

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

Содержание

1	Введение	2
2	Задание 1. Вещественное	3
2.1	Прямоугольная функция	3
2.2	Треугольная функция	5
2.3	Кардинальный синус	5
2.4	Функция Гаусса	6
2.5	Двустороннее затухание	6
3	Задание 2. Комплексное	7
4	Задание 3. Музыкальное	8

1 Введение

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

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

$$\hat{f}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt$$

Далее будет использоваться программная реализация, приведенная ниже

```

1 def find_fimg(f_t, lim1, lim2):
2     integrand = f_t * E ** (-I * omega * t)
3
4     result = integrate(integrand, (t, lim1, lim2))
5     return coeff * result

```

Листинг 1: Программная реализация вычисления Фурье-образа с угловой частотой ω

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

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

Для проверки равенства Парсеваля используется формула ниже

$$\|f\|_2 = \|\hat{f}\|_2,$$

где $\|f\|_2$ – вторая норма заданной функции, $\|\hat{f}\|_2$ – вторая норма Фурье-образа функции f . Для нахождения нормы используются формулы, представленные ниже

$$\|f(t)\|_2 = \sqrt{\int_a^b f(t) \cdot f^*(t) dt}, \quad \|\hat{f}(\omega)\|_2 = \sqrt{\int_a^b \hat{f}(\omega) \cdot \hat{f}^*(\omega) d\omega}$$

Далее для поиска нормы и проверки равенства Парсеваля будет по умолчанию использоваться программа, представленная ниже. Пример использования расположен на 12-14 строчках листинга

```

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    return pleft, pright
11
12 func = rectangular_function(1, 2)
13 pl, pr = find_parseval(func, find_fimg(func, -oo, oo), -oo, oo)
14 print(f'p_{1}: {pl} ?= {pr}')

```

Листинг 2: Программа для вычисления нормы и левой и правой стороны равенства Парсеваля

Для всех интегралов и графиков есть место с общими переменными и значениями – файл static.py. Основные используемые данные приведены ниже

```

1 t = Symbol('t')
2 omega = Symbol('omega')
3 interval = [-oo, oo]
4 a_b_pars = [(1, 2), (2, 3), (3, 4)]
5 consts = [-1, 0.5, 1]
6 colors_strs = ['red', 'purple', 'blue', 'cyan']

```

Листинг 3: Основные данные из файла static.py

В этом файле программно заданы функции, графики которых приводятся по ходу отчета. Также они необходимы для нахождения их Фурье-образа

```

1  def rectangular_function(a, b):
2      return Piecewise((a, Abs(t) <= b), (0, Abs(t) > b))
3
4  def triangular_function(a, b):
5      return Piecewise((a - Abs(a * t / b), Abs(t) <= b), (0, Abs(t) > b))
6
7  def cardinal_sinus(a, b):
8      return a * sinc(b * t)
9
10 def gaussian_function(a, b):
11     return a * E ** (-b * t ** 2)
12
13 def double_attenuation(a, b):
14     return a * E ** (-b * Abs(t))
15
16 def shifted_rectangular_function(a, b, shift):
17     if (shift == 0):
18         return rectangular_function(a, b)
19     return Piecewise((a, Abs(t + shift) <= b), (0, Abs(t + shift) > b))

```

Листинг 4: Программно заданные функции для заданий 1 и 2

2 Задание 1. Вещественное

2.1 Прямоугольная функция

Рассмотрим прямоугольную функцию следующего вида

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

Аналитическое выражение Фурье-образа с выводом $\hat{f}(\omega)$ для прямоугольной функции имеет вид

$$\begin{aligned} \hat{f}(\omega) &= \frac{1}{\sqrt{2\pi}} \left(\int_{-b}^b a e^{-i\omega t} dt + \int_b^{\infty} 0 \cdot e^{-i\omega t} dt \right) = \frac{a}{\sqrt{2\pi}} \int_{-b}^b e^{-i\omega t} dt = \left[\frac{u = -i\omega t}{\frac{du}{dt} = -\frac{du}{i\omega}} \right] = -\frac{a}{i\omega\sqrt{2\pi}} \int_{i\omega b}^{-i\omega b} e^u du = \\ &= -\frac{a}{i\omega\sqrt{2\pi}} e^u \Big|_{i\omega b}^{-i\omega b} = -\frac{a}{i\omega\sqrt{2\pi}} (e^{-i\omega b} - e^{i\omega b}) = \frac{a}{\omega\sqrt{2\pi}} \left(2 \cdot \frac{e^{i\omega b} - e^{-i\omega b}}{2i} \right) = \frac{2a \sin(b\omega)}{\omega\sqrt{2\pi}} = \frac{a\sqrt{2} \sin(b\omega)}{\omega\sqrt{\pi}} \end{aligned}$$

