МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ (ШКОЛА)

Департамент математического и компьютерного моделирования

ОТЧЕТ

к лабораторной работе №1 по дисциплине
«Дифференциальные уравнения и теоретическая механика»

Направление подготовки
01.03.02 «Прикладная математика и информатика»

Выполнил студент гр.
Б9120-01.03.02

Агличеев А.О. ________
(Ф.И.О.) *(подпись)*

«15» июня 2023 г.

г. Владивосток
2023

Содержание

Введение

В данной лабораторной работе мне нужно реализовать алгоритм построения интегрального преобразования Фурье для произвольно-заданных функций одной переменной.

1 Задание

1.1 Постановка задачи

Реализовать алгоритм построения интегрального преобразования Фурье для произвольно-заданных функций одной переменной. Для реализации потребуются следующие параметры:

- $f(t)$ – определение функции,
- a, b – область интегрирования функции,
- n_1 – количество разбиений области интегрирования(также можно использовать шаг h_1),
- n_2 – количество разбиений частотного диапазона(также можно использовать шаг h_2),
- m – максимальное значение частоты.

Задача сводится к построению спектрального разложения одномерного сигнала $f(t)$ на частоты составляющих его волн.

Реализация проводится с помощью численных методов расчета интегралов. Потребуется построить график функции $f(t)$, и ее вещественные и комплексные спектральные разложения ($\mathscr{F}_{\Re}\{f\}$ и $\mathscr{F}_{\Im}\{f\}$). Графики разложений строятся в диапазоне $\omega \in [0, m]$. Оси графиков разложений представляют из себя по горизонтали – частотный диапазон, по вертикали – амплитуда.

Решение оформить в среде $\LaTeX$.

1.2 Решение

Рассмотрим функцию $f(t) = t^2 \cdot e^{-4t}$ на диапазоне $[a, b] = [0, 4]$, в остальном диапазоне полагая $f(t) = 0$. Для аналитического решения, функция примет вид:

$f(t) = t^2 \cdot e^{-4t} \cdot \Pi_{0,4}(t)$. Положим $\chi = 2\pi\omega$. Тогда преобразование Фурье примет вид:

$$\mathscr{F}\left\{t^2 \cdot e^{-4t}\right\} = \int\limits_{-\infty}^{\infty} t^2 \cdot e^{-4t} \cdot e^{-i\chi t} \cdot \Pi_{0,4}\, dt = \int\limits_{0}^{4} t^2 \cdot e^{-t(4+i\chi)}\, dt =$$

$$= -\frac{16e^{-16-4i\chi}}{4+i\chi} - \frac{8e^{-16-4i\chi}}{(4+i\chi)^2} - \frac{2e^{-16-4i\chi}}{(4+i\chi)^3} + \frac{2}{(4+i\chi)^3}$$

Разложим преобразование на действительную и мнимую части:

$$\mathscr{F}_{\Re}\left\{t^2 \cdot e^{-4t}\right\} = \cos 4\chi \cdot e^{-16} \cdot \left(\frac{-56}{16+\chi^2} + \frac{24\chi^2 - 128}{(16+\chi^2)^3}\right) + \\ + \chi \sin 4\chi \cdot e^{-16} \cdot \left(\frac{16}{16+\chi^2} + \frac{64}{(16+\chi^2)^2} + \frac{96 - 2\chi^2}{(16+\chi^2)^3}\right) - \\ - \frac{24\chi^2 - 128}{(16+\chi^2)^3}$$

$$\mathscr{F}_{\Im}\left\{t^2 \cdot e^{-4t}\right\} = \sin 4\chi \cdot e^{-16} \cdot \left(\frac{-56}{16+\chi^2} + \frac{24\chi^2 - 128}{(16+\chi^2)^3}\right) + \frac{96\chi - \chi^3}{(16+\chi^2)^3}$$

В таком случае спектральный график в аналитической форме будет иметь следующий вид :

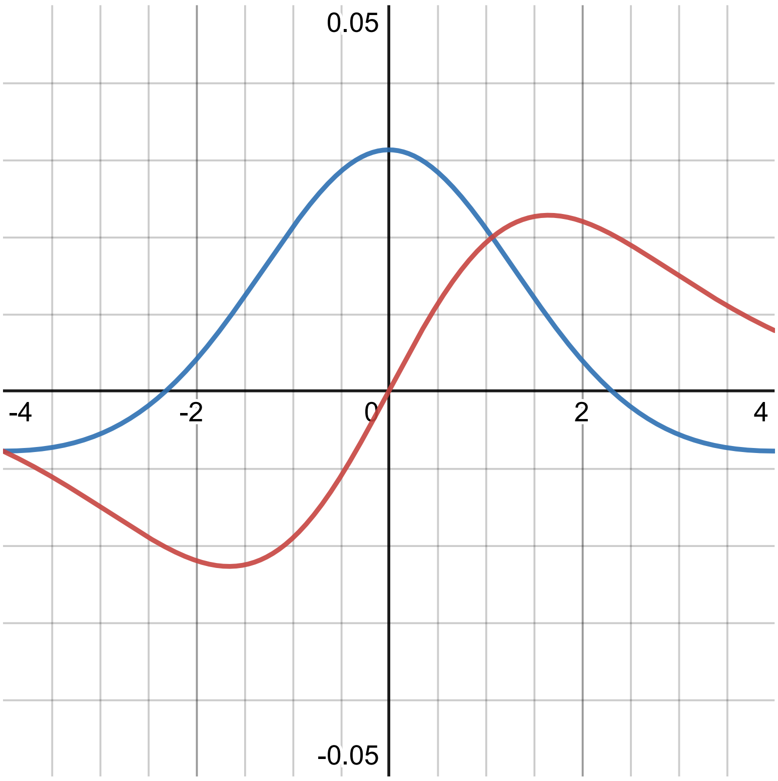

Рис. 1: Спектральный график волн синус- и косинус-преобразований (красный – синус, вещественное преобразование, синий – косинус, комплексное преобразование)

