МИНИСТЕРСТВО НАУКИ и ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение   
высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных систем и технологий

Кафедра Вычислительная техника

Дисциплина Моделирование

**КУРСОВОЙ ПРОЕКТ (РАБОТА)**

Тема Моделирование процессов в вычислительных системах. Вариант 19.

Выполнил студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Биткузин Д. Р. /

подпись инициалы, фамилия

Курс 3 Группа ДВТАСбзу-31

Направление/ специальность 09.03.01 «Информатика и вычислительная техника»

Руководитель к.т.н., доцент кафедры «Вычислительная техника»

должность, ученая степень, ученое звание

Валюх Вероника Валерьевна

фамилия, имя, отчество

Дата сдачи:

«\_16\_\_»\_\_\_\_\_Января\_\_\_\_\_2023\_\_г.

Дата защиты:

«\_26\_\_»\_\_\_\_\_Января\_\_\_\_\_2023\_\_г.

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ульяновск

2023 г.

МИНИСТЕРСТВО НАУКИ и ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ   
федеральное государственное бюджетное образовательное учреждение   
высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных систем и технологий

Кафедра Вычислительная техника

Дисциплина Моделирование

**ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ (РАБОТУ)**

Студенту ДВТАСбзу-31 Биткузин Д. Р

группа фамилия, инициалы

Тема проекта (работы) Моделирование процессов в вычислительных системах. Вариант 19.

Срок сдачи законченного проекта (работы) «\_16\_» \_\_Января\_\_\_\_\_\_\_\_\_\_2023\_\_г.

Исходные данные к проекту (работе) материалы лекций

Разработка концептуальной модели. Моделирование потока сообщений, заявок и характеристик на языке программирования Python. Расчёт характеристик стохастических сетевых моделей. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Содержание пояснительной записки (перечень подлежащих разработке вопросов) курсовая работа состоит из следующих пунктов: Линейный конгруэнтный метод генератора случайных чисел, Генерация случайных величин по закону Пуассона, моделирование источников сообщений в вычислительных системах, стохастические сетевые модели вычислительных систем.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Перечень графического материала (с точным указанием обязательных чертежей)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель \_\_\_\_ к.т.н \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_В.В.Валюх\_\_\_\_\_\_\_/

должность подпись инициалы, фамилия

«\_\_\_\_\_»\_\_январь\_\_\_\_\_\_\_\_\_\_2023\_\_г

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **/\_\_**Д.Р.\_\_Биткузин**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/**

подпись инициалы, фамилия

«\_16\_\_\_»\_\_январь\_\_\_\_\_\_\_\_\_\_2023\_\_г

МИНИСТЕРСТВО НАУКИ и ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ   
федеральное государственное бюджетное образовательное учреждение   
высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

**ОТЗЫВ  
руководителя на курсовой проект (работу)**

студента Биткузина Динислама Ренатовича

фамилия, имя и отчество

Факультет ИСТ группа ДВТАСбзу-31 курс 3

Дисциплина Моделирование

Тема проекта (работы) Моделирование источников сообщений в вычислительных системах

Моделирование потока.

Отмечаются следующие моменты: актуальность темы исследования; соответствие содержания и структуры курсовой работы ее теме; степень разработанности проблемы, наиболее интересно исследованные вопросы. Оценивается степень самостоятельности и инициативы студента; умение пользоваться различными источниками информации; уровень его теоретической подготовки; умение анализировать научные материалы, делать практические выводы; знание основных концепций, научной и специальной литературы по избранной теме. Содержится оценка проекта (работы) руководителем.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель к.т.н\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_/ В.В.\_Валюх

должность, учёная степень, ученое звание подпись инициалы, фамилия

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_\_г.

## Введение

Часто приходится сталкиваться с системами, предназначенными для многоразового использования при решении однотипных задач. Возникающие при этом системы получили название – системы массового обслуживания. Примеры таких систем можно встретить повсюду.

С помощью средств имитационного моделирования возможно имитировать процесс, проходящий в СМО.

Целью данной работы является – разработка программы с помощью языка программирования моделирующий процесс работы обслуживающих систем.

Моделирование является одним из широко используемых методов анализа процессов, а также проведения эксперимента на модели с целью анализа поведения системы. Так как зачастую на реальном объекте весьма затруднительно продумать и рассчитать все процессы.

Формулировка проблемы

Целью данного курсового проекта является программирование конгруэнтного метода случайных чисел и по закону Пуассона.

Моделирование потока сообщений, заявок и характеристик.

Рассчитать характеристики стохастических сетевых моделей вычислительных систем.

### Задачи

* Запрограммировать линейный конгруэнтный метод генератора случайных чисел и выполнить для него задание.
* Запрограммировать генерацию случайных величин по заданному закону распределения Пуассона, определяемую вариантом задания.
* Смоделировать поток сообщений (заявок) в вычислительной системе. Изучить виды и характеристики потоков заявок. Решить поставленные задачи.
* Рассчитать основные характеристики и построить структурную схему разомкнутой стохастической сети.

1. **Конгруэнтный метод.**

Для линейного конгруэнтного метода мы будем использовать числа A = 1103515245, C = 12345. Эти числа по умолчанию входят в некоторые методы генерации чисел и были выбраны с помощью статистических тестов NIST и протестированы так как они располагают к равномерному распределению.

Сгенерируем числа рис 1.

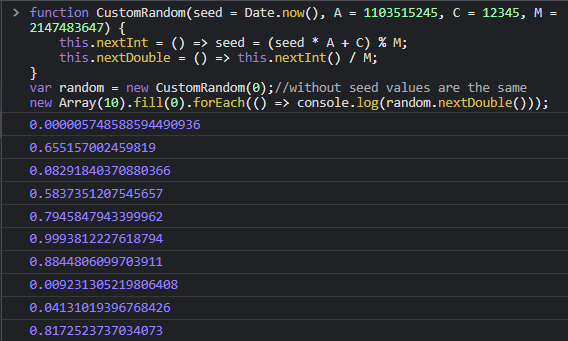


Рис 1. - Генерация псевдослучайных чисел Линейным конгруэнтным методов

Получим числа существующим методом в javascript рис 2.

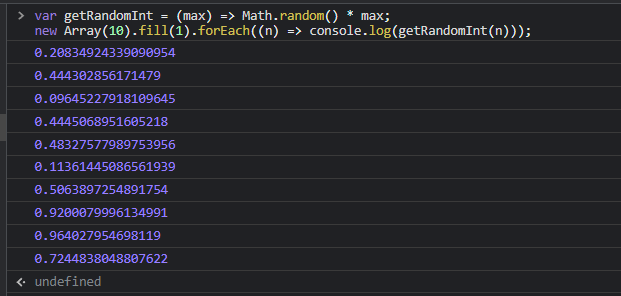


Рис 2. - Генерация чисел методом javascript.

