

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Московский государственный технический
университет имени Н.Э. Баумана»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ФУНДАМЕНТАЛЬНЫЕ НАУКИ
КАФЕДРА
«ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА И МАТЕМАТИЧЕСКАЯ ФИЗИКА»

Направление: **Математика и компьютерные науки**

Дисциплина: Теория случайных процессов

Домашняя работа №4

«Моделирование гауссовского процесса с данной автоковариационной функцией»

Группа ФН11-63Б

Вариант 2

Студент: Айгистова Д.Р.

Преподаватель: Облакова Т.В.

Оценка:

Москва 2023

Задание

На отрезке $[0, T]$ с шагом h смоделируйте n траекторий гауссовского процесса ξ_t с заданным математическим ожиданием $m(t)$ и заданной автоковариационной функцией $K(t_1, t_2)$.

Выведите на печать две-три траектории в месте с графиком $m(t)$.

Выберите несколько пар сечений построенного процесса (для далёких значений t_1 и t_2 , для близких и соседних). Постройте для выбранных пар сечений диаграммы рассеяния, вычислите выборочные коэффициенты корреляции, постройте 95% доверительные интервалы. Сравните выборочные и теоретические значения коэффициентов корреляции.

Для выбранных сечений ξ_{t_i} постройте гистограммы относительных частот, совмещённые с теоретической плотностью распределения СВ ξ_{t_i} .

Входные данные:

Интервал: $[0, 6]$

Шаг: 0.05

Число траекторий: 180

Математическое ожидание $m(t)$: $m(t) = 1 - 0.2t$

Автоковариационная функция $K(t_1, t_2) = K(\tau)$, где $t_2 - t_1 = \tau$: $K(\tau) = 3e^{-2|\tau|}(1 + 2|\tau|)$

Решение

1. Реализация разложения Холецкого

Для разложения Холецкого можно использовать встроенную функцию библиотеки NumPy `linalg.cholesky()`.

Или реализовать свою функцию, исходя из следующего алгоритма:

Пусть $\Sigma = (\sigma_{ij}), i, j = \overline{0, k}$ – симметричная невырожденная положительно определённая матрица. Тогда имеет место разложение Холецкого:

$$\Sigma = L \cdot L^T,$$

где $L = (l_{ij})$ – нижнетреугольная матрица.

Произведём соотнесение элементов матриц Σ и L :

Первый столбец:

$$\begin{cases} l_{00}^2 = \sigma_{00} \\ l_{10}l_{00} = \sigma_{10} \\ \dots \\ l_{k0}l_{00} = \sigma_{k0} \end{cases} \Rightarrow \begin{cases} l_{00} = \sqrt{\sigma_{00}} \\ l_{10} = \frac{\sigma_{10}}{l_{00}} \\ \dots \\ l_{k0} = \frac{\sigma_{k0}}{l_{00}} \end{cases} \Rightarrow \begin{cases} l_{00} = \sqrt{\sigma_{00}} \\ l_{j0} = \frac{\sigma_{j0}}{l_{00}}, j = \overline{0, k}. \end{cases}$$

Второй столбец:

$$\begin{cases} l_{10}^2 + l_{11}^2 = \sigma_{11} \\ l_{20}l_{10} + l_{21}l_{11} = \sigma_{21} \\ \dots \\ l_{k0}l_{10} + l_{k1}l_{11} = \sigma_{k1} \end{cases} \Rightarrow \begin{cases} l_{11} = \sqrt{\sigma_{11} - l_{10}^2} \\ l_{21} = \frac{\sigma_{21} - l_{10}l_{20}}{l_{11}} \\ \dots \\ l_{k1} = \frac{\sigma_{k1} - l_{10}l_{k0}}{l_{11}} \end{cases} \Rightarrow \begin{cases} l_{11} = \sqrt{\sigma_{11} - l_{10}^2} \\ l_{j1} = \frac{\sigma_{j1} - l_{10}l_{j0}}{l_{11}}, j = \overline{2, k}. \end{cases}$$

