

▼ Лабораторная работа №2

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

1. Задать свою функцию распределения

$$\rho_0(x, y, z) = \sin(x) + x^2$$

Функцию плотности распределения:

$$\rho(x) = K^{-1} * \rho_0(x)$$

$$\rho(x, y, z) = 15^{-1} * (\sin(x) + x^2) = \frac{\sin(x) + x^2}{15}$$

2. Найти решение $F(\xi) = Vd$

$$K^{-1} \left(\frac{\sin(x) + x^2}{15} \right) = V$$

```
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
class niton:
    def F(self,xi,v):#xi - вектор кси 1,2,3.   v-число случайное
        return (-np.cos(xi)+(self.xi**3)/3 +np.cos(self.a)- (self.a**3)/3 )/self.K-v
    def __init__(self):
        self.size=1
        self.coun=0
        self.alpha=0
        self.a=0
        self.b=3
        self.K=-np.cos(self.b)+(self.b**3)/3 +np.cos(self.a)- (self.a**3)/3
    def solve(self,g=np.array([0.1,0.1,0.1]),v=0):
        arr=np.array([[self.dF_de1de1(g),   self.dF_de1de2(g),   self.dF_de1de3(g)],[ self.dF_de2de1(g),   self.dF_de2de2(g),   self.dF_de2de3(g)],[ self.dF_de3de1(g),   self.dF_de3de2(g),   self.dF_de3de3(g)]])
```

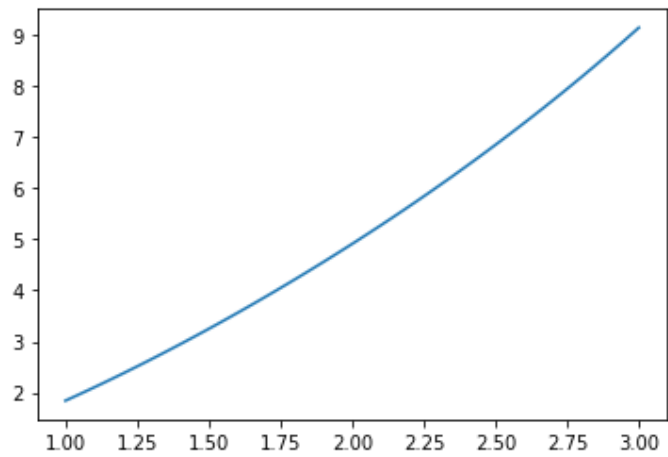
```

n_inv=np.linalg.inv(arr)
grad=np.array([self.dF_de1(g),self.dF_de2(g),self.dF_de3(g)])
g1=g-(n_inv.dot(grad))
##print(g.shape, n_inv.shape, grad.shape, (n_inv*grad).shape)
#print('fffff',self.F(g1,v))
if(np.abs(self.F(g,v)-self.F(g1,v))<0.00001):
    self.coun=0
    return g1
if self.coun>5:
    print('coun>30')
    self.coun=0
    return g1
self.coun=self.coun+1
return self.solve(g1,v)
def f0(x):
    return np.sin(x)+(x)**2
def f(xi):
    alpha=1
    a=1
    b=3
    K=-np.cos(b)+(b**3)/3 +np.cos(a)- (a**3)/3
    return(-np.cos(xi)+(xi**3)/3 +np.cos(a)- (a**3)/3 )/K-vvv
def sv(g,v):
    rt=opt.root(f,0)
    return rt

my_size = 10000 #1000*1000
V = np.random.uniform(size=my_size)
l=[ i for i in range(-100,100)]
g=np.array([1.1,1.1,1.1])
dd=[]
for i in range(len(V)):
    vvv=V[i]
    dd.append(opt.root(f,0).x)
a = 1
b = 3
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])

```





k = 10.196961469135251

[0.84157906843893, 0.8416871984068026, 0.8417953747071938, 0.8419035973357843, 0.8420118662882569, 0.8421201815602949, 0.8422285431475832, 0.842336

[0.04416508465658739, 0.02352072923160627, 0.04723110687133957, 0.05133584916479426, 0.013593735770916951, 0.03410272198386363, 0.01935464835960907

