TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Programmeerimise süvendatud algkursus ITI0140

2014

Teema

- Pilditöötlus (ingl *image processing*)
  - Fraktalid (ka fraktaalid, ingl *fractal*)
    - Julia hulgad
    - Mandelbrot'i hulk

Ääremärkus

Matemaatika filmiõhtud

Teisipäeviti, kell 19:00, Loodusteaduste majas (SCI-109)

Järgmine kord näidatakse dokumentaali

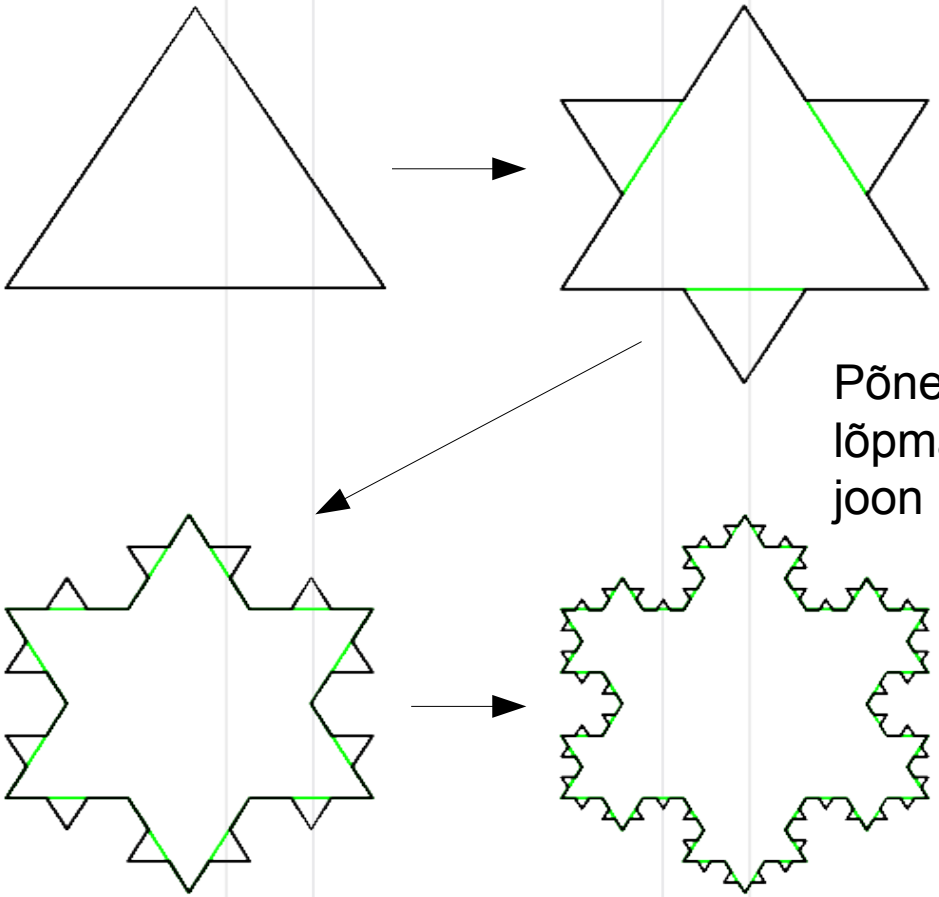
“Fractals: Hunting the Hidden Dimension”

What do movie special effects, the stock market, heart attacks, and the rings of Saturn have in common...

Benoit Mandelbrot (1924-2010)

- Poolas sündinud prantsuse-ameerika matemaatik
- Fraktalite avastaja
- Arvutite areng on otseselt seotud fraktalite avastamisega – Mandelbrot kasutas arvutigraafikat fraktaalsete geomeetriliste kujundite loomiseks ja kuvamiseks
- Ta avastas Mandelbrot'i hulga 1979. aastal IBM-is töötades
- Fraktalid aitavad seletada mitmeid nö karedaid või kaootilisi asju.
  - pilved
  - kaldajoon
  - galaktikate jaotus (?) (vt *fractal cosmology*)

