Monte Carlo simulace

153	27
14	7
215	692
55	321
403	175

Lineární kongruentní generátor

smíšený generátor

$$I_{j+1} = aI_j + c \quad (\text{mod } m)$$

$$a, c, m \in \mathbf{N}$$

$$x_j = I_j / m$$

$$0 \le x_j < 1 \quad x_j \in U(0,1)$$

$m \approx 2^{32}$

maximální perioda m

I_0 : semínko

korelace

čistě multiplikativní generátor

$$I_{j+1} = aI_j \quad (\text{mod } m)$$

$$a, m \in \mathbf{N}$$

$$x_j = I_j / m$$

$$0 \le x_j < 1 \quad x_j \in U(0,1)$$

$a = 7^5 = 16807$

$m = 2^{31} - 1 = 2147483647$

perioda $2^{31} - 2 \approx 2.1 \times 10^9$

Monte Carlo simulace

Provedeme simulaci N hodnot I_1 ....I_N

semínko I_0

uložíme poslední hodnotu I_N jako semínko pro příští simulaci

• pokud je to úplně první simulace, vymyslím si semínko

jinak

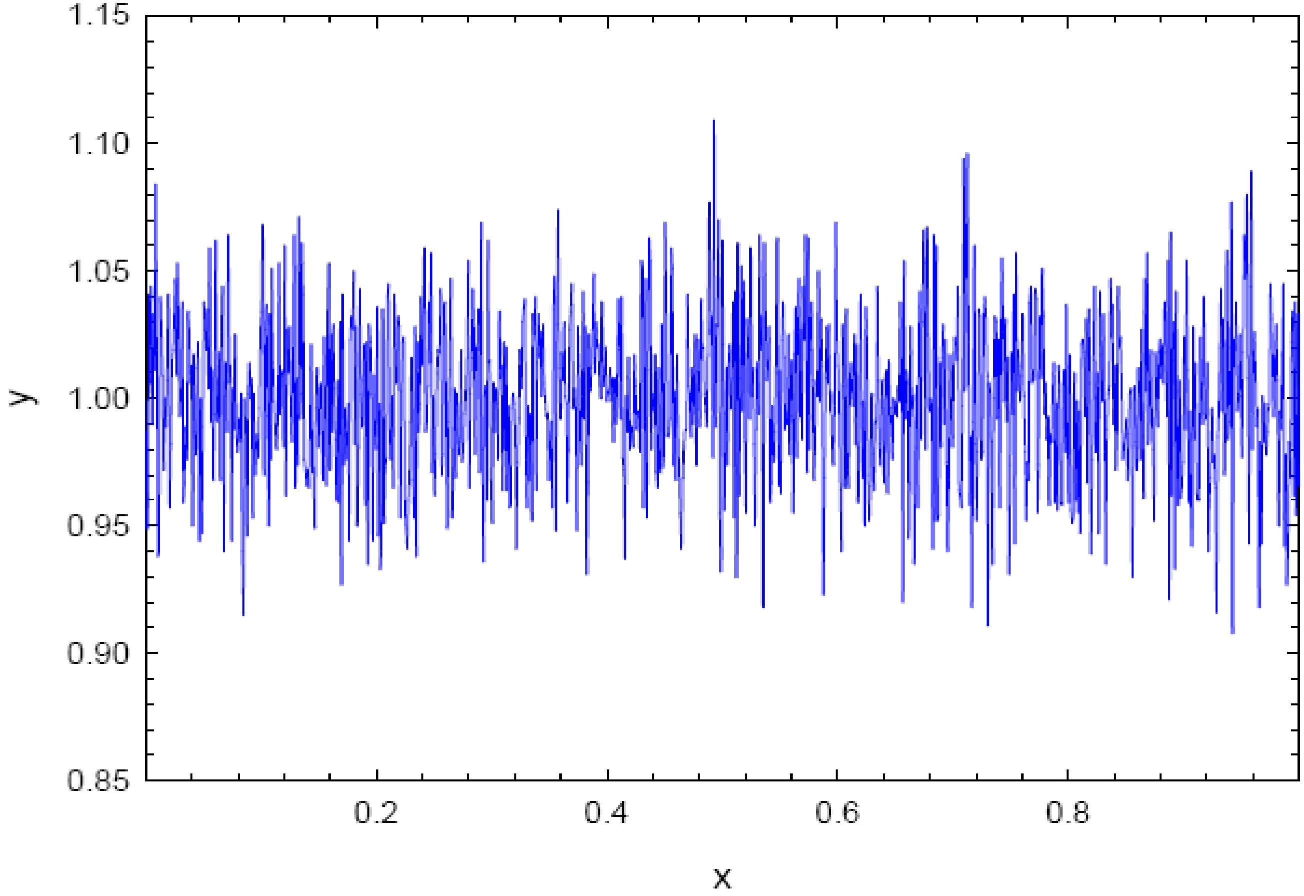
• načtu semínko
(poslední hodnotu I_j uloženou z předchozí simulace)