N -ый столбец:

$$\begin{cases} l_{N0}^2 + l_{N1}^2 + \dots + l_{NN}^2 = \sigma_{NN} \\ l_{N+1,0}l_{N0} + \dots + l_{N+1,N}l_{NN} = \sigma_{N+1,N} \\ \dots \\ l_{k0}l_{N0} + \dots + l_{kN}l_{NN} = \sigma_{kN} \end{cases} \Rightarrow \begin{cases} l_{NN} = \sqrt{\sigma_{NN} - l_{N0}^2 - \dots - l_{N,N-1}^2} \\ l_{jN} = \frac{\sigma_{jN} - l_{j0}l_{N0} - l_{j1}l_{N1} - \dots - l_{j,N-1}l_{N,N-1}}{l_{11}}, j = \overline{N+1, k}. \end{cases}$$

Тогда элементы матрицы L можно вычислить, начиная с верхнего левого угла матрицы по формулам:

$$\begin{aligned} l_{00} &= \sqrt{\sigma_{00}}; \\ l_{j0} &= \frac{\sigma_{j0}}{l_{00}}, j = \overline{1, k}; \\ l_{ii} &= \sqrt{\sigma_{ii} - \sum_{p=1}^{i-1} l_{ip}^2}, i = \overline{1, k}; \\ l_{ji} &= \frac{1}{l_{ii}}(\sigma_{ji} - \sum_{p=1}^{i-1} l_{ip}l_{jp}), i = \overline{1, k-1}, j = \overline{i, k}. \end{aligned}$$

Код реализации алгоритма Холецкого:

```
def MyCholesky(A):
    n = len(A)
    L = np.zeros((n, n))

    for i in range(n):
        for j in range(i+1):
            s = sum(L[i][k] * L[j][k] for k in range(j))
            if (i == j):
                L[i][j] = np.sqrt(A[i][i] - s)
            else:
                L[i][j] = (1.0 / L[j][j] * (A[i][j] - s))
    return L
```

Убедимся, что функция `def MyCholesky()` и встроенный метод `linalg.cholesky()` работают одинаково на тестовых данных:

```
A = np.array([[4, 12, -16],
               [12, 37, -43],
               [-16, -43, 98]])
L = MyCholesky(A)
print(L)
```

```
[[ 2.  0.  0.]
 [ 6.  1.  0.]
 [-8.  5.  3.]]
```

```
print(np.linalg.cholesky(A))
```

```
[[ 2.  0.  0.]
 [ 6.  1.  0.]
 [-8.  5.  3.]]
```

Рис. 1: Пример работы функции `def MyCholesky()` и встроенного метода `linalg.cholesky()` на тестовых данных

2. Моделирование Гауссовского процесса

Найдём размерность k моделируемой траектории $\{\xi_0, \xi_h, \dots, \xi_{kh}\}$:

$$k = \lfloor \frac{b-a}{h} \rfloor = \frac{6-0}{0.05} = 120,$$

где $\lfloor \cdot \rfloor$ – обозначение целой части числа.

Вычислим вектор математических ожиданий $M\xi = m$ и ковариационную матрицу Σ :

$$m = (m_i) = m(i), i = \overline{0, k}^T, \Sigma = (\sigma_{ij}) = cov(\xi_{hi}, \xi_{hj}) = K(|i-j| \cdot h), i, j = \overline{0, k}.$$

Выведем срез вектора m и ковариационной матрицы Σ для $i, j = \overline{0, 10}$:

$$m_i = 1.0, 0.99, 0.98, 0.97, 0.96, 0.95, 0.94, 0.93, 0.92, 0.91^T,$$

$$\sigma_{ij} = \begin{pmatrix} 3.0 & 2.986 & 2.9474 & 2.8892 & 2.8153 & 2.7294 & 2.6343 & 2.5326 & 2.4264 & 2.3174 \\ 2.986 & 3.0 & 2.986 & 2.9474 & 2.8892 & 2.8153 & 2.7294 & 2.6343 & 2.5326 & 2.4264 \\ 2.9474 & 2.986 & 3.0 & 2.986 & 2.9474 & 2.8892 & 2.8153 & 2.7294 & 2.6343 & 2.5326 \\ 2.8892 & 2.9474 & 2.986 & 3.0 & 2.986 & 2.9474 & 2.8892 & 2.8153 & 2.7294 & 2.6343 \\ 2.8153 & 2.8892 & 2.9474 & 2.986 & 3.0 & 2.986 & 2.9474 & 2.8892 & 2.8153 & 2.7294 \\ 2.7294 & 2.8153 & 2.8892 & 2.9474 & 2.986 & 3.0 & 2.986 & 2.9474 & 2.8892 & 2.8153 \\ 2.6343 & 2.7294 & 2.8153 & 2.8892 & 2.9474 & 2.986 & 3.0 & 2.986 & 2.9474 & 2.8892 \\ 2.5326 & 2.6343 & 2.7294 & 2.8153 & 2.8892 & 2.9474 & 2.986 & 3.0 & 2.986 & 2.9474 \\ 2.4264 & 2.5326 & 2.6343 & 2.7294 & 2.8153 & 2.8892 & 2.9474 & 2.986 & 3.0 & 2.986 \\ 2.3174 & 2.4264 & 2.5326 & 2.6343 & 2.7294 & 2.8153 & 2.8892 & 2.9474 & 2.986 & 3.0 \end{pmatrix}$$

