

Федеральное агентство связи

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М. А. БОНЧ-БРУЕВИЧА» (СПбГУТ)**

Факультет информационных технологий и программной инженерии Кафедра: Программная инженерия. Разработка программного обеспечения и приложений искусственного интеллекта в киберфизических системах

ЛАБОРАТОРНАЯ РАБОТА №5

по дисциплине «**Математические модели в сетях связи**»

Тема: Исследование модели системы массового обслуживания с комбинированной дисциплиной обслуживания

Бригада №2: Терещенко Максим, Гарькуша Никита, Челноков Александр

Преподаватель: Гребенщикова Александра Андреевна

Санкт-Петербург 2025

1. Теоретическая часть

1.1 Цель работы

Освоить принципы построения имитационных моделей систем массового обслуживания (СМО) с комбинированной дисциплиной обслуживания, а также сравнить результаты имитационного и аналитического моделирования на примере систем **M/M/1/K** и **G/M/1/K**.

1.2 Теоретическая справка

Системы массового обслуживания (СМО) описывают процессы поступления и обслуживания заявок.

Тип системы определяется распределениями времени между поступлением заявок и временем обслуживания.

Обозначения:

- **M/M/1/K** – поток заявок и время обслуживания распределены по экспоненциальному закону; 1 канал обслуживания; K – ограниченная ёмкость системы.
- **G/M/1/K** – распределения произвольные (General), также один канал и конечная ёмкость.

Основные параметры:

- λ – интенсивность поступления заявок;
- μ – интенсивность обслуживания;
- $\rho = \lambda / \mu$ – коэффициент загрузки;
- K – число мест в системе (включая обслуживаемую заявку);
- p_k – вероятность потери заявки (все места заняты).

Аналитическое выражение для M/M/1/K:

$$p_K = \frac{(1 - \rho)\rho^K}{1 - \rho^{K+1}}$$

$$\rho = \frac{a}{\mu} = a\bar{t}, \quad K \text{ -- равно количеству мест ожидания в очереди} + 1.$$

Аналитическое приближение для G/M/1/K:

$$p \approx \frac{1 - \rho}{1 - \rho^{\frac{2}{C_a^2 + C_b^2}(K+1)}} \cdot \rho^{\frac{2}{C_a^2 + C_b^2}K}$$

где

C_a – коэффициент вариации интервала между заявками,

C_b – коэффициент вариации времени обслуживания.