Для построения графиков $f(t)$ используется метод, представленный ниже. На строках 7-8 приведен пример использования – сначала задается функция, принимающая параметры a и b , после чего вызывается метод `build_f_t`. Далее в отчете эта программа используется по умолчанию

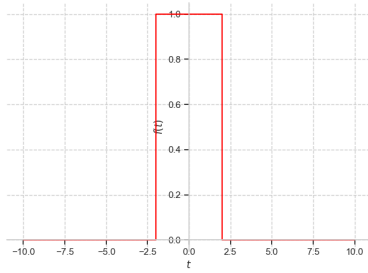
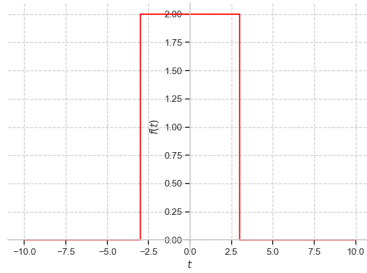
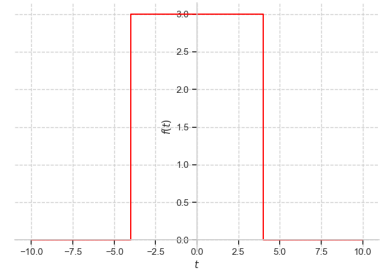
```

1  def build_f_t(f_t, clr, lbl):
2      if (lbl == None):
3          plot(f_t, line_color=clr, xlabel=r'$t$', ylabel=r'$f(t)$')
4      else:
5          plot(f_t, line_color=clr, xlabel=r'$t$', ylabel=r'$f(t)$', label=lbl, legend=True)
6
7  func = rectangular_function(1, 2)
8  build_f_t(func, 'red', None)

```

Листинг 5: Программа для построения графика $f(t)$

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

(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

Рассмотрим программу. Методом `find_fimg`, приведенным в введении, находится Фурье-образ заданной функции. Далее методом `build_fimg` строится график Фурье-образа. На 10-11 строчках располагается пример использования кода

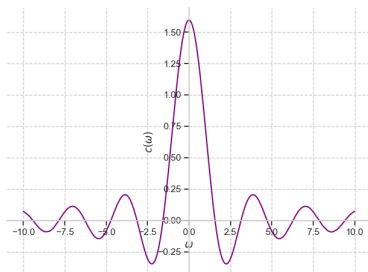
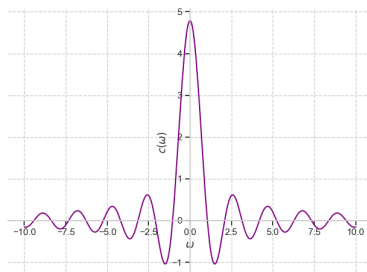
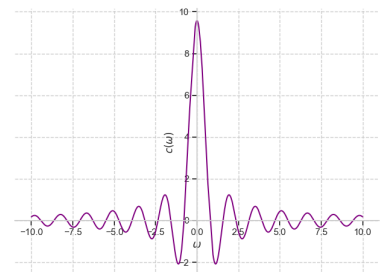
```

1  def build_fimg(fimg, clr, lbl):
2      if (lbl == None):
3          plot(fimg, line_color=clr,
4              xlabel=r'$\omega$', ylabel=r'$c(\omega)$')
5      else:
6          plot(fimg, line_color=clr,
7              xlabel=r'$\omega$', ylabel=r'$c(\omega)$',
8              label=lbl, legend=True)
9
10 rectfimg = find_fimg(rectangular_function(1, 2), -oo, oo)
11 build_fimg(rectfimg, 'purple', None)

```

Листинг 6: 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

Программа для вычисления левой и правой сторон равенства Парсеваля вывела в консоль результаты, представленные ниже

```

1  p_1: 2.0000000000000000 ?= 2.0 + 0.e-114*I
2  p_2: 4.89897948556636 ?= 4.89897948556636 + 0.e-114*I
3  p_3: 8.48528137423857 ?= 8.48528137423857 + 0.e-114*I

