МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА № 43

ОТЧЁТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| доц., канд. техн. наук | |  |  | | | |  | | С. И. Колесникова |
| должность, уч. степень, звание | |  | подпись, дата | | | |  | | инициалы, фамилия |
| ОТЧЁТ О ЛАБОРАТОРНОЙ РАБОТЕ №3 | | | | | | | | | | |
| Моделирование случайных величин с произвольным распределением на основе равномерного распределения | | | | | | | | | | |
| по дисциплине: КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ | | | | | | | | | | |
|  | | | | | | | | | | |
| РАБОТУ ВЫПОЛНИЛА | | | | | | | | | | |
| СТУДЕНТКА ГР. | 4931 | | |  | 23.10.2022 |  | | Е.Ю. Ильченко | | |
|  |  | | |  | подпись, дата |  | | инициалы, фамилия | | |
|  |  | | |  |  |  | |  | | |

Санкт-Петербург 2022

1. **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 λ1, λ2. Осуществить проверку статистической  
   гипотезы о соблюдении свойства аддитивности пуассоновского потока (сумма  
   пуассоновских потоков есть поток пуассоновский).
2. Ход работы
3. Ознакомиться со справочными сведениями; сформулировать особенности пуассоновского потока событий; указать связь (дискретного) пуассоновского потока и (непрерывного) показательного распределения.
4. Запрограммировать предложенный алгоритм генерации пуассоновского потока с использованием MatLab или Python.
5. Создать графическую интерпретацию потока событий.
6. Осуществить проверку гипотезы о виде распределения для суммарного потока.
7. Сравнить интенсивности выборочных и теоретических интенсивностей  
   потоков.
8. 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 = 11, 𝑇2 = 11 + 100 = 111, λ1 = 11+8/ 11+24 =19/35, λ2 = 11+9 /11+25 = 20/36

1. **Код программы**

import random  
import math  
import matplotlib.pyplot as plt  
import numpy as np  
import scipy.stats.distributions as sc  
  
#вар  
lmbd1 = 19 / 35  
lmbd2 = 20 / 36  
T1 = 11  
T2 = 111  
dt = (T2 - T1)/25  
x1 = []  
x2 = []  
Xnp = []  
X = []  
  
def get\_help\_table(x): #вспом табл,содержит кол-во значений,попавших в опред интервал  
 t = T1 + 4  
 help\_table = []  
 k = 0  
 while t <= T2:  
 help\_table.append([0] \* 50)  
 for i in range(len(x)):  
 for j in range(len(x[i])):  
 if (x[i][j] >= t - 4 and x[i][j] < t):  
 help\_table[k][i] += 1  
 k += 1  
 t += 4  
 return help\_table  
  
def get\_h1\_nl(help\_table):  
 h1 = [] #табл с уник значениями из вспомогательной табл  
 nl = [] #кол-во повторений уник значений  
  
 for i in range(len(help\_table)):  
 for j in range(len(help\_table[i])):  
 if (not h1.\_\_contains\_\_(help\_table[i][j])):  
 h1.append(help\_table[i][j])  
 nl.append(1)  
 else:  
 ind = h1.index(help\_table[i][j])  
 nl[ind] += 1  
 return h1, nl  
  
def get\_Mn(h1, nl): #мат ожидание  
 Mn = 0  
 for i in range(len(h1)):  
 Mn += h1[i] \* nl[i]  
 N = sum(nl)  
 Mn = Mn / N  
  
 return Mn  
  
def get\_Dn(h1, nl, Mn): #дисперсия  
 Dn = 0  
 for i in range(len(h1)):  
 Dn += ((h1[i] - Mn) \*\* 2) \* nl[i]  
 N = sum(nl)  
 Dn = Dn / N  
  
 return Dn  
  
def get\_yp(Mn):#интенсивность  
 return Mn/dt  
  
def get\_pp(yp,h1):#оценка теор вероятн по формуле пуассона  
 p\_p = []  
 for i in range(len(h1)):  
 p\_p\_cur = (np.power((yp \* dt), h1[i]) \* np.exp((-yp) \* dt) / math.factorial(h1[i]))  
 p\_p.append(p\_p\_cur)  
 return p\_p  
  
def get\_nl\_t(p\_p, N):#оценка теор частот  
 nl\_t = []  
 for i in range(len(p\_p)):  
 nl\_t\_cur = p\_p[i] \* N  
 nl\_t.append(nl\_t\_cur)  
 return nl\_t  
  