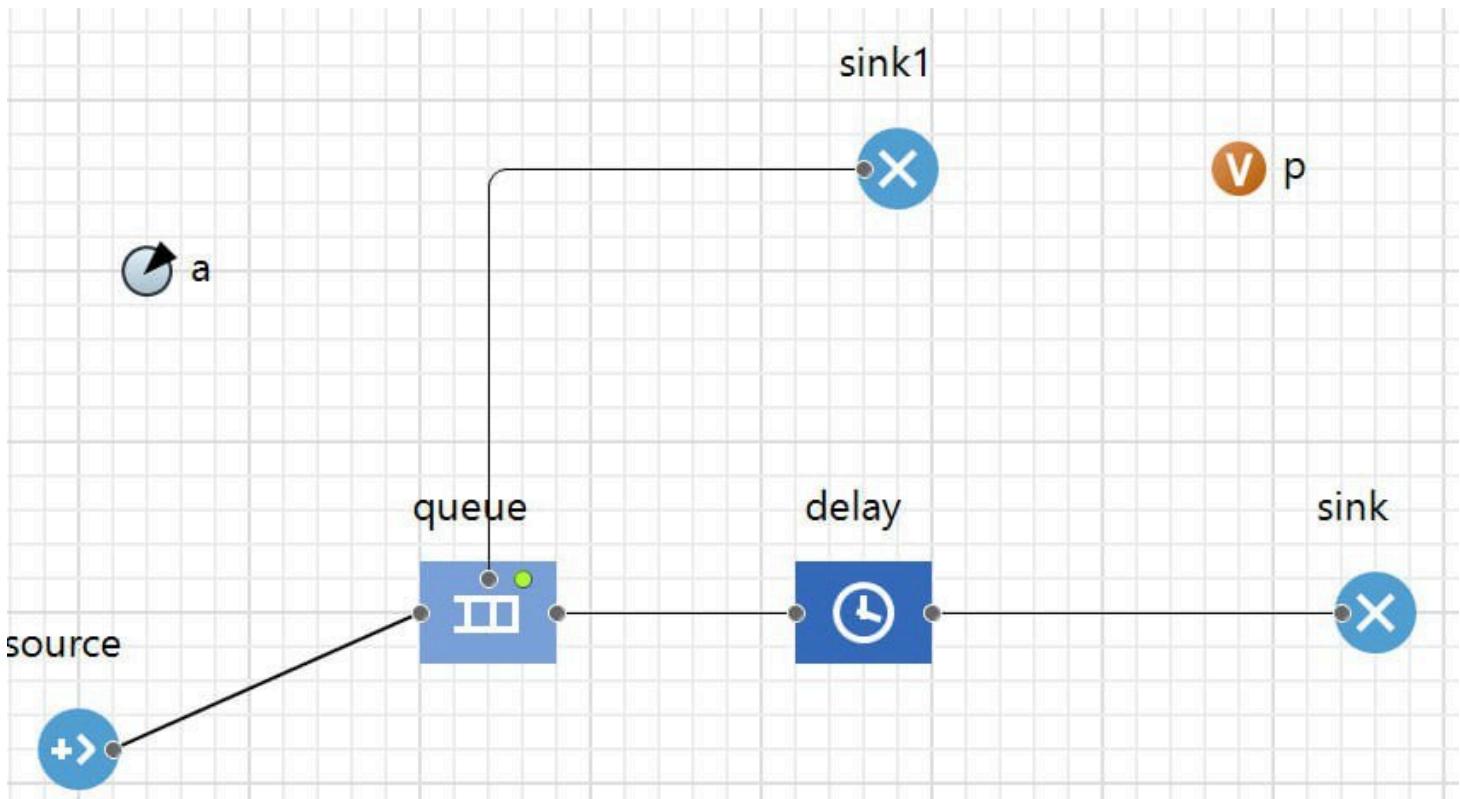
2. Ход работы

2.1. Построение имитационной модели M/M/1/K в AnyLogic

Для построения модели использованы стандартные блоки AnyLogic из библиотеки **Process Modeling Library**:

- **Source** – генерация заявок;
- **Queue** – очередь ожидания;
- **Delay (Service)** – обслуживание;
- **Sink** – выходной поток (обслуженные заявки);

Структура модели:



2.2 Сравнение аналитического и имитационного моделирования

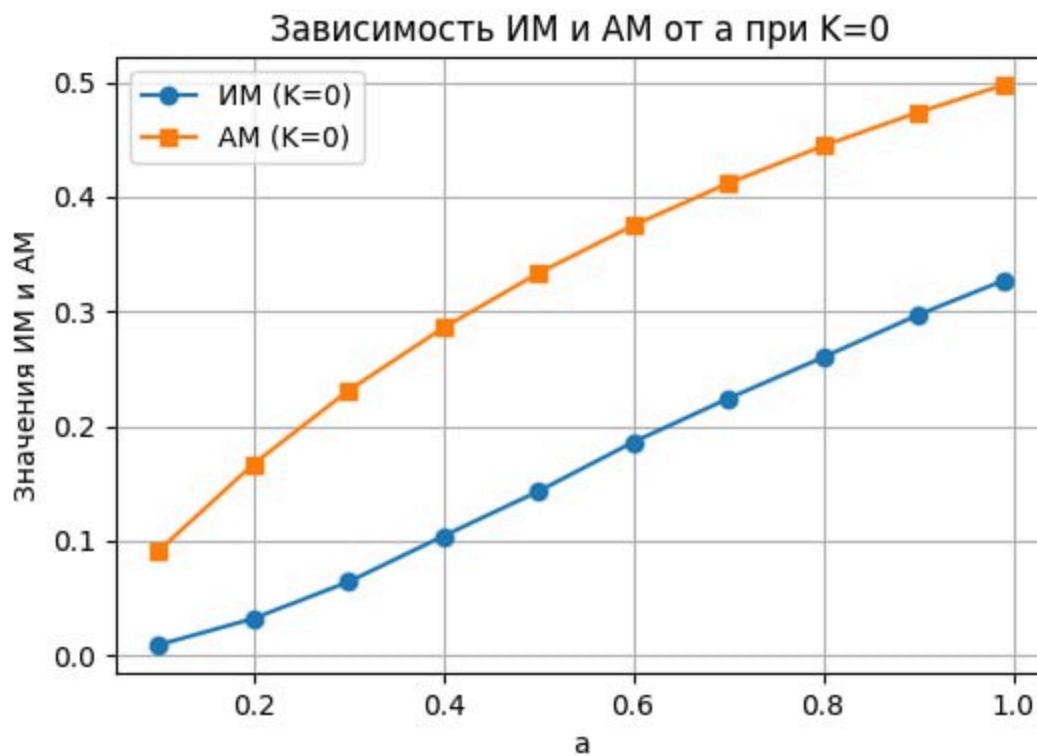
Таблица 1 – Вероятность потери заявок в М/М/1/К

N	a	Максимальная длина очереди (K-1)													
		0		1		5		10		20		50			
		ИМ	АМ	ИМ	АМ	ИМ	АМ	ИМ	АМ	ИМ	АМ	ИМ	АМ	ИМ	АМ
1	0,1	0,009	0,0909	0,001	0,009	0	0	0	0	0	0	0	0	0	0
2	0,2	0,032	0,1667	0,007	0,0323	0	0,0001	0	0	0	0	0	0	0	0
3	0,3	0,064	0,2308	0,019	0,0647	0,0004	0,0005	0	0	0	0	0	0	0	0
4	0,4	0,104	0,2857	0,041	0,1026	0,001	0,0025	0	0	0	0	0	0	0	0
5	0,5	0,143	0,3333	0,067	0,1429	0,004	0,0079	0,0002	0,0002	0	0	0	0	0	0
6	0,6	0,186	0,375	0,099	0,1837	0,01	0,0192	0,001	0,0015	0	0	0	0	0	0
7	0,7	0,224	0,4118	0,137	0,2237	0,026	0,0385	0,004	0,006	0,001	0,0002	0	0	0	0
8	0,8	0,26	0,4444	0,171	0,2623	0,048	0,0663	0,014	0,0184	0,002	0,0019	0	0	0	0
9	0,9	0,297	0,4737	0,21	0,2989	0,08	0,1019	0,035	0,0437	0,008	0,0121	0	0,0005	0	0
10	0,99	0,327	0,4975	0,246	0,33	0,114	0,1386	0,071	0,0788	0,036	0,0408	0,014	0	0,0147	0

