**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ**

**Кафедра теории вероятностей и математической статистики**

**ОТЧЕТ**

по лабораторной работе №2

«Ковариационная функция, семивариограмма и спектральная плотность стационарного в широком смысле случайного процесса»

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.** Для стационарных в широком смысле случайных процессов с известными ковариационными функциями найти аналитический вид их семивариограмм и спектральных плотностей. Сделать вывод о свойствах процессов.

**Необходимо:**

Рассмотреть требуемые модели ковариационных функций стационарных случайных процессов с непрерывным временем . Указать, к какому классу относятся исследуемые модели: *колебательному, монотонно убывающему, …*

* Модель ковариационной функции представить в общем виде с указанием всех параметров.
* Найти аналитический вид семивариограммы
* Найти аналитический вид спектральной плотности
* Построить графики функций и при различных сочетаниях параметров (минимум 3 значения для каждого параметра). Сделать сравнительный анализ.
* Вычислить время корреляции по представленным ниже формулам:

а) б) в)

Сделать сравнительный анализ длин интервалов корреляции.

* Вычислить эффективную ширину спектра по формуле:

Если основная мощность процесса сосредоточена в точке т. е. . Проверить выполнение неравенства неопределенности:

Если основная мощность процесса сосредоточена вблизи экстремальной частоты спектральной плотности, т. е. , то ширину спектра вычислить по формуле:

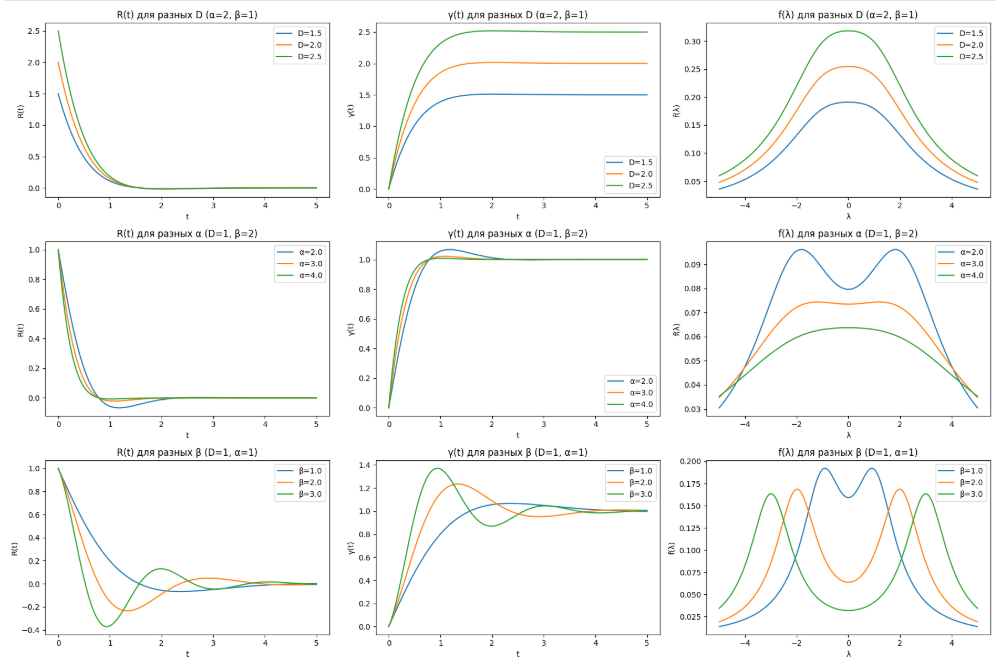
**Исходные данные (алгоритм выполнения).**

1. Дана ковариационная функция

Получим аналитический вид спектральной плотности:

Получим аналитический вид семивариограммы:

Построим графики функций и при различных сочетаниях параметров:



Вычислим время корреляции по представленным выше формулам:

Вычислим эффективную ширину спектра (эффективная мощность спектра сосредоточена в нуле для всех значений параметра):

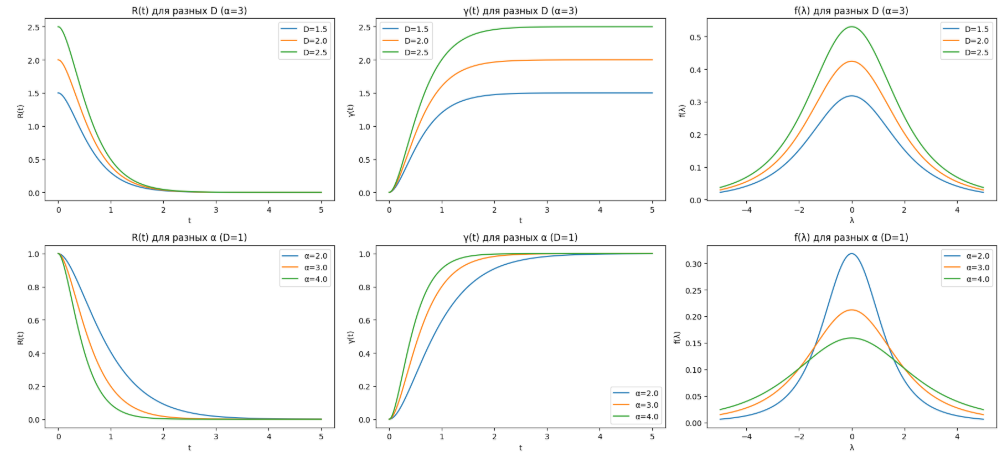
Неравенство неопределенности выполняется для (

1. Дана ковариационная функция

Получим аналитический вид спектральной плотности:

Получим аналитический вид семивариограммы:

Построим графики функций и при различных сочетаниях параметров:



Вычислим время корреляции по представленным выше формулам:

Вычислим эффективную ширину спектра (эффективная мощность спектра сосредоточена в нуле для всех значений параметра):

Неравенство неопределенности выполняется:

**Вывод. Часть первая.**

**Анализ графиков.**

**Параметр D.**

* Ковариационная функция: Увеличение D приводит к увеличению амплитуды функции, что указывает на более выраженные корреляции.
* Спектральная плотность: Высота графиков спектральной плотности увеличивается с ростом D, что указывает на усиление мощности сигнала.
* Семивариограмма: Увеличение D приводит к увеличению значений семивариограммы, указывая на большую вариативность данных.

**Параметры альфа и бета.**

* Ковариационная функция: При фиксированном D увеличение относительно делает графики более узкими и смещает их влево, указывая на более быстрое затухание ковариации.
* Спектральная плотность: Увеличение параметра приводит к расширению и сглаживанию функции распределения, в то время как большие значения соответствуют более узким и острым пикам, указывающим на концентрацию значений вокруг центра распределения.
* Семивариограмма: Увеличение приводит к более быстрой потере корреляции, а увеличение может сделать затухание более плавным, указывая на долгосрочные корреляции.

**Сравнительный анализ длин интервалов корреляции.** Все три формулы показывают, что увеличение параметра приводит к увеличению длины интервала корреляции, тогда как увеличение может вызвать его уменьшение.

**Часть вторая.**

**Анализ графиков.**

**Параметр D.**

* Ковариационная функция: Увеличение D приводит к более выраженному затуханию функции. Графики показывают, что с увеличением D корреляция становится менее выраженной на коротких временных интервалах.
* Спектральная плотность: С ростом D наблюдается увеличение амплитуды, что указывает на более высокую мощность сигнала на определенных частотах.
* Семивариограмма: Увеличение D ведет к увеличению значений семивариограммы, что свидетельствует о более высокой вариативности данных.