Koch'i (H. von Koch) lumehelbeke

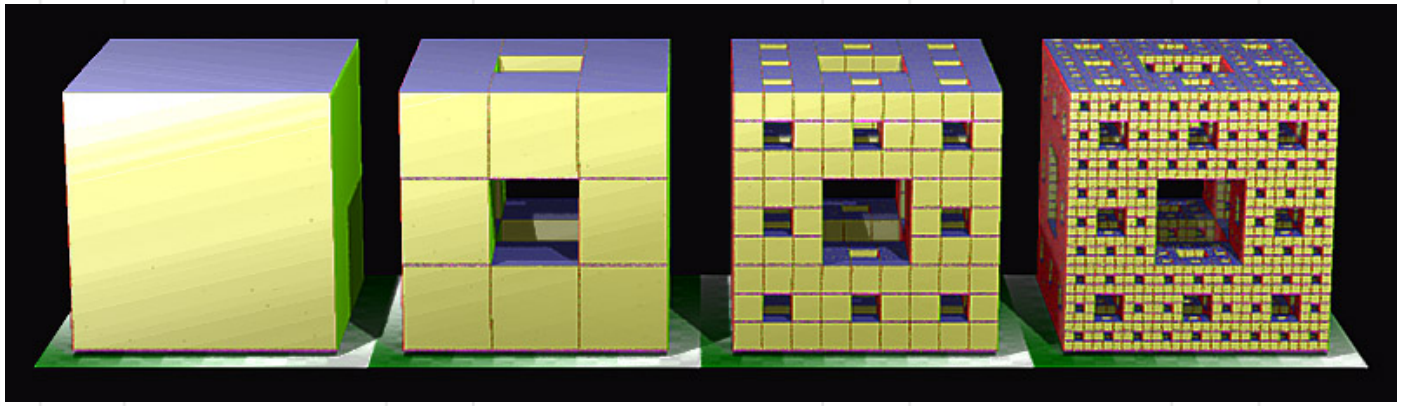


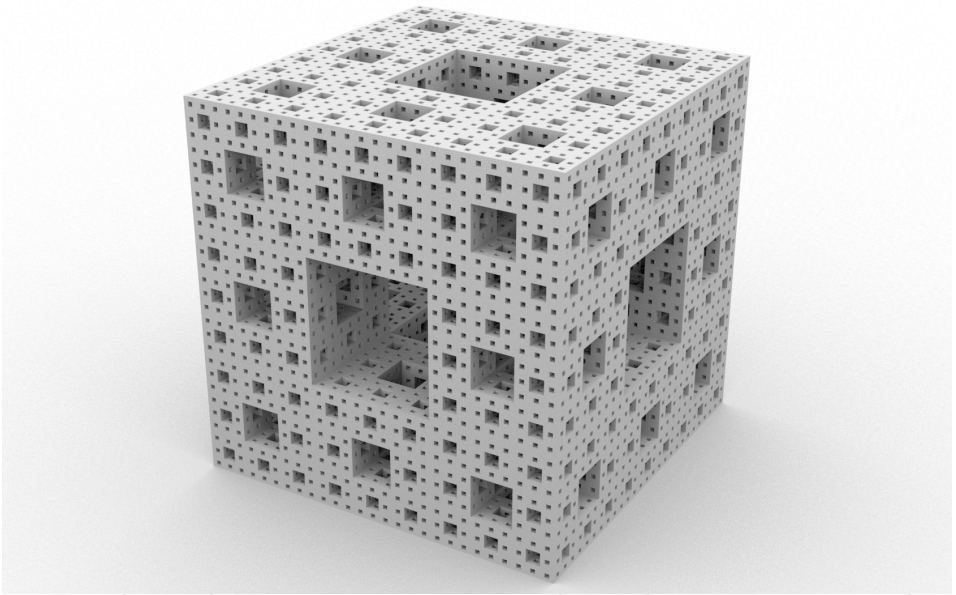
Põnev omadus: lõpmatu pikkusega joon (kui n->inf)

