**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:

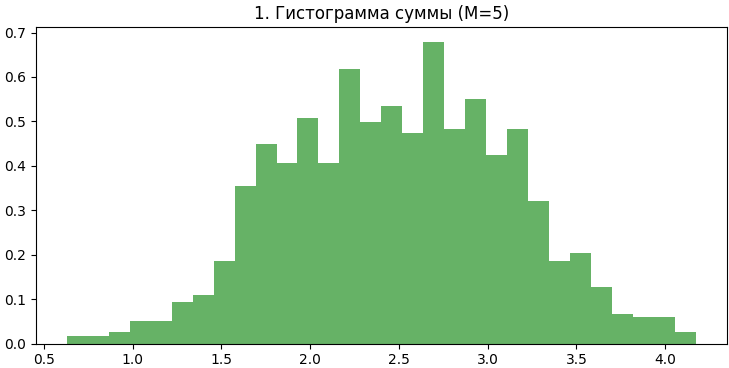
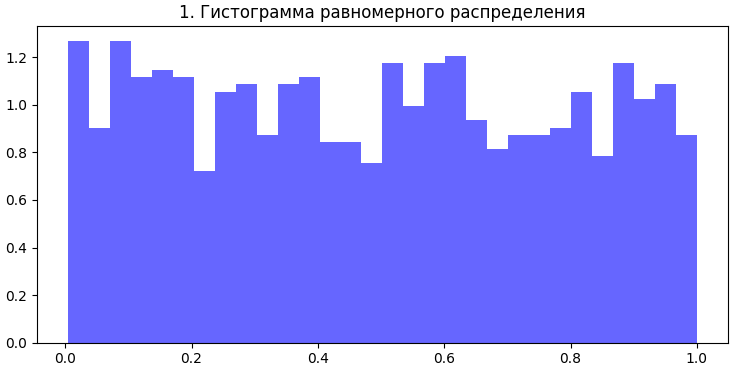
- Исходного равномерного распределения.

- 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).

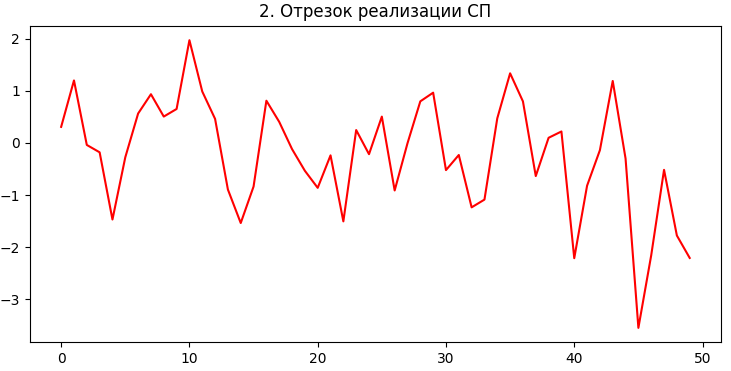
- Вычислена АКФ для различных значений временного сдвига \tau = \{0, 3, 5, 7\}.

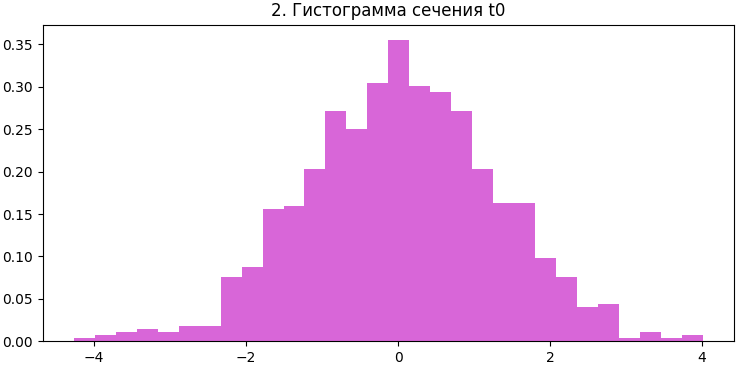
**Результаты:**

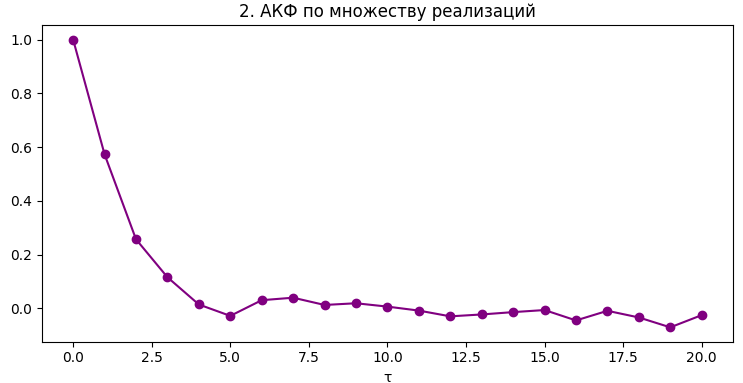
1. Временная диаграмма показывает коррелированный характер процесса (рис. 3).

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).







Интервал корреляции, определенный как \tau\_0 = \sum\_{\tau} B(\tau) / B(0), составил:

τ₀ = 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].

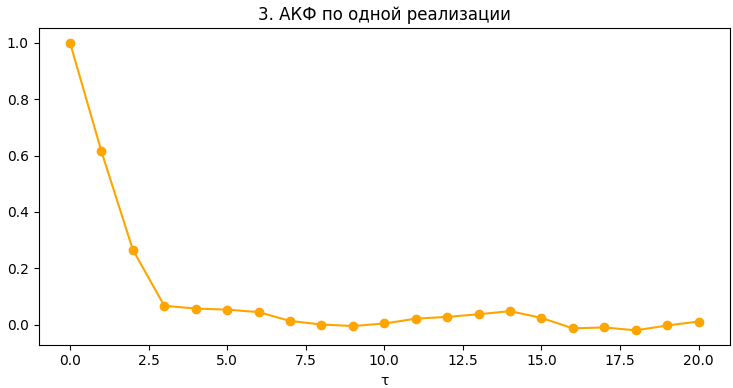
- АКФ вычислялась по формуле:

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')  *# 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*  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}') |