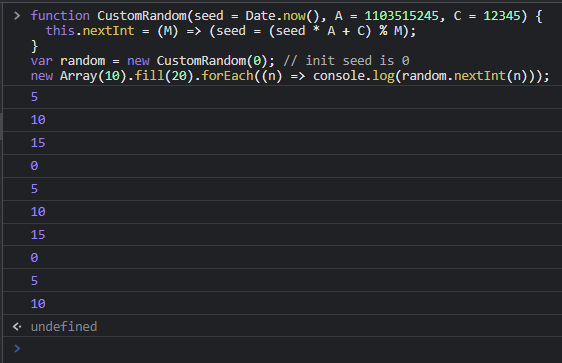
Так же мы можем получить линейную конгруэнтной последовательность рис 3.  


Рис. 3- линейная конгруэнтная последовательность

Сравним данные нашего метода и метода javascript рис 4.

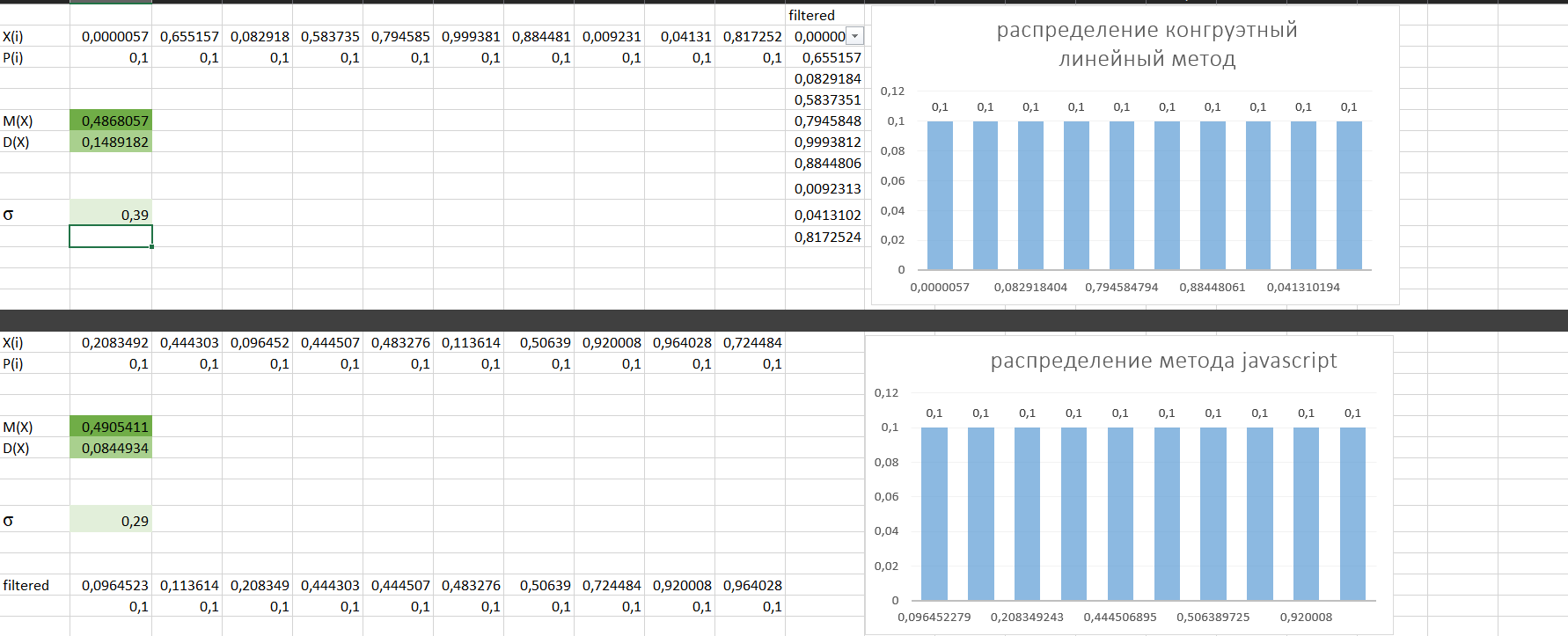
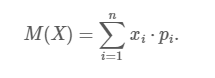
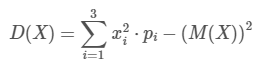


Рис 4. – Распределение чисел генерации конгруэнтного метода и метода javascript

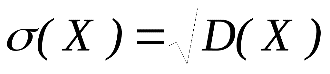
Математическое ожидание М(X) высчитываем по формуле:



дисперсия D(X) Используем формулу



Среднеквадратичное отклонение используя формулу



Сгенерируем 1000 чисел конгруэтным методом рис 5.

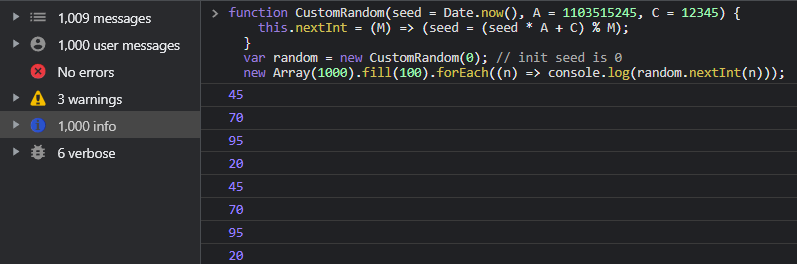


Рис 5 – Генерация конгруэнтным методом

Сгенерируем 1000 чисел методом javascript рис 6.  


