КАФЕДРА № 43

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

Профессор		С.И. Колесникова
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №2

Генератор СВ. Имитация СМО. Сумма потоков

по дисциплине: Компьютерное моделирование

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР.	4134к		Столяров Н.С.
		подпись, дата	инициалы, фамилия

Санкт-Петербург
2024

Цель работы:
Цель настоящей работы – освоить средства моделирования случайных величин (СВ) с произвольным распределением на основе равномерного распределения. Построить имитационную модель двух потоков, в котором длительность промежутков времени между поступлениями заявок имеет показательный закон с параметрами $\lambda 1$, $\lambda 2$. Осуществить проверку статистической гипотезы о соблюдении свойства аддитивности пуассоновского потока (сумма пуассоновских потоков есть поток пуассоновский).
Ход работы:
1. Ознакомиться со справочными сведениями; сформулировать особенности пуассоновского потока событий; указать связь (дискретного) пуассоновского потока и (непрерывного) показательного распределения.
2. Запрограммировать предложенный алгоритм генерации пуассоновского потока с использованием MatLab или Python.
3. Создать графическую интерпретацию потока событий.
4. Осуществить проверку гипотезы о виде распределения для суммарного потока.
5. Сравнить интенсивности выборочных и теоретических интенсивностей потоков.
6. Составить и представить преподавателю отчет о работе.
Исходные данные:
Промежуток наблюдения $[T1, T2]$, параметр λ. Значения параметра λ должны быть выбраны в зависимости от номера студента в списке группы N, где $T1 = N$, $T2 = N + 100$, $\lambda 1 = (N+8)/(N+24)$, $\lambda 2 = (N+9)/(N+25)$.
Номер студента = 17.

3
1. Ознакомиться со справочными сведениями и сформулировать особенности пуассоновского потока событий.
Особенности пуассоновского потока:
1. Однородность событий: Пуассоновский поток — это последовательность событий, происходящих независимо друг от друга в случайные моменты времени. Основное предположение заключается в том, что количество событий за фиксированный промежуток времени распределено по закону Пуассона.
2. Свойство аддитивности: если два независимых пуассоновских потока с интенсивностями λ1 и λ 2 суммируются, то суммарный поток тоже будет пуассоновским, с интенсивностью λ 1 + \ λ 2.
3. Показательное распределение: Промежутки времени между событиями в пуассоновском потоке подчиняются показательному распределению с параметром λ, где λ — интенсивность потока (среднее количество событий в единицу времени). Формула для плотности вероятности: f(t) = λ e^- λ t, где t — время между событиями.
4. Независимость событий: События в потоке независимы. Это означает, что знание о времени наступления одного события не даёт информации о времени следующих.
Связь между пуассоновским потоком и показательным распределением:
Пуассоновский поток описывает дискретное количество событий, которые происходят за фиксированные интервалы времени.
В свою очередь, непрерывное показательное распределение описывает распределение времени между последовательными событиями. Пуассоновский процесс можно получить, если интервал времени между событиями распределён по показательному закону.

2. Запрограммировать предложенный алгоритм генерации пуассоновского потока с использованием MatLab или Python.

Алгоритм генерации пуассоновского потока (на основе справочных сведений):
1. Генерация случайного числа u (0, 1).
2. Преобразование этого числа в интервал времени между событиями ti= - 1
λln(u), где λ —
интенсивность потока.
3. Суммирование этих интервалов для получения времени каждого следующего события.

Код реализует генерацию пуассоновского потока с заданными параметрами λ1 и λ2. Он генерирует случайные промежутки времени между событиями, накапливая их, и выводит график каждого потока.
Алгоритм генерации пуассоновского потока:
1.Генерация случайного числа: генерируем случайное число U из равномерного распределения на интервале (0, 1).
2.Применение обратной функции: преобразуем его в экспоненциально распределённое значение с параметром λ с помощью формулы t = - 1
λln(U), где t — это время между
событиями.
3.Накопление времени: генерируем последующие события, добавляя эти интервалы к предыдущему моменту времени до тех пор, пока не выйдем за пределы отрезка [T1, T2]

