Федеральное государственное автономное образовательное учреждение высшего образования
«Национальный Исследовательский Университет ИТМО»

ЛАБОРАТОРНАЯ РАБОТА №2
ПО ПРЕДМЕТУ «ЧАСТОТНЫЕ МЕТОДЫ»
ПО ТЕМЕ «ПРЕОБРАЗОВАНИЕ ФУРЬЕ»

Лектор: Перегудин А. А.
Практик: Пашенко А. В.
Студент: Румянцев А. А.
Поток: ЧАСТ.МЕТ. 1.3

Факультет: СУиР
Группа: R3241

Санкт-Петербург
2024

Содержание

1 Введение

В заданиях 1 и 2 используется унитарное преобразование Фурье к угловой частоте ω. Подсчет Фурье--образа производится по формуле ниже

$$c(\omega) = \frac{1}{\sqrt{2\pi}} \int\limits_{-\infty}^{\infty} f(t)e^{-i\omega t}\,dt$$

В задании 3 используется преобразование Фурье к обыкновенной частоте ν. В общем виде формула имеет вид

$$\hat{f}(\nu) = \int\limits_{-\infty}^{\infty} f(t)\,e^{-2\pi i\nu t}\,dt$$

Все графики строятся программой, написанной на языке программирования python. В 1 и 2 заданиях используется библиотека sympy, в задании 3 numpy и matplotlib. По ходу отчета приводится код для каждого задания. Для всех интегралов и графиков есть место с общими переменными и значениями – файл static.py. Основные данные приведены ниже

```
from sympy import Symbol, Piecewise, Abs, sinc, E, oo

t = Symbol('t')
omega = Symbol('omega')

interval = [(-oo, oo)]

a_b_pars = [(1, 2), (2, 3), (3, 4)]
consts = [-1, 0.5, 1]
colors_strs = ['red', 'purple', 'blue', 'cyan']
```

Листинг 1: Основные данные из файла static.py

Для проверки равенства Парсеваля используется формула ниже

$$||f||_2 = ||Ff||_2,$$

где $||f||_2$ – вторая норма заданной фукнции, $||Ff||_2$ – вторая норма Фурье-образа функции $f(t)$

2 Задание 1. Вещественное

2.1 Прямоугольная функция

Рассмотрим прямоугольную функцию следующего вида

$$f(t) = \begin{cases} a, & |t| \le b, \\ 0, & |t| > b. \end{cases}$$

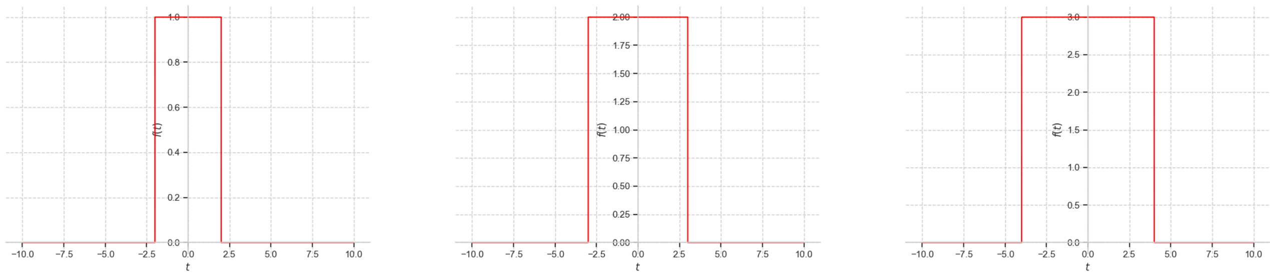
Рассмотрим программу. Сначала задается функция, принимающая параметры a и b, после чего методом build_f_t строится график $f(t)$. На строке 10 приведен пример использования кода

```
1  def rectangular_function(a, b):
2      return Piecewise((a, Abs(t) <= b), (0, Abs(t) > b))
3  
4  def build_f_t(f_t, clr, lbl):
5      if (lbl == None):
6          plot(f_t, line_color=clr, xlabel=r'$t$', ylabel=r'$f(t)$')
7      else:
8          plot(f_t, line_color=clr, xlabel=r'$t$', ylabel=r'$f(t)$', label=lbl, legend=True)
9  
10 build_f_t(rectangular_function(1, 2), 'red', None)
```