Рис 6 – генерация методом javascript

Сравним данные рис.7

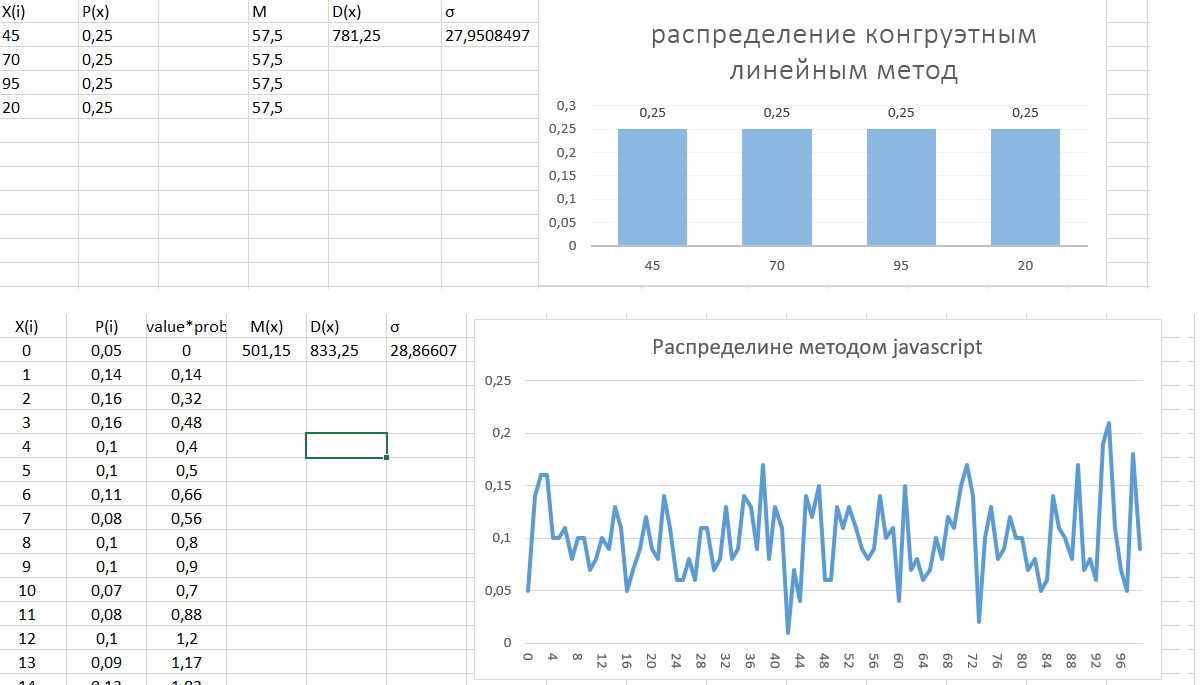


Рис. 7 - Распределение чисел генерации конгруэнтным метода и метода javascript.

Сгенерируем 10000 чисел конгруэнтным методом рис. 8

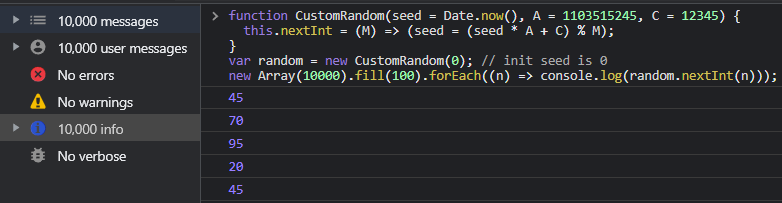


Рис 8 – генерация чисел конгруэнтным методом.

Сгенерируем 10000 чисел методом javascript рис 9

  
Рис 9 – Генерация методом javascript.

Сравним данные на рис. 10



Рис. 10 - Распределение чисел генерации конгруэнтным метода и метода javascript.

Для оценки равномерности генератора случайных чисел выполнить расчет математического ожидания Mi для i последовательностей из 1000 случайных чисел (i=1,2,..., 10) и для i последовательностей случайных чисел переменной длины (длина  i-ой  последовательности задается как i\*1000;  i = 1, 2, ..., 10). Построить графики зависимости разности (М-Мi) от номера последовательности i, где М - теоретическое математическое ожидание равномерного распределения случайных чисел, Мi - расчетное математическое ожидание для i-й последовательности случайных чисел, полученных от генератора. По данным результатам определить Р{|М-Мi|<σ} - вероятность того, что отклонения расчетного математического ожидания от теоретического не превышают величину теоретического среднеквадратичного отклонения.

Построим 1000 случайных чисел при нарастающим i.

Сгенерируем эти числа рис 11.

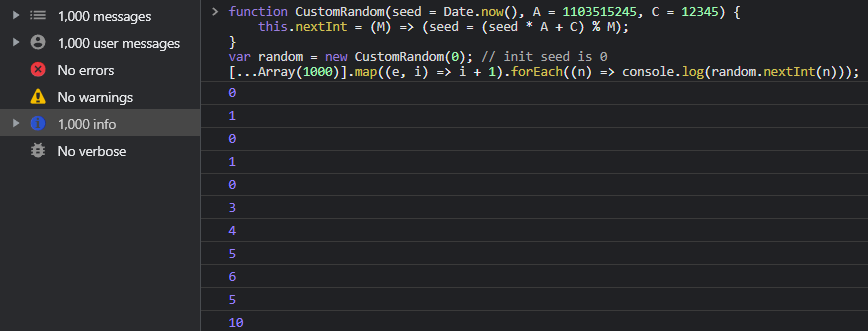


Рис. 11. Генерация 1000 чисел с шагом i = 1

График для последовательности с шагом i = 1, 2…1000 показан на рис 12.