Külgede arv $N_n = N_{n-1} * 4 = 3 * 4^n$

Mengeri käsn

1 → 2 → 3 → 4

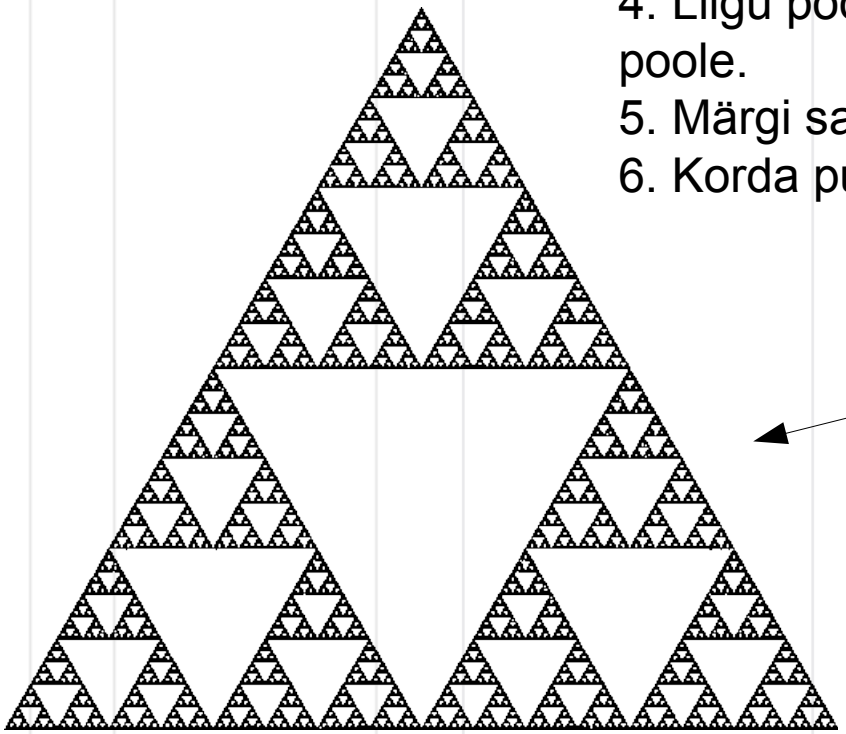




Sierpinski kolmnurk

Kaose mängu meetod
(ingl *chaos game*)

1. Vali suvaliselt 3 punkti (kolmnurk)
2. Vali suvaliselt üks punkt, mis jääb kolmnurga sisse.
3. Vali suvaliselt üks kolmnurga tipp.
4. Liigu pool teed punktist valitud tipu poole.
5. Märgi saadud punkt!
6. Korda punktist 3.



Mida enam iteratsioone, seda "teravamaks" pilt muutub.

Sierpinski triangle with Pillow

```
from PIL import Image
import random


def main():
    size = (1000, 1000)
    img = Image.new("RGB", size, 0)
    triangle = [(size[0]//2, 50), (50, size[1]-50), (size[0]-50, size[1]-
50)]
    point = (300, 600)
    for _ in range(1000000):
        random_vertex = random.randint(0, 2)
        point = (point[0] + ((triangle[random_vertex][0] - point[0]) // 2),
                 point[1] + ((triangle[random_vertex][1] - point[1]) // 2))
        img.putpixel(point, (255, 0, 0))
    img.save("sierpinski.png")


if __name__ == "__main__":
    main()
```

Itereeritud funktsioonisüsteem ehk huvitav mis siit võiks tulla?

```
from PIL import Image
import random

def main():
    size = (1000, 1000)
    img = Image.new("RGB", size)
    point = (0, 0)
    for _ in range(10000000):
        img.putpixel((int(point[0]*100+500), int(size[1] - 1 - point[1]*100)), (0, 255, 0))
        r = random.randint(0, 3)
        if r == 0:
            point = (0, 0.16*point[1])
        elif r == 1:
            point = (0.85*point[0] + 0.04*point[1], -0.04*point[0] + 0.85*point[1] + 1.6)
        elif r == 2:
            point = (0.2*point[0] - 0.26*point[1], 0.23*point[0] + 0.22*point[1] + 1.6)
        else:
            point = (-0.15*point[0] + 0.28*point[1], 0.26*point[0] + 0.25*point[1] + 0.44)
    img.save("barnsley.png")


if __name__ == "__main__":
    main()
```

