Przykładowe programy w Pythonie

Zuzanna Szester
Magdalena Syrek
Natalia Donocik

Wydział Fizyki, Matematyki i Informatyki
Politechnika Krakowska im. T.Kościuszki

19 czerwca 2018

Spis treści

Żuk Mandelbrota - fraktale

Zbiór Mandelbrota

Jest to podzbiór płaszczyzny zespolonej, którego brzeg jest jednym z najbardziej znanych fraktali ("najsłynniejszy obiekt współczesnej matematyki"). Nazwa tego obiektu została wprowadzona dla uchonorowania jego odkrywcy, francuskiego matematyka Benoit Mandelbrota.

fraktal

W znaczeniu potocznym oznacza zwykle obiekt samo-podobny. Ze względu na olbrzymią różnorodność przykładów matematycy obecnie unikają podawania ścisłej definicji i proponują określony fraktal jak zbiór, który posiada nietrywialną strukturą w każdej skali, a jego wymiar Hausdorffa jest większy niż jego wymiar topologiczny. Zwykle ma naturalny "poszarpany, kłębiasty itp." wygląd.

Historia odkrycia żuka

W 1982 Mandelbrot spopularyzował geometrię fraktalną publikując swoje dzieło The Fractal Geometry of Nature. Uświadomiło to społeczeństwu, że fraktale są "wśród nas" i mogą przybierać kształty podobne do tych naturalnych. Oprócz tych rozważań podał też bardzo prostą metodę na utworzenie fraktalu (zbioru) Mandelbrota.

Konstrukcja

Zbiór tworzący punkty p dla których ciąg opisany równaniem rekurencyjnym:

$$z_0 = 0$$

$$z_{n+1} = z_n^2 + p$$

Nie dąży do nieskończoności.

$$\lim_{n \to \infty} z_n \neq \infty$$

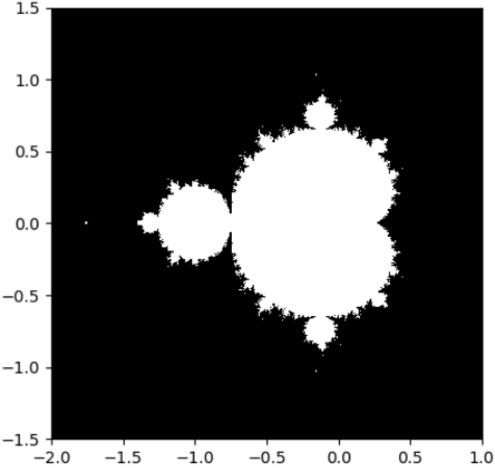
Podsumowując jednym zdaniem:

$$M = p \in \mathbb{C} \bigvee_{n \in \mathbb{N}} |z_n| < 2$$

Kod w Pythonie - fraktal

```
import numpy as np
import matplotlib.pyplot as plt
from numpy import newaxis
def funkcja_mandelbrot(N_max, plaszczyzna, nx,ny):
# Siatka wartosci c
x = np.linspace(-2, 1, nx)
y = np.linspace(-1.5, 1.5, ny)
c = x[:,newaxis] + 1j*y[newaxis,:]
# iteracja z = c for j in range(N_max): z = z**2 + c
mandelbrot_set = (abs(z) < plaszczyzna)
return mandelbrot_set
mandelbrot_set = funkcja_mandelbrot(50, 50., 601, 401)
plt.imshow(mandelbrot_set.T, extent=[-2, 1, -1.5, 1.5])
plt.gray()
plt.show()
```

Wykres



Pole wektorowe

Pole wektorowe jest to funkcja, która każdemu punktowi przestrzeni przyporządkowuje pewną wielkość wektorową. Formalnie definicja pola wektorowego odwołuje się do teorii miary i teorii przestrzeni Hilberta.

Przykłady pól wektorowych:

- pole grawitacyjne
- pole elektryczne
- pole magnetyczne
- pole prędkości i potencjał zespolony przepływu

Pole wektorowe

Pole wektorowe jest to funkcja, która każdemu punktowi przestrzeni przyporządkowuje pewną wielkość wektorową. Formalnie definicja pola wektorowego odwołuje się do teorii miary i teorii przestrzeni Hilberta.

Przykłady pól wektorowych:

- pole grawitacyjne
- pole elektryczne
- pole magnetyczne
- pole prędkości i potencjał zespolony przepływu

Pole wektorowe

Pole wektorowe jest to funkcja, która każdemu punktowi przestrzeni przyporządkowuje pewną wielkość wektorową. Formalnie definicja pola wektorowego odwołuje się do teorii miary i teorii przestrzeni Hilberta.
Przykłady pól wektorowych:

- pole grawitacyjne
- pole elektryczne
- pole magnetyczne
- pole prędkości i potencjał zespolony przepływu

Pole wektorowe

Pole wektorowe jest to funkcja, która każdemu punktowi przestrzeni przyporządkowuje pewną wielkość wektorową. Formalnie definicja pola wektorowego odwołuje się do teorii miary i teorii przestrzeni Hilberta.