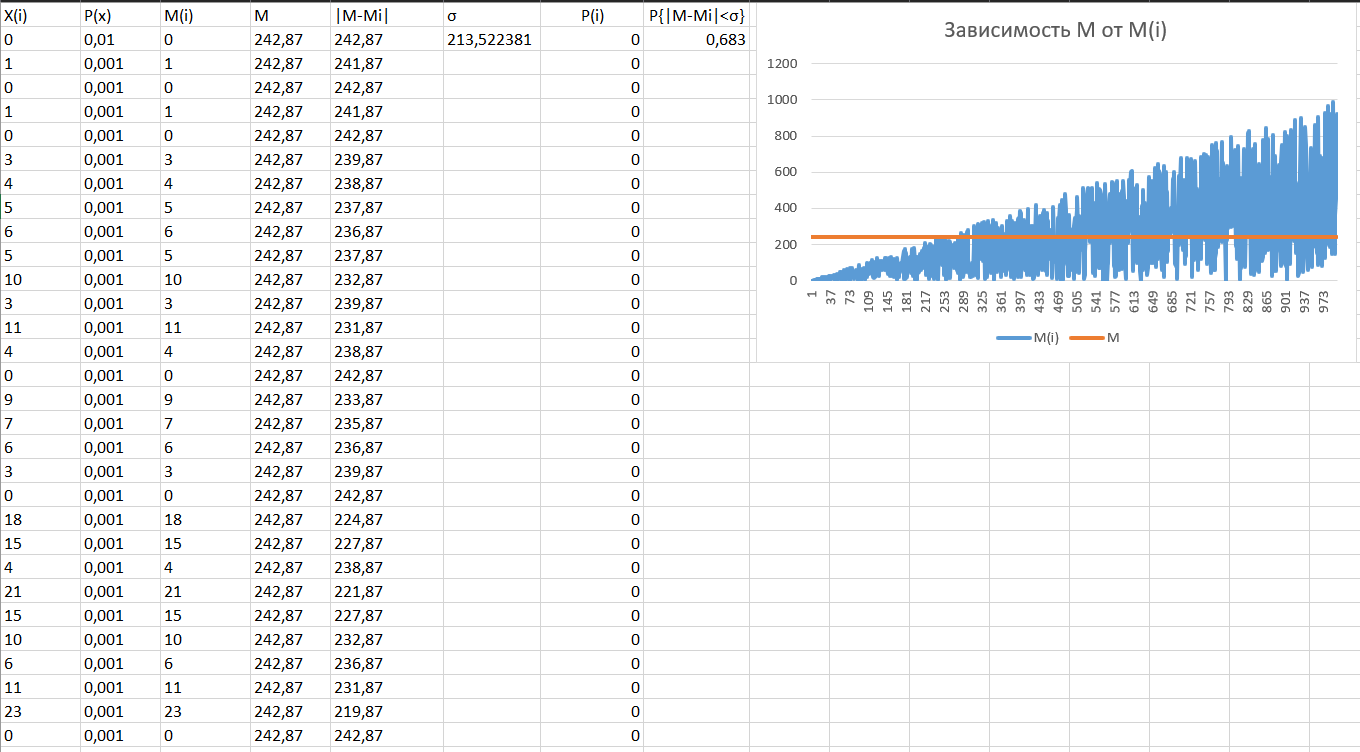


Рисунок 12. Зависимость M от M(i)

Для i = 1, 2…1000 при i\*1000;

Сгенерируем 1000 чисел с шагом i \* 1000 рис. 13.

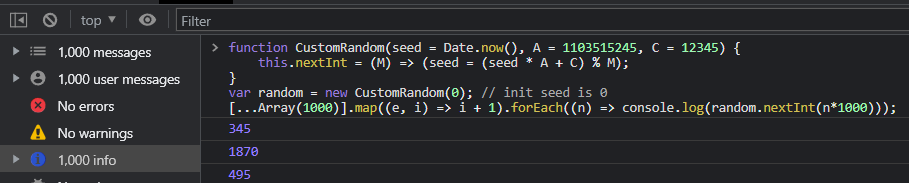


Рис. 13. Генерация 1000 чисел с шагом i= 1000

Построим график рис. 14.

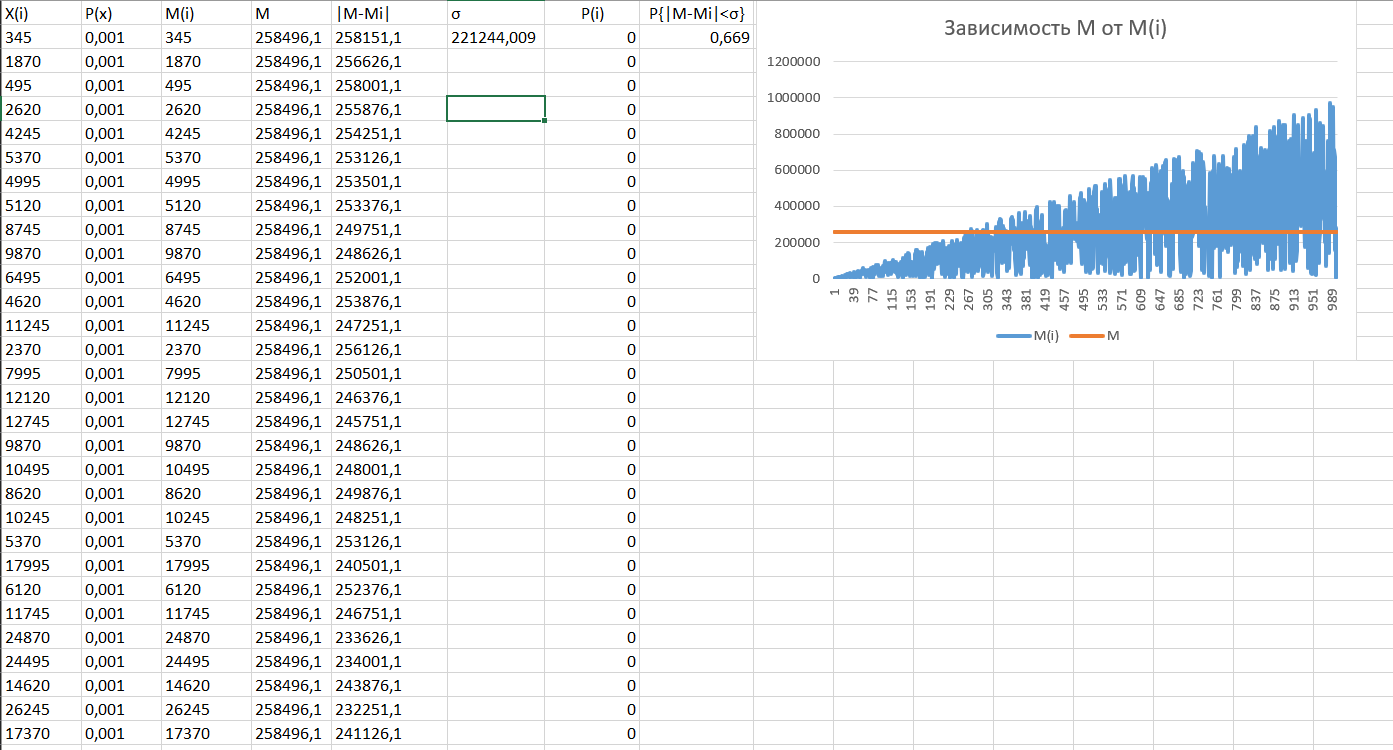


Рис. 14. Зависимость M от M(i) с шагом 1000

## Генерация по закону Пуассона

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер варианта | Вид закона распределения | Характеристики | |  |
|  |  | Мат. Ожидание | Max | p |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 19 | З-н Пуассона | 28 | 84 |  |

Запрограммируем данный вариант с параметрами max = 84, M(x) = 28

Выполним данную программу в python, подключим библиотеки, которые предоставляют функции расчёта.

Используем сид для воспроизведения результата.

Распределим 100 чисел между 0 и 84 и вызовем функции для расчета.

Подключим библиотеку matplotlib для визуализации данных. Библиотеки scipy и numpy для выполнения расчетов. Подробнее на рис 15 и рис 16.

