МИНИСТЕРСТВО НАУКИ ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

*Федеральное государственное автономное образовательное учреждение высшего образования Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики*

*Мегафакультет трансляционных информационных технологий*

*Факультет информационный технологий и программирования*

**По дисциплине:***«Прикладная математика»* **Лабораторная работа №3**

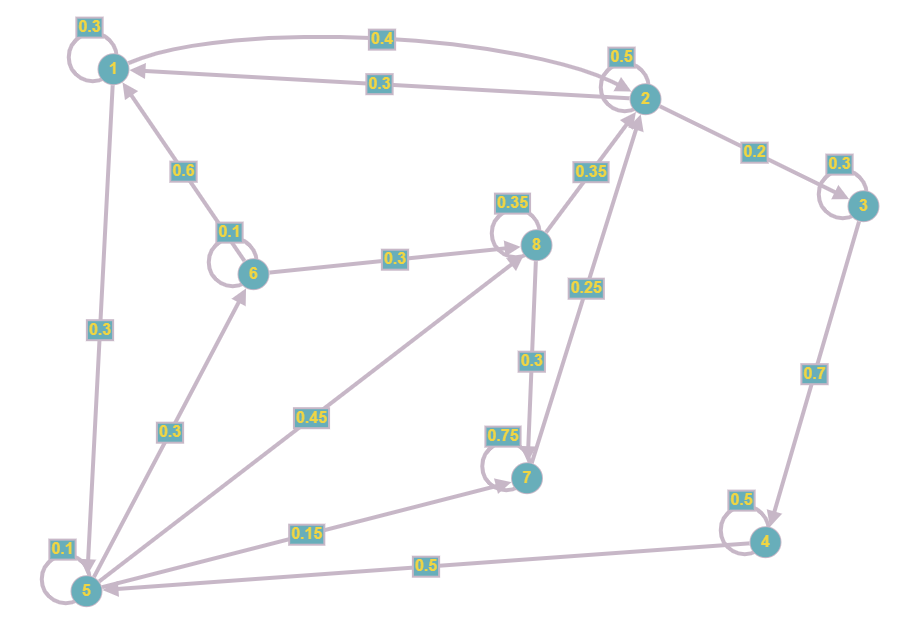
***Выполнил:***

*М33091 Ларин В. Д.*

*САНКТ-ПЕТЕРБУРГ*

*2021*

**1.Исходный граф (Эргодическая цепь Маркова):**

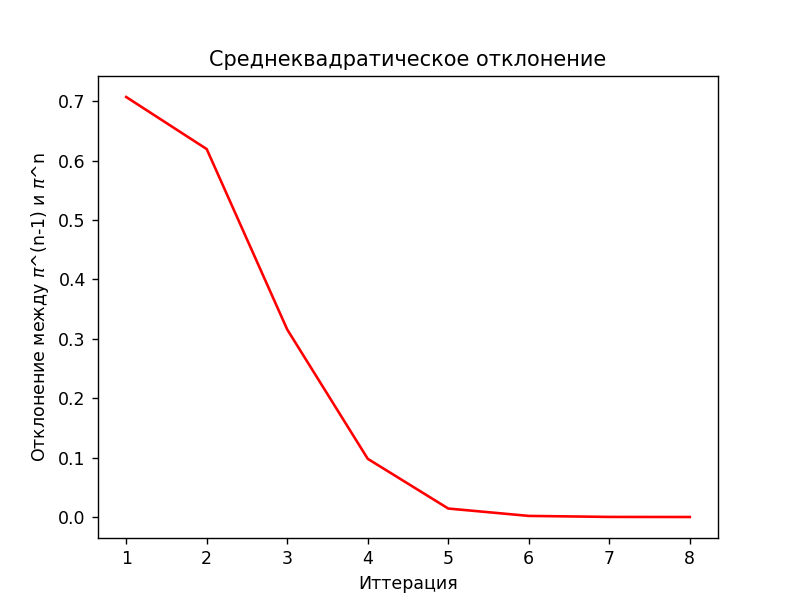


Matrix = 0.3, 0.4, 0, 0, 0.3, 0, 0, 0,  
 0.3, 0.5, 0.2, 0, 0, 0, 0, 0,  
 0, 0, 0.3, 0.7, 0, 0, 0, 0,  
 0, 0, 0, 0.5, 0.5, 0, 0, 0,  
 0, 0, 0, 0, 0.1, 0.3, 0.15, 0.45,  
 0.6, 0, 0, 0, 0, 0.1, 0, 0.3,  
 0, 0.25, 0, 0, 0, 0, 0.75, 0,  
 0, 0.35, 0, 0, 0, 0, 0.3, 0.35,

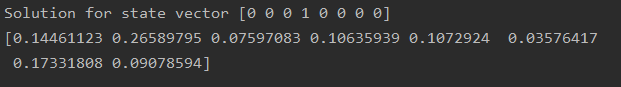
**2. Распределение по состояниям при ∞ численно**

**2.1 при изначальном векторе вероятности состояний**

График Среднеквадратического отклонения:



Полученный результат:

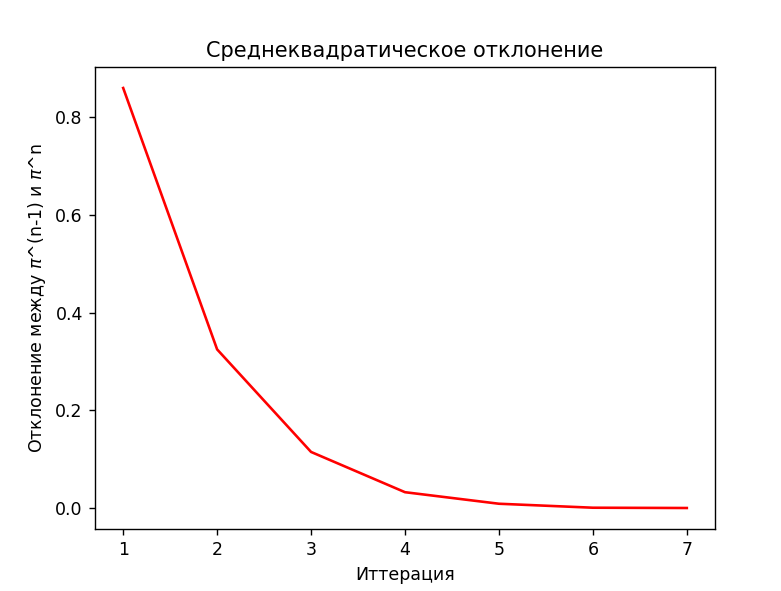


**2.2 при изначальном векторе вероятности состояний**

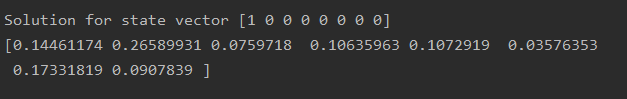
График Среднеквадратического отклонения:

при изначальном векторе вероятности состояний

График Среднеквадратического отклонения:



Полученный результат:



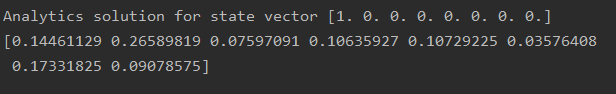
Вывод:



Таким образом -вероятность того, что система находится в j-м состоянии. не зависит от начального состояния, с которого начинается имитационное моделирование (финальные вероятности).

**3. Распределение по состояниям при ∞ аналитически:**

Полученный результат:



**Листинг кода:**

import numpy as np  
import matplotlib.pyplot as plt  
  
matrix = np.array([[0.3, 0.4, 0, 0, 0.3, 0, 0, 0,],  
 [0.3, 0.5, 0.2, 0, 0, 0, 0, 0,],  
 [0, 0, 0.3, 0.7, 0, 0, 0, 0,],  
 [0, 0, 0, 0.5, 0.5, 0, 0, 0,],  
 [0, 0, 0, 0, 0.1, 0.3, 0.15, 0.45,],  
 [0.6, 0, 0, 0, 0, 0.1, 0, 0.3,],  
 [0, 0.25, 0, 0, 0, 0, 0.75, 0,],  
 [0, 0.35, 0, 0, 0, 0, 0.3, 0.35,]])  
vector1 = np.array([0, 0, 0, 1, 0, 0, 0, 0])  
vector2 = np.array([1, 0, 0, 0, 0, 0, 0, 0])  
epsilon = 0.0001  
step = 1  
mse = 1  
  
  
def solve(vector, matrix, step, mse, epsilon):  
 print('Solution for state vector', vector)  
 mse\_values = []  
 x\_values = []  
 while mse > epsilon:  
 powered\_matrix = np.linalg.matrix\_power(matrix, step) # возведение матрицы в степень  
 new\_vector = vector.dot(powered\_matrix) # скалярное произведенеие  
 mse = np.linalg.norm(vector - new\_vector)  
  
 mse\_values.append(mse)  
 x\_values.append(step)  
  
 vector = new\_vector  
 step += 1  
  
 plt.title("Среднеквадратическое отклонение")  
 plt.xlabel("Иттерация")  
 plt.ylabel("Отклонение между $\pi$^(n-1) и $\pi$^n")  
 plt.plot(x\_values, mse\_values, 'r')  
 plt.show()  
  
 print(vector, '\n')  
 return vector  
  
  
def analytics(matrix, n):  
 matrix\_this = matrix.T  
 matrix\_this -= np.eye(n)  
 matrix\_this[0] = np.ones(n)  
 vector\_analytics = np.zeros((n,))  
 vector\_analytics[0] = 1  
 result = np.linalg.solve(matrix\_this, vector\_analytics) # решение системы

**Ответы на вопросы:**

1.Случайная величина(дискретная) - это функция из множества элементарных исходов в множество вещественных не отрицательных чисел. Множество элементарных исходов должно быть конечным или счётным множеством чисел.

2. Модель Марковского процесса представляет собой граф, где узлы обозначают состояние

моделируемого объекта, а дуги – вероятность перехода из одного состояния в другое.

3. Марковские процессы делятся на два вида:

1. Дискретные цепи Маркова, где система меняет свое состояние в определенные

такты времени (P-схема)

2. Непрерывные цепи Маркова, где система меняет свое состояние в произвольный

момент времени (q-схема)

4. - вероятность переходи из i-го состояния в j-ое

5. Стохастическая матрица в теории вероятностей — это неотрицательная матрица, в которой сумма элементов любой строки или любого столбца равна единице.

6. Если существует такое k >0 , что при любых допустимых состояниях Xi и X j вероятности πij(k)

положительны, то цепь эргодична.