Для моделирования траектории гауссовского процесса необходимо применить разложение Холецкого к ковариационной матрице Σ для получения нижнетреугольной матрицы L :

$$\Sigma = L \cdot L^T.$$

Выведем срез матрицы L для $i, j = \overline{0, 10}$:

$$l_{ij} = \begin{pmatrix} 1.7321 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 1.7239 & 0.1674 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 1.7017 & 0.3127 & 0.0804 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 1.6681 & 0.4288 & 0.1659 & 0.0784 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 1.6254 & 0.52 & 0.2344 & 0.1628 & 0.0783 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 1.5758 & 0.5899 & 0.2884 & 0.2304 & 0.1626 & 0.0783 & 0.0 & 0.0 & 0.0 & 0.0 \\ 1.5209 & 0.6419 & 0.3299 & 0.2837 & 0.2301 & 0.1626 & 0.0783 & 0.0 & 0.0 & 0.0 \\ 1.4622 & 0.6786 & 0.361 & 0.3247 & 0.2834 & 0.2301 & 0.1626 & 0.0783 & 0.0 & 0.0 \\ 1.4009 & 0.7025 & 0.3831 & 0.3554 & 0.3244 & 0.2833 & 0.2301 & 0.1626 & 0.0783 & 0.0 \\ 1.338 & 0.7157 & 0.3978 & 0.3773 & 0.355 & 0.3244 & 0.2833 & 0.2301 & 0.1626 & 0.0783 \end{pmatrix}$$

Посчитаем невязку полученного разложения:

$$\max(L \cdot L^T - \Sigma) = 8.8818 \cdot 10^{-16}.$$

По теореме о конечномерных распределениях гауссовского процесса для центрированной последовательности η получим:

$$\text{cov}(\eta, \eta) = M\eta\eta^T = ML\varepsilon\varepsilon^T L^T = LM\varepsilon\varepsilon^T L^T = LL^T = \Sigma.$$

Следовательно, матрица Σ будет ковариационной для центрированной последовательности η :

$$\eta = L \cdot \varepsilon,$$

где $\eta = (\eta_0, \eta_h, \eta_{2h}, \dots, \eta_{kh})^T$, $\varepsilon = \varepsilon_0, \varepsilon_1, \dots, \varepsilon_k^T$ – последовательность из $k + 1$ независимой стандартной гауссовской случайной величины.

Для завершения моделирования траектории гауссовского процесса остаётся добавить нужное математическое ожидание:

$$\xi_i = \eta_i + m_i, i = \overline{0, k}.$$

Внизу представлен код, реализующий моделирование траектории гауссовского процесса:

```

def modeling(a, b, h, m, K):
    # Находим размерность k случайного вектора
    k = np.modf((b - a) / h)[1]
    k = int(k)
    # Вычисляем вектор математических ожиданий
    expected = [m(i * h) for i in range(k + 1)]
    # Вычисляем матрицу ковариаций
    Sigma = []
    for i in range(k + 1):
        tmp = []
        for j in range(k + 1):
            tmp.append(K(abs(i - j) * h))
        Sigma.append(tmp)
    # Применяем разложение Холецкого к ковариационной матрице
    Sigma = np.array(Sigma)
    L = MyCholesky(Sigma)
    # Ищем L транспонированную
    L_t = L.T
    # Считае невязку полученного разложения
    LL = L.dot(L_t)
    residual = np.abs(LL - Sigma)
    max_residual = np.max(residual)
    # Генерируем последовательность независимых стандартных Гауссовских величин
    eps = np.random.normal(size = k + 1)
    # Центрируем последовательность
    etta = L @ eps
    # Добавляем необходимое математическое ожидание
    ksi = []
    for i in range(k + 1):
        ksi.append(etta[i] + expected[i])
    return ksi

