БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра интеллектуальных информационных технологий

ОТЧЁТ  
по лабораторной работе №1  
по дисциплине

«ОБРАБОТКА ИЗОБРАЖЕНИЙ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ»

Выполнил студент гр. 321701

Романов К.В.

Проверил

Сальников Д. А.

Минск 2025

**Тема:** Быстрое преобразование Фурье.

**Задание:**

1. Изучить алгоритм БПФ.

2. Выполнить программную реализацию алгоритма БПФ.

3. На вход подать функцию sin(x) или cos(x) для N частоты и показать правильность работы преобразования

**Листинг кода:**

import cmath

import math

import matplotlib.pyplot as plt

import numpy as np

def fft(x):

N = len(x)

if N <= 1:

return x

even = fft(x[0::2])

odd = fft(x[1::2])

T = [cmath.exp(-2j \* math.pi \* k / N) \* odd[k] for k in range(N // 2)]

return [even[k] + T[k] for k in range(N // 2)] + \

[even[k] - T[k] for k in range(N // 2)]

if \_\_name\_\_ == "\_\_main\_\_":

N = 16

fs = 64 # частота дискретизации

t = [n / fs for n in range(N)]

# Сигнал: x(t) = sin(2π⋅10t) + 0.5sin(2π⋅20t)

x = [math.sin(2 \* math.pi \* 10 \* ti) + 0.5 \* math.sin(2 \* math.pi \* 20 \* ti) for ti in t]

# --- Ручное FFT ---

X\_manual = fft([complex(val, 0) for val in x])

# --- Библиотечное FFT (NumPy) ---

X\_numpy = np.fft.fft(x)

print("Результат БПФ (амплитуда и фаза):\n")

print(f"{'k':>3} {'Частота (Гц)':>15} {'Ампл.(ручн.)':>15} {'Ампл.(numpy)':>15} {'Фаза(рад)':>15}")

print("-" \* 75)

amplitudes\_manual = []

amplitudes\_numpy = []

freqs = []

for k in range(N // 2):

freq = k \* fs / N

amp\_manual = abs(X\_manual[k]) / (N / 2)

amp\_numpy = abs(X\_numpy[k]) / (N / 2)

phase = math.atan2(X\_manual[k].imag, X\_manual[k].real)

freqs.append(freq)

amplitudes\_manual.append(amp\_manual)

amplitudes\_numpy.append(amp\_numpy)

print(f"{k:3d} {freq:15.2f} {amp\_manual:15.4f} {amp\_numpy:15.4f} {phase:15.4f}")

# --- Графики ---

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

# 1. Сигнал во времени

plt.subplot(1, 2, 1)

plt.plot(t, x, marker='o', linestyle='-', color='blue')

plt.title("Сигнал во времени")

plt.xlabel("Время (с)")

plt.ylabel("Амплитуда")

plt.grid(True)

# 2. Сравнение спектров

plt.subplot(1, 2, 2)

plt.plot(freqs, amplitudes\_manual, marker='o', linestyle='-', color='red', label="Самописный FFT")

plt.plot(freqs, amplitudes\_numpy, marker='x', linestyle='--', color='green', label="NumPy FFT")

plt.title("Сравнение амплитудных спектров")

plt.xlabel("Частота (Гц)")

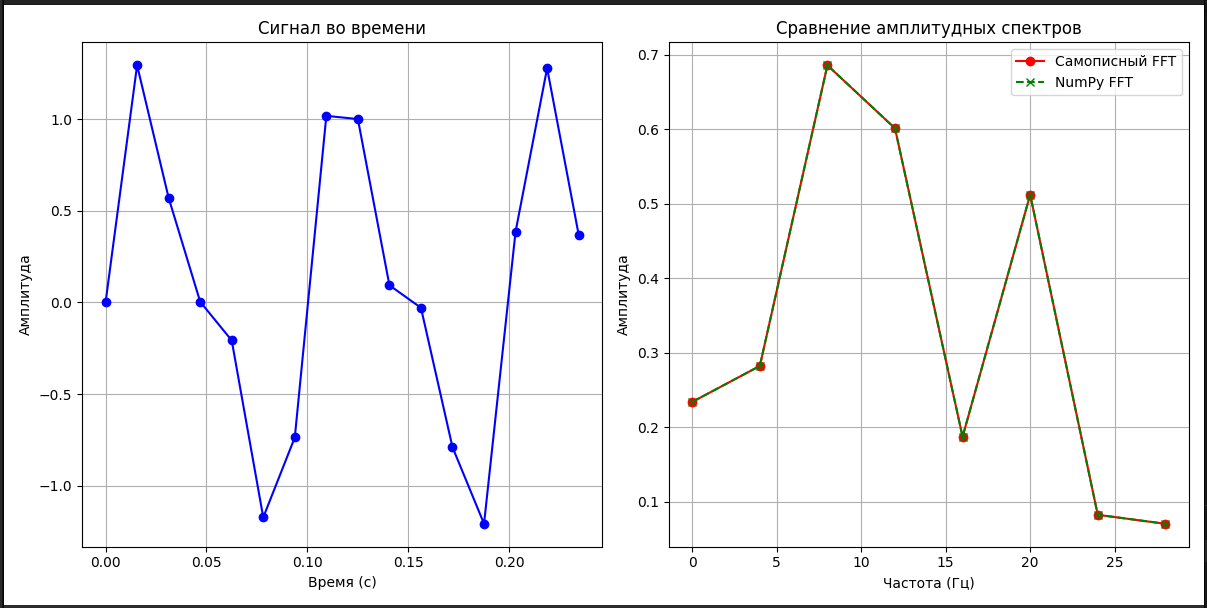
plt.ylabel("Амплитуда")

plt.legend()

plt.grid(True)

plt.tight\_layout()

plt.show()



**Вывод:** в рамках лабораторной работы был изучен и программно реализован на алгоритм быстрого преобразования Фурье, который значительно ускоряет вычисления по сравнению с прямым ДПФ, сокращая сложность с O(N²) до O(N log N).

На примере анализа синусоидального сигнала была продемонстрирована эффективность алгоритма. Спектр, полученный с помощью собственной реализации БПФ, полностью совпал с результатами стандартной функции numpy.fft.fft, что подтвердило корректность работы программы.