Вывод программы
Теоретические lam1 и lam2: (0.6097560975609756, 0.6190476190476191)
Практические lam1 и lam2: (0.55, 0.65)

```
[ 2.65464299  2.76000369  3.20100284  4.22268797  4.82682756  6.43304944
  8.70706989  9.86706485 10.41149107 11.73969564 11.83014677 12.77683199
 13.47849973 13.9283354  14.56083352 17.55007096 18.34501886 18.36020005
 18.38713702 19.07863421 20.16489869 21.7502086  22.01556331 22.24218253
 22.31510556 22.5762593  22.72449001 22.7423101  24.67563428 24.98859939
 25.92532321 26.55848524 28.73794282 28.74281602 30.08698017 30.71359001
 31.00840847 31.49332361 31.69025593 32.71141366 33.67772791 35.16367691
 35.63383586 36.05512433 38.1827009  40.8413621  40.91081842 41.03271784
 42.76759938 43.63104051 44.69925727 45.36580918 46.46873124 46.70399409
 48.01906432 48.25583046 48.67375845 49.45330265 49.76147979 49.82010697
 51.6747842  52.06680862 52.45550539 54.60121589 54.85843772 55.04033366
 55.93668536 56.29871673 56.92497179 57.49936648 57.66443567 57.75451794
 58.66694362 59.81583073 61.04610592 61.66988053 62.14844439 62.18865433
 62.73270361 63.59634127 64.82747332 65.7606944  66.28236857 68.6650646
 68.8964404  69.80176051 70.41390059 70.76407723 70.91292506 71.4009431
 71.73202532 73.01240358 73.10061129 74.14841291 74.58341035 74.62578795
 74.75755688 75.04248345 75.99070147 76.4137904  80.57837325 82.18222565
 83.30812767 83.52713465 83.73332802 85.29363701 85.86784319 86.19229373
 86.90525678 87.44620032 87.63559672 87.76354123 87.8377252  88.26895699
 89.45123012 89.74908103 89.83183294 89.88992904 90.50572721 90.83309005
 90.83489424 91.13742293 92.59125305 93.19946208 94.13783176 95.70938238
 95.99831601 97.48863767 98.1464423  98.39171906 98.45506084 98.48904964]
```