```

Листинг 7: 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

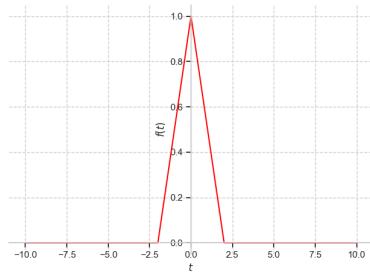
2.2 Треугольная функция

Рассмотрим треугольную функцию следующего вида

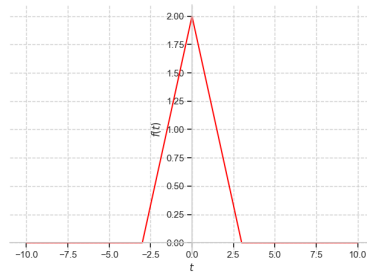
$$f(t) = \begin{cases} a - \left| \frac{at}{b} \right|, & |t| \leq b, \\ 0, & |t| > b. \end{cases}$$

Аналитическое выражение с выводом Фурье-образа $\hat{f}(\omega)$ для треугольной функции имеет вид

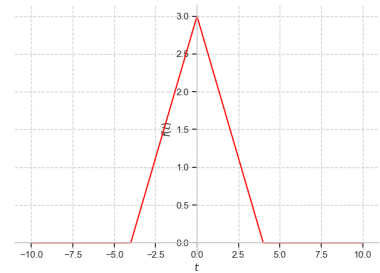
Построенные графики функций $f(t)$ и их Фурье-образов $\hat{f}(\omega)$ для нескольких значений параметров $a, b > 0$ расположены ниже



(a) $a = 1, b = 2$

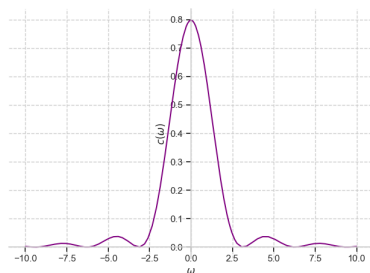


(b) $a = 2, b = 3$

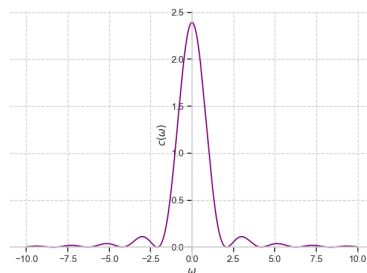


(c) $a = 3, b = 4$

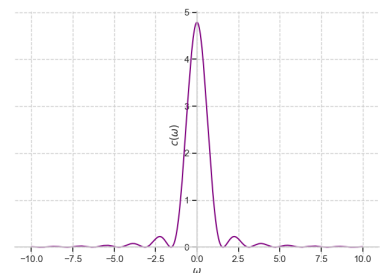
Рис. 3: Треугольные функции при различных значениях a и b



(a) $a = 1, b = 2$



(b) $a = 2, b = 3$



(c) $a = 3, b = 4$

Рис. 4: Фурье-образы треугольных функций при различных значениях a и b

Проверим программой выполнение равенства Парсеваля

```
1 p_1: 1.15470053837925 ?= 1.15470053837925
2 p_2: 2.82842712474619 ?= 2.82842712474619
3 p_3: 4.89897948556636 ?= 4.89897948556636
```