Листинг 2: Программа для построения графика прямоугольной функции

Построенные графики $f(t)$ для нескольких значений параметров $a, b > 0$ расположены ниже

Рассмотрим программу. Методом find_fimg находится Фурье-образ заданной функции. Далее методом build_fimg2 строится график Фурье-образа. На 16-17 строчках находится пример использования кода



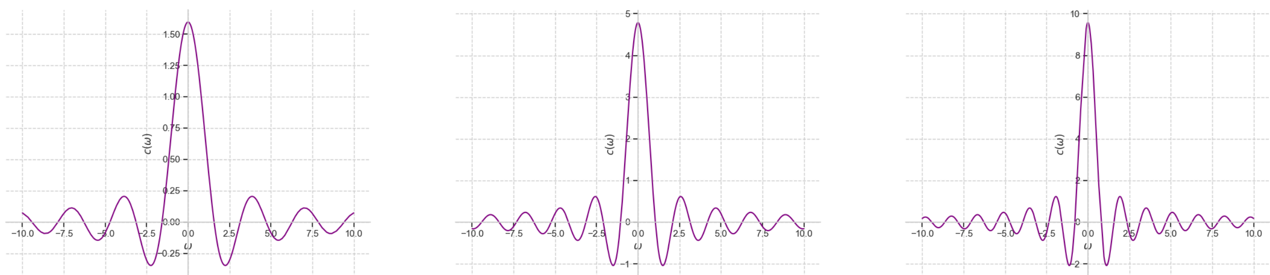
(a) $a = 1,\ b = 2$ (b) $a = 2,\ b = 3$ (c) $a = 3,\ b = 4$

Рис. 1: Прямоугольные функции при различных значениях a и b

```
def find_fimg(f_t, lim1, lim2):
    integrand = f_t * E ** (-I * omega * t)

    result = integrate(integrand, (t, lim1, lim2))
    return coeff * result

def build_fimg2(fimg, clr, lbl):
    if (lbl == None):
        plot(fimg, line_color=clr,
             xlabel=r'$\omega$', ylabel=r'$c(\omega)$')
    else:
        plot(fimg, line_color=clr,
             xlabel=r'$\omega$', ylabel=r'$c(\omega)$',
             label=lbl, legend=True)

rectfimg = find_fimg(rectangular_function, -oo, oo)
build_fimg2(rectfimg, 'purple', None)
```

Листинг 3: Программа для построения графика Фурье-образа некоторой функции $f(t)$

Построенные графики $\hat{f}(\omega)$ для тех же значений a и b расположены ниже



(a) $a = 1,\ b = 2$ (b) $a = 2,\ b = 3$ (c) $a = 3,\ b = 4$

Рис. 2: Фурье-образы прямоугольных функций при различных значениях a и b

Рассмотрим программу. Методом find_norm2 находится вторая норма переданной функции. Метод find_parseval считает левую и правую части равенства Парсеваля. Пример использования кода расположен на 13-14 строчках листинга ниже

```
1  def find_norm2(f, lim1, lim2, var):
2      integrand = f * conjugate(f)
3
4      result = integrate(integrand, (var, lim1, lim2)).evalf()
5      return sqrt(result).evalf()
6
7  def find_parseval(f, fimg, lim1, lim2):
8      pleft = find_norm2(f, lim1, lim2, t)
9      pright = find_norm2(fimg, lim1, lim2, omega)
10
11     return pleft, pright
12
13 pl, pr = find_parseval(rectangular_function, rectfimg, -oo, oo)
14 print(f'p_{1}: {pl} ?= {pr}')
```

Листинг 4: Программа для нахождения левой и правой сторон равенства Парсеваля

Программа вывела в консоль результаты, представленные ниже

```
    p_1: 2.00000000000000 ?= 2.0 + 0.e-114*I
    p_2: 4.89897948556636 ?= 4.89897948556636 + 0.e-114*I
    p_3: 8.48528137423857 ?= 8.48528137423857 + 0.e-114*I
```

Листинг 5: Результат выполнения программы для вычисления равенства Парсеваля

Мнимыми частями в правой части равенства Парсеваля пренебрежем вследствие их стремления к нулю. В таком случае равенство Парсеваля выполняется, что можно объяснить тем, что интеграл позволяет рассмотреть норму непрерывно на заданном промежутке, а ряд только дискретно, вследствие чего теряются какие-то члены ряда, которых не хватает для выполнения равенства Парсеваля