Przykłady pól wektorowych:

- pole grawitacyjne
- pole elektryczne
- pole magnetyczne
- pole prędkości i potencjał zespolony przepływu

Kołczan

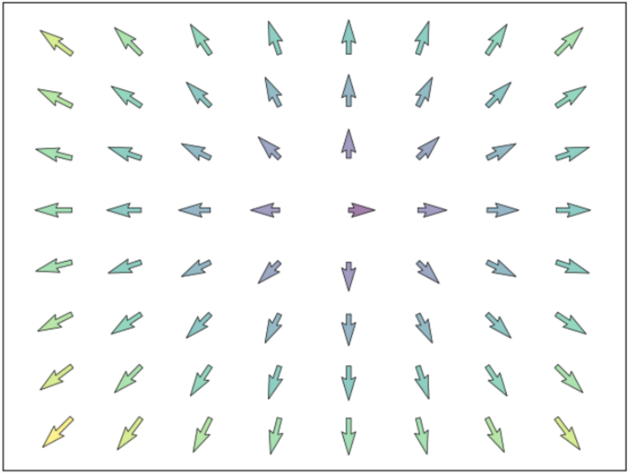
Kołczan (Quiver)

W matematyce kołczan jest skierowanym wykresem, w którym dozwolone są pętle i wiele strzałek między dwoma wierzchołkami, tj. multigraf. Jest on powszechnie stosowany w teorii reprezentacji: reprezentacja V kołczanu przypisuje przestrzeń wektorową V(x) do każdego wierzchołka x kołczanu i liniową mapę V(a) do każdej strzałki a. W teorii kategorii kołczan można rozumieć jako podstawową strukturę kategorii, ale bez morfizmów i kompozycji tożsamości.

Kod w Pythonie - pole wektorowe

```
import numpy as np
import matplotlib.pyplot as plt
n = 8
X, Y = np.mgrid[0:n, 0:n]
T = np.arctan2(Y - n / 2., X - n/2.)
R = 10 + np.sqrt((Y - n / 2.0) ** 2 + (X - n / 2.0) ** 2)
U, V = R * np.cos(T), R * np.sin(T)
plt.axes([0.025, 0.025, 0.95, 0.95])
plt.quiver(X, Y, U, V, R, alpha=.5)
plt.quiver(X, Y, U, V, edgecolor='k', facecolor='None',
linewidth=.5)
plt.xlim(-1, n)
plt.xticks(())
plt.ylim(-1, n)
```

Wykres



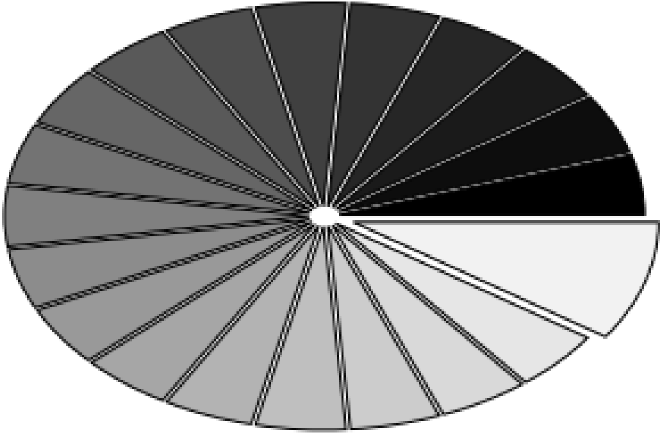
Wykres kołowy

Wykresy kołowe najczęściej wykorzystywane są w celu graficznego zaprezentowania charakterystyki badanych osób. Wykorzystywany jest dla zobrazowania częstości / frekwencji wariantów danej zmiennej. Wykres najczęściej tworzony jest dla jednej zmiennej, a poszczególne elementy koła obrazują procent występowania danego wariantu zmiennej.W prezentowanym przykładzie wykres kołowy składa się z 20 wycinków w odcieniach szarości.

Kod w Pythonie - wykres kołowy

```
import numpy as np
import matplotlib.pyplot as plt
n=20
Z=np.ones(n)
Z[-1]*=2
plt.axes([0.025,0.025,0.95,0.95])
plt.pie(Z, explode=Z*.05, colors = [' procent f' procent (i/float(n)
for i in range(n)])
plt.axis('equal')
plt.xticks(())
plt.yticka(())
plt.show()
```

Okrąg z wycinków w odcieniach szarości



Wykres Shmita

Jednym z zastosowań wykresu kołowego jest przedstawienie impedancji we współrzędnych biegunowych. Każdy punkt wykresu odpowiada określonej liczbie zespolonej interpretowanej fizycznie jako impedancja. Wykres Shmita, to diagram kołowy powstały w wyniku homograficznego przekształcenia siatki prostych równoległych do osi rzeczywistej i równoległych do osi urojonej na paszczyźnie zespolonej Z w układ wzajemnie ortogonalnych okręgów.

Wykres Shmita