Листинг 8: Равенство Парсеваля для треугольных функций

Равенство Парсеваля выполняется для треугольной функции

2.3 Кардинальный синус

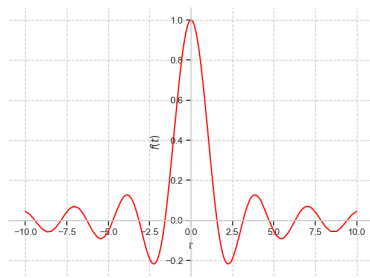
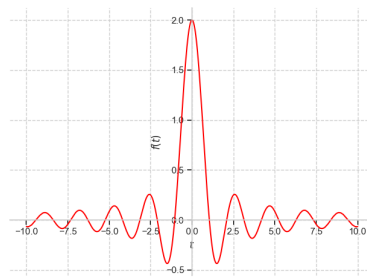
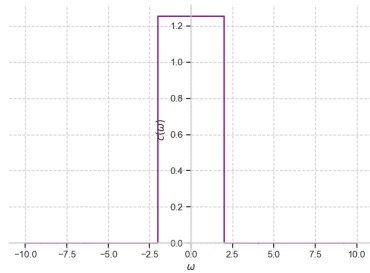
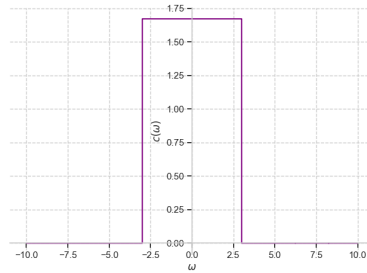
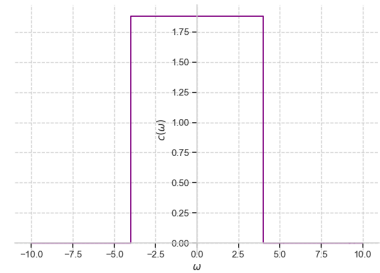
Рассмотрим кардинальный синус следующего вида

$$f(t) = a \cdot \text{sinc}(bt)$$

Аналитическое выражение Фурье-образа $\hat{f}(\omega)$ для кардинального синуса имеет вид

Построенные графики $f(t)$ и $\hat{f}(\omega)$ для нескольких значений параметров $a, b > 0$ расположены ниже

Проверим программой выполнение равенства Парсеваля

(a) $a = 1, b = 2$ (b) $a = 2, b = 3$ (c) $a = 3, b = 4$ Рис. 5: Кардинальные синусы при различных значениях a и b (a) $a = 1, b = 2$ (b) $a = 2, b = 3$ (c) $a = 3, b = 4$ Рис. 6: Фурье-образы кардинальных синусов при различных значениях a и b

```

1  p_1: 1.253331412734194  ?= 1.253331412734194
2  p_2: 2.04665340503514  ?= 2.04665340503514
3  p_3: 2.65868076577972  ?= 2.65868076577972

```

Листинг 9: Равенство Парсеваля для кардинальных синусов

Результат программы показывает, что равенство Парсеваля выполняется для кардинального синуса

2.4 Функция Гаусса

Рассмотрим функцию Гаусса следующего вида

$$f(t) = ae^{-bt^2}$$

Аналитическое выражение Фурье-образа $\hat{f}(\omega)$ для функции Гаусса имеет вид

$$\hat{f}(\omega) = \frac{a}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-bt^2} e^{-i\omega t} dt = \frac{a}{e^{\frac{\omega^2}{4b}} \sqrt{2b}}$$

Построенные графики $f(t)$ и $\hat{f}(\omega)$ для нескольких значений параметров $a, b > 0$ расположены ниже

Проверим программой выполнение равенства Парсеваля

```

1  p_1: 0.941396263776715  ?= 0.941396263776715
2  p_2: 1.70129510027892  ?= 1.70129510027892
3  p_3: 2.37485023062924  ?= 2.37485023062924

```

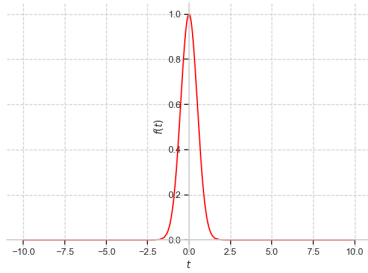
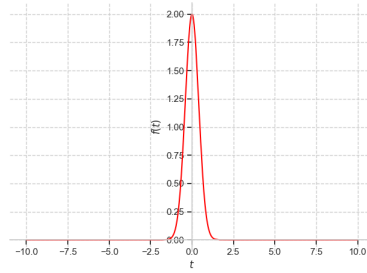
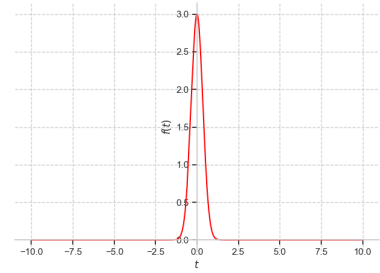
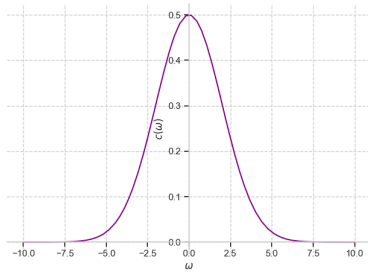
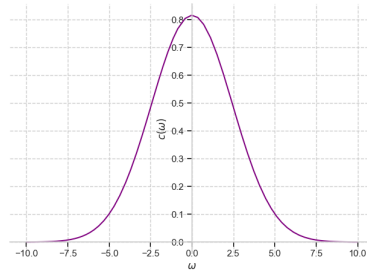
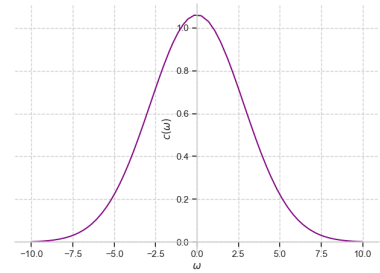
Листинг 10: Равенство Парсеваля для функции Гаусса