nebo

• semínko nějak vytvořím, tak aby to bylo pokaždé jiné číslo
(lze požít např. aktuální datum a čas atd.)

Lineární kongruentní generátor

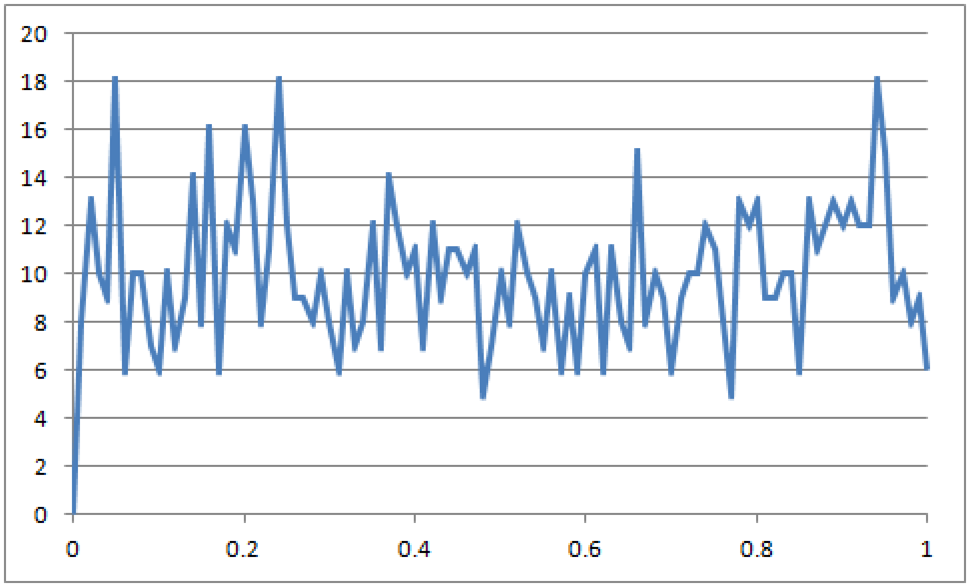
N = 1000000
multiplikativní generátor, a = 16807, m = 2^{31}-1



Monte Carlo simulace v Excelu

`generator-nahodnych-cisel.xlsx`

1. Vygenerujte v Excelu 1000 náhodných čísel z rovnoměrného rozdělení U(0,1) a nakreslete histogram nasimulovaných hodnot.



generátor náhodných čísel U(0,1)

`Ai=NÁHČÍSLO()`

histogram

`{=ČETNOSTI(A1:A1000,B1:B100)}`

oblast vygenerovaných hodnot

biny

maticové vzorce v Excelu:

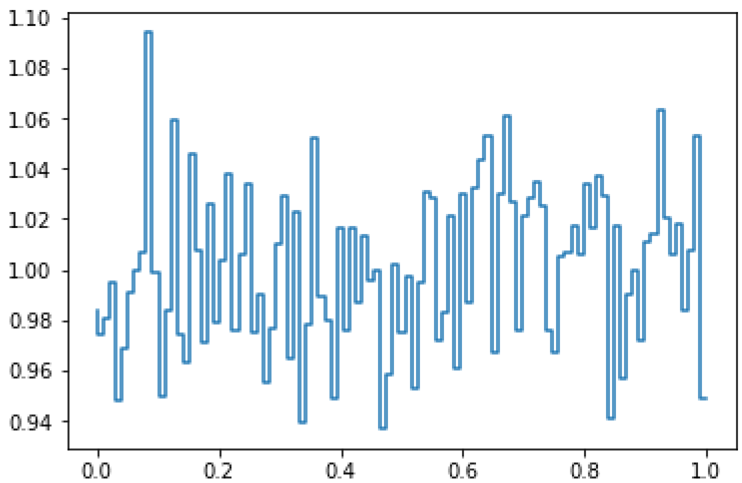
1. označit výstupní oblast
2. napsat vzorec a Ctrl+Shift+ENTER

Monte Carlo simulace v Pythonu

rand-histogram-1.py

vytvoření histogramu „bod po bodu“

```
import numpy as np
import matplotlib.pyplot as plt
n=100000 #pocet dat
nbins=100 #pocet binu
x=np.linspace(0,1,nbins) #x-ova souradnice grafu
y=np.zeros(nbins) #y-ova souradnice grafu
for i in range(0,n):
    ibin=int(nbins*np.random.random()) #generuje nahodne cislo
    y[ibin]=y[ibin]+1 #inkrementace histogramu
y=y/(n/nbins) #normalizace
plt.step(x,y) #vykresleni grafu
```

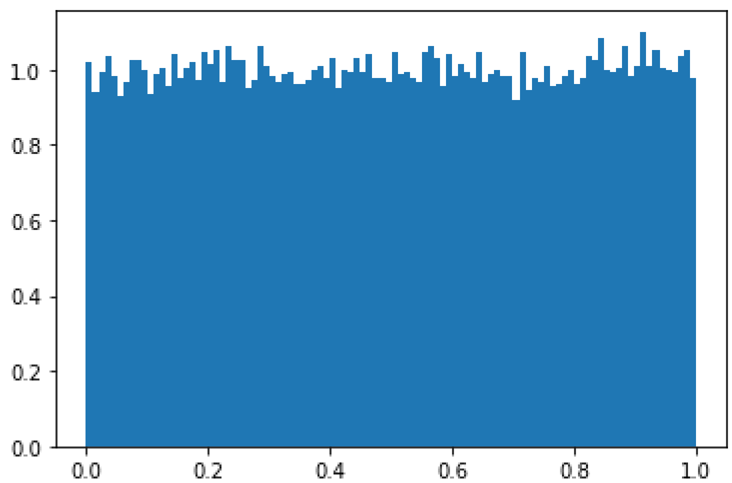


Monte Carlo simulace v Pythonu

`rand-histogram-2.py`

vytvoření a vykreslení histogramu pomocí metody plt.histogram

```
import numpy as np
import matplotlib.pyplot as plt
n=100000 #pocet dat
nbins=100
data=np.array(n) #deklarace pole x-souradnic
data=np.random.random_sample(n) #naplni pole data nahodnymi hodnotami
plt.hist(data,bins=nbins,density=True) #udela a vykresli histogram
```