```

3. Визуализация траекторий

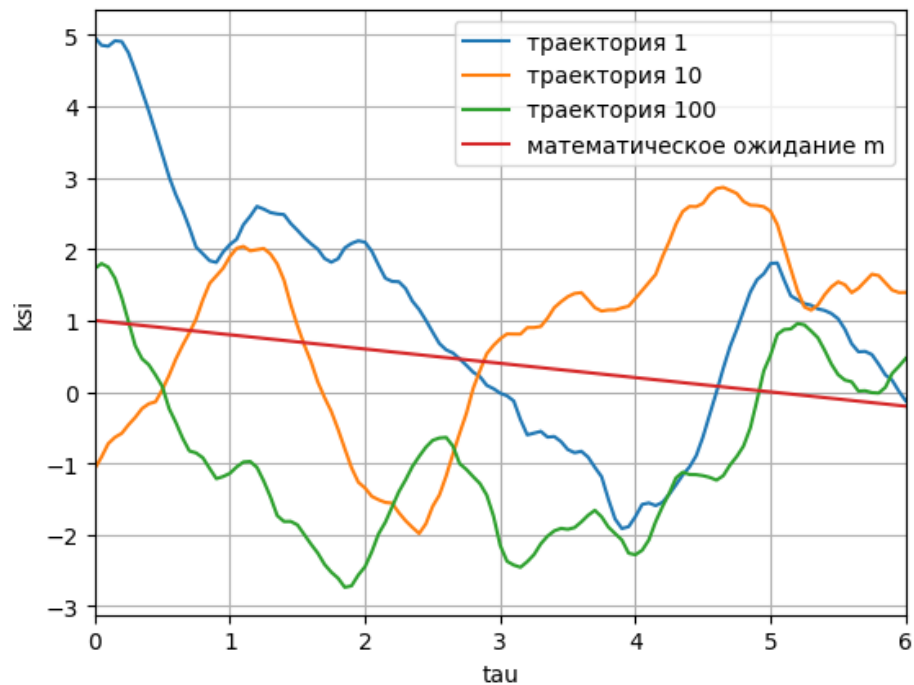


Рис. 2: Визуализация 1, 10 и 100 смоделированных траекторий гауссовского процесса и теоретического математического ожидания $m(\tau)$

Код для визуализации:

```
def viz(a, b, h, m, one = 1, two = 10, tree = 100):  
    x = np.arange(a, b+h, h)  
    y1 = tracks[one - 1]  
    plt.plot(x, y1, label = f"траектория {one}")  
    y2 = tracks[two - 1]  
    plt.plot(x, y2, label = f"траектория {two}")  
    y3 = tracks[tree - 1]  
    plt.plot(x, y3, label = f"траектория {tree}")  
    plt.plot(x, m(x), label = f"математическое ожидание m")  
    plt.grid(which = "major")  
    plt.legend()  
    plt.xlabel("tau")  
    plt.ylabel("ksi")  
    plt.xlim([a, b])  
    plt.show()  
viz(a, b, h, m)
```

4. Проверка качества моделирования траекторий

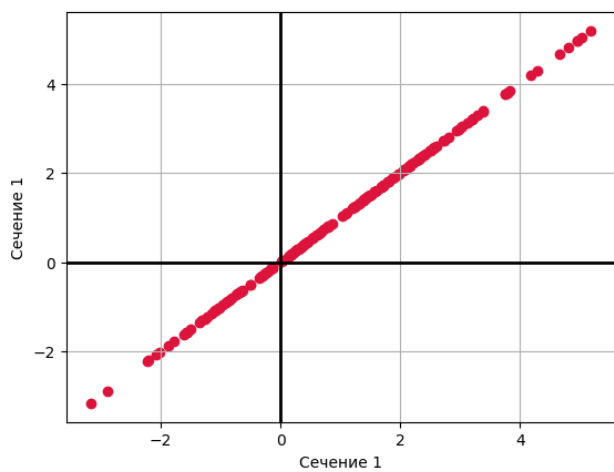
N -ым сечением S^N процесса ξ будем называть последовательность размерности n , состоящую из N -ых элементов каждой траектории, т.е.:

$$S^N = \{\Xi_{iN}, i = \overline{0, n-1}\} = \xi_N^1, \xi_N^2, \dots, \xi_N^n.$$

