   1. $z_0 = 0$
   2. $z_1 = z_0^2 + c = 0^2 + 0 = 0$
   3. $z_2 = z_1^2 + c = 0^2 + 0 = 0$

Редицата остава нула за всяка стойност на n, така че $c = 0$ принадлежи към множеството на Манделброт.

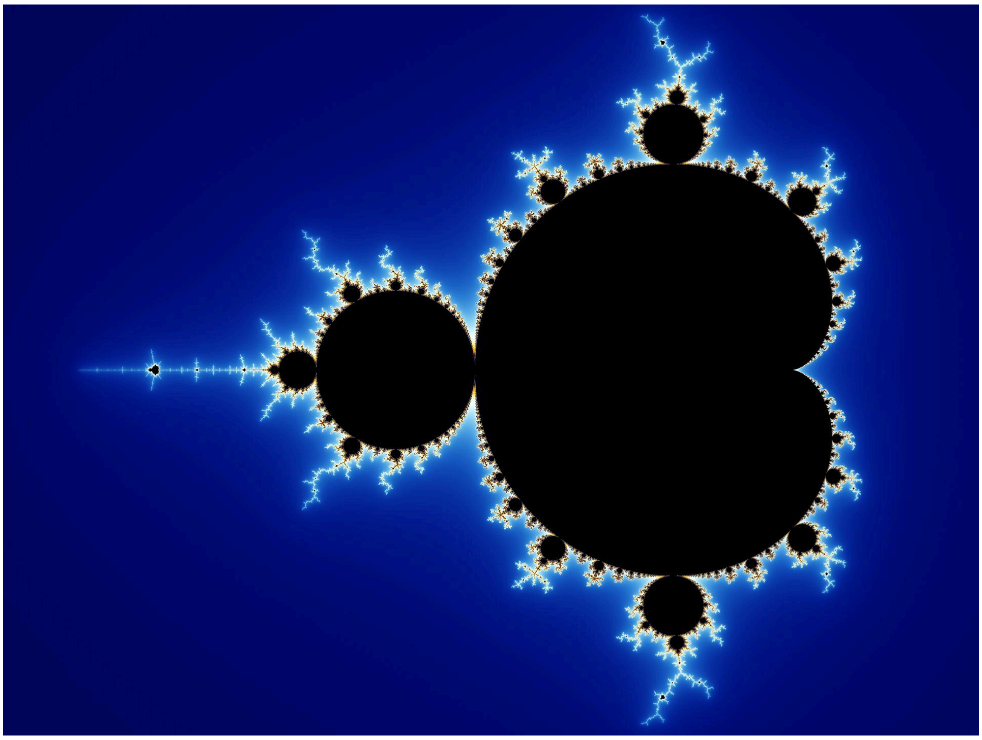
6. Множество на Манделброт - графично представяне

За да визуализираме множеството на Манделброт, обикновено трябва да бъдат изпълнени няколко стъпки, които ще изброим в следващите редове:

i. Разделяме комплексната равнина на решетка от точки, всяка от които представлява комплексно число c.
ii. За всяка точка c, започваме с $z_0 = 0$ и изчисляваме следващите стойности на z до определен брой итерации.
iii. Ако след тези итерации модулът на z_n не е надхвърлил определена граница, тогава точката c се счита за част от множеството и се оцветява в черно.

iv. Ако модулът на z_n надхвърли границата, точката c не е част от множеството и се оцветява в цвят, който зависи от броя на итерациите, необходими за това.