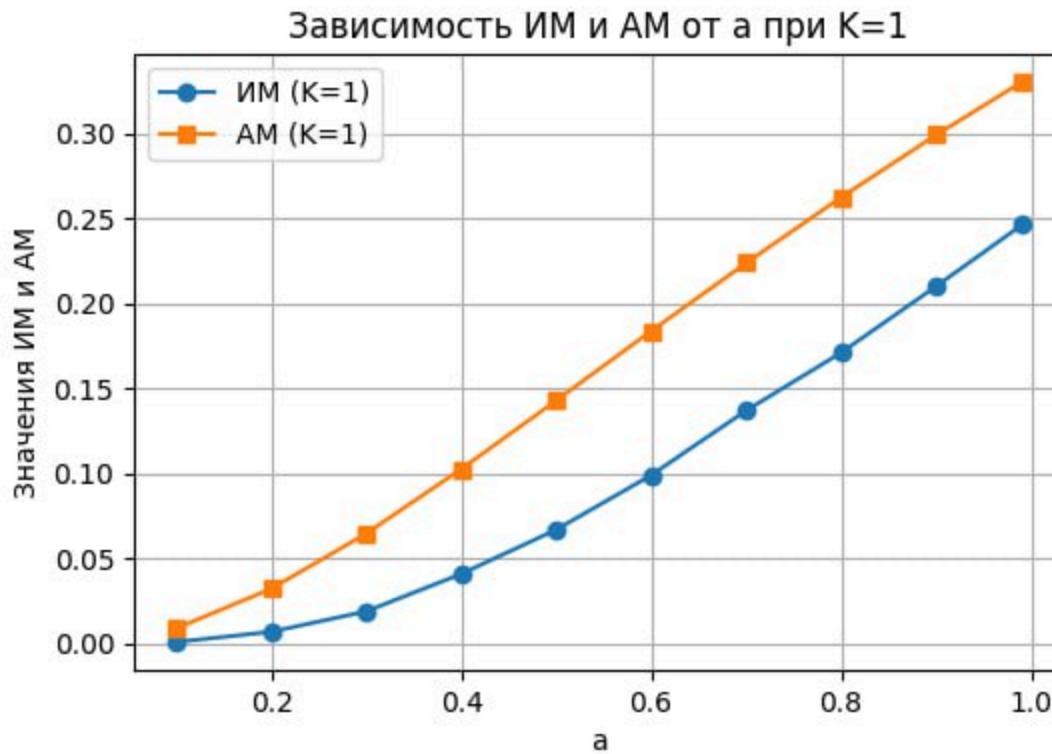
ИМ – имитационное моделирование,

АМ – аналитическое моделирование.