Результат говорит о выполнении равенства Парсеваля для функции Гаусса

2.5 Двустороннее затухание

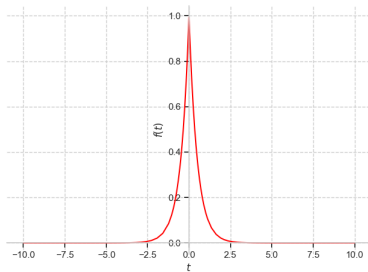
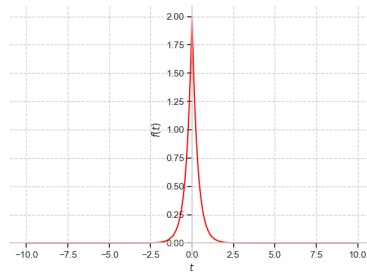
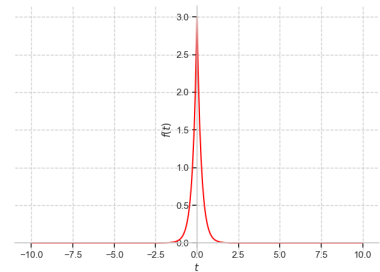
Рассмотрим двустороннее затухание следующего вида

$$f(t) = ae^{-b|t|}$$

(a) $a = 1, b = 2$ (b) $a = 2, b = 3$ (c) $a = 3, b = 4$ Рис. 7: Функции Гаусса при различных значениях a и b (a) $a = 1, b = 2$ (b) $a = 2, b = 3$ (c) $a = 3, b = 4$ Рис. 8: Фурье-образы функций Гаусса при различных значениях a и b

Аналитическое выражение с выводом Фурье-образа $\hat{f}(\omega)$ для двустороннего затухания имеет вид

Построенные графики $f(t)$ и $\hat{f}(\omega)$ для нескольких значений параметров $a, b > 0$ расположены ниже

(a) $a = 1, b = 2$ (b) $a = 2, b = 3$ (c) $a = 3, b = 4$ Рис. 9: Двусторонние затухания при различных значениях a и b

Проверим программой выполнение равенства Парсеваля

```
1 p_1: 0.700601299282005 ?= 0.700601299282005
2 p_2: 1.15326854220130 ?= 1.15326854220130
3 p_3: 1.49974838192519 ?= 1.49974838192519
```

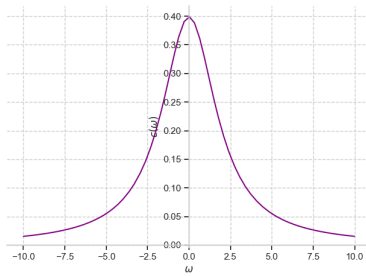
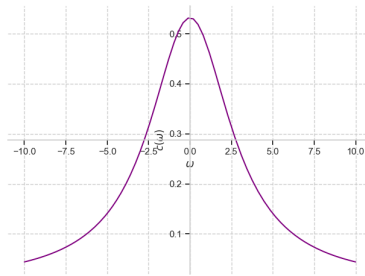
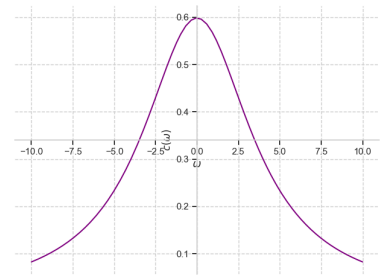
Листинг 11: Равенство Парсеваля для функции Гаусса