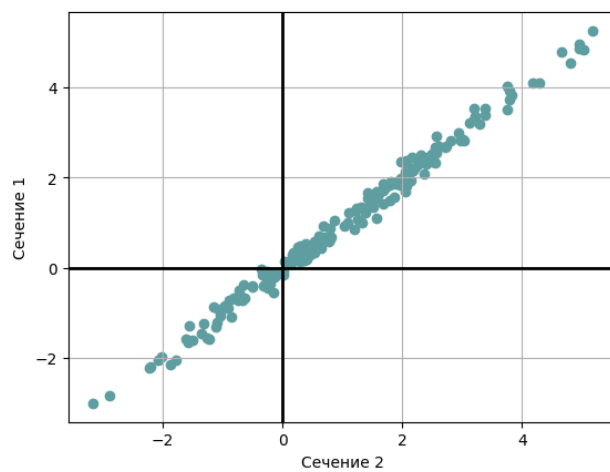
Выберем несколько пар сечений построенного процесса: $(1, 1)$, $(1, 2)$, $(10, 2)$, $(100, 2)$ и построим для них диаграмму рассеяния (*Scatterplot*) (см. Рис.3):

```
# Проверим качество моделирования траекторий
tracks = np.array(tracks)
tracks_t = tracks.T # Для работы с сечениями

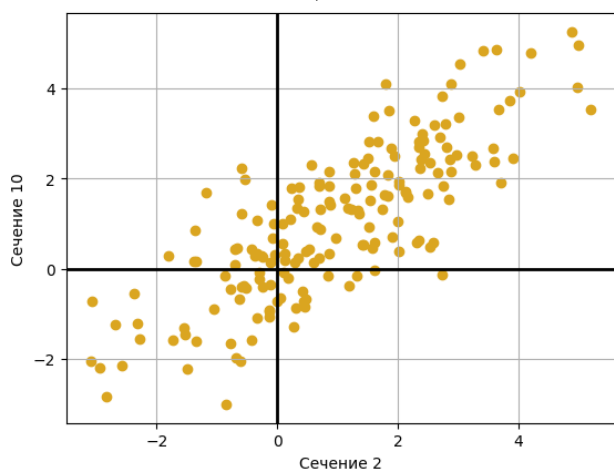
def scatter_par(a, b, h, m, K, main = 2, sub = 1, color = "crimson"):
    plt.scatter(x = tracks_t[sub - 1], y = tracks_t[main - 1], color = color)
    plt.xlabel(f"Сечение {main}")
    plt.ylabel(f"Сечение {sub}")
    plt.axhline(0, color='black', linewidth=2)
    plt.axvline(0, color='black', linewidth=2)
    plt.grid()
    plt.show()
```

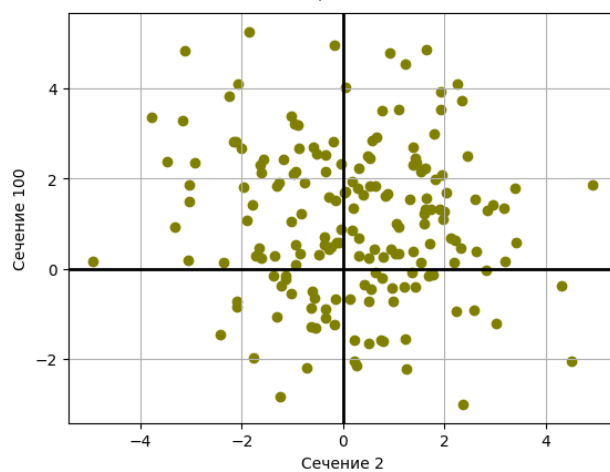
a)



b)



c)



d)

Рис. 3: Диаграммы рассеяния для сечений: a) (1,1), b) (1,2), c) (10,2), d) (100,2).

Видно, что для близких сечений диаграмма рассеяния менее разбросана по плоскости, что соответствует ковариационной матрице.