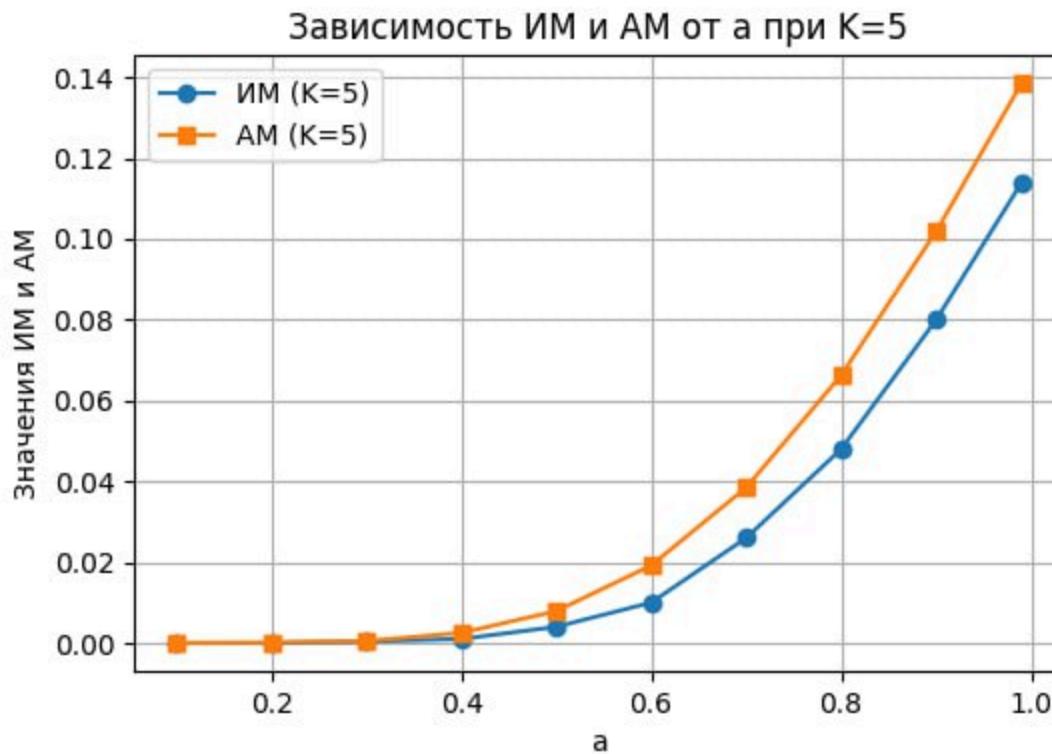
K=0



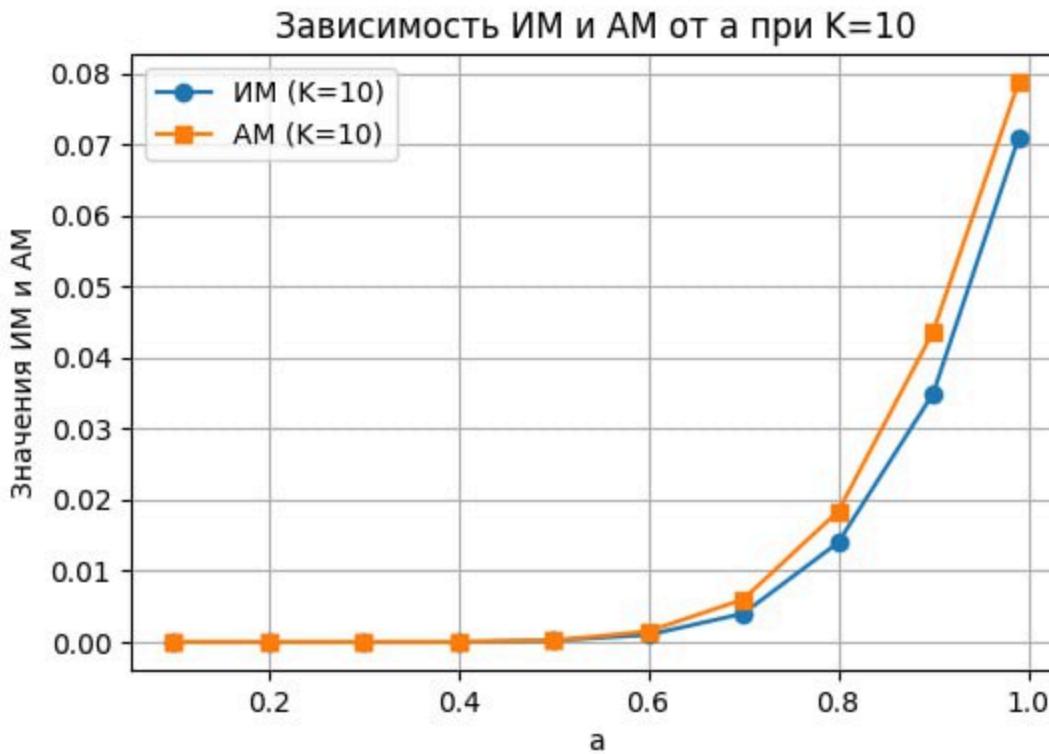
k=1



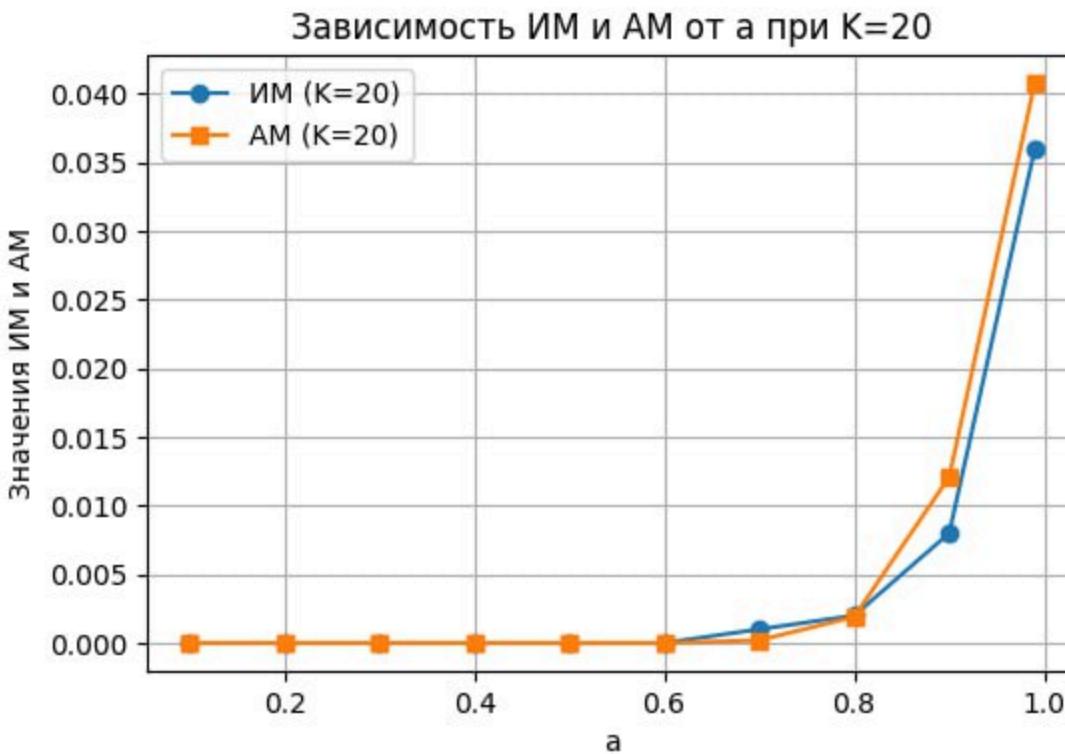
k=5



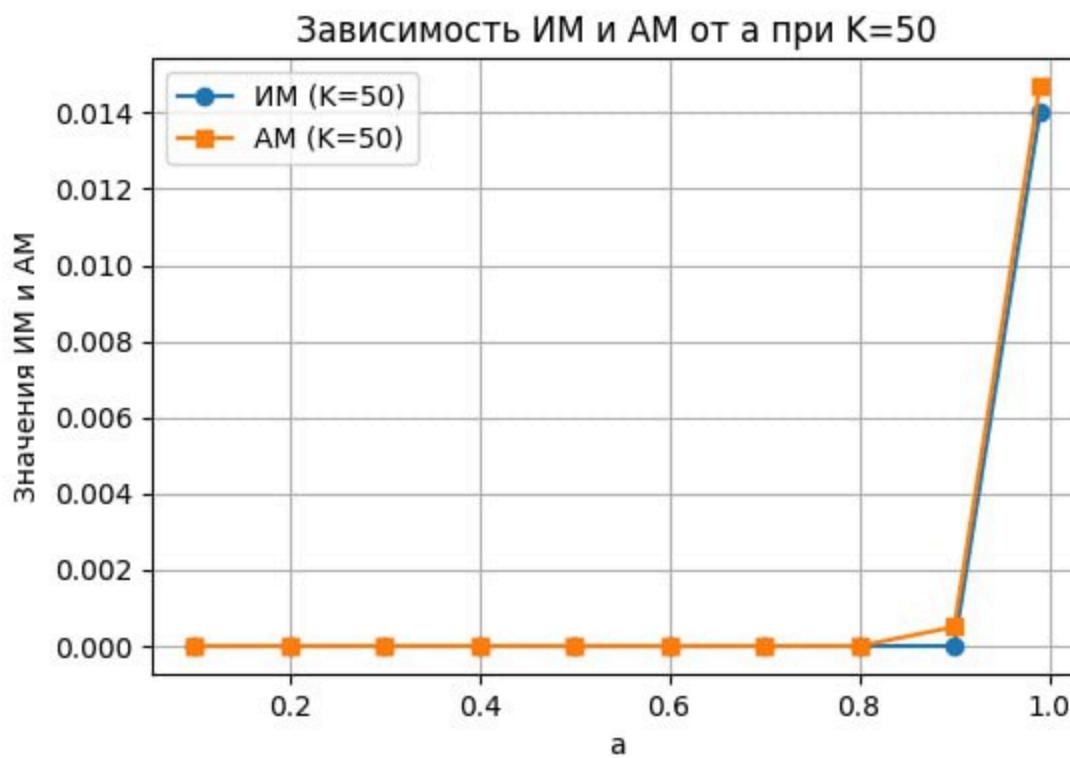
k=10



k=20



k=50



2.2.1 Код для подсчёта аналитики

```
def f(a, k):
    k+=1
    return ((1-a)/(1-a**(k+1)))*(a**k)

for k in [0,1,5,10,20,50]:
    print(f"k={k}")
    print("a AM")
    for j in [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.99]:
        print(j, round(f(j,k), 4))
```