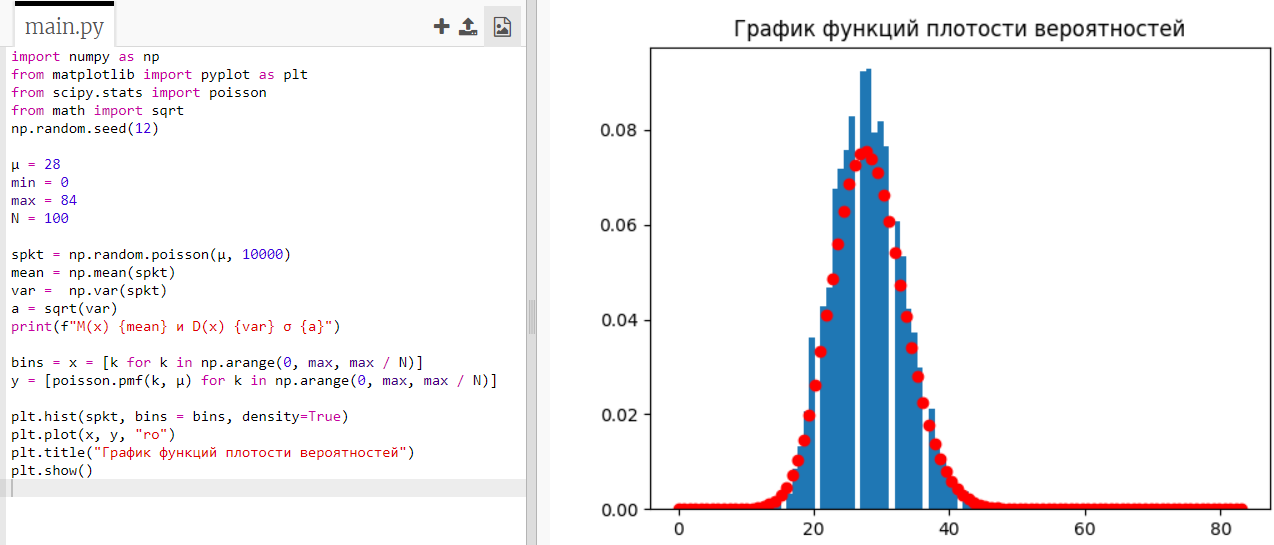


Рисунок 15. График функции f(x) p(x)

Получаем над графиков все 3 характеристики

Математическое ожидание - M(x) 27.9649

Дисперсия - D(x) 27.817067989999998

Отклонение - σ 5.274188846637935

Так как M(x) - 28 то и большинство сгенерированных чисел появились в этой последовательности. Случайная последовательность сгенерировалась качественно.

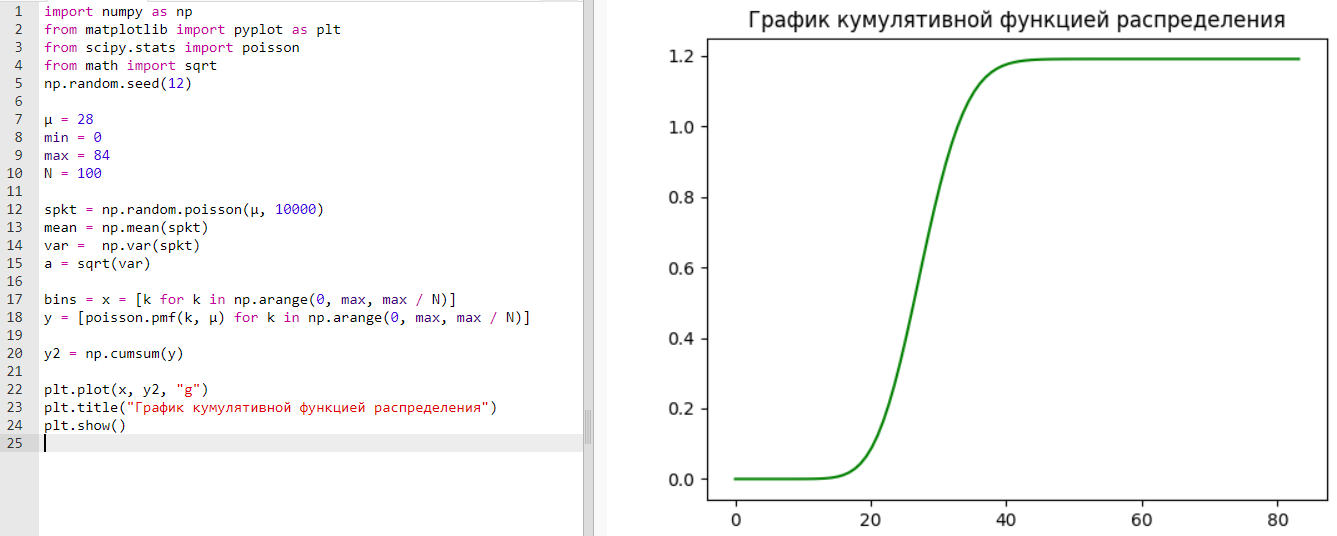


Рисунок 16. График кумулятивной функции распределения F(x)

1. **Моделирование заявок.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Номер варианта** | **З-н распределения заявок** | | **З-н распределения заявок** | | | | |
| **Тип** | **P(i)** | **1** | **2** | **3** | **4** | **5** |
| **19** | **1**  **2**  **3**  **4** | **0,53**  **0,09**  **0,02**  **0,36** | **0,48**  **0,07**  **0,12**  **0,02** | **0,28**  **0,22**  **0,27**  **0,72** | **0,09**  **0,41**  **0,07**  **0,01** | **0**  **0,17**  **0,39**  **0,07** | **0,15**  **0,13**  **0,15**  **0,18** |

|  |  |  |
| --- | --- | --- |
| **Номер варианта** | **Характеристики** | |
| **A** | **B** |
| **19** | **21** | **237** |

Рис. 1 - Характеристики потока.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Номер варианта** | **Вид закона распределения** | **Характеристики** | | | | | | |
| **Средний промежуток времени** | **Коэффициент Зорланга** | **Дисперсия** | **a** | **b** | **m** | **l** |
| **19** | **4** |  |  |  | **3** | **0,9** |  |  |

На рис. 17 показана таблица полученных сообщений. В ней отражена информация о типе и длине сообщения, адресе абонента, времени поступления сообщения

На рис. 17 показана таблица полученных сообщений. В ней отражена информация о типе и длине сообщения, адресе абонента, времени поступления сообщения