def get\_x2\_check(nl, nl\_t):#проверка гипотезы  
 x2\_p = 0  
 for i in range(len(nl)):  
 x2\_p\_cur = ((nl[i] - nl\_t[i]) \*\* 2) / nl\_t[i]#практ  
 x2\_p += x2\_p\_cur  
 x2\_t = sc.chi2.ppf(1-0.05, 23)  
  
 return x2\_p < x2\_t, x2\_p, x2\_t  
  
for k in range(50): #генер 50 реализ каждого из потоков  
 x1.append([])  
 x2.append([])  
 X.append([])  
 u1 = []  
 u2 = []  
 u3 = []  
 while True: #генер поток х1 на интс 1  
 eps = random.uniform(0,1)  
 u\_cur1 = -math.log(eps) / lmbd1  
 u1.append(u\_cur1)  
 t\_cur1 = T1 + sum(u1)  
 t\_cur1 = round(t\_cur1, 2)  
 if (t\_cur1 <= T2):  
 x1[k].append(t\_cur1)  
 else:  
 break  
  
 while True: #генер поток х2 на интс 2  
 eps = random.uniform(0,1)  
 u\_cur2 = -math.log(eps) / lmbd2  
 u2.append(u\_cur2)  
 t\_cur2 = T1 + sum(u2)  
 t\_cur2 = round(t\_cur2, 2)  
 if (t\_cur2 <= T2):  
 x2[k].append(t\_cur2)  
 else:  
 break  
  
 while True: #ген поток сумар интен  
 eps = random.uniform(0,1)  
 u\_cur3 = -math.log(eps) / (lmbd1 + lmbd2)  
 u3.append(u\_cur3)  
 t\_cur3 = T1 + sum(u3)  
 t\_cur3 = round(t\_cur3, 2)  
 if (t\_cur3 <= T2):  
 X[k].append(t\_cur3)  
 else:  
 break  
  
for k in range(50): #накладываем х1 на х2 друг на друга  
 Xnp.append([])  
 for i in range(len(x1[k])):  
 Xnp[k].append(x1[k][i])  
 for i in range(len(x2[k])):  
 Xnp[k].append(x2[k][i])  
 Xnp[k].sort()  
l = [x1, x2, Xnp, X] # поток Xnp (сложение) X поток с сумарной ин-тенсив (лямбда 1 и 2)  
  
# проверяем в цикле гипотезу для каждого потока  
for v in l:  
 help\_table\_v = get\_help\_table(v) # формир вспомаг таблицу  
 h1\_v, nl\_v = get\_h1\_nl(help\_table\_v) # форм табл уникальных знач для поиска мат ожидан  
 N\_v = sum(nl\_v) # сумма накоплен частот  
 Mn\_v = get\_Mn(h1\_v, nl\_v) # наход мат ожид  
 Dn\_v = get\_Dn(h1\_v, nl\_v, Mn\_v) # наход дисперсию  
 yp\_v = get\_yp(Mn\_v) # поиск интесив вывод из мат ожид(практик)  
 p\_p\_v = get\_pp(yp\_v, h1\_v) # оценка теор вероятн по форм Пуассона  
 nl\_t\_v = get\_nl\_t(p\_p\_v, N\_v) # оценка теор частот по варианту  
 res\_v, x2\_p\_v, x2\_t\_v = get\_x2\_check(nl\_v, nl\_t\_v) # проверка гипотезы о том что поток пуссоновский по хи квадрат  
  
 # принято/непринято значен хи2 крит значение хи 2(переменные)  
 if (v == x1): # вывод  
 print('X1 : ')  
 print(f'\tlmbd1 = {(lmbd1)} lmbd1^ = {yp\_v}')  
 if (v == x2):  
 print('x2 : ')  
 print(f'\tlmbd2 = {(lmbd2)} lmbd2^ = {yp\_v} ')  
 if (v == X):  
 print('X : ')  
 print(f'\tlmbd1\_2 = {(lmbd1 + lmbd2)} lmbd1\_2^ = {yp\_v} ')  
 if (v == Xnp):  
 print('Xnp : ')  
 print(f'\tlmbd1\_2 = {(lmbd1 + lmbd2)} lmbd1\_2^ = {yp\_v} ')  
  