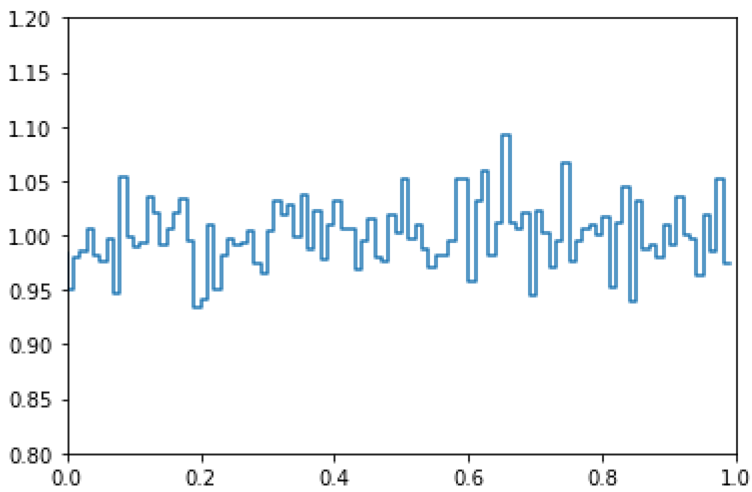


Monte Carlo simulace v Pythonu

rand-histogram-3.py

vytvoření histogramu pomocí metody np.histogram

```
import numpy as np
import matplotlib.pyplot as plt
n=100000 #pocet dat
nbins=100
data=np.array(n) #deklarace pole x-souradnic
data=np.random.random_sample(n) #naplni pole nahodnymi hodnotami
hist,bin_edges=np.histogram(data,bins=nbins,density=True) #udela histogram
x=bin_edges[0:nbins] #x-ova souradnice
y=hist  #y-ova souradnice
fig,ax=plt.subplots() #vytvoreni obrazku
ax.step(x,y) #vykresleni grafu
plt.xlim(0,1) #nastaveni rozmezi osy x: 0,1
plt.ylim(0.8,1.2) #nastaveni rozmezi osy y> 0.5,1.5
```

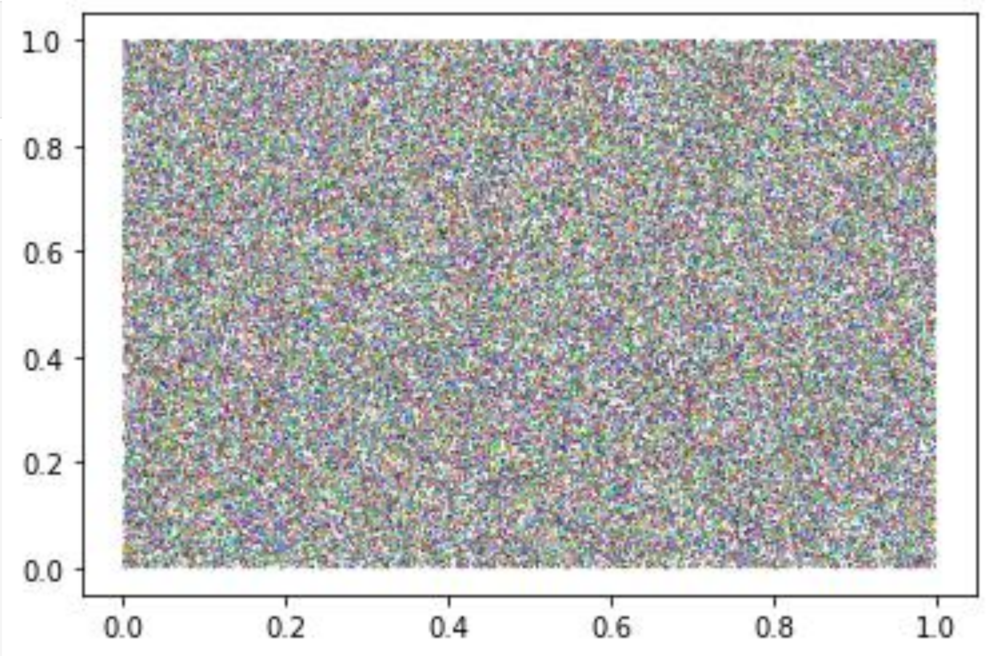


Barevný test v Pythonu

color-test.py

```
import numpy as np
import matplotlib.pyplot as plt
```

```
ndata=100000
#pole x,y souřadnic
x=np.empty(ndata)
y=np.empty(ndata)
#pole obsahující barevné složky rgb
color=np.empty([ndata,3])
for i in range(ndata):
    x[i]=np.random.random()
    y[i]=np.random.random()
    color[i,0]=np.random.random()
    color[i,1]=np.random.random()
    color[i,2]=np.random.random()
```




```
plt.scatter(x,y,c=color,s=1)
plt.show()
```

