

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ «БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»
(БГТУ им. В.Г.Шухова)**

Лабораторная работа №2
дисциплина «Системный анализ»
по теме «Разработка датчика случайных чисел, распределенных с заданным
законом распределения»

Выполнил: студент группы ВТ-31
Проверил:

Макаров Д.С.
Полунин А. И.

Белгород 2020

Лабораторная работа №2

«Разработка датчика случайных чисел, распределенных с заданным законом распределения»

Цель работы:изучить универсальный метод получения случайной величины для произвольного закона распределения случайной величины. Реализовать датчик случайных чисел с заданным законом распределения.

Задание: Зададим функцию $f_0(x) = x^5 + x^2 + 7$ на интервале $[1, 2]$.

Ход работы

Для получение функции распределения из $f_0(x)$ нормируем ее.

$$f(x) = k^{-1} * f_0(x)$$

$$k = \int_a^b f_0(x)dx = (x^5 + x^2 + 7) \Big|_1^2 \approx 19.83$$

Случайная величина ξ распределенная по закону распределения f , находится по уравнению:

$$\int_a^{\xi} f(x)dx = v$$

,где v - равномерно распределенная случайная величина.

Для удобства применения численного метода представим уравнение в виде:

$$\int_a^{\xi} f(x)dx - v = 0$$

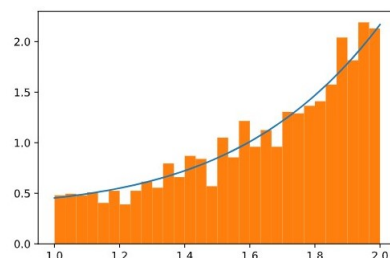


Рис. 1: График функции $f(x)$ и гистограмма полученных случайных величин, $n = 2000$

Приложение

Содержимое файла main.py

```
import numpy as np
import math
import matplotlib.pyplot as pl
import scipy.integrate as integrate
import scipy.optimize as optimize
k = 19.833333333333336
a = 1
b = 2
count_of_num = 2000
arr = []
result_arr = []

def f(x):
    return (x**5+x**2+7)
def f_(x):
    return (x**5+x**2+7)/k

def prob_num(xi,v):
    s1 = integrate.quad(f,a,xi)[0]
    #print('integral =',s1,' v =',v,"int-v =",s1/k-v," xi =",xi)
    return (s1/k - v)

for i in range(0,count_of_num):
    arr.append(np.random.random_sample())

for i in arr:
    def t(x):
        return prob_num(x,i)
    solve = optimize.fsolve(t,b)
    result_arr.append(solve[0])
    #print("xi =",solve)
    np.round(solve,3,solve)
    #print("xi =",solve)
    #print('---')
x_axis = np.linspace(a,b)
fig,ax = pl.subplots(1,1)
pl.plot(x_axis,np.apply_along_axis(f_,0,x_axis))
pl.hist(result_arr,30,density=True)
pl.show()
```