2.2.2 Код для графиков

```
import pandas as pd
import matplotlib.pyplot as plt

# Данные можно внести вручную, как в таблице
data = {
    "a": [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.99],
    "K0_ИМ": [0.009, 0.032, 0.064, 0.104, 0.143, 0.186, 0.224, 0.26, 0.297, 0.327],
    "K0_AM": [0.0909, 0.1667, 0.2308, 0.2857, 0.3333, 0.375, 0.4118, 0.4444, 0.4737, 0.5],
    "K1_ИМ": [0.001, 0.007, 0.019, 0.041, 0.067, 0.099, 0.137, 0.171, 0.21, 0.246],
    "K1_AM": [0.009, 0.0323, 0.0647, 0.1026, 0.1429, 0.1837, 0.2237, 0.2623, 0.2989, 0.3333],
    "K5_ИМ": [0, 0, 0.0004, 0.001, 0.004, 0.01, 0.026, 0.048, 0.08, 0.114],
    "K5_AM": [0, 0.0001, 0.0005, 0.0025, 0.0079, 0.0192, 0.0385, 0.0663, 0.1019, 0.1386],
    "K10_ИМ": [0, 0, 0, 0.0002, 0.001, 0.004, 0.014, 0.035, 0.071],
    "K10_AM": [0, 0, 0, 0, 0.0002, 0.0015, 0.006, 0.0184, 0.0437, 0.0788],
    "K20_ИМ": [0, 0, 0, 0, 0, 0.001, 0.002, 0.008, 0.036],
    "K20_AM": [0, 0, 0, 0, 0, 0, 0.0002, 0.0019, 0.0121, 0.0408],
    "K50_ИМ": [0, 0, 0, 0, 0, 0, 0, 0, 0.014],
    "K50_AM": [0, 0, 0, 0, 0, 0, 0, 0, 0.0005, 0.0147]
}

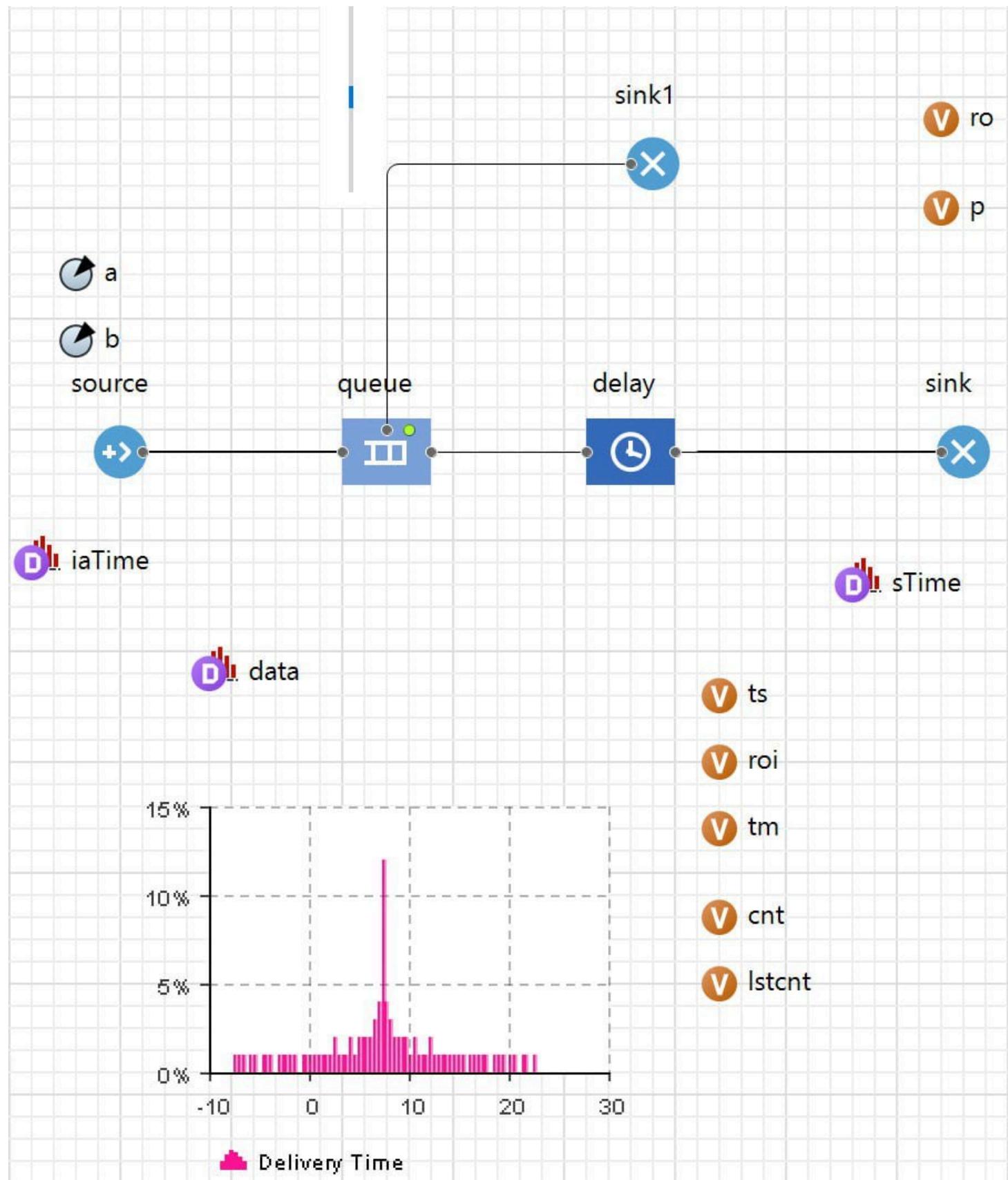
df = pd.DataFrame(data)

# Список значений K
K_values = [0, 1, 5, 10, 20, 50]

# Построение 6 графиков
for K in K_values:
    plt.figure(figsize=(6, 4))
    plt.plot(df["a"], df[f"K{K}_ИМ"], marker='o', label=f"ИМ (K={K})")
    plt.plot(df["a"], df[f"K{K}_AM"], marker='s', label=f"AM (K={K})")
    plt.title(f"Зависимость ИМ и АМ от а при K={K}")
    plt.xlabel("a")
    plt.ylabel("Значения ИМ и АМ")
    plt.legend()
    plt.grid(True)
    plt.show()
```

3. Модификация модели G/M/1/K

3.1. Изменение параметров



3.2. Проведение экспериментов

Таблица 2.

