Numerické výpočty, simulace a vizualizace
(Ukázky)

Jan Kybic

http://cmp.felk.cvut.cz/~kybic
kybic@fel.cvut.cz

2016–2017

Lineární interpolace metodou nejmenších čtverců

Mějme N naměřených bodů (x_i, y_i)

```
import numpy as np     # http://www.numpy.org/

x=np.array([10,20,30,40,50])
y=np.array([17.1,21.9,24.5,29.5,35.8])
```

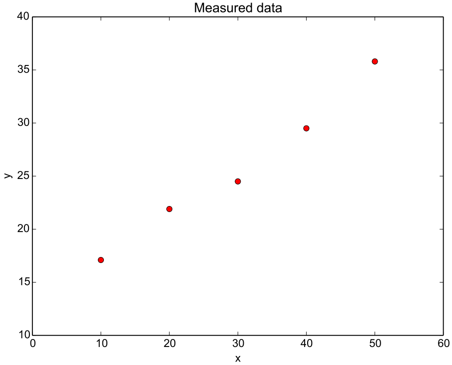
Například teplota (y) v čase (x)

Jaká je hodnota mezi naměřenými body?

Úkol: Body vykreslete a proložte přímkou.

Soubor `linfit.py`.

Vykreslení
knihovna Matplotlib




```
import matplotlib.pyplot as plt

plt.plot(x,y,'ro')
plt.xlabel('x')
plt.ylabel('y')
plt.title('Measured data')
plt.axis((0,60,10,40))

plt.savefig('namerena_data.pdf')
plt.show()
```

Interpolace metodou nejmenších čtverců

Minimalizujme chybu

$$J = \sum_{i=1}^{n} \bigl(f(x_i; a, b) - y_i\bigr)^2, \qquad f(x; a, b) = ax + b$$

Pro optimální hodnotu parametrů platí:

$$0 = \frac{\partial J}{\partial a} = 2 \sum_i x_i (a x_i + b - y_i)$$

$$0 = \frac{\partial J}{\partial b} = 2 \sum_i (a x_i + b - y_i)$$

To je lineární soustava rovnic

$$a \sum_i x_i^2 + b \sum_i x_i = \sum_i x_i y_i$$

$$a \sum_i x_i + b \sum_i 1 = \sum_i y_i$$

Interpolace metodou nejmenších čtverců

To je lineární soustava rovnic

$$
\begin{aligned}
a \sum_i x_i^2 + b \sum_i x_i &= \sum_i x_i y_i \\
a \sum_i x_i + b \sum_i 1 &= \sum_i y_i
\end{aligned}
$$

Pomocí matic píšeme

$$
\underbrace{\begin{bmatrix} \sum_i x_i^2 & \sum_i x_i \\ \sum_i x_i & \sum_i 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} a \\ b \end{bmatrix}}_{\theta} = \underbrace{\begin{bmatrix} \sum_i x_i y_i \\ \sum_i y_i \end{bmatrix}}_{\mathbf{r}}
$$

$$
A\,\theta = \mathbf{r}
$$

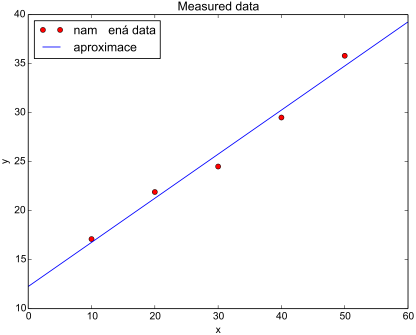
Fitování

$$\underbrace{\begin{bmatrix} \sum_i x_i^2 & \sum_i x_i \\ \sum_i x_i & \sum_i 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} a \\ b \end{bmatrix}}_{\theta} = \underbrace{\begin{bmatrix} \sum_i x_i y_i \\ \sum_i y_i \end{bmatrix}}_{\mathbf{r}}$$

```
n=len(x)
sx=np.sum(x)
sy=np.sum(y)
sxy=np.dot(x,y)
sxx=np.dot(x,x)
A=np.array([[sxx,sx],[sx,n]])
r=np.array([sxy,sy])
theta=np.linalg.solve(A,r) # obsahuje a,b
print("theta=",theta)
```

```
theta= [  0.45  12.26]
```

Interpolace a vykreslení




```
t=np.arange(0,60,0.1)
z=t*theta[0]+theta[1]
plt.plot(t,z,'b-')
plt.legend(['naměřená data','aproximace'],loc='upper left')
#plt.savefig('figs/aproximace.pdf')
#plt.show()
```

Soubor `linfit.py`.

