**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ**

**Кафедра теории вероятностей и математической статистики**

**ОТЧЕТ**

по лабораторной работе №1

«Моделирование стационарных случайных процессов с заданными корреляционными свойствами»

учебной дисциплины

«Математические методы анализа данных»

Вариант №3

**Выполнила:**

Лавринович Анна Павловна,

3 курс 7а группа, специальность «прикладная математика»

**Преподаватель:**

Цеховая Татьяна Вячеславовна,

кандидат физико-математических наук, доцент

Минск, 2025

**Постановка задачи.** На отрезке времени [0, T] смоделировать n, n = 150, отсчетов случайного процесса Х(t), имеющего математическое ожидание μ = 0, ковариационную функцию R(t) и спектральную плотность f(λ). Необходимо:

1. Выбрать самостоятельно алгоритм моделирования отсчетов случайного процесса.

2. Отсчеты Х(1), Х(2), …, Х(n) случайного процесса представить графически.

3. Выполнить предварительный статистический анализ данных (вычислить описательные статистики, построить гистограмму, проверить гипотезу на нормальность, осуществить тренд-анализ). Сделать вывод.

4. Вычислить оценку ковариационной функции. В качестве оценки ковариационной функции рассмотреть статистику

Положить и для

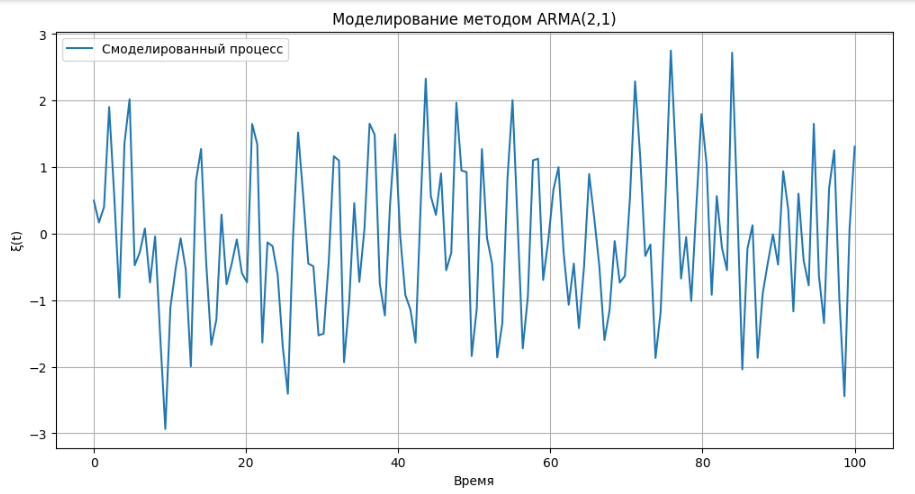
5. На одном рисунке представить графики ковариационной функции и ее оценки для лага h = 0, …, .

**Исходные данные (алгоритм выполнения).** Дана ковариационная функция .

Алгоритм моделирования (ARMA (2, 1) – метод авторегрессии-скользящего среднего):

Положим параметры

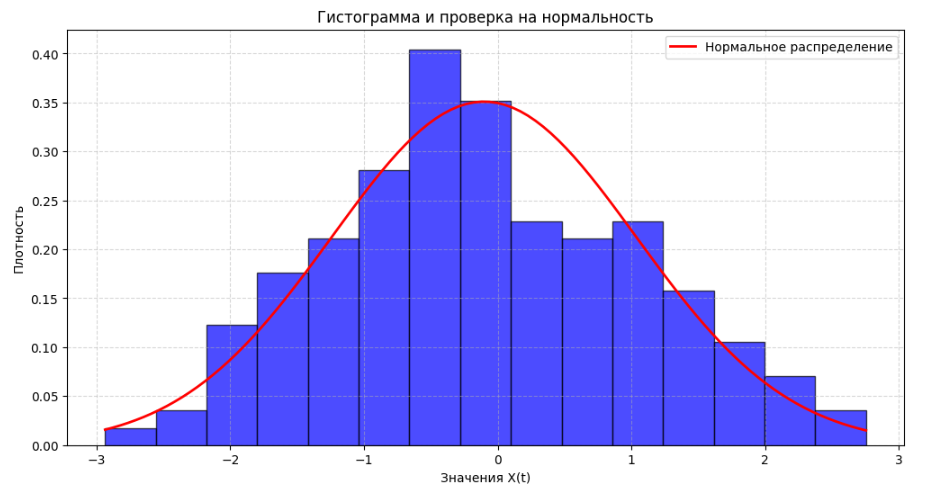
**Результат.** График отсчётов случайного процесса:

****

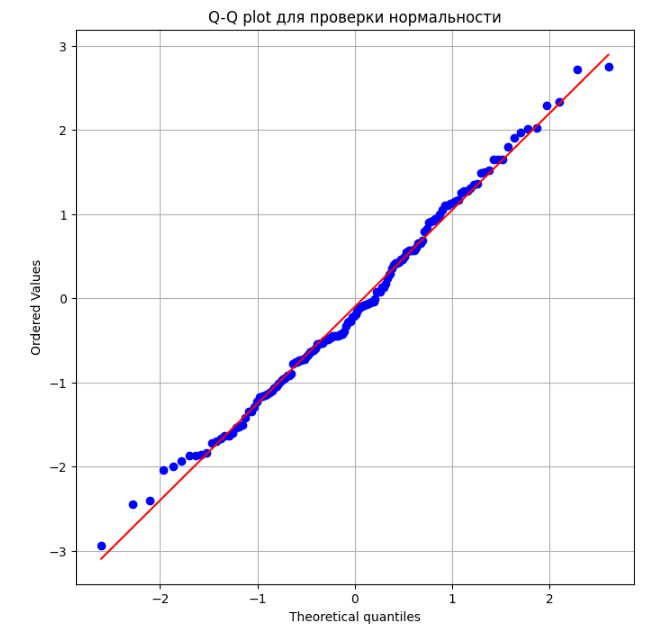
Описательные статистики:

* Среднее: -0.1025;
* Дисперсия: 1.2928;
* Медиана: -0.1979;
* Первый квартиль (Q1): -0.9123;
* Третий квартиль (Q3): 0.6558;
* Асимметрия: 0.1856;
* Эксцесс: -0.3822.

Гистограмма:

****

Проверка гипотезы на нормальность (тест Шапиро-Уилка):

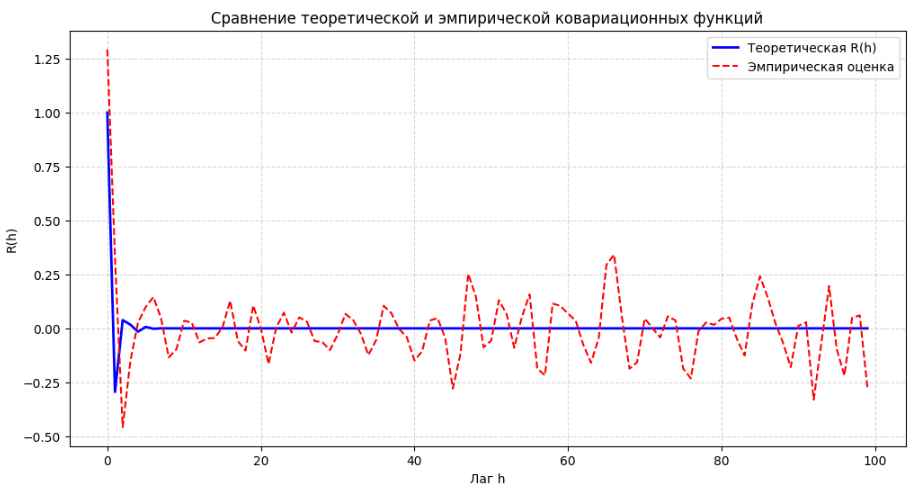
****

Тест Шапиро-Уилка: p-value = 0.5466. Не отвергаем гипотезу о нормальности (p > 0.05).

Тренд-анализ (тест Манна-Кендалла):

Тест Манна-Кендалла на тренд: p-value = 0.7268. Значимый тренд не обнаружен.

График ковариационной функции и ее оценки:

****

**Вывод.** Описательные статистики. Распределение скошено вправо, малая правосторонняя асимметрия. Распределение имеет плосковершинную форму и является низко вершинным по отношению к нормальному распределению.

Проверка на нормальность. Гистограмма показывает распределение значений с наложенной кривой нормального распределения. Визуально можем подтвердить, что данные могут быть нормально распределены. Тест Шапиро-Уилка гипотезу о нормальности не отверг. Это говорит о том, что данные могут следовать нормальному распределению.

Тренд-анализ. Тест Манна-Кендалла не обнаружил значимый тренд в данных. Значение p-value выше 0.05 указывает на то, что тренд может отсутствовать.

Графики оценки ковариационной функции и теоретической близки на малых значениях лагов, можем предположить, что модель корректно воспроизводит процесс, параметры подобраны верно.

Результаты показывают, что СП, смоделированный по методу авторегрессии-скользящего среднего, демонстрирует нормальность распределения, значимый тренд отсутствует.

