Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Отчет по лабораторной работе №2 по курсу «Моделирование»

Тема Марковские процессы

Студент Прянишников А. Н.

Группа ИУ7-75Б

Оценка (баллы) ____________

Преподаватель Рудаков И. В.

Москва — 2022 г.

Условия лабораторной работы

Написать программу, позволяющую рассчитать предельную вероятность пребывания в каждом из состояний системы по заданным интенсивностям перехода между состояниями. Количество состояний не может быть больше 10.

Теоретическая часть

В этом разделе будет рассмотрено определение марковских систем, уравнение Колмогорова и способы расчёта предельной вероятности для каждого состояния.

Марковская система

Случайный процесс, протекающий в системе S, называется марковским, если он обладает следующим свойством: для каждого момента времени t_0 вероятность любого состояния системы в будущем (при $t > t_0$) зависит только от ее состояния в настоящем (при $t = t_0$) и не зависит от того, когда и каким образом система пришла в это состояние.

Вероятностью i-го состояния называется вероятность p_i того, что в момент t система будет находиться в состоянии S_i. Для любого момента t сумма вероятностей всех состояний равна единице.

Уравнения Колмогорова

Для решения поставленной задачи, необходимо составить систему уравнений Колмогорова по следующим принципам:

- в левой части каждого из уравнений стоит производная вероятности i-го состояния;
- в правой части — сумма произведений вероятностей всех состояний (из которых идут стрелки в данное состояние), умноженная на интенсивно-

сти соответствующих потоков событий, минус суммарная интенсивность всех потоков, выводящих систему из данного состояния, умноженная на вероятность данного (i-го состояния).

Пусть в системе возможно три состояния. Таблица интенсивности переходов между состояниями представлена на таблице 1:

Таблица 1: Матрица интенсивности переходов в системе

λ_{00}	λ_{01}	λ_{02}
λ_{10}	λ_{11}	λ_{12}
λ_{20}	λ_{21}	λ_{22}

Тогда уравнения Колмогорова соответствуют формуле 1:

$$\begin{cases} p_0^{'} = -(\lambda_{00} + \lambda_{01} + \lambda_{02}) * p_0 + \lambda_{10} * p_1 + \lambda_{20} * p_2 \\ p_1^{'} = -(\lambda_{10} + \lambda_{11} + \lambda_{12}) * p_1 + \lambda_{01} * p_0 + \lambda_{21} * p_2 \\ p_2^{'} = -(\lambda_{20} + \lambda_{21} + \lambda_{22}) * p_2 + \lambda_{02} * p_0 + \lambda_{12} * p_1 \end{cases} \tag{1}$$

Вычисление предельной вероятности

Для вычисления предельной вероятности, нужно приравнять левую часть уравнений к 0, а также учесть, что сумма вероятностей должна равняться 1.

Тогда уравнения приводятся к системе, представленной на формуле 2:

$$\begin{cases} -(\lambda_{00} + \lambda_{01} + \lambda_{02}) * p_0 + \lambda_{10} * p_1 + \lambda_{20} * p_2 = 0 \\ -(\lambda_{10} + \lambda_{11} + \lambda_{12}) * p_1 + \lambda_{01} * p_0 + \lambda_{21} * p_2 = 0 \\ -(\lambda_{20} + \lambda_{21} + \lambda_{22}) * p_2 + \lambda_{02} * p_0 + \lambda_{12} * p_1 = 0 \\ p_0 + p_1 + p_2 = 1 \end{cases} \tag{2}$$

Эту систему можно решить методами линейной алгебры, предварительно убрав из системы первое уравнение. Вектор вероятностей представим как P, вектор коэффициентов как A, а вектор решений уравнений – B. Тогда требуется решить уравнение $AP = B$.

Вычисление необходимого времени

После того, как предельные вероятности будут найдены, необходимо найти время. Для этого необходимо с интервалом dt находить каждую вероятность в момент времени $t+dt$. Когда найденная вероятность будет равна соответствующей финальной с точностью до заданной погрешности, тогда можно завершить вычисления.

На каждом шаге необходимо вычислять приращения для каждой вероятности (как функции). Уравнение вычисления представлено на формуле 3:

$$dp_i = \frac{-\sum_{j=0}^{N} \lambda_{ij} * p_i + \sum_{j=0}^{N} \lambda_{ji} * p_j, j \neq i}{dt} \quad (3)$$

Начальные вероятности можно посчитать по формуле 4:

$$p_i = 1/N \quad (4)$$

Практическая часть

В этой части будет обоснован выбор средств разработки ПО, приведены листинги кода и демонстрация работы программы.