Simulace házení

Házíme předmět pod úhlem φ z bodu $(0,0)$ rychlostí v_0. Najděte trajektorii $x(t), y(t)$.

Spojité rovnice

$$\begin{aligned} x''(t) &= 0 \\ y''(t) &= -g \\ x(0) &= 0 \\ y(0) &= 0 \\ x'(0) &= v_0 \cos\varphi \\ y'(0) &= v_0 \sin\varphi \end{aligned}$$

Eulerova diskretizace:

$$\begin{aligned} x(t+\Delta t) &= x(t) + \Delta t\, v_x(t) \\ y(t+\Delta t) &= y(t) + \Delta t\, v_y(t) \\ v_x(t+\Delta t) &= v_x(t) \\ v_y(t+\Delta t) &= v_y - \Delta t\, g \end{aligned}$$

Simulace házení — implementace

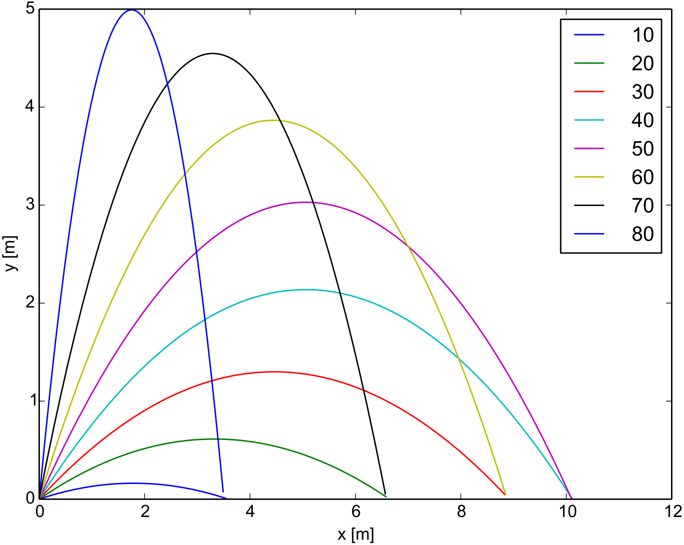
```
def hod(uhel,rychlost,dt=0.01,tmax=1000):
  xs=[]
  ys=[]
  x=0. ; y=0. ; t=0.
  anglerad=uhel*math.pi/180.
  vx=math.cos(anglerad)*rychlost
  vy=math.sin(anglerad)*rychlost
  while y>=0. and t<tmax:
    xs+=[x]
    ys+=[y]
    t+=dt
    x+=dt*vx
    y+=dt*vy
    vy-=9.81*dt
  return xs,ys
```

Soubor `hazeni.py`.

Simulace házení — implementace

```
def test_hod():
  rychlost=10.
  plt.figure(1)
  plt.clf()
  for uhel in [10,20,30,40,50,60,70,80]:
    xs,ys=hod(uhel, rychlost)
    plt.plot(xs,ys,label='%3.0f' % uhel)
  plt.xlabel('x [m]')
  plt.ylabel('y [m]')
  plt.legend(loc='top right')
  #plt.savefig('figs/hody.pdf')
  #plt.show()
```

Simulace házení — výsledky



Simulace gravitačního pohybu
n-body simulation

Rovnice gravitačního pohybu

$$\mathbf{F}_{ij} = G\frac{m_i m_j}{r^3}\mathbf{r}, \quad G \approx 6.67408 \cdot 10^{-11}\,\mathrm{m}^3\,\mathrm{kg}^{-1}\,\mathrm{s}^{-2}$$

$$\text{where} \quad \mathbf{r} = \mathbf{x}_j - \mathbf{x}_i$$

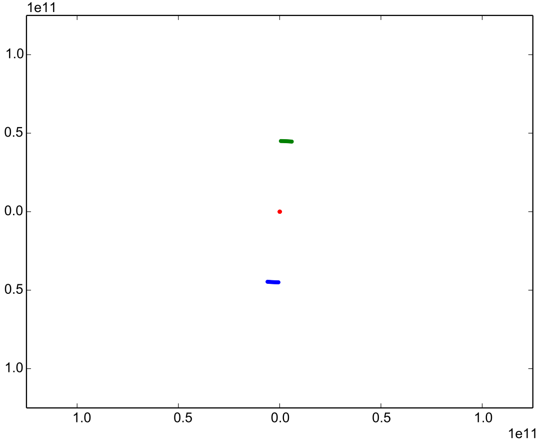
$$\mathbf{x}_i'' = \frac{1}{m_i}\sum_j \mathbf{F}_{ij}$$