Вычислим теоретические и выборочные коэффициенты корреляции для построенных сечений процесса по формулам:

$$r_{NM} = \frac{\sigma_{NM}}{\sqrt{\sigma_{NN}\sigma_{MM}}},$$

$$\hat{r}_{NM} = \frac{\sum_{i=1}^n (S_i^N - \bar{S}^N)(S_i^M - \bar{S}^M)}{\sqrt{\sum_{i=1}^n (S_i^N - \bar{S}^N)^2 \sum_{i=1}^n (S_i^M - \bar{S}^M)^2}}, \quad N, M = \overline{1, n}.$$

Ниже приведён фрагмент кода, реализующий эти вычисления:

```
def teor_cor(a, b, h, K, main, sub):
    main -= 1
    sub -= 1
    r = Sigma[main][sub] / np.sqrt(Sigma[main][main] * Sigma[sub][sub])
    return r

def emper_cor(n, main, sub):
    chisl = 0
    sum_1 = 0
    sum_2 = 0
    main -= 1
    sub -= 1
    for i in range(n):
        chisl += ((tracks_t[main][i] - np.mean(tracks_t[main])) *
                  (tracks_t[sub][i] - np.mean(tracks_t[sub])))
        sum_1 += (tracks_t[main][i] - np.mean(tracks_t[main])) ** 2
        sum_2 += (tracks_t[sub][i] - np.mean(tracks_t[sub])) ** 2
    znam = np.sqrt(sum_1 * sum_2)
    r = chisl / znam
    return r
```

Получим:

Корреляция	(2, 1)	(2, 10)	(2, 100)
r	0.9953	0.8088	0.0006
\hat{r}	0.9954	0.8091	-0.0923

Построим 95% доверительные интервалы на основе выборочных коэффициентов корреляции по формуле:

$$\rho_{\mp}(\hat{r}) = \tanh \frac{\log(\frac{1+\hat{r}}{1-\hat{r}})}{2} - \frac{\hat{r}}{2(n-1)} \mp \frac{u_{0.975}}{\sqrt{n-2}}.$$

Ниже приведён фрагмент кода, реализующий эти вычисления:

```
def intervals(n):
    confidence = 0.95
    quantile = norm.ppf((1 + confidence) / 2)
    rho = []
    dist = norm()
    for i in range(3):
        rho_i_m = np.tanh(np.log((1 + emper[i]) / (1 - emper[i]))) / 2 -
                        emper[i] / (2 * (n - 1)) -
                        quantile / np.sqrt(n - 2))
        rho_i_p = np.tanh(np.log((1 + emper[i]) / (1 - emper[i]))) / 2 -
                        emper[i] / (2 * (n - 1)) +
                        quantile / np.sqrt(n - 2))
```

Получим значения 95% доверительных интервалов для рассматриваемых пар сечений:

(N, M)	$\rho_-(\hat{r})$	r	$\rho_+(\hat{r})$
(2, 1)	0.9938	0.9953	0.9965
(2, 10)	0.7510	0.8088	0.8535
(2, 100)	-0.2348	0.0006	0.0545

Видно, что теоретические коэффициенты корреляции удовлетворяют полученным 95% доверительным интервалам.

Следовательно, можно сделать вывод о том, что гауссовский процесс смоделирован с достаточно хорошей точностью.

5. Проверка элементов сечений на принадлежность нормальному закону распределения

Для того, чтобы убедиться в том, что сечения рассматриваемых траекторий действительно распределены согласно нормальному закону распределения, необходимо проверить, что $\Xi_{iN} \sim N(m_N, \sqrt{\sigma_{NN}})$. Для этого построим гистограммы относительных частот N -ых сечений и наложим их на функции плотности нормального распределения с характеристиками m_N и $\sqrt{\sigma_{NN}}$.