Выбор средств разработки ПО

В качестве языка программирования выбран Python 3.9, так как имеется опыт разработки проектов на этом языке. Также будет использована библиотека numpy для решения уравнений для вычисления предельных вероятностей.

Листинги кода

На листинге 1 представлена реализация алгоритма создания матрицы вероятностей по заданной матрице интенсивностей переходов.

На листинге 2 представлена реализация алгоритма решения уравнения для нахождения предельных вероятностей.

На листинге 3 представлена реализация алгоритма поиска времени для нахождения вероятностей системы.

Листинг 1: Реализация алгоритма создания матрицы вероятностей

```
def create_probs_matrix(matrix_of_intensity: np.array) -> np.array:
    numbers_of_state = matrix_of_intensity.shape[0]

    matrix_of_probs = np.zeros(shape=(numbers_of_state + 1, numbers_of_state +
        1))

    for cur_state_row in range(numbers_of_state):
        for i in range(numbers_of_state):
            if i != cur_state_row:
                matrix_of_probs[cur_state_row][i] = matrix_of_intensity[i][
                    cur_state_row]
        matrix_of_probs[cur_state_row][cur_state_row] = -sum(matrix_of_intensity
            [cur_state_row])

    for i in range(numbers_of_state + 1):
        matrix_of_probs[numbers_of_state][i] = 1

    return matrix_of_probs
```

Листинг 2: Реализация алгоритма решения уравнения для нахождения предельных вероятностей

```
def solve_equation(matrix_of_probs: np.array) -> np.array:
    a = build_a_matrix(matrix_of_probs)
    b = build_b_matrix(matrix_of_probs)

    return np.linalg.solve(a, b)
```

Листинг 3: Реализация алгоритма поиска времени для нахождения вероятностей системы

```
        for cur_state_row in range(number_of_states):
            for i in range(number_of_states):
                if i != cur_state_row:
                    delta[cur_state_row] += intensity_matrix[i][cur_state_row] *
                        cur_probs[i]
            delta[cur_state_row] -= cur_probs[cur_state_row] * sum(
               intensity_matrix[cur_state_row, :])
        cur_probs += delta * TIME_DELTA

        all_times.append(system_time)
        system_time += TIME_DELTA
```

Демонстрация работы программы

Таблица интенсивности переходов между состояниями представлена на таблице 2:

Таблица 2: Матрица интенсивности переходов в системе

0	0.5	0	0	0
0	0	2	0	0
0	0	0.2	1.3	1.5
0.8	0	0	0	0
2	0	0	0	0

Пользователь может выбрать в меню количество элементов системы, а также задать матрицу интенсивности. Демонстрация меню представлена на рисунке 1:

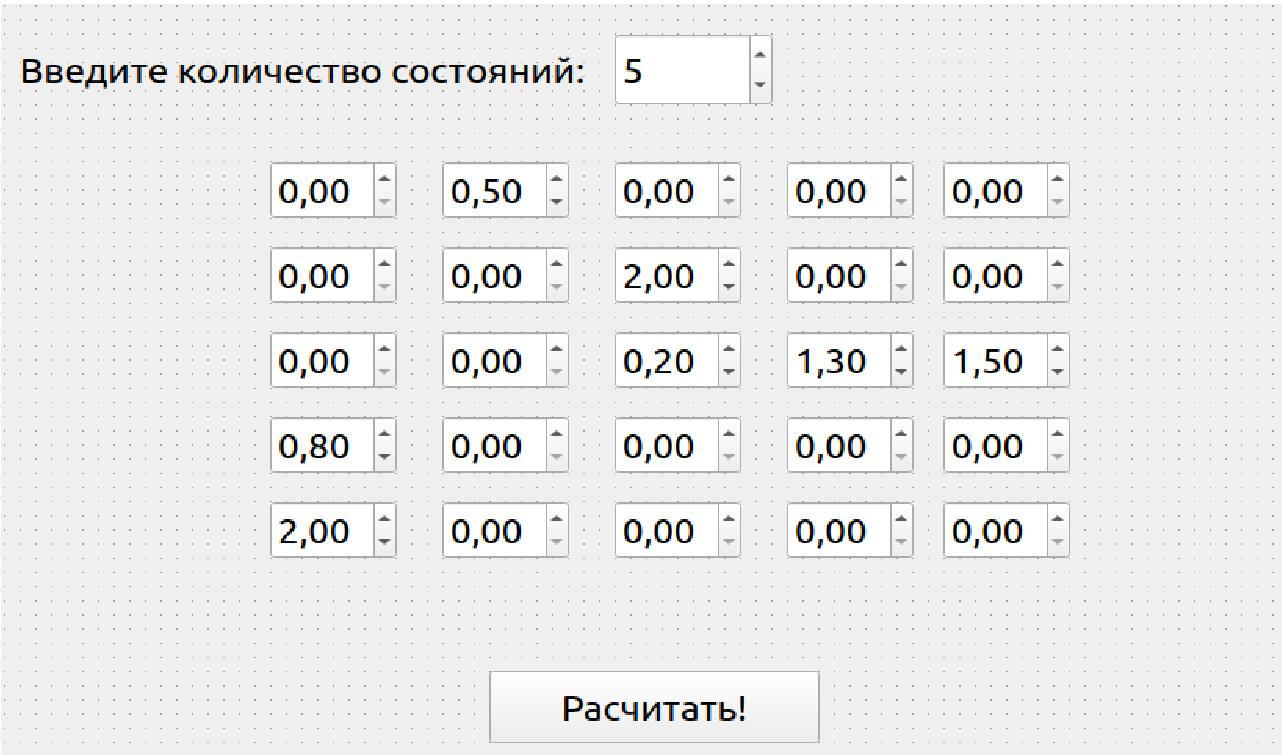

Рисунок 1: Демонстрация работы программы

Получившиеся результаты представлены в таблице 3:

Таблица 3: Матрица интенсивности переходов в системе

p_i	0.55172414	0.13793103	0.09195402	0.14942529	0.06896552
t_i	1.65	0.31	0.833	2.791	1.731

График времени стабилизации вероятности состояний изображен на рисунке 2:

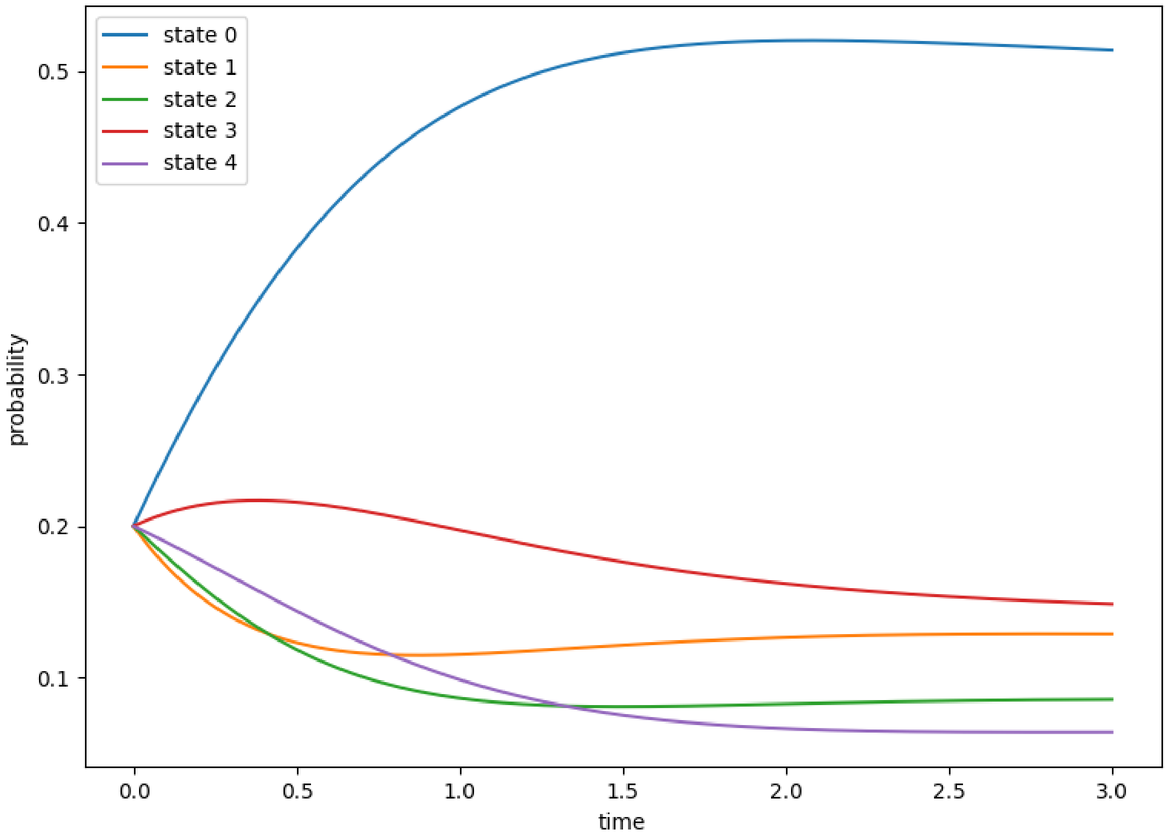

Рисунок 2: Демонстрация работы программы