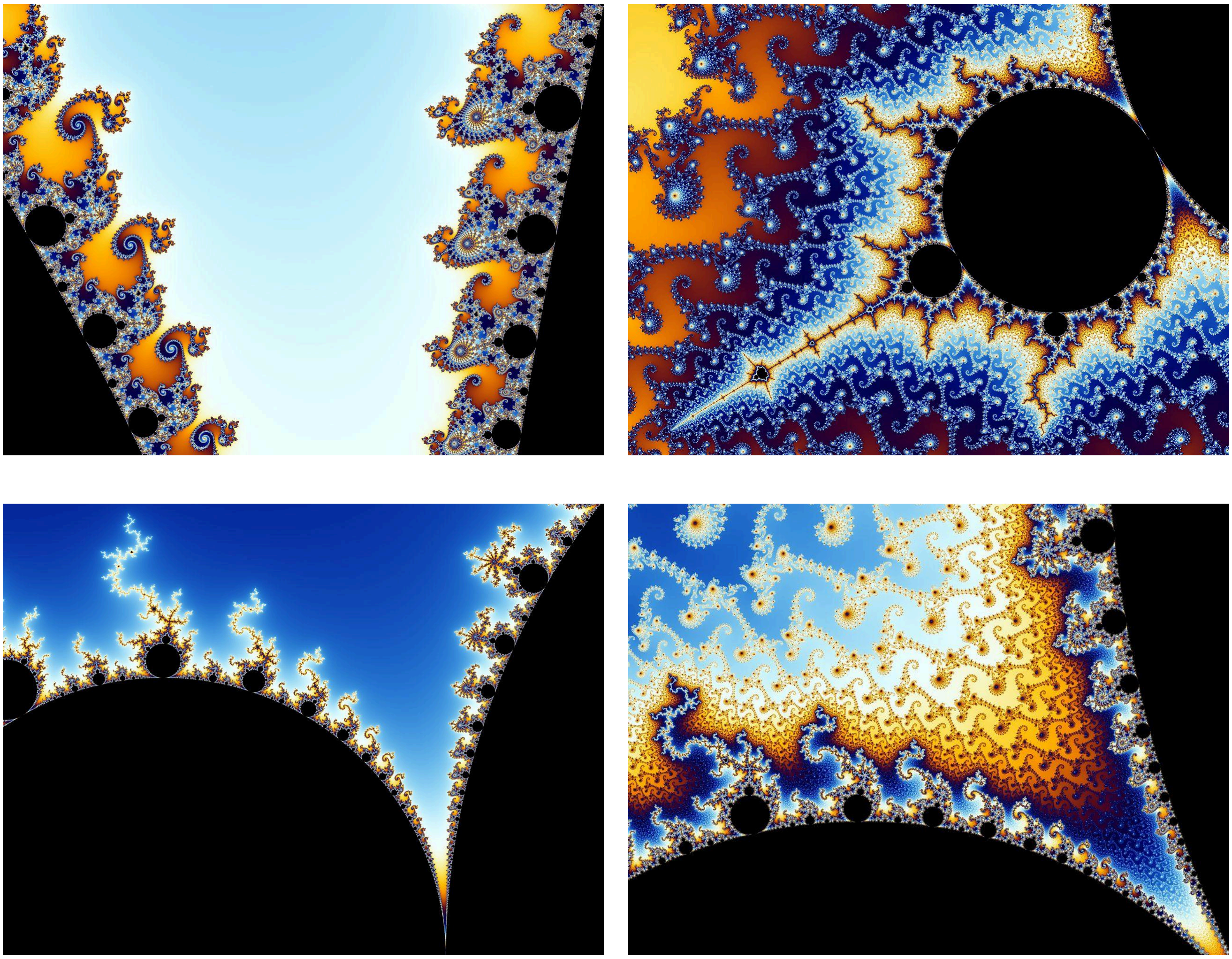
По този начин се представят формите на множеството на Манделброт, които показват как различните точки в комплексната равнина се разпределят спрямо множеството.



Фигура 5. Множество на Манделброт

7. Множество на Манделброт - графични примери за увеличение

Разглеждайки различни увеличения върху дадени стойности на c в множеството на Манделброт, откриваме впечатляващо разнообразие от геометрични структури. Всяка стойност на c отваря врати към свой собствен свят от фрактални детайли.