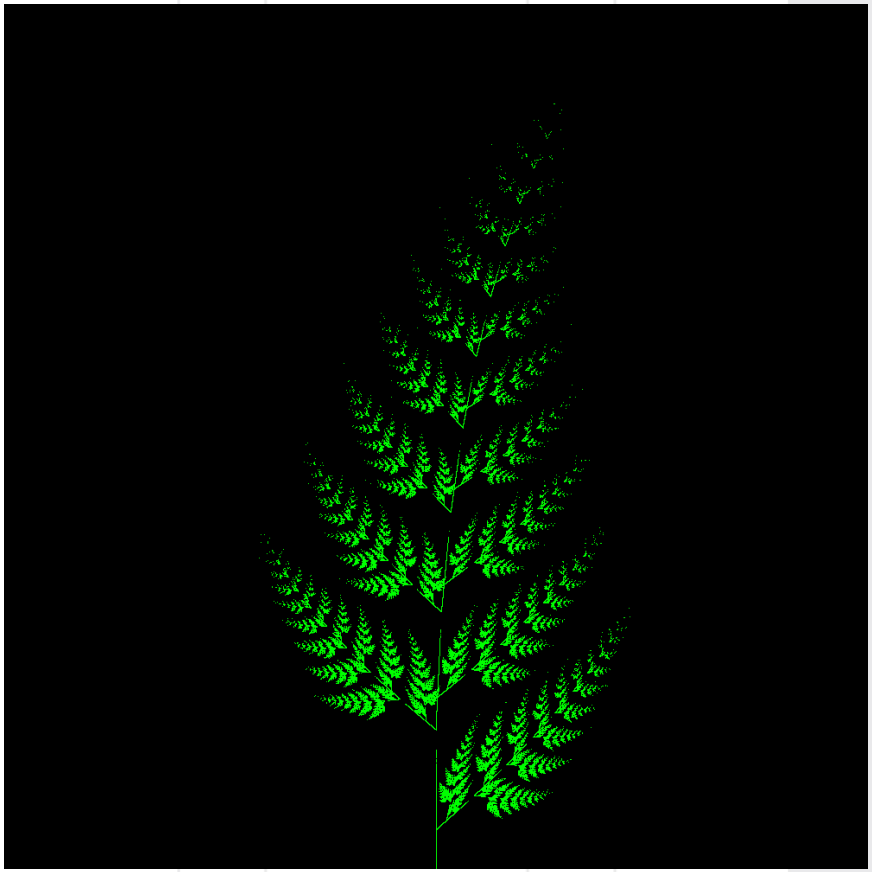
Itereeritud funktsioonisüsteem ehk huvitav mis siit võiks tulla?

```
from PIL import Image
import random

def main():
    size = (1000, 1000)
    img = Image.new("RGB", size)
    point = (0, 0)
    for _ in range(10000000):
        img.putpixel((int(point[0]*100+500), int(size[1] - 1 - point[1]*100)), (0, 255, 0))
        r = random.randint(0, 3)
        if r == 0:
            point = (0, 0.16*point[1])
        elif r == 1:
            point = (0.85*point[0] + 0.04*point[1], -0.04*point[0] + 0.85*point[1] + 1.6)
        elif r == 2:
            point = (0.2*point[0] - 0.26*point[1], 0.23*point[0] + 0.22*point[1] + 1.6)
        else:
            point = (-0.15*point[0] + 0.28*point[1], 0.26*point[0] + 0.25*point[1] + 0.44)
    img.save("barnsley.png")


if __name__ == "__main__":
    main()
```

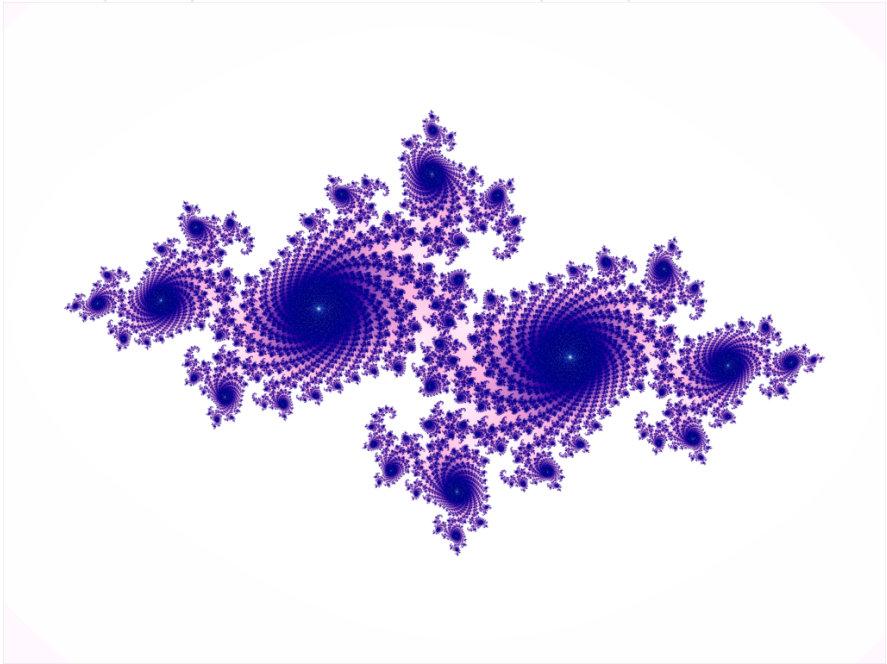
Barnsley sõnajalg

"Lõplik" kuju
(VisSim plot)