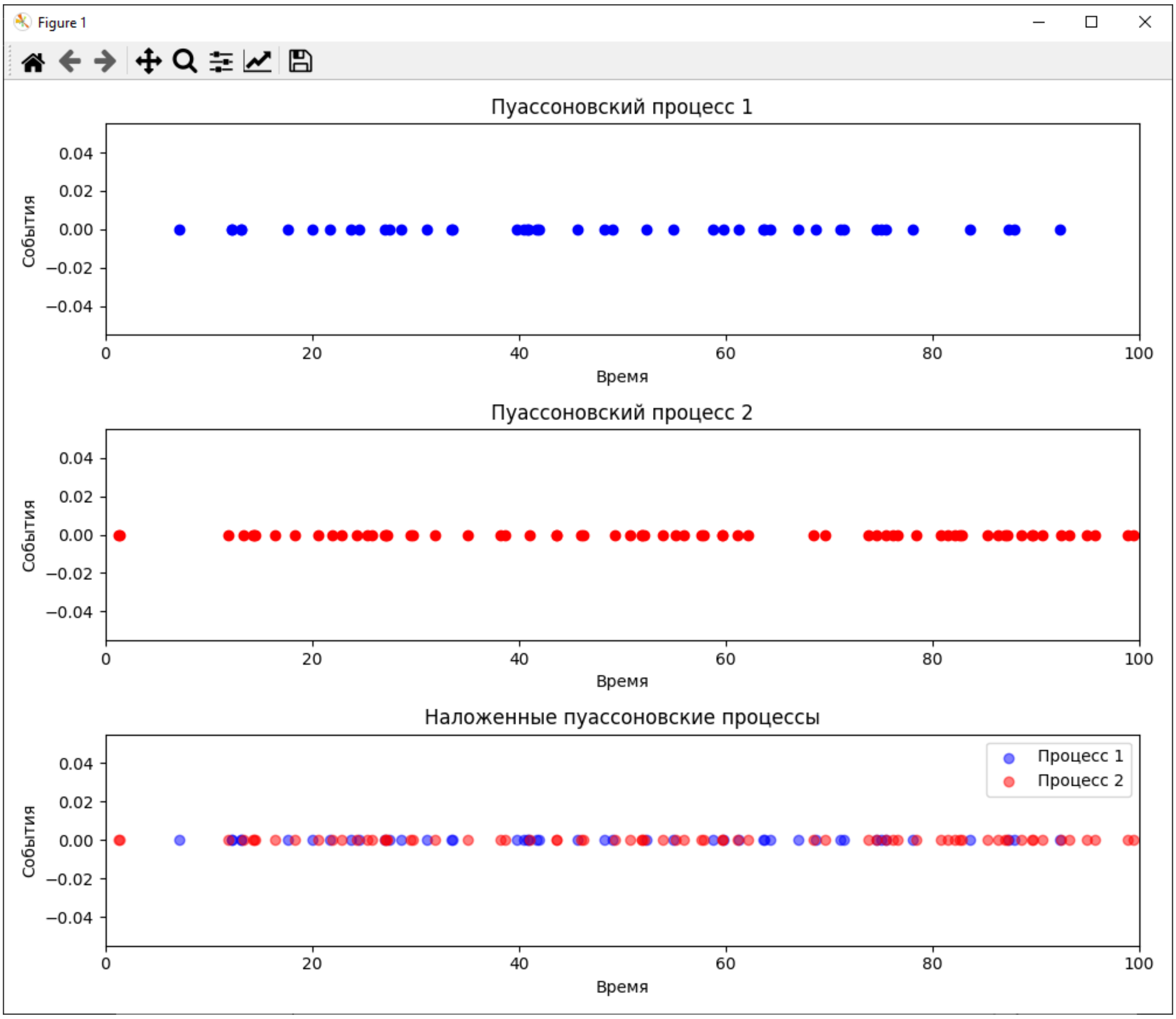
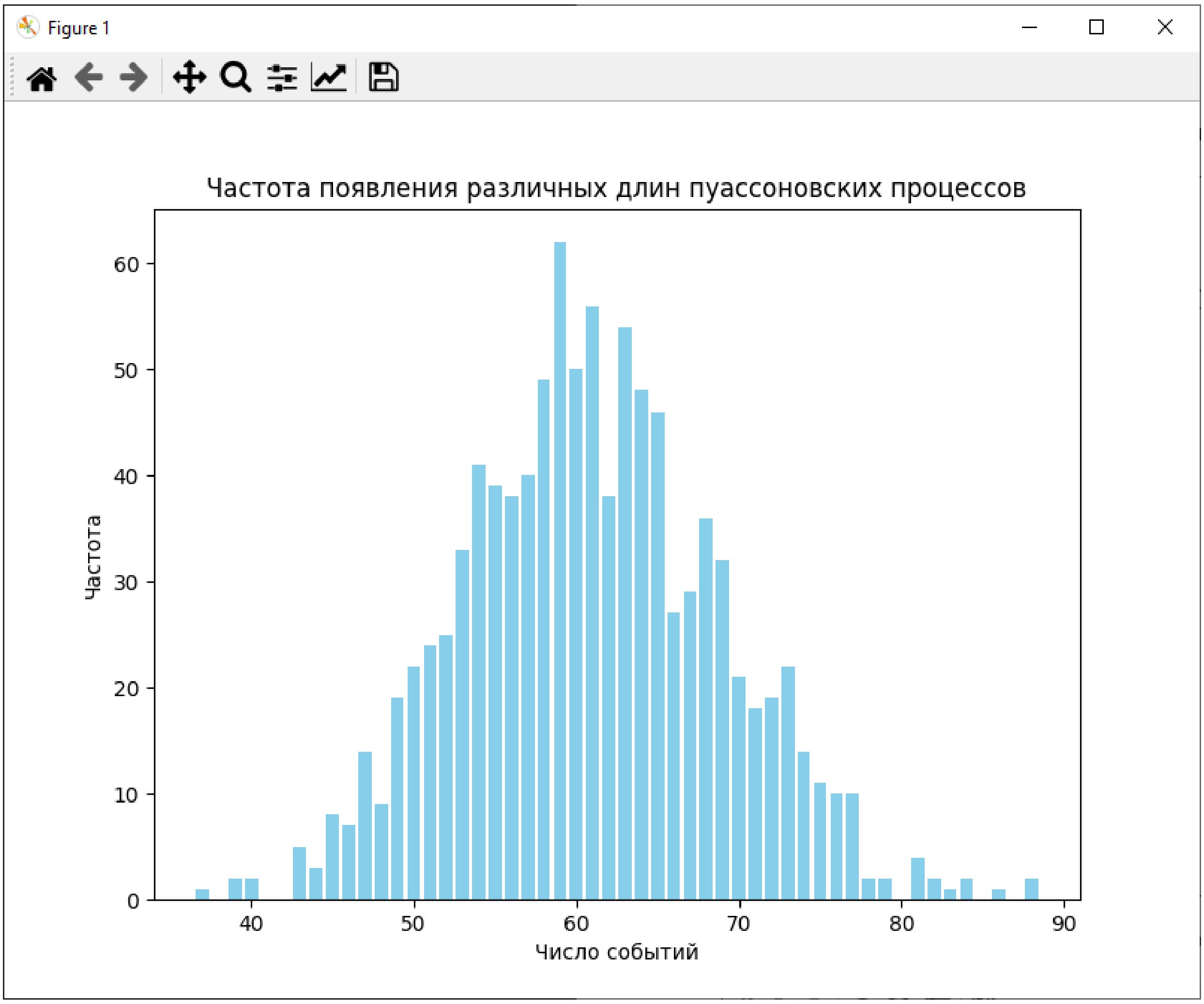


