Michał Gradzik 253880

Big Data Analytics

Rejection Method presented by Von Neumann

The task was to make an algorithm which generating X distributed as F.

In this model I had 2 functions:

$$f(x) = 4x^2e^{-2x}$$

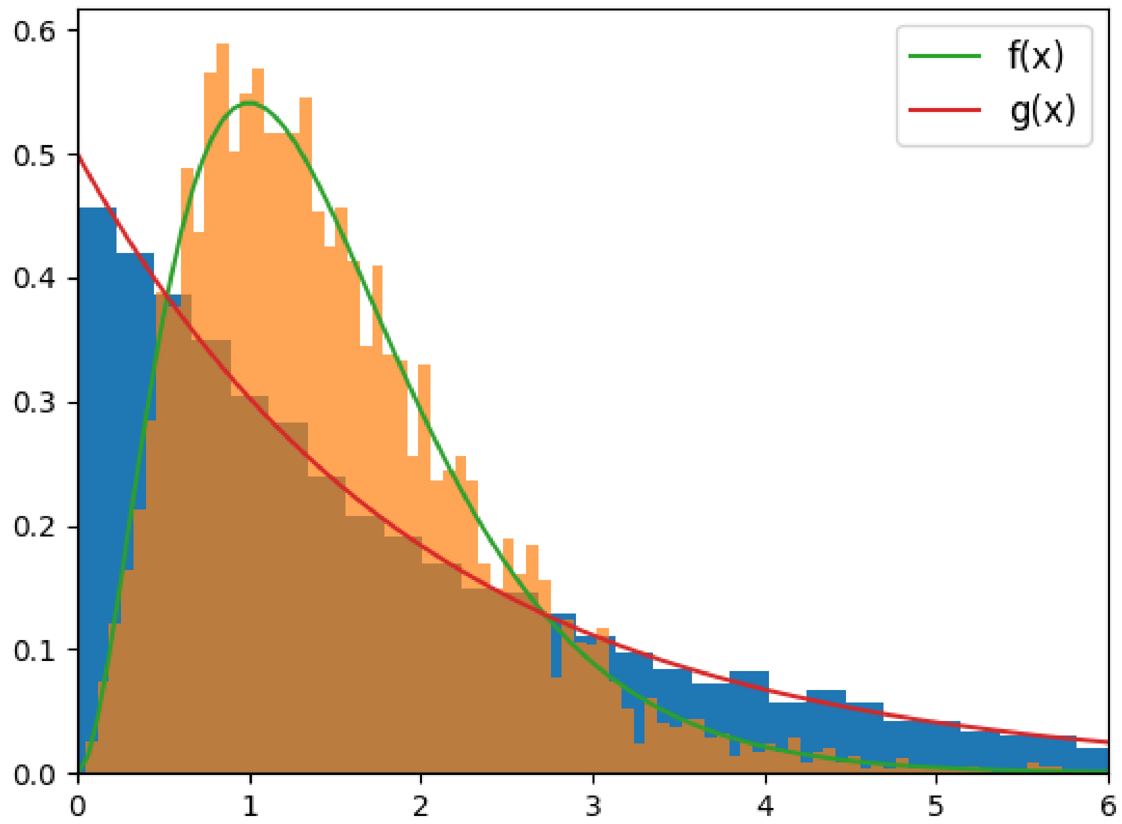
$$g(x) = \frac{1}{2}e^{-\frac{x}{2}}$$

$$C = 2$$

Algorith for this method is as below:

1. Generate Y distributed as g(x)
2. Generate u (must be independent from Y)
3. Check If $C * g(y) * u \le f(y)$ is true
   a. If true, accept y
   b. Otherwise go back to 1.

Running this algorithm, I got:



On the graph all Y are put into blue bins. The x-axis is limited for better clarity.

With orange color I printed bins which contains numbers from distribution with density function f.

In simulation I used 1 loops to generate 10000 numbers.

In each loop I'm generating y, u and then checking inequality.

The u variable is from uniform distribution where y is generated using equation from labs.

Source code:

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

pdf = lambda x: 4 * (x ** 2) * np.exp(-2 * x)
g_func = lambda x: 0.5 * np.exp(-x / 2)
C = 2
loops = 10000
number_array = []
y_pdf = [pdf(i) for i in np.linspace(0, 6, 100)]
y_g_func = [g_func(i) for i in np.linspace(0, 6, 100)]
uniform = []
i = 0
for i in range(loops):
    u = np.random.uniform(0, 1)
    y = - 2 * math.log(1 - np.random.uniform(0, 1), math.e)
    uniform.append(y)
    if C * g_func(y) * u < pdf(y):
        number_array.append(y)


plt.hist(uniform, bins=100, density=True)
plt.hist(number_array, bins=100, density=True, alpha=0.7)
plt.xlim([0, 6])
plt.plot(np.linspace(0, 6, 100), y_pdf, label='f(x)')
plt.plot(np.linspace(0, 6, 100), y_g_func, label='g(x)')
plt.legend(loc='upper right', fontsize='large')
plt.show()
```