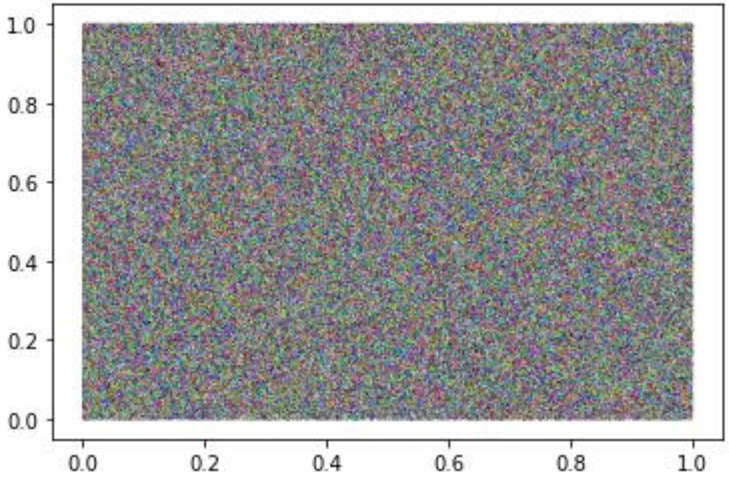
Lineární kongruentní generátor   kongruentni-generator.py

čistě mutiplikativní generátor

$$I_{j+1} = a\, I_{j+1} \pmod{m}$$

$$a = 16807$$

$$m = 2^{31} - 1 = 2147483647$$



počet dat $N = 10^6$

```
import numpy as np
import matplotlib.pyplot as plt
#ciste multiplikativni generator
a=16807
m=2147483647
#IBM RANDU
#a=65539
#m=2147483648
i_seed=1234
n=1000000
x = np.empty(n, dtype=float) #deklarace pole x-souradnic
y = np.empty(n, dtype=float) #deklarace pole y-souradnic
color=np.empty([n,3],dtype=float) #deklarace pole barva RGB
#ciste multiplikativni generator
i_old=i_seed
for i in range(0,n):
   i_next=(a*i_old) % m
   i_old=i_next
   x[i]=i_next/m
   i_next=(a*i_old) % m
   i_old=i_next
   y[i]=i_next/m
   i_next=(a*i_old) % m
   i_old=i_next
   color[i,0]=i_next/m
   i_next=(a*i_old) % m
   i_old=i_next
   color[i,1]=i_next/m
   i_next=(a*i_old) % m
   i_old=i_next
   color[i,2]=i_next/m
plt.scatter(x,y,s=1,c=color,edgecolors="none") #nakresli graf
```