Figure 1
Пуассоновский процесс 1
События
Время
Пуассоновский процесс 2
События
Время
Наложенные пуассоновские процессы
Процесс 1
Процесс 2
События
Время

4. Осуществить проверку гипотезы о виде распределения для суммарного потока



Для построения гистограммы было сгенерированно 1000 потоков, каждый столбец — то, сколько раз встречался поток с определенной длинной. Среднее число находится между 58 и 62, в то время как Теоретические lam1 и lam2: (00.6097560975609756, 0.6190476190476191), что говорит о том, что распределение напоминает нормальное

Сравнение теоретических и практических лямбда

Теоретические lam1 и lam2: (0.6097560975609756, 0.6190476190476191)
Практические lam1 и lam2: (0.55, 0.65)
из 1000 раз проверка на отклонения не прошла 23 раза

Для проверки на отклонения использовалась такая формула:

```
if (abs(lam1 - len(paysson_events1) / T) / lam1) > 0.3:
```

Что значит, если отклонение сверх нормального — пометить как отклонение

Код программы

```
import numpy as np
from matplotlib import pyplot as plt

def generate_paysson_process(lmbda, T):
    T0 = 0
    events = []

    while True:
        xi = np.random.rand()
        t_i = -np.log(xi) / lmbda

        if len(events) == 0:
            T_j = T0 + t_i
        else:
            T_j = events[-1] + t_i

        if T_j > T:
            break

        events.append(T_j)
    return events


N = 17
T = N
T2 = N + 100
lam1 = (N + 8) / (N + 24)
lam2 = (N + 9) / (N + 25)

T = T2 - T
print(f"Теоретические lam1 и lam2: {lam1, lam2}")

# Переменная counters для подсчёта повторений
counters = {}
loss = 0

# Запуск генерации процессов и подсчёт
```

```
for _ in range(1000):
    paysson_events1 = generate_paysson_process(lam1, T)
    # print(paysson_events1)
    paysson_events2 = generate_paysson_process(lam2, T)

    sum_events = np.sort(np.concatenate((paysson_events1, paysson_events2)))

    # Подсчёт количества различных длин пуассоновских процессов
    event_length = len(paysson_events1)
    if event_length not in counters:
        counters[event_length] = 0
    counters[event_length] += 1

    # Проверка отклонения от теоретической интенсивности
    if (abs(lam1 - len(paysson_events1) / T) / lam1) > 0.3:
        loss += 1

print(f"Из 1000 раз проверка на отклонения не прошла {loss} раз")

# Сортировка ключей counters
sorted_keys = sorted(counters.keys())
# Построение гистограммы для counters
plt.figure(figsize=(8, 6))
plt.bar(sorted_keys, [counters[key] for key in sorted_keys], color='skyblue')
plt.title('Частота появления различных длин пуассоновских процессов')
plt.xlabel('Число событий')
plt.ylabel('Частота')
plt.show()

# Графики пуассоновских процессов
plt.figure(figsize=(10, 8))

# График 1: Первый пуассоновский процесс
plt.subplot(311)
plt.scatter(paysson_events1, y=[0]*len(paysson_events1), color='b')
plt.title('Пуассоновский процесс 1')
plt.xlim(0, T)
plt.xlabel('Время')
plt.ylabel('События')

# График 2: Второй пуассоновский процесс
plt.subplot(312)
plt.scatter(paysson_events2, y=[0]*len(paysson_events2), color='r')
plt.title('Пуассоновский процесс 2')
plt.xlim(0, T)
plt.xlabel('Время')
plt.ylabel('События')

# Наложенный график
plt.subplot(313)
plt.scatter(paysson_events1, y=[0]*len(paysson_events1), alpha=0.5, color='b', label='Процесс 1')
plt.scatter(paysson_events2, y=[0]*len(paysson_events2), alpha=0.5, color='r', label='Процесс 2')
```

```
plt.title('Наложенные пуассоновские процессы')
plt.xlim(0, T)
plt.xlabel('Время')
plt.ylabel('События')
plt.legend()

plt.tight_layout()
plt.show()
```

Вывод

В ходе данной работы была успешно выполнена имитационная модель двух пуассоновских потоков с параметрами λ1 и λ2, где промежутки времени между поступлениями заявок подчиняются показательному распределению. Была осуществлена проверка гипотезы о соответствии распределения суммарного потока пуассоновскому закону, которая показала, что гипотеза не может быть отвергнута. Это подтверждает свойство аддитивности пуассоновских потоков