10 miljonit iteratsiooni



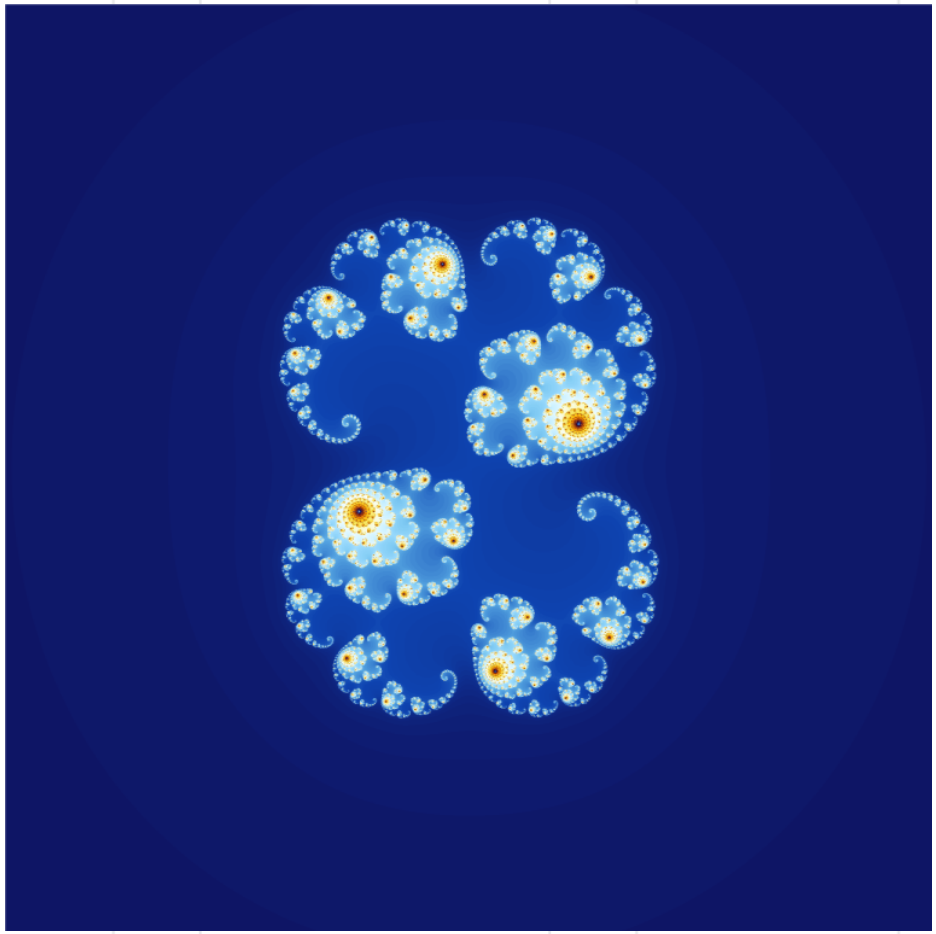
Julia hulgad



$$z_n = z_{n-1}^2 + c$$

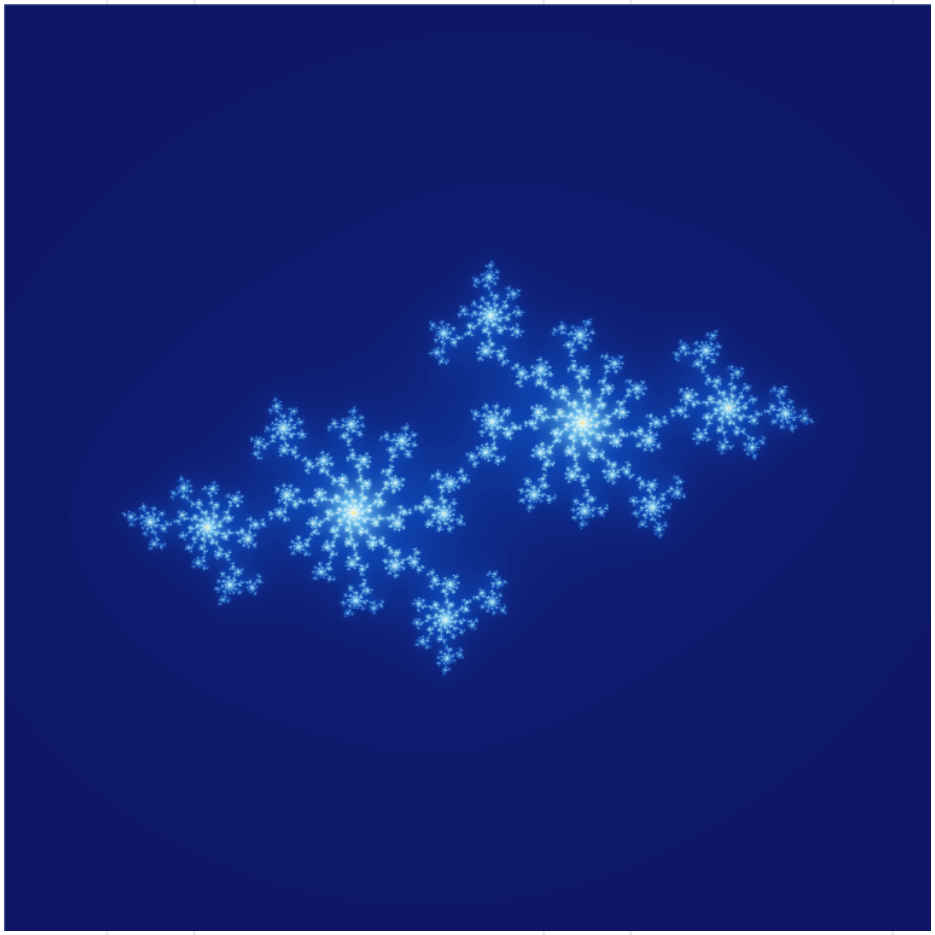
Kompleksarvuline parameeter, mis määrab fraktali "kuju"