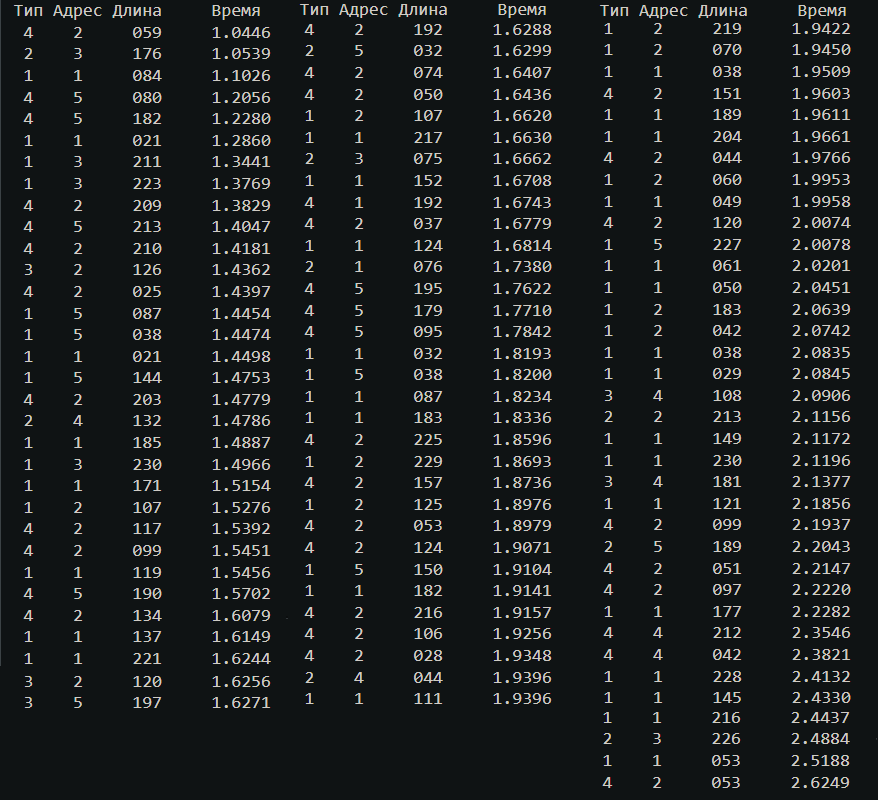


Рис. 17 - Поток сообщений.

На рис. 18 показана таблица вероятностей появления заявок.

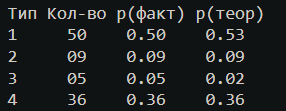


Рис. 18 - Сравнение вероятностей.

На рис. 19 показана таблица длин типов заявок.

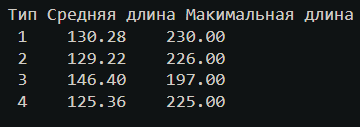


Рис. 19 - Сравнение средних длин.

На рис. 20 показана таблица средних частот заявок каждого типа.

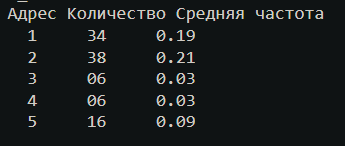


Рис. 20 - Сравнение средних частот.

Объединим полученные данные в таблицу 1.

Таблица 1 - Данные о сообщениях потока.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Тип сообщения | Характеристики | Номер абонентского аппарата | | | | |
| 1 | 2 | 3 | 4 | 5 |
| 1 | Число заявок | 32 | 9 | 3 | 0 | 6 |
| Вероятность | 0.64 | 0.18 | 0.06 | 0 | 0.12 |
| Теор. вероятность | 0.48 | 0.28 | 0.09 | 0 | 0.15 |
| 2 | Число заявок | 1 | 1 | 3 | 2 | 2 |
| Вероятность | 0.11 | 0.11 | 0.33 | 0.22 | 0.22 |
| Теор. вероятность | 0.07 | 0.22 | 0.41 | 0.17 | 0.13 |
| 3 | Число заявок | 0 | 2 | 0 | 2 | 1 |
| Вероятность | 0 | 0.4 | 0 | 0.4 | 0.2 |
| Теор. вероятность | 0.12 | 0.27 | 0.07 | 0.39 | 0.15 |
| 4 | Число заявок | 1 | 26 | 0 | 2 | 7 |
| Вероятность | 0.03 | 0.72 | 0 | 0.05 | 0.2 |
| Теор. вероятность | 0.02 | 0.72 | 0.01 | 0.07 | 0.18 |

На рис. 21 показаны числовые характеристики каждого из типов сообщений.

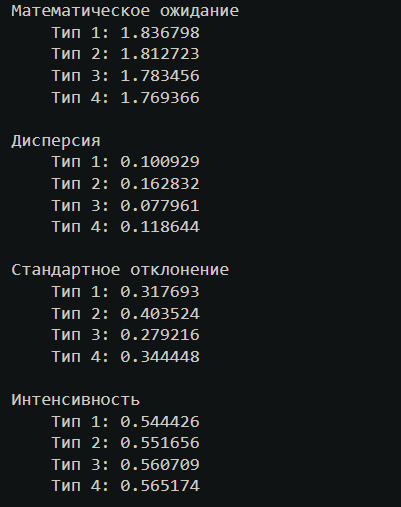


Рис. 21 – характеристики сообщений.

Исходя из полученных данных можно сделать вывод, что реализованная программа достаточно точно моделирует поток заявок, характеристики которого не так далеки от его теоретической модели.

1. **разомкнутая стохастическая сеть.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Элементы матрицы вероятностей передач | o c-1 | Vi | K1       K2      K3      K4     K5 |
| 19 | P12        P21       P32       P42       P54 | 2 | 2.5 | 1      1      2      3      3 |

Значение вероятности Р01 принимается равным единице. Значение Р00, Р02, P03, P04 равны нулю рис 22.