Фигура 6. Увеличение на множеството на Манделброт

8. Програмна реализация на множеството на Манделброт чрез HTML, CSS, JavaScript

```
<!DOCTYPE html>
<html>
<head>
  <style>
    body, html {
      margin: 0;
      padding: 0;
      font-family: 'Arial';
      text-align: center;
```

```
        }
        section.guide {
            position: absolute;
            top: 7em;
            right: 2em;
            width: 20em;
            color: white;
            background: rgba(0, 0, 0, 0.5);
            padding: 1em;
            box-sizing: border-box;
            border-radius: 2em;
        }
        footer, header {
            font-weight: bold;
            padding: 3px;
            background-color: #333;
            color: white;
        }
    </style>
    <title>Mandelbrot Set</title>
</head>
<body>
    <header>
        <h1>Множество на Манделброт</h1>
    </header>
    <canvas id="myCanvas" width="1488" height="596"></canvas>
    <section class="guide">
        <aside>
            <h2>Упътване за работа с програмата:</h2>
            С кликване на левия бутон на мишката върху фрактала на желаната
позиция, той се приближава. Когато задържите бутона ctrl и кликнете с левия
бутон на мишката, фракталът се отдалечава. С едновременно натискане на
```

```
бутони ctrl и R от клавиатурата, екранът се обновява и фракталът връща
началния си облик.
    </aside>
  </section>
  <footer>
    <p>Фрактали, летен семестър 2023/2024</p>
  </footer>
  <script>
      var canvas = document.getElementById("myCanvas");
      var context = canvas.getContext("2d");

      var canvasWidth = canvas.width;
      var canvasHeight = canvas.height;
      var imagedata = context.createImageData(canvasWidth, canvasHeight);

      var offsetx = -canvasWidth / 2;
      var offsety = -canvasHeight / 2;
      var panx = -100;
      var pany = 0;
      var zoom = 150;
      var palette = [];
      var maxiterations = 250;

      canvas.addEventListener("mousedown", onMouseDown);
      generatePalette();
      drawFractal();

      function generatePalette() {
        var roffset = 24, goffset = 16, boffset = 0;
        for (var i = 0; i < 256; i++) {
          palette[i] = { r: roffset, g: goffset, b: boffset };
          if (i < 64) roffset += 3;
```

```
      else if (i < 128) goffset += 3;
      else if (i < 192) boffset += 3;
    }
  }

  function drawFractal() {
    for (var y = 0; y < canvasHeight; y++) {
      for (var x = 0; x < canvasWidth; x++) {
        mandelbrot(x, y, maxiterations);
      }
    }
    context.putImageData(imagedata, 0, 0);
  }

  function mandelbrot(x, y, maxiterations) {
    var x0 = (x + offsetx + panx) / zoom;
    var y0 = (y + offsety + pany) / zoom;
    var a = 0, b = 0, rx = 0, ry = 0, iterations = 0;
    while (iterations < maxiterations && (rx * rx + ry * ry <= 4)) {
      rx = a * a - b * b + x0;
      ry = 2 * a * b + y0;
      a = rx;
      b = ry;
      iterations++;
    }
    var color = (iterations == maxiterations) ? { r: 0, g: 0, b: 0 } :
palette[Math.floor((iterations / (maxiterations - 1)) * 255)];
    var pixelindex = (y * canvasWidth + x) * 4;
    imagedata.data[pixelindex] = color.r;
    imagedata.data[pixelindex + 1] = color.g;
    imagedata.data[pixelindex + 2] = color.b;
    imagedata.data[pixelindex + 3] = 255;
```

```
        }

        function zoomFractal(x, y, factor, zoomin) {
          zoom = zoomin ? zoom * factor : zoom / factor;
          panx = zoomin ? factor * (x + offsetx + panx) : (x + offsetx + panx) / factor;
          pany = zoomin ? factor * (y + offsety + pany) : (y + offsety + pany) / factor;
        }

        function onMouseDown(e) {
          var pos = getMousePos(canvas, e);
          var zoomin = !e.ctrlKey;
          var zoomfactor = e.shiftKey ? 1 : 2;
          zoomFractal(pos.x, pos.y, zoomfactor, zoomin);
          drawFractal();
        }

        function getMousePos(canvas, e) {
          var rect = canvas.getBoundingClientRect();
          return {
                x:Math.round((e.clientX - rect.left) / (rect.right - rect.left) *
canvas.width),
                y:Math.round((e.clientY - rect.top) / (rect.bottom - rect.top) *
canvas.height)
          };
        }
    </script>
</body>
</html>
```