Лабораторная работа № 2

Тема: Случайные процессы

Ошлаков К.К.
ИА-232

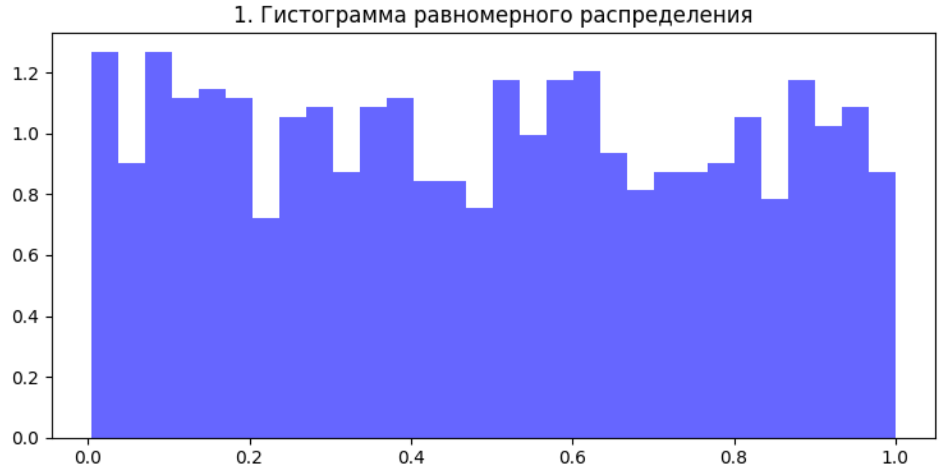
1. Проверка центральной предельной теоремы

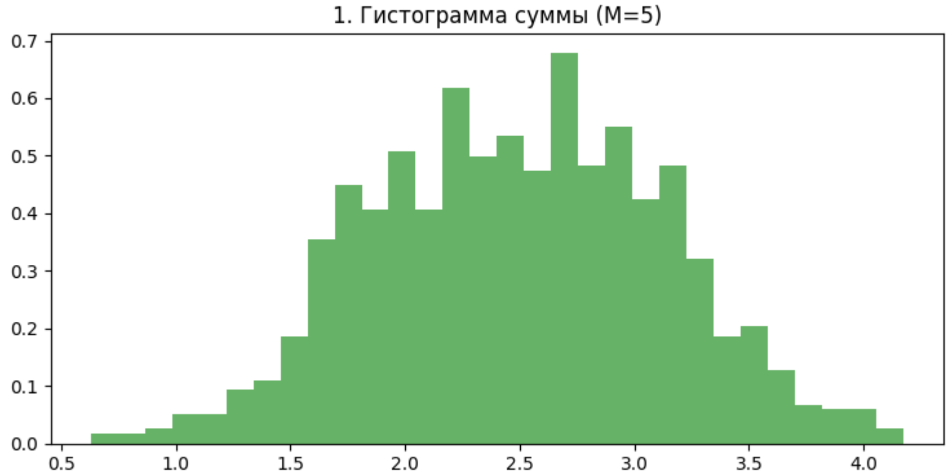
Цель: проверить, что распределение суммы большого числа независимых случайных величин (СВ) с произвольным законом распределения стремится к нормальному.

Методика выполнения:
- Сгенерирована выборка из 1000 равномерно распределенных СВ на интервале [0, 1].
- Вычислена сумма M = 5 таких СВ для каждой реализации.
- Построены гистограммы распределений:
  - Исходного равномерного распределения.
  - Распределения суммы Y_n.

Результаты:

1. Гистограмма равномерного распределения показывает характерную прямоугольную форму (рис. 1).
2. Гистограмма суммы Y_n демонстрирует приближение к нормальному распределению с увеличением числа слагаемых (рис. 2).





Вычисление автокорреляционной функции (АКФ) по множеству реализаций

Цель: изучить корреляционные свойства нормального случайного процесса (СП) и вычислить АКФ.

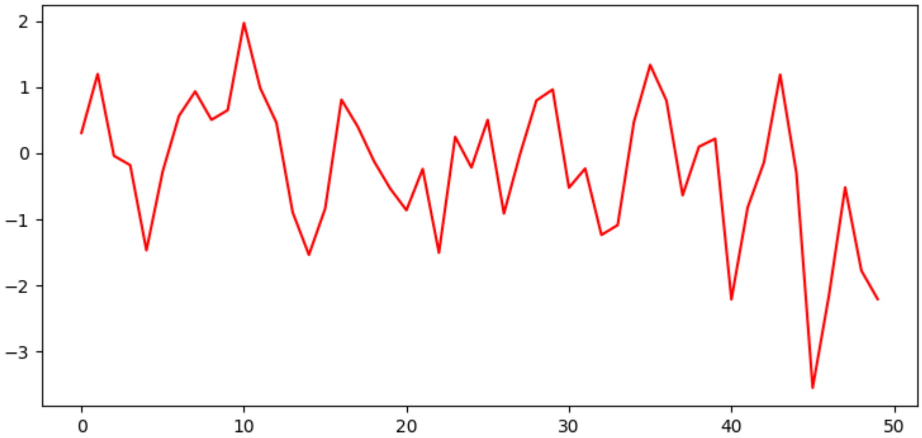
Методика выполнения:

- Сгенерировано 1000 реализаций нормального СП длиной 100 точек каждая.
- Каждая реализация была преобразована с использованием фильтра [1, 0.7, 0.3, 0.1, 0.05], чтобы получить коррелированный СП.
- Для анализа выбраны значения СВ в момент времени t_0 = 50 (сечение СП).
- Построены:
  - Временная диаграмма отрезка одной реализации (рис. 3).
  - Гистограмма распределения значений СВ в сечении t_0 (рис. 4).
- Вычислена АКФ для различных значений временного сдвига \tau = \{0, 3, 5, 7\}.

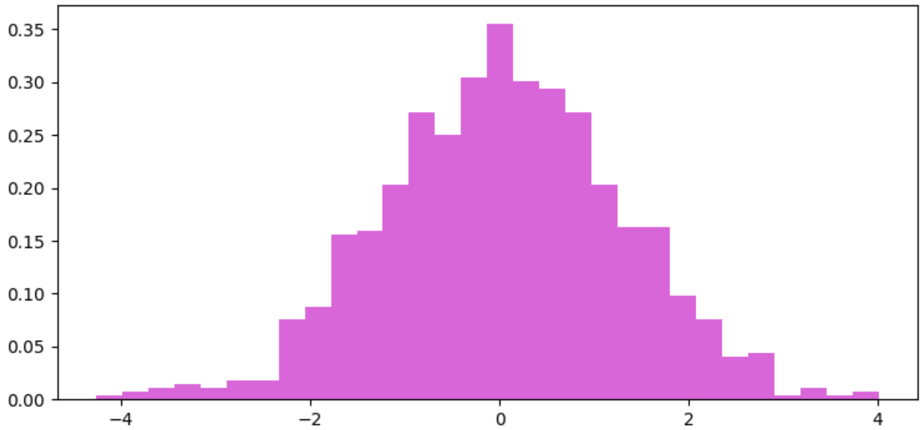
Результаты:

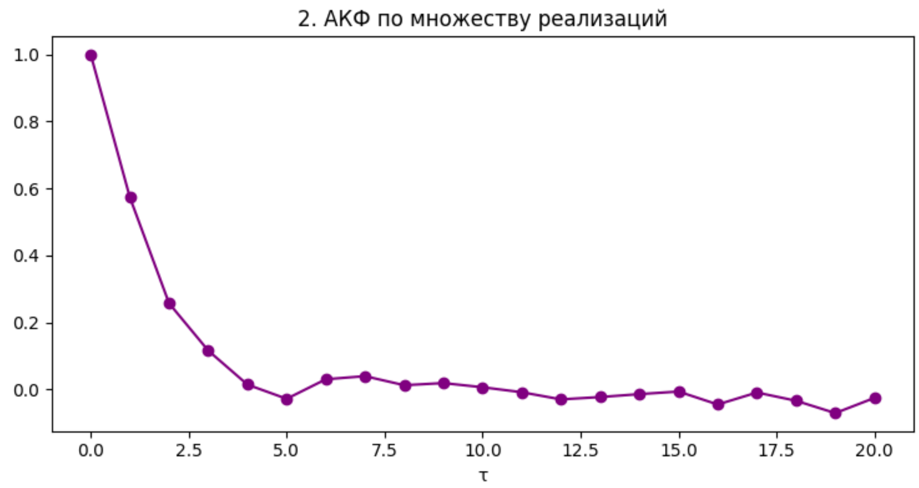
1. Временная диаграмма показывает коррелированный характер процесса (рис. 3).
2. Гистограмма сечения подтверждает нормальное распределение значений СВ в момент времени t_0 (рис. 4).
3. График АКФ демонстрирует экспоненциальное затухание корреляции с увеличением tau (рис. 5).