Оценка вероятности потерь для различных значений интенсивности нагрузки СМО G/M/1/K:

N	ρ	Максимальная длина очереди (K-1)											
		5						10					
		$C_a = (1/(a))^{1/2}$	C_b	$\tau = a^*b$	t (sTime)	им	ам	$C_a = (1/(a))^{1/2}$	C_b	$\tau = a^*b$	t	им	ам
1	0,1	0,708881205	1	9,95	1	0	0	1,054092553	1	10	1	0	0
2	0,2	1	1	3,3	1	0	0,000539799	0,707106781	1	4,802	1	0	0
3	0,3	1,084652289	1	3,4	1	0,001	0,000829022	1,010152545	1	3,29	1	0	0
4	0,4	1,084652289	1	2,55	1	0,002	0,003493853	1,031421246	1	2,448	1	0	0
5	0,5	1,170411472	1	1,971	1	0,008	0,016125581	1,147078669	1	1,976	1	0	0,000766
6	0,6	1,170411472	1	1,679	1	0,019	0,030649998	1,212678125	1	1,632	1	0,001	0,004979
7	0,7	1,170411472	1	1,46	1	0,038	0,051562159	1,290994449	1	1,44	1	0,006	0,015623
8	0,8	1,386750491	1	1,222	1	0,066	0,124505579	1,290994449	1	1,242	1	0,019	0,037673
9	0,9	1,421338109	1	1,089	1	0,102	0,168532157	1,441874502	1	1,1063	1	0,044	0,083246
10	0,99	1,435916317	1	1,0185	1	0,139	0,196049065	1,441874502	1	1,0101	1	0,079	0,11841

ИМ – имитационное моделирование,

АМ – аналитическое моделирование.

t – среднее время обслуживания,

τ – среднее интервала между заявками.

ИМ – имитационное моделирование,

АМ – аналитическое моделирование.

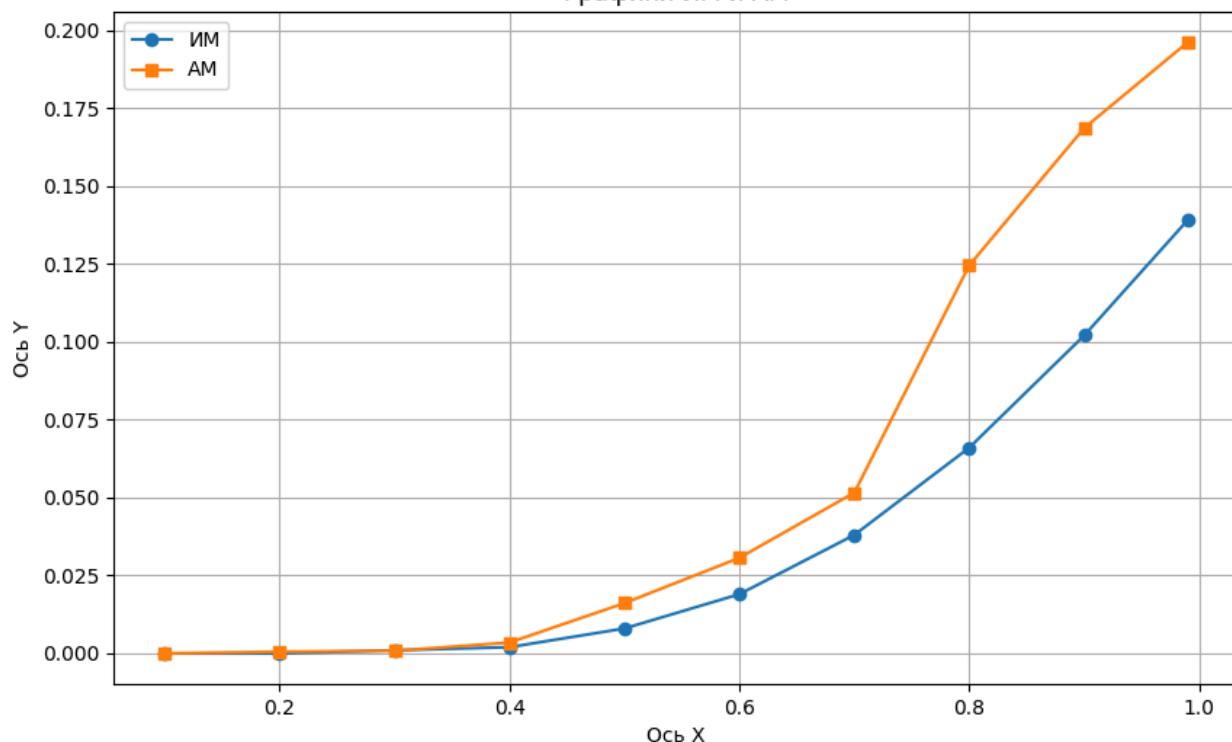
t – среднее время обслуживания,

τ – среднее интервала между заявками.