Lineární kongruentní generátor

`kongruentni-generator.py`

čistě mutiplikativní generátor

$I_{j+1} = a\, I_{j+1} \pmod{m}$

$a = 16807$

$m = 2^{31} - 1 = 2147483647$

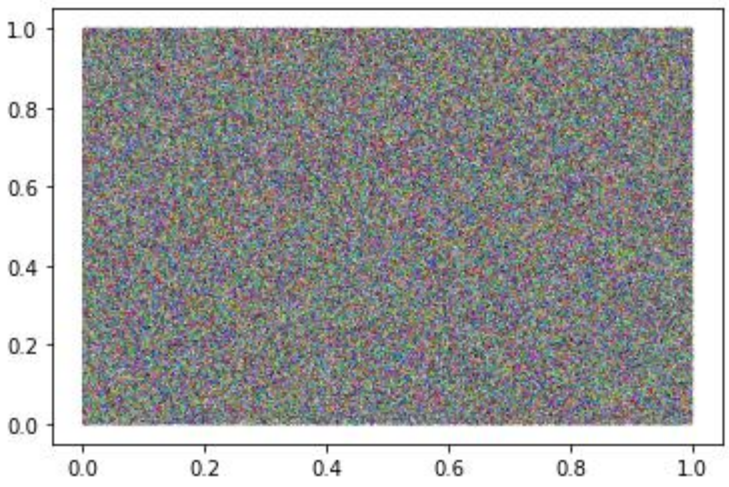
IBM RANDU

$I_{j+1} = a\, I_{j+1} \pmod{m}$

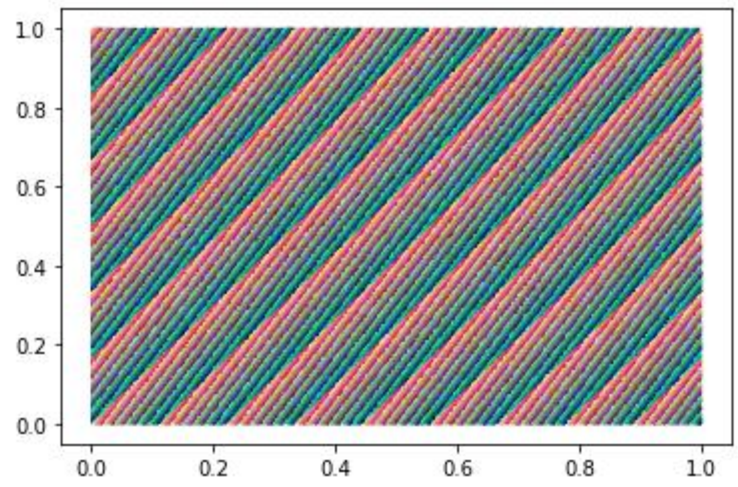
$a = 65539$

$m = 2^{31} = 2147483648$

“ We guarantee that each number is random individually,
but we don’t guarantee that more than one of them is random.”

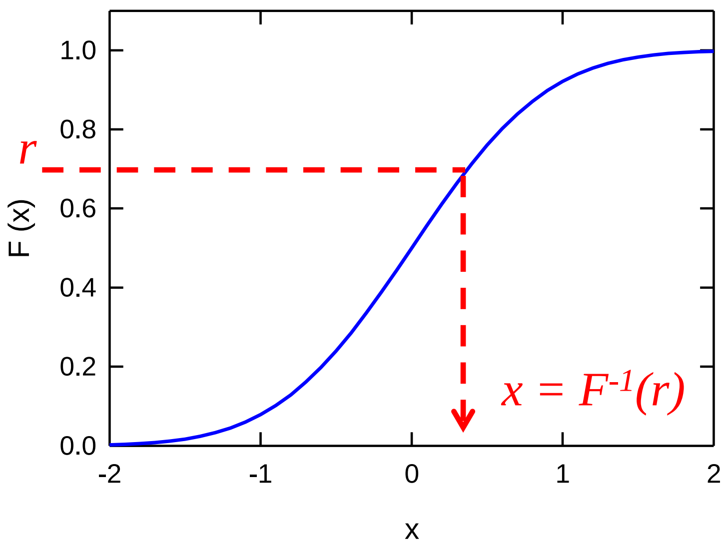


počet dat $N = 10^6$



počet dat $N = 10^6$

Monte Carlo simulace – metoda inverzní funkce



Metoda inverzní funkce

1. Vygeneruj náhodnou proměnnou r z U(0,1)

2. Vypočítej $x = F^{-1}(r)$,
   kde F^{-1} je inverzní funkce k distribuční funkci $F(x)$ požadovaného rozdělení

Nechť x je náhodná proměnná s rozdělením popsaným hustotou pravděpodobnosti $f(x)$ a distribuční funkcí $F(x)$ potom náhodná proměnná $r = F(x)$ má rovnoměrné rozdělní U(0,1)

Monte Carlo simulace

2. Doba života vybuzeného stavu elektronu je 100 μs. Při rozpadu se emituje foton. Proveďte v Excelu simulaci měření fotoluminiscence (200 hodnot). Nakreslete histogram naměřených hodnot.