 print('\tN = ' + str(N\_v))  
 print('\tМат. ожидание = ' + str(Mn\_v))  
 print('\tДисперсия = ' + str(Dn\_v))  
 print('\tX2\_Практ - ' + str(x2\_p\_v) + ' X2\_Крит - ' + str(x2\_t\_v))  
 print('\tЭто распределение Пуассона? - ' + str(res\_v))  
 print('\n')  
  
fig = plt.figure(1, figsize=(10, 10))  
  
plt.subplot(4, 1, 1)  
  
for k in range(50):  
 plt.plot([T1, T2], [k, k], linewidth=0.4, color='yellow')  
 for i in range(len(x1[k])):  
 plt.scatter(x1[k][i], k, color='blue', s=4, marker='o')  
plt.subplot(4,1,2)  
  
for k in range(50):  
 plt.plot([T1, T2], [k, k], linewidth=0.4, color='yellow')  
 for i in range(len(x2[k])):  
 plt.scatter(x2[k][i], k, color='violet', s=4, marker='o')  
  
plt.subplot(4,1,3)  
  
for k in range(50):  
 plt.plot([T1, T2], [k, k], linewidth=0.4, color='yellow')  
 for i in range(len(Xnp[k])):  
 plt.scatter(Xnp[k][i], k, color='blue', s=4, marker='o')  
  
plt.subplot(4,1,4)  
  
for k in range(50):  
 plt.plot([T1, T2], [k, k], linewidth=0.4, color='yellow')  
 for i in range(len(X[k])):  
 plt.scatter(X[k][i], k, color='darkviolet', s=4, marker='o')  
  
plt.show()  
  
def findVariants(thread):  
 variants = list()  
 delta = dt  
 start = T1  
 while start < T2:  
 for item in thread:  
 variants.append(0)  
 for i in range(0, len(item)):  
 if (item[i] >= start) and (item[i] < start + delta):  
 variants[-1] += 1  
 start += delta  
 return variants  
  
def getFreq(thread):  
 N = len(thread)  
 unique = list(set(thread))  
 L = len(unique)  
 frequency = list()  
 for i in range(0, L):  
 count = 0  
 for j in range(0, N):  
 if thread[j] == unique[i]:  
 count += 1  
 frequency.append(count)  
 # print(thread)  
 # print(variant)  
 return frequency  
  
def getTeor(thread, frequency):  
 N = len(thread)  
 unique = list(set(thread))  
 L = len(unique)  
  
 selectiveIntensivity = 0  
 for i in range(0, L):  
 selectiveIntensivity += unique[i] \* frequency[i]  
 selectiveIntensivity /= N \* dt  
 evaluation = list()  
 teoreticalEvaluation = list()  
 xi = 0  
 for i in range(0, L):  
 evaluation.append((selectiveIntensivity \* dt)\*\*unique[i] /  
 math.factorial(unique[i]) \* math.exp(-selectiveIntensivity \* dt))  
 teoreticalEvaluation.append(evaluation[i] \* N)  
  
 return teoreticalEvaluation  
  
  
def buildPlotFrequency(variants, freqs, teors):  
 fig = plt.figure(2, figsize=(8, 9))  
  
 for i in range(0, len(variants)):  
 plt.subplot(4,1,i+1)  
 plt.plot(variants[i], freqs[i], 'b')  
 plt.plot(variants[i], teors[i], 'k')  
 plt.grid()  
 plt.show()  
  
freqs = list()  
teors = list()  
variants = list()  
for i in l:  
 variant = findVariants(i)  
 variants.append(list(set(variant)))  
 freq = getFreq(variant)  
 freqs.append(freq)  
 teors.append(getTeor(variant, freq))  
  
buildPlotFrequency(variants, freqs, teors)