Визуализируем эти графики для сечений $N = 1, 50, 100$:

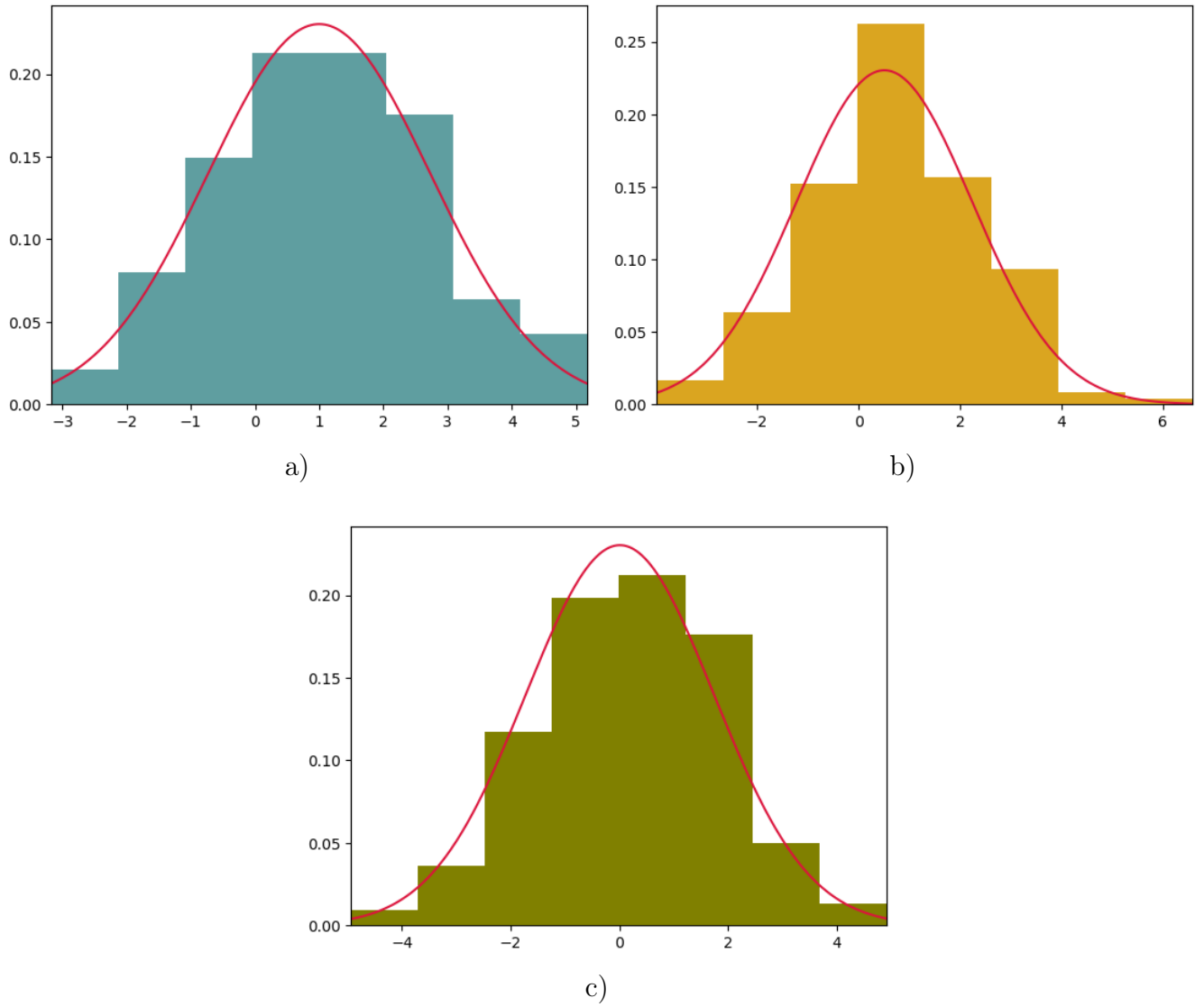


Рис. 4: Совмещённые графики гистограмм относительных частот N -ых распределений и плотностей нормального распределения с параметрами m_N и $\sqrt{\sigma_{NN}}$ для: а) $N = 1$; б) $N = 50$; в) $N = 100$.

Посчитаем выборочные характеристики рассматриваемых сечений и сравним их с теоретическими:

N	\bar{S}^N	$\sqrt{S^{2N}}$	m_N	$\sqrt{\sigma_{NN}}$
1	1.0567	2.8233	1.0	1.7321
50	0.7807	2.9395	0.51	1.7321
100	0.2144	2.9431	0.01	1.7321

Таким образом, элементы сечений Ξ_{iN} гауссовского процесса ξ распределены нормально с математическим ожиданием m_N и дисперсией σ_{NN} .