Результат говорит о выполнении равенства Парсеваля для двустороннего затухания

3 Задание 2. Комплексное

Для выполнения задания выбрана прямоугольная функция из задания 1 при зафиксированных параметрах $a = 1, b = 2$

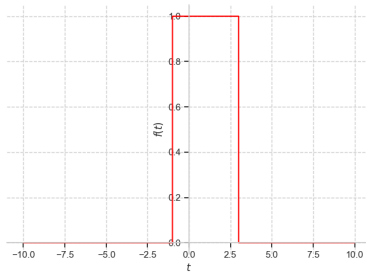
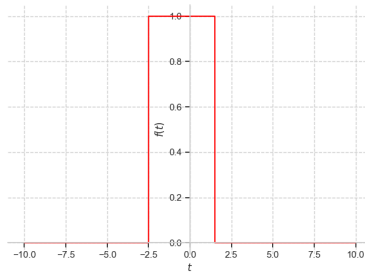
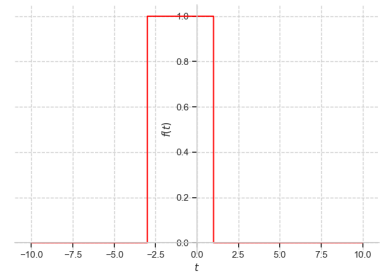
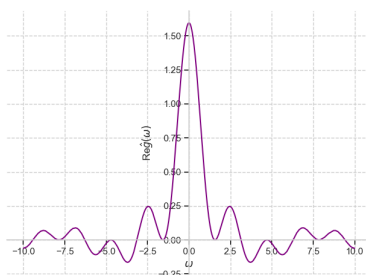
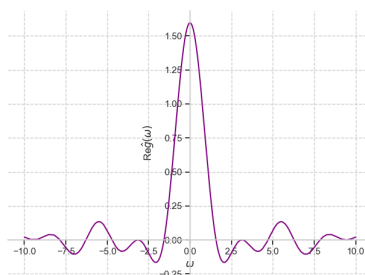
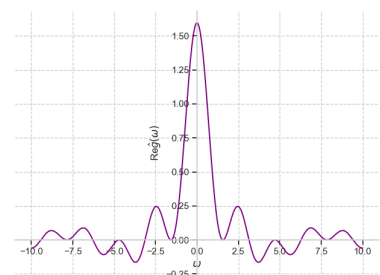
$$g(t) = f(t + c) = \begin{cases} 1, & |t + c| \leq 2, \\ 0, & |t + c| > 2. \end{cases}$$

(a) $a = 1, b = 2$ (b) $a = 2, b = 3$ (c) $a = 3, b = 4$ Рис. 10: Фурье-образы двусторонних затуханий при различных значениях a и b

Аналитическое выражение Фурье-образа $\hat{f}(\omega)$ для прямоугольной функции со сдвигом имеет вид

$$\hat{f}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-2-c}^{2-c} 1 \cdot e^{-i\omega t} dt =$$

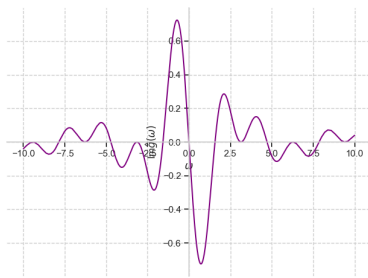
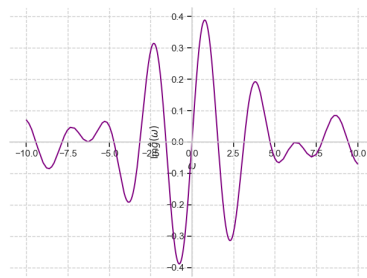
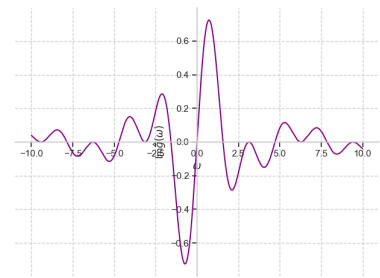
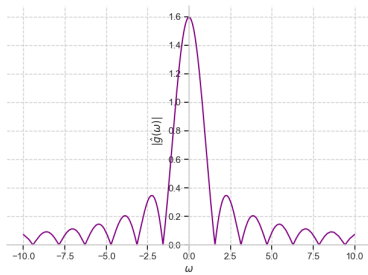
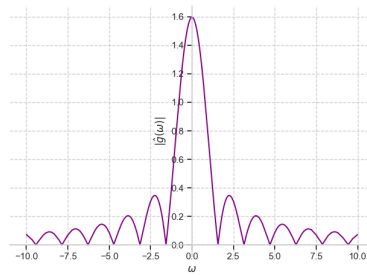
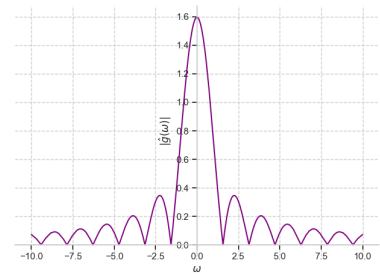
Построенные графики $g(t)$, $\text{Re } \hat{g}(\omega)$, $\text{Im } \hat{g}(\omega)$ и $|\hat{g}(\omega)|$ для нескольких значений параметра c расположены ниже