1. **Графическое представление**

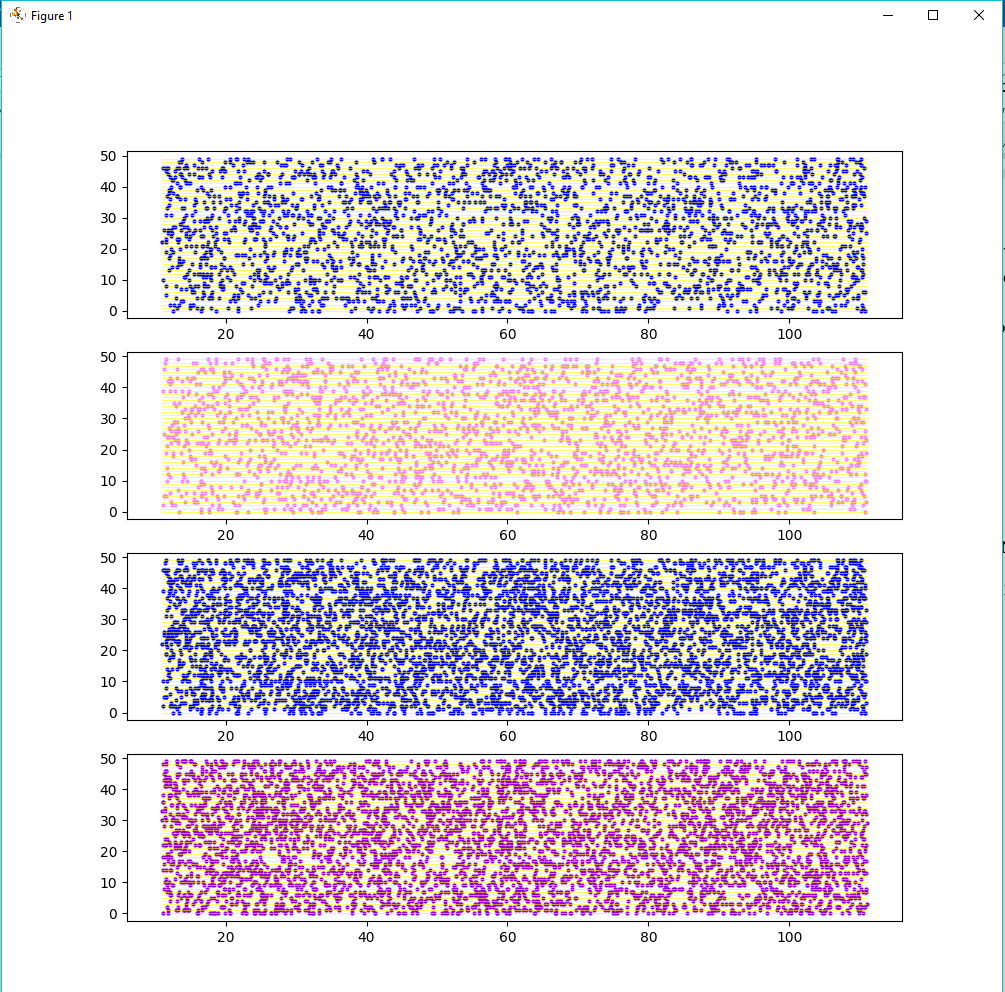


Рис. 1 – графическая интерпретация потоков событий

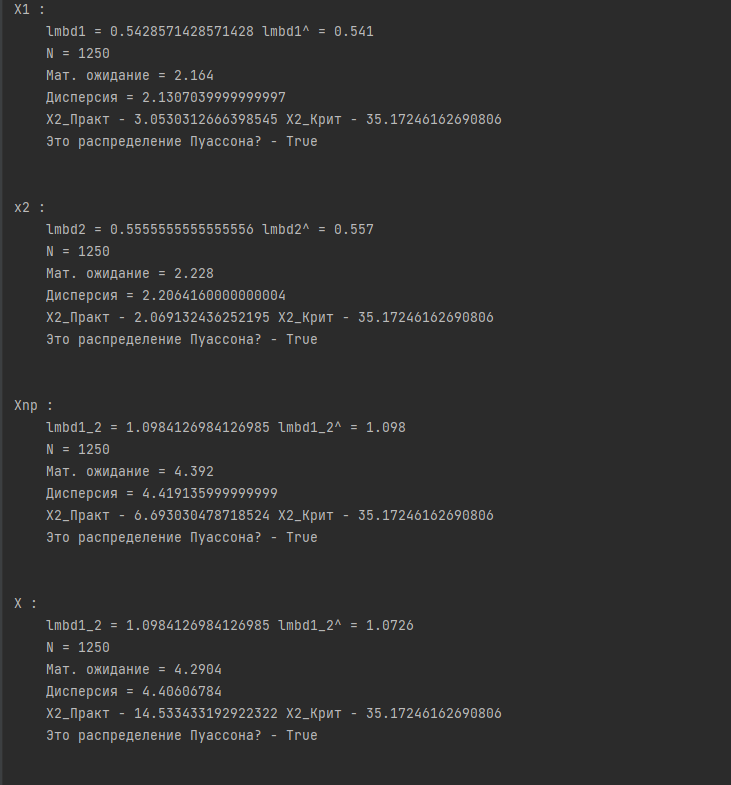


Рис.2 – Результат работы программы

1. **Сравним полученные значения**

|  |  |
| --- | --- |
| = 0.5436=0.544 | λ1 = 0.5428571428571428 = 0.543 |
| = 0.539 | λ2 = 0.5555555555555556 = 0.556 |
| ((Xпр(t)) = 1.0826 | λ((X(t)) = λ1 + λ2 = 1.0984126984126985= 1.098 |