**Листинг программы**

import numpy as np

import matplotlib.pyplot as plt

from scipy import stats

#Параметры

D = 1.0

alpha = 0.5

beta = 2

n = 150

T = 100.0

delta\_t = T / n

gamma = alpha \* delta\_t

gamma\_0 = beta \* delta\_t

alpha0 = np.exp(-gamma) \* (np.exp(-2\*gamma) - 1) \* np.cos(gamma0)

alpha1 = 1 - np.exp(-4\*gamma)

a0 = np.sqrt(D) \* alpha

a1 = np.sqrt(D) \* (alpha\_0 / alpha)

b1 = 2 \* np.exp(-gamma) \* np.cos(gamma\_0)

b2 = -np.exp(-2 \* gamma)

# Генерация белого шума

np.random.seed(42)

x = np.random.normal(0, 1, n)

# Инициализация процесса

xi = np.zeros(n)

xi[0] = a0 \* x[0]

xi[1] = a0 \* x[1] + a1 \* x[0] + b1 \* xi[0]

for i in range(2, n):

xi[i] = a0 \* x[i] + a1 \* x[i-1] + b1 \* xi[i-1] + b2 \* xi[i-2]

# График

t = np.linspace(0, T, n)

plt.figure(figsize=(12, 6))

plt.plot(t, xi, label='Смоделированный процесс')

plt.title('Моделирование методом ARMA(2,1)')

plt.xlabel('Время')

plt.ylabel('ξ(t)')

plt.grid()

plt.legend()

plt.show()

#Описательные статистики:

mu\_hat = np.mean(xi)

var\_hat = np.var(xi)

skew\_hat = stats.skew(xi)

kurt\_hat = stats.kurtosis(xi)

# Добавленные статистики

median = np.median(xi)

q1 = np.percentile(xi, 25) # Первый квартиль (25-й процентиль)

q3 = np.percentile(xi, 75) # Третий квартиль (75-й процентиль)

print("Описательные статистики:")

print(f"Среднее: {mu\_hat:.4f}")

print(f"Дисперсия: {var\_hat:.4f}")

print(f"Медиана: {median:.4f}")

print(f"Первый квартиль (Q1): {q1:.4f}")

print(f"Третий квартиль (Q3): {q3:.4f}")

print(f"Асимметрия: {skew\_hat:.4f}")

print(f"Эксцесс: {kurt\_hat:.4f}")

# Гистограмма с нормальным распределением

plt.figure(figsize=(12, 6))

plt.hist(xi, bins=15, density=True, alpha=0.7, color='blue', edgecolor='black')

x\_norm = np.linspace(min(xi), max(xi), 100)

plt.plot(x\_norm, stats.norm.pdf(x\_norm, mu\_hat, np.sqrt(var\_hat)),

'r-', lw=2, label='Нормальное распределение')

plt.title('Гистограмма и проверка на нормальность')

plt.xlabel('Значения X(t)')

plt.ylabel('Плотность')

plt.legend()

plt.grid(True, linestyle='--', alpha=0.5)

plt.show()

plt.figure(figsize=(8, 8))

stats.probplot(xi, dist="norm", plot=plt)

plt.title('Q-Q plot для проверки нормальности')

plt.grid(True)

plt.show()

# Проверка на нормальность (тест Шапиро-Уилка)

shapiro\_test = stats.shapiro(xi)

print(f"\nТест Шапиро-Уилка: p-value = {shapiro\_test.pvalue:.4f}")

if shapiro\_test.pvalue > 0.05:

print("Не отвергаем гипотезу о нормальности (p > 0.05)")

else:

print("Отвергаем гипотезу о нормальности (p ≤ 0.05)")

# Проверка на тренд (тест Манна-Кендалла)

from scipy.stats import kendalltau

tau, p\_value = kendalltau(t, xi)

print(f"\nТест Манна-Кендалла на тренд: p-value = {p\_value:.4f}")

if p\_value > 0.05:

print("Значимый тренд не обнаружен")

else:

print("Обнаружен значимый тренд")

# Эмпирическая ковариационна функция

def empirical\_covariance(x, max\_lag):

n = len(x)

mu = np.mean(x)

R = np.zeros(max\_lag)

for h in range(max\_lag):

R[h] = np.sum((x[h:] - mu) \* (x[:n-h] - mu)) / (n - h)

return R

max\_lag = 2 \* n // 3

R\_emp = empirical\_covariance(xi, max\_lag)

# Теоретическая ковариационная функция

def theoretical\_cov(h, D, alpha, beta):

return D \* np.exp(-alpha\*np.abs(h)) \* np.cos(beta\*h)

R\_theory = [theoretical\_cov(h, D, alpha, beta) for h in range(max\_lag)]

#Сравнение теоретической и эмпирической ковариационных функций

plt.figure(figsize=(12, 6))

plt.plot(range(max\_lag), R\_theory, 'b-', linewidth=2, label='Теоретическая R(h)')

plt.plot(range(max\_lag), R\_emp, 'r--', linewidth=1.5, label='Эмпирическая оценка')

plt.title('Сравнение теоретической и эмпирической ковариационных функций')

plt.xlabel('Лаг h')

plt.ylabel('R(h)')

plt.legend()

plt.grid(True, linestyle='--', alpha=0.5)

plt.show()