ГУАП

КАФЕДРА № 41

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ассистент |  |  |  | М.Н. Шелест |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ №3 |
| МОДЕЛИРОВАНИЕ ВХОДНОГО ПОТОКА ЗАПРОСОВ |
| по курсу: ПОСТРОЕНИЕ И АНАЛИЗ ГРАФОВЫХ МОДЕЛЕЙ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4917 |  |  |  | В.С. Тихонов |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург

2022

Цель работы: изучение алгоритмов получения на ЭВМ чисел с заданным законом распределения и построения гистограмм.

Вариант 20

|  |  |
| --- | --- |
| Порядок эрланговского потока | Параметр λ |
| 4 | 20 |

Ход работы:

Плотность распределения Эрланга имеет вид (Рисунок 1):

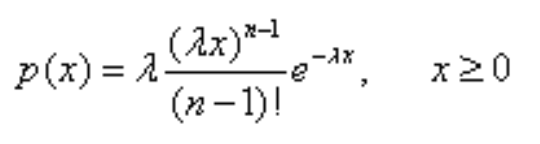


Рисунок 1 - Плотность распределения Эрланга

Графики функции и плотности распределения (Рисунки 2-3):

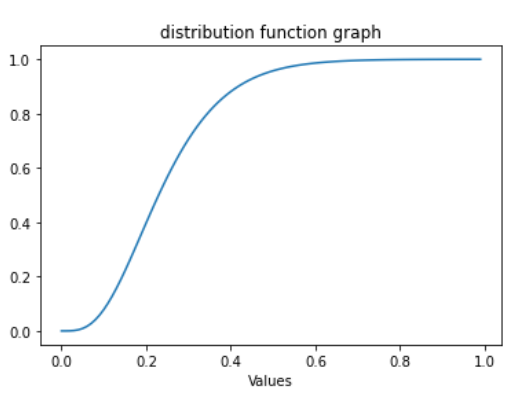


Рисунок 2 – График функции распределения

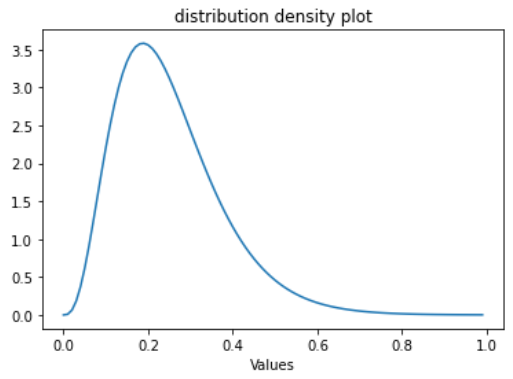


Рисунок 3 – График плотности распределения

Расчет теоретического значения интенсивности λ и вариации νu.

Напишем программу, реализующую методику оценки интенсивности потока, и с её помощью произведём оценку интенсивности и коэффициента вариации заданного потока (Рисунок 5):



Рисунок 5 – Результат работы алгоритма моделирования

Построим графики зависимости оценок и от величины N (рис. 6, 7):

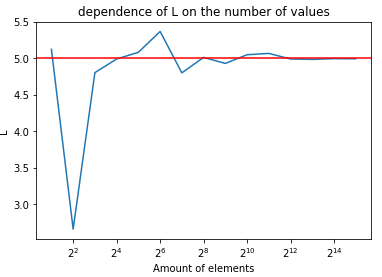


Рисунок 6 - График зависимости оценки от величины N

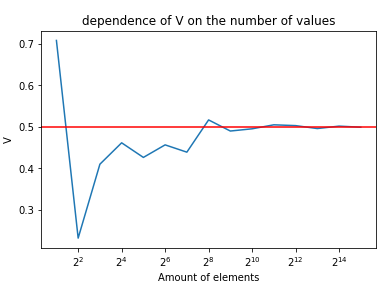


Рисунок 7 - График зависимости оценки от величины N

Вывод: в ходе выполнения лабораторной работы были изучены методы исследования основных характеристик входных потоков заявок, а также базовые принципов моделирования СМО по событиям. Была написана программа, реализующая методику оценки интенсивности потока и на е основе построены графики, демонстрирующие увеличение точности вычисления оценок λ ̂ и (ν\_u ) ̂ при увеличении значения N.

Листинг 1 – Код программы

|  |
| --- |
| import numpy, random, math  import matplotlib.pyplot as plt  # Генерация значений распределения  def erlang\_spec(k, L):  values = []  for i in range(k):  values.append(-1 \* numpy.log(random.uniform(0, 1)) / L)  return sum(values)  def flow\_intensity\_estimate(N):  N\_List = []  L\_List = []  V\_List = []  t = 0 # Время моделирования  k = 0 # Счётчик  Lo = 9999999999 # Старая величина интенсивности L  Vo = 9999999999 # Старая величина вариации V  while True:  u\_list = [] # Список значений Uk  while k<N:  k = k+1  Uk = erlang\_spec(4, 20) # Случайное число, распределённое по заданному закону  t = t + Uk  u\_list.append(Uk)  Mu = t/N # Мат. ожидание  sum\_in\_SKO = 0  for u in u\_list:  sum\_in\_SKO += (u - Mu) \*\* 2  SKO = math.sqrt((1 / (N - 1)) \* sum\_in\_SKO) # Среднеквадратическое отклонение  Ln = 1/Mu  Vn = SKO/Mu  N\_List.append(N)  L\_List.append(Ln)  V\_List.append(Vn)  if (abs((Ln - Lo) / Lo) > 0.01) or (abs((Vn - Vo) / Vo) > 0.01):  Lo = Ln # Новая величина L  Vo = Vn # Новая величина V  t = 0  k = 0  N = 2\*N  else:  return Ln, Vn, N\_List, L\_List, V\_List  L, N, N\_List, L\_List, V\_List = flow\_intensity\_estimate(10000)  print(L, N)  L, N, N\_List, L\_List, V\_List = flow\_intensity\_estimate(2)  plt.plot(N\_List, L\_List)  plt.xscale('log', base=2)  plt.axhline(y=5, color='red')  plt.title('dependence of L on the number of values')  plt.xlabel('Amount of elements')  plt.ylabel('L')  plt.show()  plt.plot(N\_List, V\_List)  plt.xscale('log', base=2)  plt.axhline(y=0.5, color='red')  plt.title('dependence of V on the number of values')  plt.xlabel('Amount of elements')  plt.ylabel('V')  plt.show() |