Julia hulgad



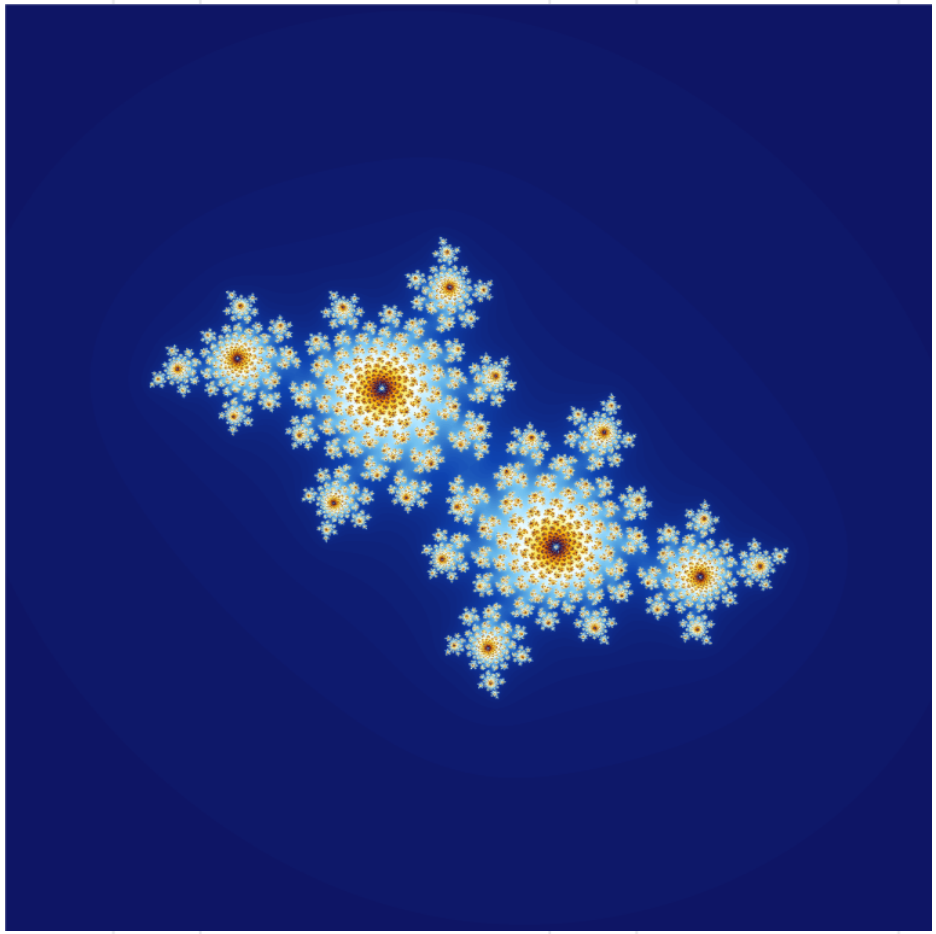
c = 0.285 + 0.01i

Julia hulgad



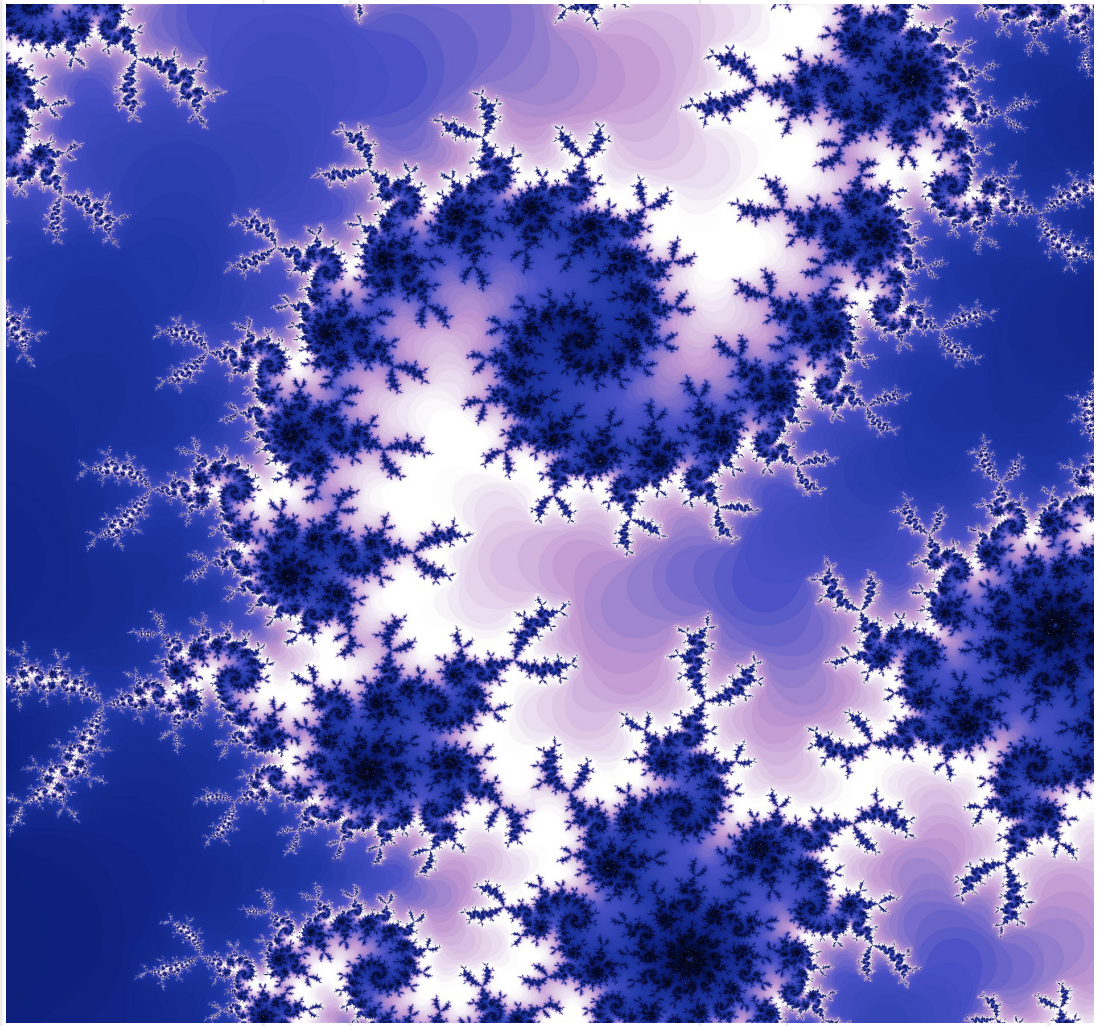
c = -0.70176 - 0.3842i

Julia hulgad

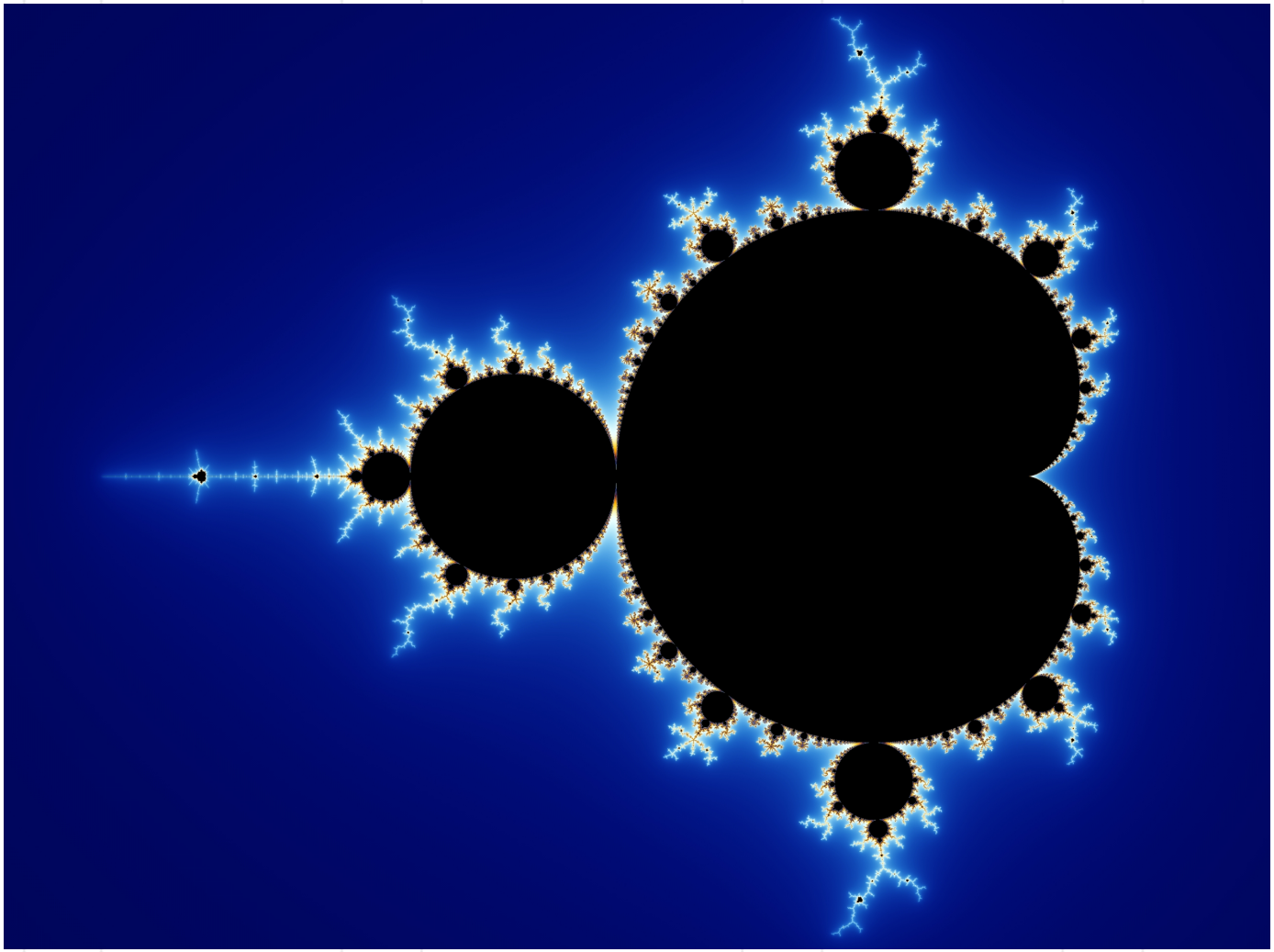


c = -0.4 + 0.6i

Julia hulgad

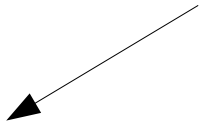


Mandelbrot'i hulk: "Piparkoogimehike"



Kompleksarvud Pythonis

i asemel j



```
import math

c = 1.23 + 4.56j
norm_c = math.sqrt(c.real**2 + c.imag**2)
print(norm_c)
```

```
>>> 4.722975756871931
```

On olemas ka cmath nimeline pakett

(Mathematical functions for complex numbers)
https://docs.python.org/3/library/cmath.html

Ülesanne

1) Genereeri ja joonista ise **Mandelbrot'i hulga graafik**.

2) Loo **Julia hulkade graafikute** genereerimise funktsioon ***julia(c)***, millele saab anda kaasa argumendina kompleksarvulise parameetri c. Funktsioon peaks tagastama kahemõõtmelise järjendi graafiku pildi pikslivääртustega.

Joonistamiseks kasuta eelmises tunnis tutvustatud Pillow või PythonMagick paketti. Joonistamise jaoks vajaliku matemaatika sammhaaval kirjelduse leiad:

http://www.wikihow.com/Plot-the-Mandelbrot-Set-By-Hand