2. Отрезок реализации СП



2. Гистограмма сечения t0





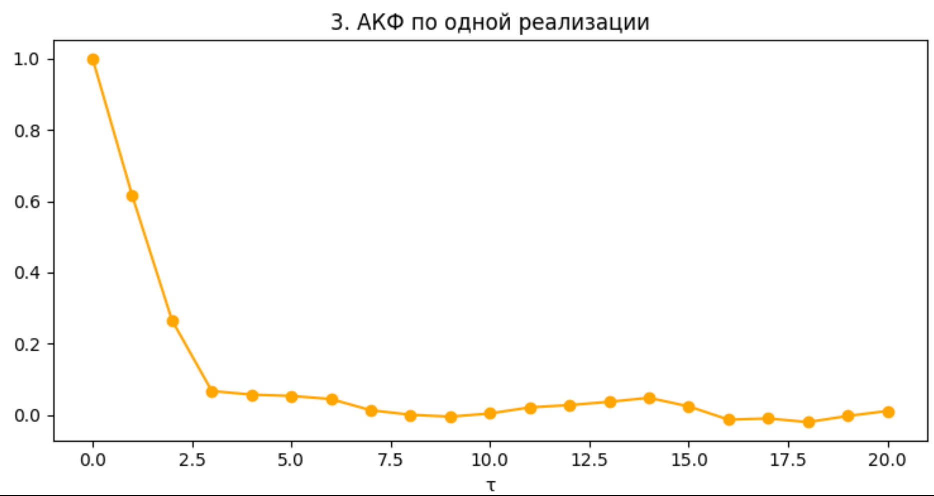
Интервал корреляции, определенный как \tau_0 = \sum_{\tau} B(\tau) / B(0), составил:
$\tau_0 = 1.77$

3. Вычисление АКФ по одной реализации

Цель: проверить возможность вычисления АКФ путем усреднения одной длинной реализации.

Методика выполнения:
- Сгенерирована одна длинная реализация нормального СП длиной 1000 точек.
- Реализация была преобразована с использованием того же фильтра [1, 0.7, 0.3, 0.1, 0.05].
- АКФ вычислялась по формуле:
B(n) = \frac{1}{N-n} \sum_{k=1}^{N-n} x_k x_{k+n},
где n — временной сдвиг.

Результаты:
График АКФ, полученный по одной реализации, совпадает с графиком АКФ, вычисленным по множеству реализаций (рис. 6). Это подтверждает корректность метода усреднения одной реализации.




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

fig, axs = plt.subplots(3, 2, figsize=(15, 12))
fig.suptitle('Результаты анализа СВ и СП')
```

```python
# 1: Проверка ЦПТ
np.random.seed(42)
N = 1000
M = 5

# Равномерное распределение
xn_uniform = np.random.uniform(0, 1, N)
axs[0, 0].hist(xn_uniform, bins=30, density=True, alpha=0.6, color='b')
axs[0, 0].set_title('1. Гистограмма равномерного распределения')

# Сумма M равномерных СВ
Yn = np.sum(np.random.uniform(0, 1, (N, M)), axis=1)
axs[0, 1].hist(Yn, bins=30, density=True, alpha=0.6, color='g')
axs[0, 1].set_title('1. Гистограмма суммы (M=5)')

# 2: АКФ по множеству реализаций
m, sigma = 0, 1
num_realizations = 1000
realization_length = 100
kernel = [1, 0.7, 0.3, 0.1, 0.05]

# Генерация реализаций
realizations = []
section_t0 = []
for _ in range(num_realizations):
    xn = np.random.normal(m, sigma, realization_length)
    xn1 = np.convolve(xn, kernel, mode='same')
    realizations.append(xn1)
    section_t0.append(xn1[50])

# Временная диаграмма
axs[1, 0].plot(realizations[0][:50], 'r')
axs[1, 0].set_title('2. Отрезок реализации СП')

# Гистограмма сечения
axs[1, 1].hist(section_t0, bins=30, density=True, alpha=0.6, color='m')
axs[1, 1].set_title('2. Гистограмма сечения t0')

# Вычисление АКФ
max_tau = 20
taus = np.arange(max_tau + 1)
B_many = []
for tau in taus:
    products = []
    for realization in realizations:
        if 50 + tau < len(realization):
            products.append(realization[50] * realization[50 + tau])
    B_many.append(np.mean(products) if products else 0)

# Нормировка
```

```python
B0 = B_many[0]
B_many_norm = [b / B0 for b in B_many]
axs[2, 0].plot(taus, B_many_norm, 'o-', color='purple')
axs[2, 0].set_title('2. АКФ по множеству реализаций')
axs[2, 0].set_xlabel('τ')

# 3: АКФ по одной реализации
N_single = 1000
xn_single = np.random.normal(m, sigma, N_single)
xn1_single = np.convolve(xn_single, kernel, mode='same')

max_shift = 20
B_single = []
for n in range(max_shift + 1):
    sum_prod = np.sum(xn1_single[:N_single - n] * xn1_single[n:N_single])
    B_single.append(sum_prod / (N_single - n))

# Нормировка
B0_single = B_single[0]
B_single_norm = [b / B0_single for b in B_single]
axs[2, 1].plot(range(max_shift + 1), B_single_norm, 'o-', color='orange')
axs[2, 1].set_title('3. АКФ по одной реализации')
axs[2, 1].set_xlabel('τ')

plt.tight_layout()
plt.show()

# Интервал корреляции
tau_0 = np.sum(B_many_norm) / B_many_norm[0]
print(f'Интервал корреляции: τ₀ = {tau_0:.2f}')
```