В процессе реализации численного алгоритма, максимальное значение частоты выберем $m = 2$. Таким образом, численно найденный спектральный график имеет вид:

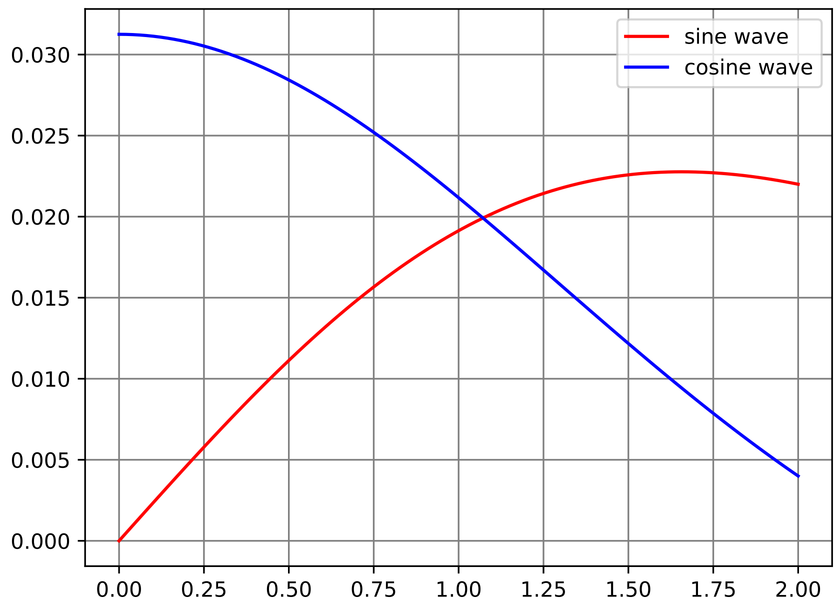

Рис. 2: Численно рассчитанный спектральный график волн синус- и косинус-преобразований (красный – синус, синий – косинус)

```
import numpy as np
from typing import Callable
from matplotlib import pyplot as plt


def left_rectangles(f: Callable, a: float, b: float, n: int, args
   =()):
    h = (b-a)/n
    res = 0
    for i in range(n):
        res += h * f(a + h*i, *args)
    return res


def fourier_transform(f: Callable, a: float, b: float, m: float,
   v_n=100, int_n=100):
    vs = np.linspace(0, m, v_n)
    sine = [left_rectangles(lambda t, _v: f(t) * np.sin(_v * t), a,
        b, int_n, args=(v,)) for v in vs]
    cosine = [left_rectangles(lambda t, _v: f(t) * np.cos(_v * t),
       a, b, int_n, args=(v,)) for v in vs]
    return vs, sine, cosine


def sine_wave(v):
    return np.sin(4*v)*(-64*np.exp(-16)/(16+v*v) + 8*np.exp(-16)
       /(16+v*v) + 2*(12*v*v-64)*np.exp(-16)/np.power((16+v*v), 3))
        \
```

```
        + 2*(48*v - v*v*v)/np.power((16+v*v), 3)


def cosine_wave(v):
    return np.cos(4*v)*(-64*np.exp(-16)/(16+v*v) + 8*np.exp(-16)
       /(16+v*v) + 2*(12*v*v-64)*np.exp(-16)/np.power((16+v*v), 3))
        \
    + np.sin(4*v)*(16*v*np.exp(-16)/(16+v*v) + 64*v*np.exp(-16)/np.
       power(16+v, 2) + 2*(48*v - v*v*v)*np.exp(-16)/np.power(v*v
       +16, 3)) \
    + 2*(-12*v*v+64)/np.power(v*v+16, 3)


if __name__ == '__main__':
    vs, sine, cosine = fourier_transform(lambda t: t*t*np.exp(-4*t)
       , 0, 4, 2)
    plt.plot(vs, sine, color='red', label='sine wave')
    plt.plot(vs, cosine, color='blue', label='cosine wave')
    plt.grid(color='grey', linestyle='-')
    plt.legend(loc="upper right")
    plt.show()

    plt.plot(vs, sine_wave(vs), color='red', label='sine wave')
    plt.plot(vs, cosine_wave(vs), color='blue', label='cosine wave
       ')
    plt.grid(color='grey', linestyle='-')
    plt.legend(loc="upper right")
    plt.show()
```

Листинг 1: Код алгоритма построения интегрального преобразования Фурье

Заключение

Я написал программу на языке «Python», реализующую алгоритм преобразования Фурье. Оформлял отчет по работе в «TEXstudio».