



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 № 3
по курсу «Моделирование»
на тему: «Генератор псевдослучайных чисел»

Студент ИУ7-71Б
(Группа)

(Подпись, дата)

Мицевич М. Д.
(И. О. Фамилия)

Преподаватель

(Подпись, дата)

Рудаков И. В.
(И. О. Фамилия)

2022 г.

1 Теоретический раздел

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

Для Марковских процессов обычно составляют уравнения Колмогорова.

Общий вид:

$$F = (p'(t), p(t), \Lambda) = 0,$$

где $\Lambda = \lambda_1, \lambda_2, \dots, \lambda_n$ - набор коэффициентов.

Вероятностью i -го состояния называется вероятность $p_i(t)$ того, что в момент t система будет находиться в состоянии S_i . К системе может быть добавлено условие нормировки: для любого момента t сумма вероятностей всех состояний равна единице:

$$\sum_{i=1}^n p_i = 1.$$

Для того, чтобы решить поставленную задачу, необходимо составить систему уравнений Колмогорова. Все уравнения составляются по определённым правилам.

- В левой части каждого уравнения стоит производная вероятности i -ого состояния.
- В правой части содержится столько членов, сколько переходов, связанных с данным состоянием. Если переход из состояния, то соответствующий член имеет знак минус, а, если наоборот, то плюс.
- Каждый член равен произведению плотности вероятности перехода (т.е. интенсивности), соответствующей рассматриваемой стрелки, на вероятность того состояния, из которого исходит стрелка.

2 Практическая часть

На листинге 2.1 представлен код для моделирования работы системы.

Листинг 2.1 – Построение графиков для равномерного распределения

```
from math import fabs

EPS = 1e-5

def get_system_times(matrix):
    print(matrix)

    pArray = list()
    pArray.append(1)
    for i in range(1, len(matrix[0])):
        pArray.append(0)

    step = 0.01
    t = 0
    times = [0 for _ in pArray]

    while (True):
        pd = [0 for _ in pArray]

        allZeros = True

        for i in range(len(matrix[0])):
            for j in range(len(matrix[0])):
                pd[i] += matrix[j][i] * pArray[j] - matrix[i][j]
                    * pArray[i]

            pArray[i] += step * pd[i]

            if fabs(pd[i]) < EPS:
                if fabs(times[i]) < EPS:
                    times[i] = t
            else:
                allZeros = False

        if allZeros:
            break
```

```

        t += step

    return times, pArray

```

На листинге 2.2 представлен код для создания интерфейса.

Листинг 2.2 – Построение графиков для равномерного распределения

```

class MainWindow(QWidget):
    def __init__(self):
        super(MainWindow, self).__init__()
        self.ui = uic.loadUi("window.ui", self)
        self.ui.tableWidgetMatrix.horizontalHeader().
            setDefaultSectionSize(30)
        self.tableWidgetMatrix.itemChanged.connect(lambda x: self
            ._item_changed(x))
        self.spinBoxStatesCount.setValue(3)

    def _item_changed(self, value):
        try:
            if value.text() != "":
                float(value.text())
        except ValueError:
            QtWidgets.QMessageBox.critical(None, "Invalid input",
                "Please, enter a float number!")
            value.setText("")

    @pyqtSlot()
    def on_pushButtonSolve_clicked(self):
        table = self._get_matrix_from_table()
        self.ui.listWidgetSolution.clear()
        i = 1
        result = markov.get_system_times(table)
        if len(result) == 0:
            QtWidgets.QMessageBox.critical(None, "Invalid input",
                "Please, enter correct intensities!")
        else:
            for i in range(len(result[0])):
                QListWidgetItem("{n}: {time:0.5f} {p:0.5f}".
                    format(n = i + 1, time = round(result[0][i],
                        5), p = round(result[1][i], 5)), self.ui.
                    listWidgetSolution)

```

```

@pyqtSlot('int')
def on_spinBoxStatesCount_valueChanged(self, value):
    self.ui.tableWidgetMatrix.setRowCount(value)
    self.ui.tableWidgetMatrix.setColumnCount(value)
    self.ui.tableWidgetMatrix.clearContents()

def _get_matrix_from_table(self):
    res = []
    try:
        for i in range(self.ui.tableWidgetMatrix.rowCount()):
            row = []
            for j in range(self.ui.tableWidgetMatrix.
                columnCount()):
                item = self.ui.tableWidgetMatrix.item(i, j)
                val = item.text() if item and item.text() !=
                    "" else "0"
                row.append(float(val))
            res.append(row)
    except KeyError:
        print(res)
        QtWidgets.QMessageBox.critical(None, "Invalid input",
            "Please, enter a float number!")
    return res

```

Результат работы программы для системы из 3 состояний представлен на рисунке 2.1.