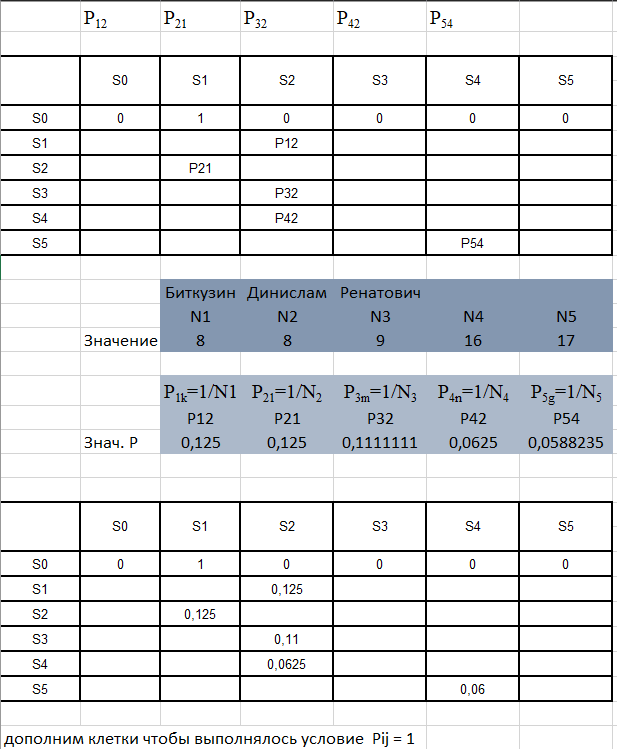


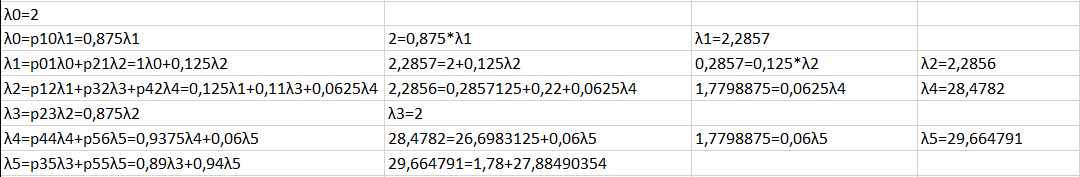
Рис.22 – Элементы матрицы

Вставим вычисляемые значения рис 23.

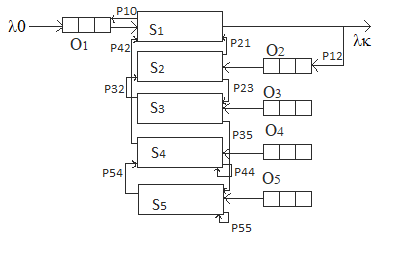
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | S0 | S1 | S2 | S3 | S4 | S5 |
|
| S0 | 0 | 1 | 0 | 0 | 0 | 0 |
| S1 | 0,875 |  | 0,125 |  |  |  |
| S2 |  | 0,125 |  | 0,875 |  |  |
| S3 |  |  | 0,11 |  |  | 0,89 |
| S4 |  |  | 0,0625 |  | 0,9375 |  |
| S5 |  |  |  |  | 0,06 | 0,94 |

Рис. 23 – Элементы матрицы

Вычислим значения

  
Итого λ0=2; λ1=2,2857; λ2=2,2856; λ3=2; λ4=28,4782; λ5=29,664791

Структурная схема сети на основе матрицы   коэффициентов передач имеет вид:



Коэффициенты передач:

α1 = λ1/ λ0, α2 = λ2/ λ0, α3 = λ3/ λ0, α4 = λ4/ λ0, α5 = λ5/ λ0;

α1 = 1,14285; α2 = 1,1428; α3 = 1; α4 = 14,2391; α5 = 14,8323955

|  |
| --- |
| K1K2K3K4K5 |
| 1      1      2      3      3 |

Значения 1 приведены в таблице 1  = 2.5

все СМО имеют одинаковую среднюю длительность обслуживания заявок, т.е. 1 = 2 = 3 = 4 =5).

i = 2.5

λ0 <min

λ0 = 2c-1 < min (2.18; 2,18; 5; 0,5267; 0,5056)

Подставляя полученные значения определим вероятности простоя каждой СМО сети:

На основе полученных вероятностей состояний определяют все остальные характеристики систем в сети, используя теорию массового обслуживания.

Средняя длина очереди заявок, ожидающих обслуживания в системе Si:

l1 = 0.035.; l2= 0.035;   l3 = 0.74:   l4 = 0.07; l5 = 0.002

Среднее число заявок в системе Si :

***mi = li+i***

m1 = 0.035 + 5.71 = 5.745;

m2 = 0.035 + 5.74 = 5.775;

m3 = 0.74 + 5 = 5.74;

m4 = 0.07 + 71.19 = 71.26;

m5 = 0.002 + 74.16 = 74.162;

Среднее время ожидания заявки в очереди системы Si:

***i= li/ i***

1 = 0.035 /2.2857= 0.015 c;

2= 0.035 /2.2856= 0.015 с:

3 = 0.74 / 2 = 0.37 с;

4 = 0.07 / 28.4782= 0.002 с.

5 = 0.002 / 29.664791= 0.0001 с.

Среднее время пребывания заявки в системе Si :

***ui= mi/i***

следует возможность определения     ui = i+ i.

u1 = 0.015 + 2.5 = 2.515 c;

u2 = 0.015 + 2.5 = 2.515 с;

u3 = 0.37 + 2.5 = 2.87 c;

u4 = 0.002 + 2.5 = 2.502 с.

u5 = 0.0001 + 2.5 = 2.5001 с.

Среднее число заявок, стоящих на очереди в сети:

Среднее число заявок, находящихся на обслуживании в сети:

Среднее время ожидания в сети:

Среднее время пребывания заявки в сети:

## Заключение

Итак, моделирование позволяет заранее предвидеть ход событий и тенденции развития, присущие управляемой системе, выяснить условия ее существования и установить режим деятельности с учетом влияния разных факторов.

Результате данной работы стало построение программы, моделирующей процесс функционирования заданной системы. Были рассчитаны (аналитически и при помощи построенного моделирующего алгоритма), сравнение фактической и теоретической вероятности появления заявок, сравнение длин, количество заявок и средних частот.

Было высчитано математическое ожидание, дисперсия, стандартное отклонение и интенсивность.

## Список литературы

1. Дворецкий, С.И. Моделирование систем: Учебник / С.И. Дворецкий. - М.: Академия, 2019.  [дата обращения: 14.01.2023]
2. Коротаев, А.В. Законы истории: Математическое моделирование развития Мир-Системы. Демография, экономика, культура / А.В. Коротаев, А.С. Малков, Д.А. Халтурина. - М.: Ленанд, 2019 [дата обращения: 14.01.2023]
3. Рыжиков, Ю.И. Имитационное моделирование. Авторская имитация систем и сетей с очередями: Учебное пособие / Ю.И. Рыжиков. - СПб.: Лань, 2019 [дата обращения: 14.01.2023]

**Приложение 1 Исходный код программы линейный конгруэнтный метод.**

Генерации 100 чисел

function CustomRandom(seed = Date.now(), A = 1103515245, C = 12345, M =

2147483647) {

    this.nextInt = () => seed = (seed \* A + C) % M;

    this.nextDouble = () => this.nextInt() / M;