**Параметр альфа.**

* Ковариационная функция: При фиксированном D изменение влияет на скорость затухания. При больших значениях графики становятся более пологими, указывая на замедленное затухание.
* Спектральная плотность: Увеличение приводит к смещению пиков к более низким значениям частот, делая спектры более пологими.
* Семивариограмма: Большие значения соответствуют более высоким и острым пикам функции.

**Сравнительный анализ длин интервалов корреляции.** Все три формулы показывают, что увеличение параметра вызывает уменьшение длины интервала корреляции.

**Листинг программы**

# Первая часть

import numpy as np

import matplotlib.pyplot as plt

from scipy.integrate import quad

# Параметры для построения графиков

D\_values = [1.5, 2.0, 2.5]

alpha\_values = [2.0, 3.0, 4.0]

beta\_values = [1.0, 2.0, 3.0]

# Временная ось

t = np.linspace(0, 5, 500)

# Частотная ось

lambda\_vals = np.linspace(-5, 5, 500)

# Функция ковариации R(t)

def covariance\_func(t, D, alpha, beta):

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

# Семивариограмма gamma(t) = R(0) - R(t)

def semivariogram(t, D, alpha, beta):

return D - covariance\_func(t, D, alpha, beta)

# Спектральная плотность f(lambda)

def spectral\_density(lambda\_val, D, alpha, beta):

term1 = D \* alpha \* (alpha\*\*2 + lambda\_val\*\*2 + beta\*\*2)

term2 = (alpha\*\*2 + (lambda\_val - beta)\*\*2) \* (alpha\*\*2 + (lambda\_val + beta)\*\*2)

return term1 / (np.pi \* term2)

# Построение графиков

plt.figure(figsize=(18, 12))

# Графики для разных D (alpha=2, beta=1)

plt.subplot(3, 3, 1)

for D in D\_values:

plt.plot(t, covariance\_func(t, D, 2.0, 1.0), label=f'D={D}')

plt.title('R(t) для разных D (α=2, β=1)')

plt.xlabel('t')

plt.ylabel('R(t)')

plt.legend()

plt.subplot(3, 3, 2)

for D in D\_values:

plt.plot(t, semivariogram(t, D, 2.0, 1.0), label=f'D={D}')

plt.title('γ(t) для разных D (α=2, β=1)')

plt.xlabel('t')

plt.ylabel('γ(t)')

plt.legend()

plt.subplot(3, 3, 3)

for D in D\_values:

plt.plot(lambda\_vals, spectral\_density(lambda\_vals, D, 2.0, 1.0), label=f'D={D}')

plt.title('f(λ) для разных D (α=2, β=1)')

plt.xlabel('λ')

plt.ylabel('f(λ)')

plt.legend()

# Графики для разных alpha (D=1, beta=2)

plt.subplot(3, 3, 4)

for alpha in alpha\_values:

plt.plot(t, covariance\_func(t, 1.0, alpha, 2.0), label=f'α={alpha}')

plt.title('R(t) для разных α (D=1, β=2)')

plt.xlabel('t')

plt.ylabel('R(t)')

plt.legend()

plt.subplot(3, 3, 5)

for alpha in alpha\_values:

plt.plot(t, semivariogram(t, 1.0, alpha, 2.0), label=f'α={alpha}')

plt.title('γ(t) для разных α (D=1, β=2)')

plt.xlabel('t')

plt.ylabel('γ(t)')

plt.legend()

plt.subplot(3, 3, 6)

for alpha in alpha\_values:

plt.plot(lambda\_vals, spectral\_density(lambda\_vals, 1.0, alpha, 2.0), label=f'α={alpha}')

plt.title('f(λ) для разных α (D=1, β=2)')

plt.xlabel('λ')

plt.ylabel('f(λ)')

plt.legend()