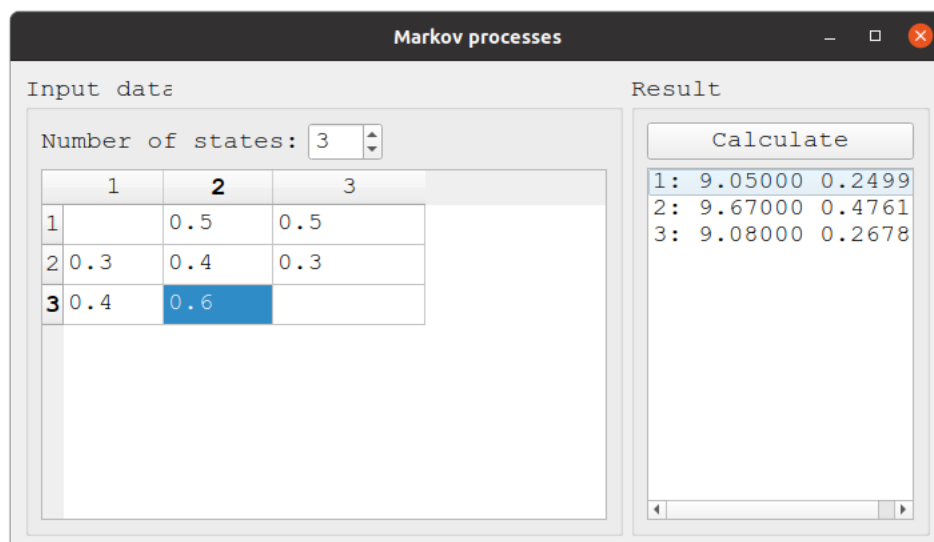


Рисунок 2.1 – Система из трех состояний

Здесь первый столбец отвечает за номер состояния, второй за время через, которое вероятность нахождения системы в этом состоянии стала стабильной, третий – эта вероятность.

Результат работы программы для системы из 5 состояний представлен на рисунке 2.2.

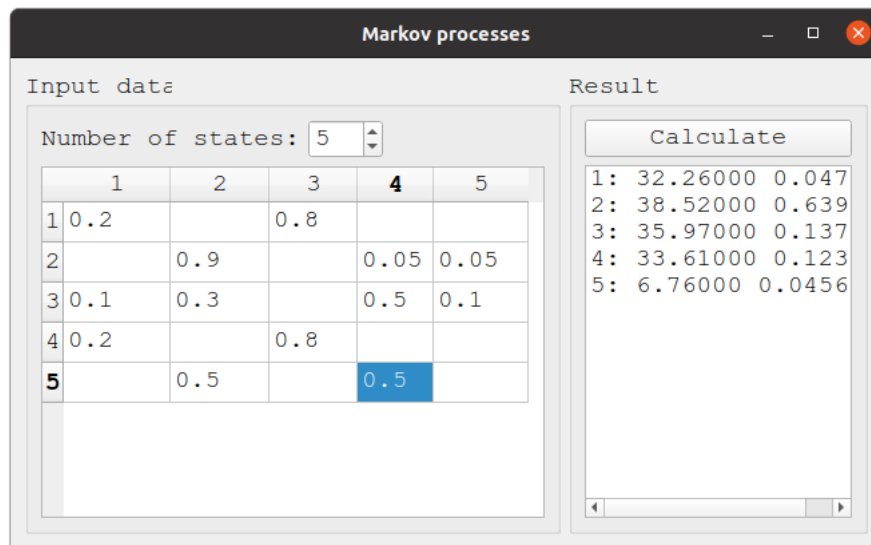


Рисунок 2.2 – Система из пяти состояний