}

var random = new CustomRandom(0);//without seed values are the same

new Array(100).fill(0).forEach(() => console.log(random.nextDouble()));

Получение линейной конгруэнтной последовательности

function CustomRandom(seed = Date.now(), A = 1103515245, C = 12345) {

  this.nextInt = (M) => (seed = (seed \* A + C) % M);

}

var random = new CustomRandom(0); // init seed is 0

new Array(10).fill(20).forEach((n) => console.log(random.nextInt(n)));

Построение 10000 случайных чисел при нарастающим i.

function CustomRandom(seed = Date.now(), A = 1103515245, C = 12345) {

    this.nextInt = (M) => (seed = (seed \* A + C) % M);

}

var random = new CustomRandom(0); // init seed is 0

[...Array(1000)].map((e, i) => i + 1).forEach((n) => console.log(random.nextInt(n)));

Для i = 1, 2…1000 при i\*1000;

function CustomRandom(seed = Date.now(), A = 1103515245, C = 12345) {

    this.nextInt = (M) => (seed = (seed \* A + C) % M);

}

var random = new CustomRandom(0); // init seed is 0

[...Array(1000)].map((e, i) => i + 1).forEach((n) => console.log(random.nextInt(n\*1000)));

Генерация 10 случайных чисел в javascript

var getRandomInt = (max) => Math.random() \* max;

new Array(10).fill(1).forEach((n) => console.log(getRandomInt(n)));

Генерация 1000 случайных чисел в javascript

var getRandomInt = (max) => Math.floor(Math.random() \* max);

new Array(1000).fill(100).forEach((n) => console.log(getRandomInt(n)));

Генерация 10000 случайных чисел в javascript

var getRandomInt = (max) => Math.floor(Math.random() \* max);

new Array(10000).fill(100).forEach((n) => console.log(getRandomInt(n)));

**Приложение 2 Исходный код программы распределения Пуассона.**

Графики функций плотности вероятностей f(X) / p(X)

import numpy as np

from matplotlib import pyplot as plt

from scipy.stats import poisson

from math import sqrt

np.random.seed(12)

μ = 28

min = 0

max = 84

N = 100

spkt = np.random.poisson(μ, 10000)

mean = np.mean(spkt)

var =  np.var(spkt)

a = sqrt(var)

print(f"M(x) {mean} и D(x) {var} σ {a}")

bins = x = [k for k in np.arange(0, max, max / N)]

y = [poisson.pmf(k, μ) for k in np.arange(0, max, max / N)]

plt.hist(spkt, bins = bins, density=True)

plt.plot(x, y, "ro")

plt.title("График функций плотости вероятностей")

plt.show()

График кумулятивной функции распределения F(x)

import numpy as np

from matplotlib import pyplot as plt

from scipy.stats import poisson

from math import sqrt

np.random.seed(12)

μ = 28

min = 0

max = 84

N = 100

spkt = np.random.poisson(μ, 10000)

mean = np.mean(spkt)

var =  np.var(spkt)

a = sqrt(var)

bins = x = [k for k in np.arange(0, max, max / N)]

y = [poisson.pmf(k, μ) for k in np.arange(0, max, max / N)]

y2 = np.cumsum(y)

plt.plot(x, y2, "g")

plt.title("График кумулятивной функцией распределения")

plt.show()

**Приложение 3 Исходный код программы моделирование потока.**

import random

import numpy as np

from scipy.stats import weibull\_min

class Order:

    def \_\_init\_\_(self):

        self.type = 0

        self.address = 0

        self.length = 0

        self.time = 0

    def \_\_str\_\_(self):

        return f" {self.type}    {self.address}     {self.length:03}     {self.time:.4f}"

orders = []

N = 100

MIN = 21

MAX = 237

a = 3

b = 0.9

weibull = weibull\_min.rvs(a, b, size=N)

types = {1: 0.53, 2: 0.09, 3: 0.02, 4: 0.36}

def createOrders(i):

    order = Order()

    order.length = random.randrange(MIN, MAX)

    order.time = weibull[i]

    type = random.random()

    if 0 <= type <= 0.53:

        order.type = 1

        order.address = np.random.choice(range(1, 6), p=(0.48, 0.28, 0.09, 0, 0.15))

    elif 0.53 < type <= 0.62:

        order.type = 2

        order.address = np.random.choice(range(1, 6), p=(0.07, 0.22, 0.41, 0.17, 0.13))

    elif 0.62 < type <= 0.65:

        order.type = 3

        order.address = np.random.choice(range(1, 6), p=(0.12, 0.27, 0.07, 0.39, 0.15))

    elif 0.65 < type <= 1:

        order.type = 4

        order.address = np.random.choice(range(1, 6), p=(0.02, 0.72, 0.01, 0.07, 0.18))

    return order

for i in range(0, N):

    orders.append(createOrders(i))

    orders.sort(key=lambda x: x.time)

print("Заявки")

print("Тип Адрес Длина  Время")

for i in range(0, N):

    print(orders[i])

print("\nТип Кол-во р(факт) р(теор)")

for i in range(1, len(types) + 1):

    s = sum(order.type == i for order in orders)

    print(f"{i}     {s:02}    {(s / 100):.2f}    {types.get(i):.2f}")

print("\nТип Средняя длина Макимальная длина")

for i in range(1, len(types) + 1):

    s = sum(order.type == i for order in orders)

    l = sum(order.length for order in orders if order.type == i) / s

    m = max(order.length for order in orders if order.type == i)

    print(f" {i}    {l:.2f}    {m:.2f}")

t\_max = sum(order.time for order in orders)

print(f"\nt\_max = {t\_max:.2f}")

print("Адрес Количество Средняя частота")

for i in range(1, 6):

    s = sum(order.address == i for order in orders)

    print(f"  {i}     {s:02}     {(s / t\_max):.2f}")

print("\nЧисловые характеристики для типов сообщений")

t = [[] for i in range(0, len(types))]

for order in orders:

    t[order.type - 1].append(order.time)

print("Математическое ожидание")

for i in range(1, len(types) + 1):

    print(f"    Тип {i}: {np.mean(t[i - 1]):.6f}")

print("\nДисперсия")

for i in range(1, len(types) + 1):

    print(f"    Тип {i}: {np.var(t[i - 1]):.6f}")

print("\nСтандартное отклонение")

for i in range(1, len(types) + 1):

    print(f"    Тип {i}: {np.std(t[i - 1]):.6f}")

print("\nИнтенсивность")

for i in range(1, len(types) + 1):

    print(f"    Тип {i}: {1 / np.mean(t[i - 1]):.6f}")