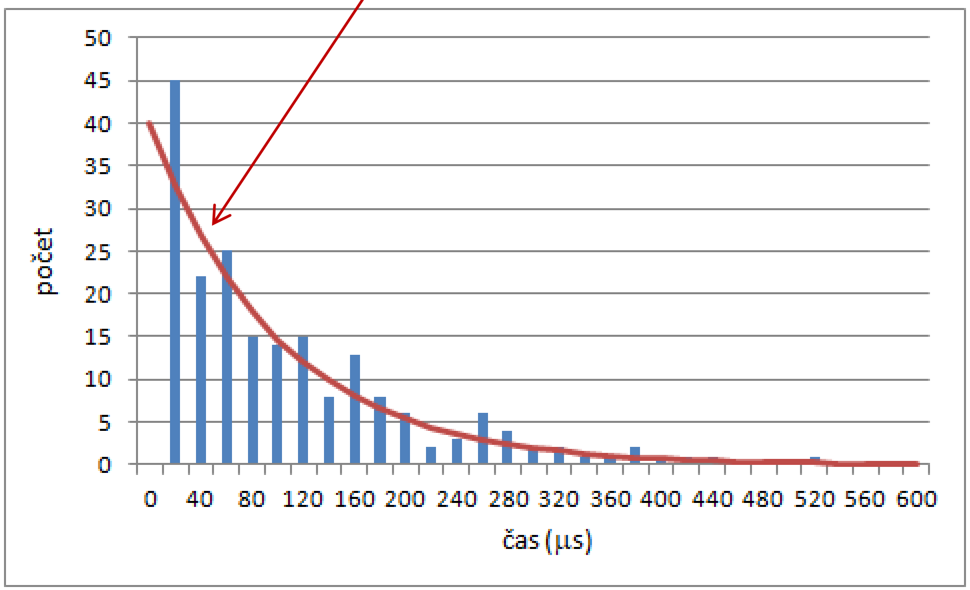
- hustota pravděpodobnosti: $f(t) = \frac{1}{\tau} e^{-\frac{t}{\tau}}$
- distribuční funkce: $F(t) = \int_0^t \frac{1}{\tau} e^{-\frac{z}{\tau}} \, dz \longrightarrow F(t) = 1 - e^{-\frac{t}{\tau}}$
- náhodná proměnná s rovnoměrným rozdělením: $r \in U(0,1)$
- inverzní funkce k distribuční funkci: $F(r)^{-1} = -\tau \ln(1-r)$
- náhodnou proměnnou s exponenciálním rozdělením získáme takto: $t = -\tau \ln(1-r)$
- ekvivalentní je $t = -\tau \ln(r)$

Monte Carlo simulace

`simulace-exp-rozpad.xlsx`

2. Doba života vybuzeného stavu elektronu je 100 μs. Při rozpadu se emituje foton. Proveďte v Excelu simulaci měření fotoluminiscence (200 hodnot). Nakreslete histogram naměřených hodnot.

modelová funkce $\frac{1}{\tau}e^{-\frac{t}{\tau}}$



generátor náhodných čísel U(0,1)

`Ai=NÁHČÍSLO()`

exponenciální rozdělení τ = 100 metodou inverzní funkce

`Bi=-100*LN(Ai)`

histogram

`{=ČETNOSTI(B2:B199,F2:F32)}`

oblast vygenerovaných hodnot

biny

maticové vzorce v Excelu:

1. označit výstupní oblast
2. napsat vzorec a Ctrl+Shift+ENTER

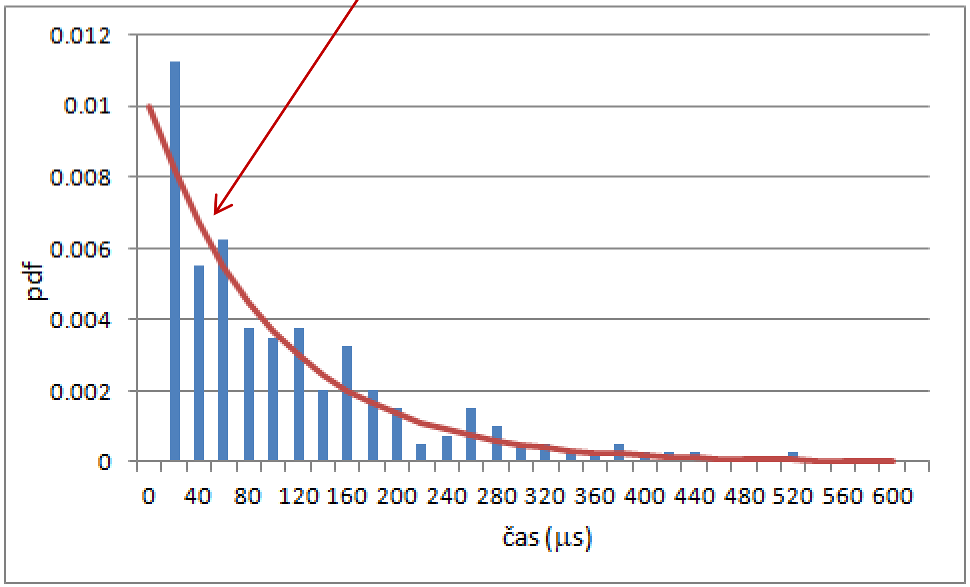
Monte Carlo simulace

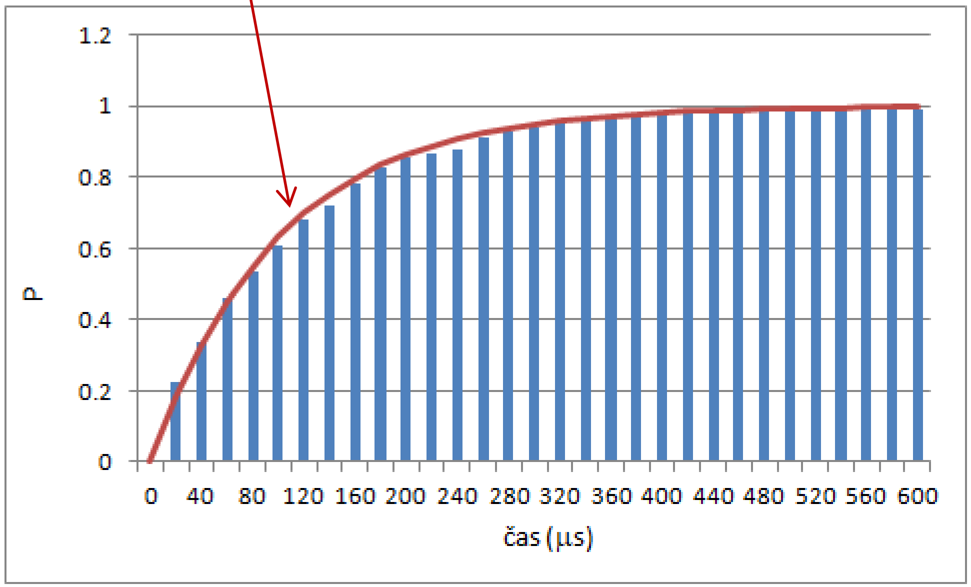
`simulace-exp-rozpad.xlsx`

3. *Z* dat vygenerovaných v předchozím příkladu udělejte normovaný histogram a srovnejte s hustotou pravděpodobnosti exponeciálního rozdělení a kumulativní histogram, který srovnejte s distribuční funkcí exponeciálního rozdělení.

hustota pravděpodobnosti $\frac{1}{\tau}e^{-\frac{t}{\tau}}$

distribuční funkce $1 - e^{-\frac{t}{\tau}}$





3. Jaká je pravděpodobnost, že vybuzená hladina bude žít déle než 200 μs?

$P(t > 200) = 1 - F(200) \approx 12\%$