Ćwiczenia VI
Generowanie i charakteryzownie fraktali

Jakub Tworzydło

Katedra Teorii Materii Skondensowanej
Instytut Fizyki Teoretycznej
telefon: (022)5532-314, pokój 344

7/11/2012 Hoża, 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 $(x = m_4, y = m_5)$, 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 kroku wybieramy jeden z zespołów parametrów $\{m\}$ z ustalonym prawdopodobieństwem p i obliczamy nowy punkt. Na wyres nanosimy wszystkie punkty z jednego, długiego ciągu iteracji.

Parametry

Naszym zadaniem jest otrzymanie w ten sposób obrazka trójkąta Sierpińskiego oraz paprotki Barnsley'a.

Transformacje i odpowiednie prawdopodobieństwa dla trójkąta Sierińskiego wynoszą

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

Przekształcenia dla paprotki podane są poniżej

```
m = [ 0.85, 0.04,-0.04, 0.85, 0.0, 1.6]  p = 0.73
m = [  0.2,-0.26, 0.23, 0.22, 0.0, 1.6]  p = 0.13
m = [-0.15, 0.28, 0.26, 0.24, 0.0, 0.44] p = 0.11
m = [  0.0,  0.0,  0.0, 0.16, 0.0, 0.0]  p = 0.03
```

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.

Należy wykonać wykres $\log(N_r)$ w funkcji $r \log(2)$ i dopasować linię prostą wybierając "rozsądny" zakres. Czy możemy na podstawie otrzymanych danych oszacować wymiar fraktalny dla trójkąta Sierpińskiego i paprotki?

Zadanie dodatkowe

Można w podobny sposób wygenerować i zbadać wymiar fraktalny:

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
```

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
```

... a nawet atraktora Duffinga

Dopasowanie prostej do danych

Poniżej podaję prosty przykład dopasowania funkcji liniowej w Pythonie.

```
from scipy import linspace, polyfit, randn
from matplotlib.pyplot import plot, title, show , legend

# Sample data creation with n=50 points
n=50
x=linspace(-5,5,n)
# known parameters
a=0.8; b=-4
# we add some noise
ydat=a*x+b+randn(n)

#Linear regressison with polyfit
(ar,br)=polyfit(x,ydat,1)
yfit= ar*x+br
```

Dopasowanie prostej c.d.

```
# print results
print('Linear regression using polyfit')
print('parameters: a=%.2f b=%.2f \t \
       regression: a=%.2f b=%.2f' % (a,b,ar,br))

# matplotlib ploting
title('Linear Fit Example')
plot(x,ydat,'k.')
plot(x,yfit,'r-')
legend(['noisy data', 'fit: a=%.2f b=%.2f' % (ar,br)],\
                                              loc='best')
show()
```

Więcej przykładów fitowania w Pythonie można znaleźć w:
http://www.scipy.org/Cookbook/FittingData