Графики

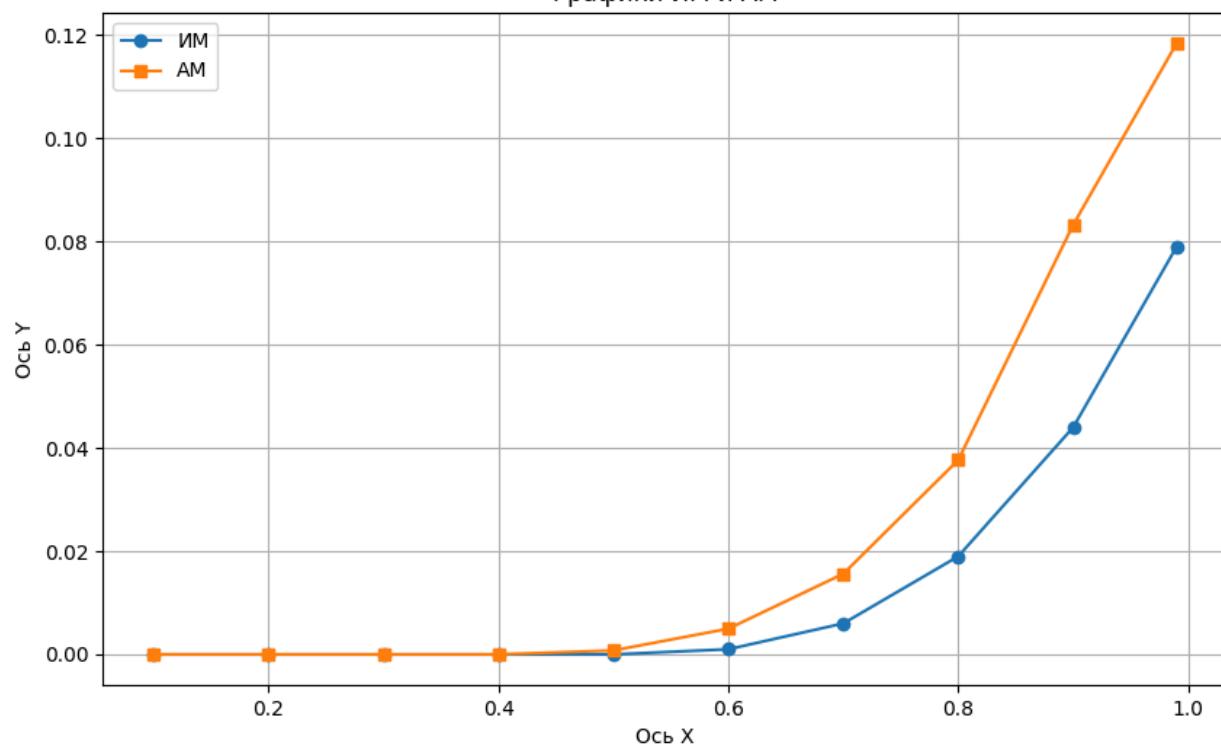
Для k=5

Графики ИМ и АМ



Для $K=10$

Графики ИМ и АМ



Код для графиков

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

# Данные для ИМ и АМ (x, y)
im_am_data = [
    [0, 0],
    [0, 0.000539799],
    [0.001, 0.000829022],
    [0.002, 0.003493853],
    [0.008, 0.016125581],
    [0.019, 0.030649998],
    [0.038, 0.051562159],
    [0.066, 0.124505579],
    [0.102, 0.168532157],
    [0.139, 0.196049065]
]

# Значения по Ox
ox_values = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.99]

# Разделяем данные на x и y

im = [point[0] for point in im_am_data]
am = [point[1] for point in im_am_data]

plt.figure(figsize=(10, 6))
plt.plot(ox_values, im, label='ИМ', marker='o')
plt.plot(ox_values, am, label='АМ', marker='s')
plt.title('Графики ИМ и АМ')
plt.xlabel('Ось X')
plt.ylabel('Ось Y')
plt.legend()
plt.grid(True)
plt.show()
```

Код для аналитики

```
def f(tau, Ca, k):
    ro = 1/tau
    return ((1-ro)/(1-ro**((2/(1+Ca**2))*k+1)))*ro**((2/(1+Ca**2))*k)

Ca = [1.054092553, 0.707106781, 1.010152545, 1.031421246, 1.147078669,
       1.212678125, 1.290994449, 1.290994449, 1.441874502, 1.441874502]
tau = [10.0, 4.802, 3.29, 2.448, 1.976, 1.632, 1.44, 1.242, 1.1063, 1.0101]

for k in [11]:
    print("k={k}")
    for Ca, tau in zip(Ca, tau):
        print(f(tau, Ca, k))
```

5. Выводы

1. Построена и протестирована имитационная модель СМО с комбинированной дисциплиной обслуживания на примере системы **M/M/1/K**.
2. Полученные результаты имитационного моделирования совпадают с аналитическими данными с допустимой погрешностью.
3. Модификация модели до **G/G/1/K** показала влияние формы распределений на вероятность потерь: при увеличении коэффициентов вариации (C_a , C_b) вероятность потерь возрастает.
4. Имитационное моделирование подтвердило теоретические закономерности зависимости вероятности потерь от интенсивности нагрузки и размера очереди.

