Министерство образования и науки РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Рыбинский государственный авиационный технический университет

имени П.А. Соловьева»

Факультет радиоэлектроники и информатики

Кафедра Математическое и программное обеспечение ЭВС

**ОТЧЕТ**

**по лабораторной работе**

**«Стохастический автомат»**

Студентка группы ИПБ-15       Семёнова Е. Е.

*(Код) (Подпись, дата)        (Фамилия И. О.)*

Рыбинск 2019

# Задача

Автомат имеет 3 состояния, имеется матрица вероятностей перехода из одного состояния в другое. На выходе выводится результат расчета математической модели и автоматной.

Матрица вероятностей перехода

0,5 0,25 0,25

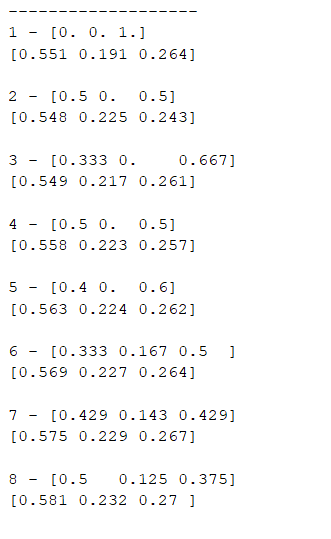
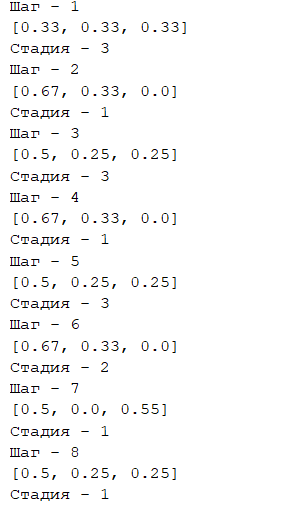
0,5 0 0,5

0,67 0,33 0

Входной вектор

vectorNorm = [0.33, 0.33, 0.33]

Результат работы программы.



Используемые формулы.

Уравнение Колмогорова-Чепмена



# Листинг программы.

import numpy as np

import random

import math

def Vectorstep(vector, matrix):

n, m = matrix.shape

x = 0

res = np.zeros((vector.\_\_len\_\_()))

for i in range(0, n, 1):

for j in range(0, m, 1):

res[i] += vector[i] \* matrix[j][i]

return res

def Matrstep(x1, x2):

n1, m1 = x1.shape

n2, m2 = x2.shape

if (m2 != n1):

return None

mResult = np.zeros((n1, m1))

for i in range(0, n1, 1):

for j in range(0, m2, 1):

for k in range(0, m1, 1):

mResult[i, j] += x1[i][k] \* x2[k][j]

return mResult

def Ymnoj(steps, matrix, vector):

matt = matrix

for i in range(1, steps, 1):

matt = Matrstep(matt, matrix)

res = Vectorstep(vector, matt) return res

def matem(vectorNorm, matrix, steps):

fullRes = np.zeros((steps, vectorNorm.\_\_len\_\_()))

for i in range(0, steps, 1):

fullRes[i] = Ymnoj(i + 1, matrix, vectorNorm)

return fullRes

def automat(vector, matrix, steps):

vect = vector

stepStat = np.zeros((steps, vector.\_\_len\_\_()))

for i in range(0, steps, 1):

zz = str(i + 1)

print("Шаг - " + zz)

rz = 0

min = 1

max = 99

dice = min + int(random.random() \* ((max - min) + 1))

dice /= 100

diceField = np.zeros(((vect.\_\_len\_\_()) + 1))

for j in range(0, vect.\_\_len\_\_(), 1):

rz += vect[j]

diceField[j + 1] = rz

rz = 1

numb = 0

znak = 0

for j in range(0, vect.\_\_len\_\_(), 1):

if (math.fabs(dice - diceField[j]) < rz):

numb = j

rz = math.fabs(dice - diceField[j])

znak = dice - diceField[j]

if znak < 0:

numb = numb - 1

for j in range(0, vect.\_\_len\_\_(), 1):

if (i > 0):

stepStat[i][j] = stepStat[i - 1][j]

stepStat[i][numb] += 1;

print(vector)

vv = numb + 1

vv = str(vv)

print("Стадия - " + vv)

for j in range(0, vector.\_\_len\_\_(), 1):

vect[j] = matrix[numb][j]

return stepStat

if \_\_name\_\_ == '\_\_main\_\_':

vectorNorm = [0.33, 0.33, 0.33]

len\_vectorNorm = vectorNorm.\_\_len\_\_()

matrix = np.loadtxt("test.txt")

print(matrix)

print("Кол-во шагов?")

steps=int(input())

matem= matem(vectorNorm, matrix, steps)

automat = automat(vectorNorm, matrix, steps)

matem=np.around(matem,decimals=3)

for i in range (0,steps,1):

for j in range (0,matrix.\_\_len\_\_(),1):

automat[i][j]/=i+1

automat = np.around(automat, decimals=3)

print()

print('-------------------')

for i in range (0,steps,1):

print(i+1,automat[i], sep=' - ')

print(matem[i])