Ćwiczenia VIII
Generowanie i charakteryzownie fraktali

Jakub Tworzydło

29 i 30/11/2022 Pasteura, Warszawa

Plan

1. Generowanie obrazków
2. Obliczanie wymiaru pudełkowego
3. Wskazówki

Plan

1. Generowanie obrazków
2. Obliczanie wymiaru pudełkowego
3. Wskazówki

Plan

1. Generowanie obrazków
2. Obliczanie wymiaru pudełkowego
3. Wskazówki

Zadanie 1

Algorytm IFS omawiany na wykładzie daje prosty sposób generowania fraktali. Zacznijmy iterację z punktu $(x = 0, y = 0)$, który poddajemy transformacji liniowej o postaci

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} m_0 x + m_1 y \\ m_2 x + m_3 y \end{pmatrix} + \begin{pmatrix} m_4 \\ m_5 \end{pmatrix}.$$

W każdym kolejnym kroku wybieramy jeden z n zespołów parametrów $\{m\}$ z ustalonym prawdopodobieństwem p i obliczamy nowy punkt. Na wykres nanosimy wszystkie punkty z jednego, długiego ciągu iteracji.

Parametry

Naszym zadaniem jest otrzymanie w ten sposób obrazka z trójkątem Sierpińskiego ($n = 3$) oraz z paprotką Barnsley'a ($n = 4$).
Transformacje i odpowiednie prawdopodobieństwa dla trójkąta Sierpińskiego wynoszą

```
p = 1./3  m = [0.5, 0, 0, 0.5, 0.25, sqrt(3.)/4]
p = 1./3  m = [0.5, 0, 0, 0.5, 0.0, 0]
p = 1./3  m = [0.5, 0, 0, 0.5, 0.5, 0]
```

Przekształcenia dla paprotki podane są poniżej

```
p = 0.02 m = [0.001,  0.0,  0.0, 0.16, 0.0, 0.0]
p = 0.09 m = [-0.15, 0.28, 0.26, 0.24, 0.0, 0.44]
p = 0.10 m = [  0.2,-0.26, 0.23, 0.22, 0.0, 1.6]
p = 0.79 m = [ 0.85, 0.04,-0.04, 0.85, 0.0, 1.6]
```

Zadanie 2

Fraktal otrzymany w poprzednim ćwiczeniu dzielimy siatką o rozmiarach $2^r \times 2^r$, przy czym $r = 1, 2, \ldots$. Obliczamy liczbę oczek siatki (pudełek) N_r, które zawierają przynajmniej jeden punkt badanego obiektu (wskazówka na tablicy).

Należy wykonać wykres $\log(N_r)$ w funkcji r **wspólnie** dla dość długiego przebiegu M iteracji oraz dla krótszego ciągu $M/4$ iteracji. Czy otrzymana zależność jest liniowa (jakościowo) w pewnym zakresie? Jak wyjaśnisz wypłaszczanie wykresu dla dużych r?

Zadanie dodatkowe (do zaliczenia całe)

Obliczyć wymiar fraktalny D_f kolejnych obiektów wykonując dopasowanie prostej

$$\log N_r = D_f \log(2)r + \text{const.}$$

Należy oszacować błędy dopasowania oraz ustalić rozsądny zakres, w którym zależność jest liniowa w skali log-log. Przedstawić dopasowania na wykresach.

Do testowania metody użyć trójkąta Sierpińskiego posiadającego znany wymiar $D_f \approx 1.585$.

Zadanie dodatkowe (c.d.)

Wykreślić i podać wynik dopasowania dla Smoka fraktalnego

```
 m = [0.824074, 0.281482, −0.212346,  0.864198, −1.882290,
                                  −0.110607], p = 0.787473
 m = [0.088272, 0.520988, −0.463889, −0.377778,  0.785360,
                                   8.095795], p = 0.212527
```

oraz dla Krzywej C Levy'ego

```
m = [0.5, −0.5, 0.5, 0.5, 0.0, 0.0], p = 0.5
m = [0.5, 0.5, −0.5, 0.5, 0.5, 0.5], p = 0.5
```

Wskazówki

Ad.1 Ciekawy efekt powstaje przy zmianie koloru tła i naniesieniu kolorowych punktów

```
plt.rcParams['axes.facecolor'] = 'black'

plt.scatter(xpoints,ypoints, s=1, marker="o", lw=0,
                                        c='green')
```

Ad.extra Proste dopasowanie liniowe można wykonać przy pomocy procedury `polyfit` z modułu `scipy` (przykład w skrypcie `dopasowanie.py`).

Przykład fitowania w Pythonie prawa potęgowego z obliczeniem błędów można znaleźć w ostanim przykładzie ("Fitting a power law") z http://scipy-cookbook.readthedocs.org/items/FittingData.html.