Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Теоретическая информатика и компьютерные технологии»

Отчет по лабораторной работе № 7 «Тригонометрическая интерполяция функций с помощью быстрого преобразования Фурье»

по курсу «Численные методы»

Выполнил:

Студент группы ИУ9-62Б

Караник А.А.

Проверила:

Домрачева А.Б.

2024 г.

Цель работы

Целью данной работы научиться строить тригонометрическую интерполяцию функций с помощью быстрого преобразования Фурье.

Постановка задачи

1) Вычислить значения функции $f(x) = (x - [x])sin2\pi x$ в узлах сетки $x_j = \frac{j}{N}, N = 128$
2) Построить тригонометрическую интерполяцию функции, пользуясь БПФ для подсчета дискретных коэффициентов Фурье.
3) Сравнивать значения тригонометрической интерполяции в средних точках всех отрезков разбиения $y_j = 0.5 + \frac{j}{N}, j = 0, \dots, N - 1$

Теоретические сведения

Пусть периодическая функция $f(x)$ с периодом I может быть разложена в ряд Фурье:

$$f(x) = \sum_{q=-\infty}^{\infty} \alpha_q \exp(2\pi i q x)$$

Рассмотрим значения этой функции на сетке из точек $x_j = \frac{j}{N}$, где j, N - целые числа, и обозначим $f(x_j) = f_j$. Тогда из-за периодичности функции $f(x)$ ряд Фурье можно записать в виде

$$f_j = \sum_{q=0}^{N-1} A_q \exp(2\pi i q x_j),$$

где

$$A_q = \sum_{q=-\infty}^{\infty} \alpha_{q+sN}$$

Верно и обратное равенство:

$$A_q = \frac{1}{N} \sum_{j=0}^{N-1} f_j \exp(-2\pi i q x_j),$$

Аппроксимация функции $f(x)$ с помощью приближенного равенства

$$f(x) \approx \sum_{q=0}^{N-1} A_q \exp(2\pi i q x)$$

Носит название тригонометрической интерполяции; коэффициенты A_q называется дискретными коэффициентами Фурье.

Быстрое преобразование Фурье (БПФ) применяют, если число узлов сетки $N = 2^r$. Представим числа q и j, входящие в предыдущие формулы и лежащие в пределах $0 \le q, j < N$, в виде двоичного разложения:

$$q = \sum_{k=1}^{r} q_k 2^{k-1}, j = \sum_{m=1}^{r} j_{r+1-m} 2^{m-1},$$

Где $q_k, j_m = 0$, I. БПФ состоит в подсчете коэффициентов A_q с помощью рекуррентных соотношений

$$A_q = A^{(r)}(q_1, \dots, q_r).$$

$$A^{(m)}(q_1, \dots, q_m, j_{m+1}, \dots, j_r)$$
$$= \frac{1}{2} \sum_{j_m=0}^{1} \exp\left(-2\pi i j_m 2^{-m} \sum_{k=1}^{m} q_k 2^{k-1}\right) A^{(m-1)}(q_1, \dots, q_{m-1}, j_m, \dots, j_r), m = 1, \dots, r,$$

где

$$A^{(d)}(j_1, \dots, j_r) = f_{j_r + j_{r-1} 2 + \dots + j_1 2^{r-1}}$$

Реализация

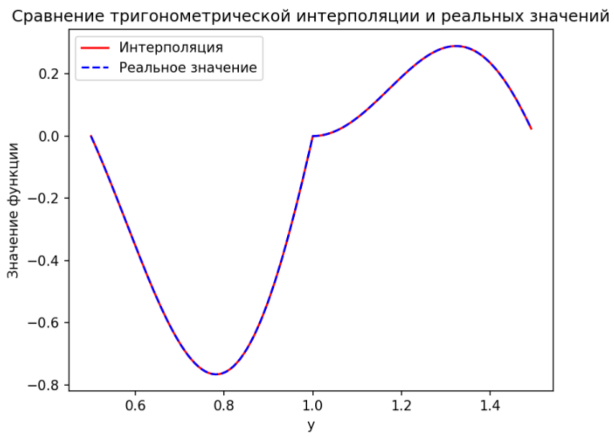
Исходный код программы:

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


def fft(x):
    N = len(x)

    if N <= 1:
        return x
    if N % 2 > 0:
        raise ValueError("Размер входного массива должен быть степенью двойки")

    even = fft(x[0::2])
    odd = fft(x[1::2])

    T = [np.exp(-2j * np.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)]

N = 128
x = np.arange(N) / N
f_x = (x - np.floor(x)) * np.sin(2 * np.pi * x)

coeffs = np.array(fft(f_x)) / N

def trig_interpolation(xi):
    M = len(xi)
    interpol_values = np.zeros(M, dtype=complex)
    for k in range(N):
        term = coeffs[k] * np.exp(2j * np.pi * k * xi)
        interpol_values += term
    return interpol_values.real

y = 0.5 + np.arange(N) / N

interpol_values = trig_interpolation(y)

true_values = (y - np.floor(y)) * np.sin(2 * np.pi * y)

for j in range(N):
    print(f"j={j}, yj={y[j]:.6f}, Интерполяция={interpol_values[j]:.6f}, Реальное значение={true_values[j]:.6f}")

e = true_values - interpol_values
error = np.max(np.abs(e))
print("Погрешность:", error)

plt.plot(y, interpol_values, 'r', label='Интерполяция')
plt.plot(y, true_values, 'b--', label='Реальное значение')
plt.xlabel('y')
plt.ylabel('Значение функции')
plt.legend()
plt.title('Сравнение тригонометрической интерполяции и реальных значений')
plt.show()
```

Тестирование

Вывод программы:

Погрешность: 2.6006974351844292e-14



Вывод

В процессе выполнения лабораторной работы была разработана программа, осуществляющая тригонометрическую интерполяцию функций с помощью быстрого преобразования Фурье. В программе на языке Python использовался тип комплексных чисел, что значительно упростило разработку. Также было произведено сравнение интерполяции и реальных значений функции. Программа работает корректно.