Eulerova diskretizace

Soubory `body.py`, `universe.py`.

http://introcs.cs.princeton.edu/python/code/

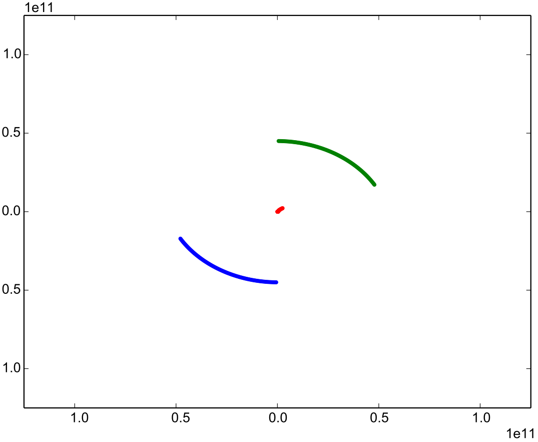
Simulace 3 těles



$n = 10$

`python3 universe.py 3body.txt 20000` n

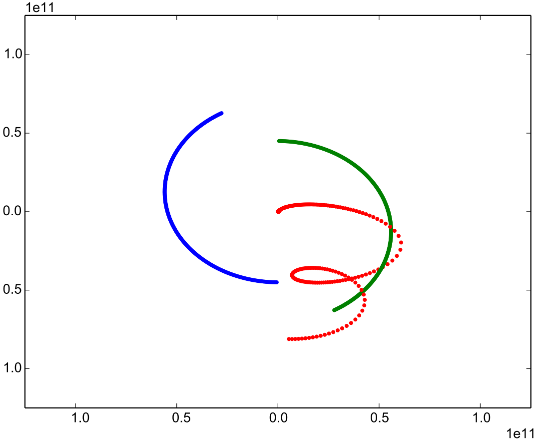
Simulace 3 těles



$n = 100$

```
python3 universe.py 3body.txt 20000 n
```

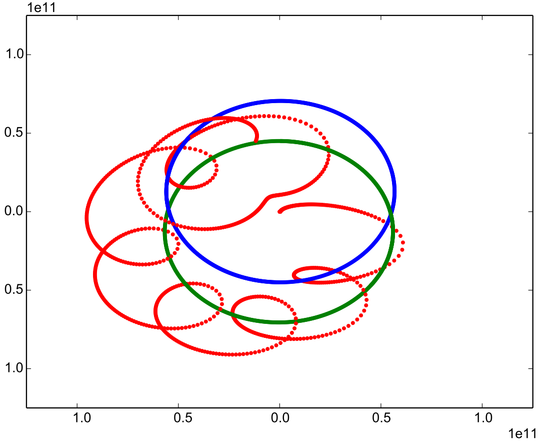
Simulace 3 těles



$n = 300$

`python3 universe.py 3body.txt 20000` n

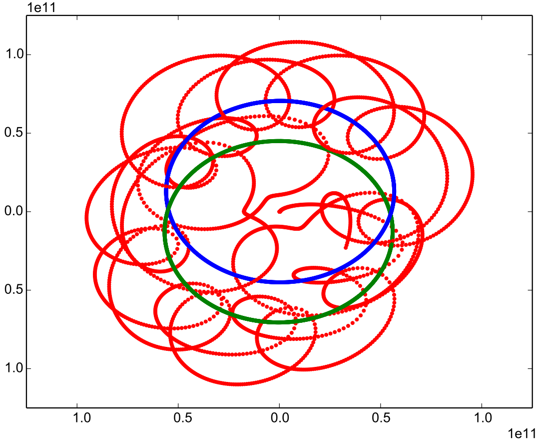
Simulace 3 těles



$n = 1000$

```
python3 universe.py 3body.txt 20000 n
```

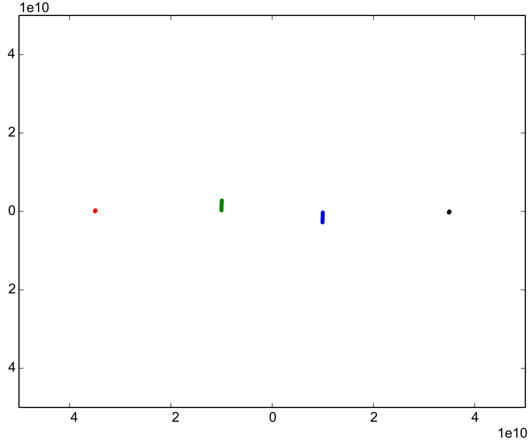