Таблица 1 – Сравнение 6.1

|  |  |  |
| --- | --- | --- |
| ((X(t)) = 1.1154 = 1.115 | ((Xпр(t)) = 1.0826 | (𝑋1(t) + 𝑋2(t)) = = 1.099 |

Таблица 2 – Сравнение 6.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | X1(t) | X2(t) | X(t) | Xпр(t) |
| Mη | 2.1744 | 2.156 | 4.3304 | 4.4616 |
| Dη | 2.10718464 | 2.198864 | 4.240435839999999 | 4.37652544 |

Таблица 3– Сравнение 6.3

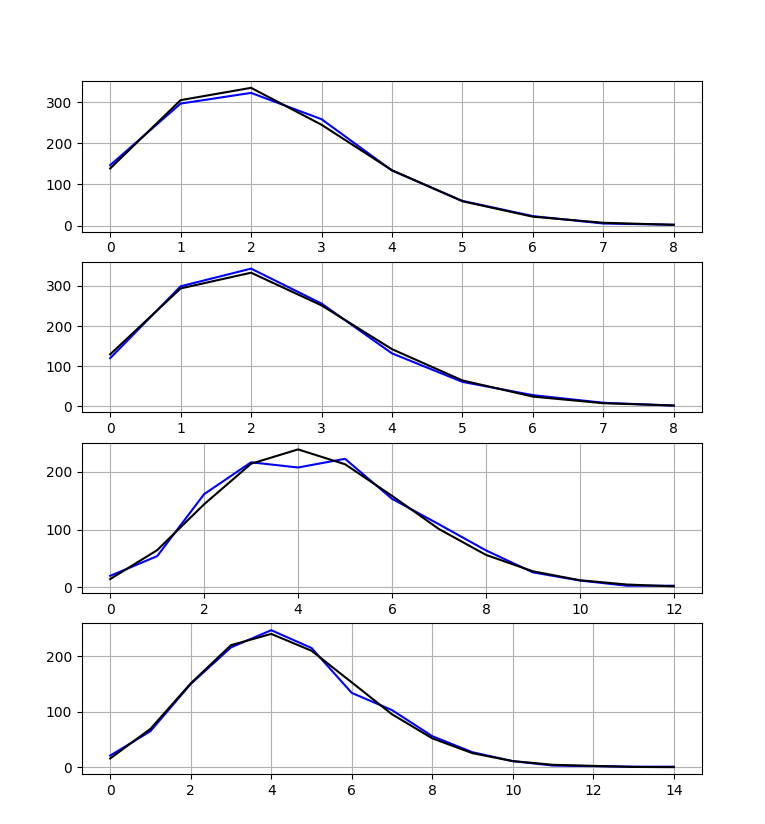
Полученные данные подтверждают статистическую гипотезу о соблюдении свойства аддитивности пуассоновского потока (сумма пуассоновских потоков есть поток пуассоновский).

**Зависимость практической и теоретической частоты от вариантов для каждого из потоков**

Ниже представлены графики зависимости практический и теоретических частот от вариантов для каждого из потоков, где:

* Синим – показана практическая частота;
* Черным – показана теоретическая частота.

для потоков ,,, соответственно:



**Проверка гипотезы о «пуассоновости» потока для каждого потока**

Ниже представлена таблица сравнений значений квантиля хи квадрат при уровне значимости α = 0.05 (1 – α = 0.95) и числе степеней свободы n = 23:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | Сравнение | Вывод |
| *i = 1* | 35.1725 | 2.4658 |  | гипотеза о «пуассоновости» потока **не** отвергается |
| *i = 2* | 35.1725 | 2.8778 |  | гипотеза о «пуассоновости» потока **не** отвергается |
| *i = 3* | 35.1725 | 14.0294 |  | гипотеза о «пуассоновости» потока **не** отвергается |
| *i = 4* | 35.1725 | 10.6274 |  | гипотеза о «пуассоновости» потока **не** отвергается |