**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ **«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ ИМ. В. Г. ШУХОВА»**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

**ЛАБОРАТОРНАЯ РАБОТА №2**

**Дисциплина: Системный анализ**

**Тема:** **Разработка датчика случайных чисел, распределенных по произвольному закону**

Выполнил: ст. группы ВТ-31

Подкопаев Антон Валерьевич

Проверил: проф. ПО и ВТАС

Полунин Александр Иванович

**Белгород 2020**

**Цель работы:** составить свою функцию распределения и получить числа заданные по закону распределения

**Задания для выполнения к работе**

Разработать датчик случайных чисел, распределенных по заданному закону. Вид закона и интервал распределения задает сам учащийся. При задании функции плотности вероятности необходимо учесть, что в области ее существования она положительна, интеграл от нее на интервале [a,b] равен единице. Для удовлетворения первому условию студент должен проанализировать выбранную функцию и выбрать интервал [a,b] там, где она положительна. Для удовлетворения второму условию поступаем следующим образом.

Вычисляем интеграл от функции на интервале [a,b]. Получаем

S=.

Задаем новую функцию плотности вероятности

Интеграл от нее на интервале [a,b] равен единице. Для этой функции разрабатываем датчик случайных чисел.

При задании использовать нелинейную зависимость.

**Результат работы программы**

**Изображение выглядит как монитор, снимок экрана, внутренний, компьютер

Автоматически созданное описание**

**Изображение выглядит как снимок экрана, монитор, внутренний, компьютер

Автоматически созданное описание**

*Приложение*

import numpy as np

import matplotlib.pyplot as plt

import math

import scipy

import scipy.optimize as opt

import scipy.integrate as integrate

global vvv

def f0(x):

return (-0.2\*x\*\*3+1.8\*x\*\*2-1.5\*x+1.3)\*0.2

def f(xi):

a=0

b=10

K=(0.02\*(-0.5\*b\*\*4+6\*b\*\*3-7.5\*b\*\*2+13\*b))-(0.02\*(-0.5\*a\*\*4+6\*a\*\*3-7.5\*a\*\*2+13\*a))

return((0.02\*(-0.5\*xi\*\*4+6\*xi\*\*3-7.5\*xi\*\*2+13\*xi))-(0.02\*(-0.5\*a\*\*4+6\*a\*\*3-7.5\*a\*\*2+13\*a)))/K-vvv

my\_size = 5000#1000000

V = np.random.uniform(size=my\_size)

dd=[]

for i in range(len(V)):

vvv=V[i]

dd.append(opt.root(f,3).x)

a = 0

b = 5

x\_axis = np.linspace(a,b)

plt.plot(x\_axis,np.apply\_along\_axis(f0,0,x\_axis))

plt.show()

print("k =",integrate.quad(f0,a,b)[0])

ni = [dd[i][0]for i in range(len(V))]

plt.hist(ni,math.ceil(np.log2(my\_size)\*4))

plt.show()