Simulace 3 těles



$n = 3000$

`python3 universe.py 3body.txt 20000` n

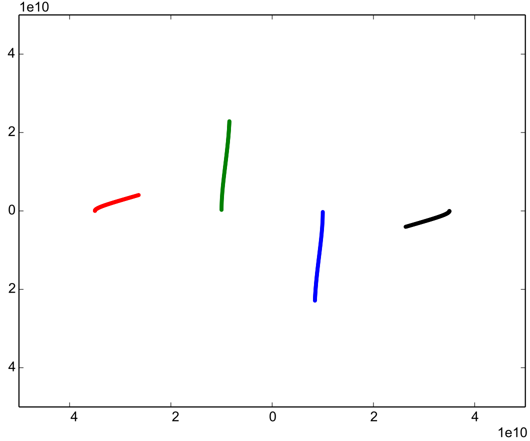
Simulace 4 těles



$$n = 10$$

```
python3 universe.py 4body.txt 20000 n
```

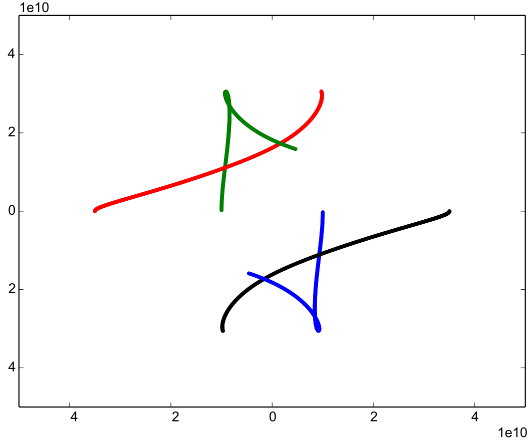
Simulace 4 těles



$$n = 100$$

`python3 universe.py 4body.txt 20000` *n*

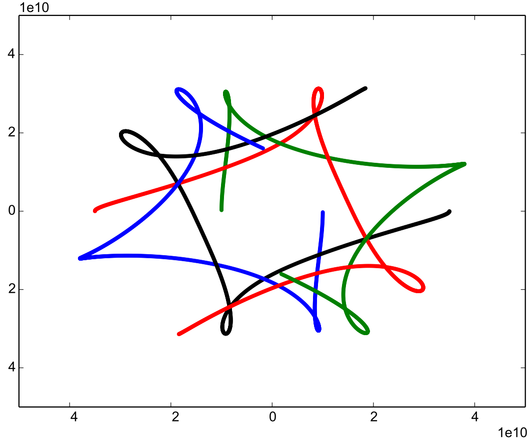
Simulace 4 těles



$n = 300$

```
python3 universe.py 4body.txt 20000 n
```

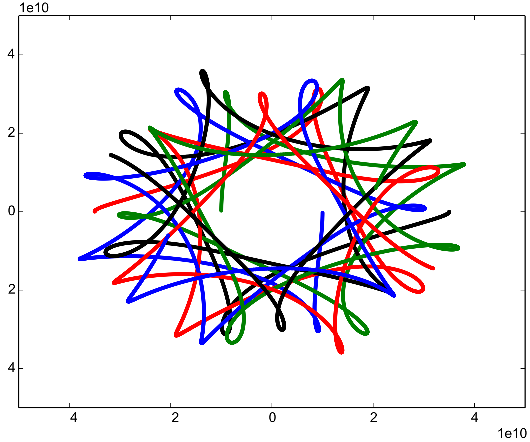
Simulace 4 těles



$n = 1000$

```
python3 universe.py 4body.txt 20000 n
```

Simulace 4 těles



$n = 3000$

`python3 universe.py 4body.txt 20000` n