# Графики для разных beta (D=1, alpha=1)

plt.subplot(3, 3, 7)

for beta in beta\_values:

plt.plot(t, covariance\_func(t, 1.0, 1.0, beta), label=f'β={beta}')

plt.title('R(t) для разных β (D=1, α=1)')

plt.xlabel('t')

plt.ylabel('R(t)')

plt.legend()

plt.subplot(3, 3, 8)

for beta in beta\_values:

plt.plot(t, semivariogram(t, 1.0, 1.0, beta), label=f'β={beta}')

plt.title('γ(t) для разных β (D=1, α=1)')

plt.xlabel('t')

plt.ylabel('γ(t)')

plt.legend()

plt.subplot(3, 3, 9)

for beta in beta\_values:

plt.plot(lambda\_vals, spectral\_density(lambda\_vals, 1.0, 1.0, beta), label=f'β={beta}')

plt.title('f(λ) для разных β (D=1, α=1)')

plt.xlabel('λ')

plt.ylabel('f(λ)')

plt.legend()

plt.tight\_layout()

plt.show()

# Вторая часть

import numpy as np

import matplotlib.pyplot as plt

from scipy.integrate import quad

# Параметры для построения графиков

D\_values = [1.5, 2.0, 2.5]

alpha\_values = [2.0, 3.0, 4.0]

# Временная ось

t = np.linspace(0, 5, 500)

# Частотная ось

lambda\_vals = np.linspace(-5, 5, 500)

# Функция ковариации R(t)

def covariance\_func(t, D, alpha):

return D \* np.exp(-alpha \* np.abs(t)) \* (1 + alpha \* np.abs(t))

# Семивариограмма gamma(t) = R(0) - R(t)

def semivariogram(t, D, alpha):

return D - covariance\_func(t, D, alpha)

# Спектральная плотность f(lambda)

def spectral\_density(lambda\_val, D, alpha):

return (D / np.pi) \* (2 \* alpha\*\*3) / (alpha\*\*2 + lambda\_val\*\*2)\*\*2

# Построение графиков

plt.figure(figsize=(18, 12))

# Графики для разных D (alpha=3)

plt.subplot(3, 3, 1)

for D in D\_values:

plt.plot(t, covariance\_func(t, D, 3.0), label=f'D={D}')

plt.title('R(t) для разных D (α=3)')

plt.xlabel('t')

plt.ylabel('R(t)')

plt.legend()

plt.subplot(3, 3, 2)

for D in D\_values:

plt.plot(t, semivariogram(t, D, 3.0), label=f'D={D}')

plt.title('γ(t) для разных D (α=3)')

plt.xlabel('t')

plt.ylabel('γ(t)')

plt.legend()

plt.subplot(3, 3, 3)

for D in D\_values:

plt.plot(lambda\_vals, spectral\_density(lambda\_vals, D, 3.0), label=f'D={D}')

plt.title('f(λ) для разных D (α=3)')

plt.xlabel('λ')

plt.ylabel('f(λ)')

plt.legend()

# Графики для разных alpha (фиксируем D=1)

plt.subplot(3, 3, 4)

for alpha in alpha\_values:

plt.plot(t, covariance\_func(t, 1.0, alpha), label=f'α={alpha}')

plt.title('R(t) для разных α (D=1)')

plt.xlabel('t')

plt.ylabel('R(t)')

plt.legend()

plt.subplot(3, 3, 5)

for alpha in alpha\_values:

plt.plot(t, semivariogram(t, 1.0, alpha), label=f'α={alpha}')

plt.title('γ(t) для разных α (D=1)')

plt.xlabel('t')

plt.ylabel('γ(t)')

plt.legend()

plt.subplot(3, 3, 6)

for alpha in alpha\_values:

plt.plot(lambda\_vals, spectral\_density(lambda\_vals, 1.0, alpha), label=f'α={alpha}')

plt.title('f(λ) для разных α (D=1)')

plt.xlabel('λ')

plt.ylabel('f(λ)')

plt.legend()

plt.tight\_layout()

plt.show()