6. Проверка качества моделирования траекторий с помощью построения поверхностей выборочных коэффициентов корреляции и 95% доверительного интервала

Визуализируем полученные матрицы нижней границы ρ_- и верхней границы ρ_+ 95% доверительного интервала и матрицу теоретических коэффициентов корреляции r .

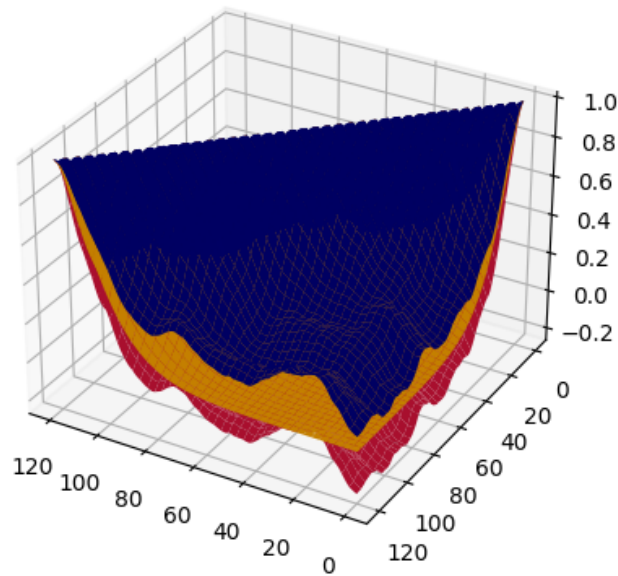


Рис. 5: Теоретический коэффициент корреляции и 95% доверительный интервал для смоделированных траекторий гауссовского процесса.

Видно, что теоретические коэффициенты корреляции лежат внутри доверительного интервала.

7. Построение коррелограммы

Построим коррелограмму (*Heatmap*) для сечений смоделированного гауссовского процесса:

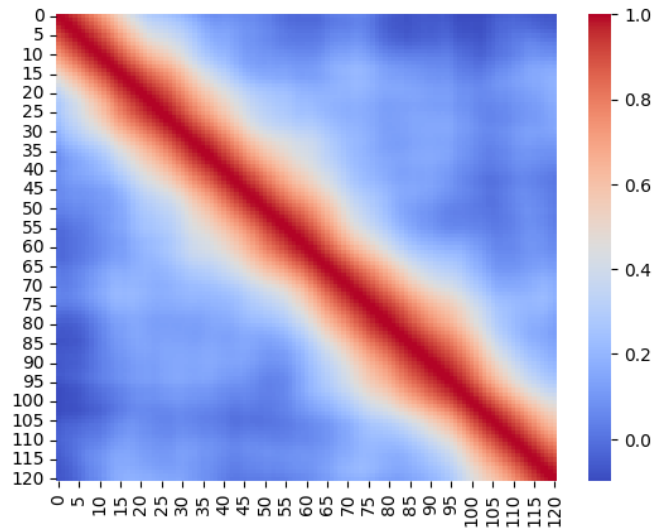


Рис. 6: *Коррелограмма сечений*

По коррелограмме видно, что наибольшие значения корреляции имеют соседние сечения, а при удалении сечений друг от друга корреляция уменьшается до нуля.

Выводы

В ходе выполнения задания мной были реализованы следующие алгоритмы: алгоритм разложения Холецкого, алгоритм моделирования гауссовского процесса и визуализации его траекторий, алгоритм построения диаграмм рассеяния для сечений гауссовского процесса, алгоритм вычисления выборочных и теоретических коэффициентов корреляции и построения по ним 95% доверительных интервалов для проверки качества моделирования процесса. Дополнительно я произвела проверку элементов сечений рассматриваемых траекторий на принадлежность нормальному закону распределения, рассмотрела расширенный способ проверки моделирования процесса с помощью визуализации матриц ρ 95% доверительного интервала и матрицы выборочных коэффициентов корреляции и построила коррелограмму для сечений смоделированного гауссовского процесса.

Подводя итоги проделанной работы, можно сказать, что с помощью предложенного алгоритма можно с достаточно высокой точностью смоделировать гауссовский процесс с заданным математическим ожиданием и заданной автоковариационной функцией.

Промежуточные выводы сформулированы по ходу выполнения работы и представлены в отчёте.