(a) $c = -1$ (b) $c = 0.5$ (c) $c = 1$ Рис. 11: Прямоугольные функции со смещением при различных значениях c (a) $c = -1$ (b) $c = 0.5$ (c) $c = 1$ Рис. 12: Действительные части Фурье-образов прямоугольных функций со смещением при различных значениях параметра c

4 Задание 3. Музыкальное

Для выполнения задания с представленного [гугл-диска](#) выбран аккорд номер 26

Для считывания аудиозаписи в список используется библиотека `librosa`. В переменную `y` запишутся значения амплитуд аудиосигнала, в переменной `sr` будет частота дискретизации аудиосигнала, а `t` – список временных отсчетов в секундах, соответствующий каждой амплитуде в списке `y`. На строках 15-17 показан пример использования метода. Указывается путь к `mp3` файлу в переменной `audio_file`. В переменной

(a) $c = -1$ (b) $c = 0.5$ (c) $c = 1$ Рис. 13: Мнимые части Фурье-образов прямоугольных функций со смещением при различных значениях параметра c (a) $c = -1$ (b) $c = 0.5$ (c) $c = 1$ Рис. 14: Модули Фурье-образов прямоугольных функций со смещением при различных значениях параметра c

`select_channel` хранится номер нужного звукового канала – выбран первый канал. Аудиозапись длится 4 секунды, последняя из которых вырезана на 19-20 строках, так как не несет в себе важной информации, аккорд заканчивается раньше

```

1  def get_y_sr_t(audio_file: str, select_channel: int):
2      y, sr = load(audio_file)
3
4      if select_channel >= y.ndim:
5          select_channel = 0
6
7      y = y[:, select_channel] if y.ndim > 1 else y
8      t = linspace(0, len(y) / sr, len(y))
9
10     return y, sr, t
11
12     audio_file = 'fm_lab2/chord/chord26.mp3'
13     select_channel = 0
14     y, sr, t = get_y_sr_t(audio_file, select_channel)
15
16     y = y[:3 * sr]
17     t = t[:3 * sr]
```

Листинг 12: Программа для считывания аудиозаписи в список

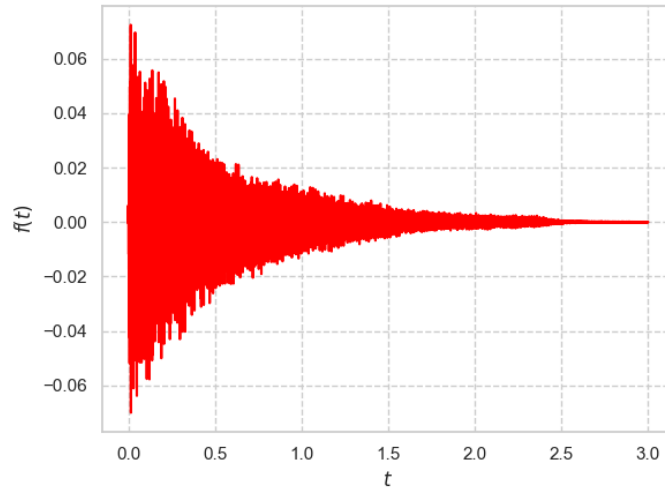
Программа ниже построит график аудиозаписи $f(t)$. На 12-13 строках расположен пример использования

```

1  def build_audio_f_t(t, y, clr=None):
2      if (clr != None):
3          plt.plot(t, y, color=clr)
4      else:
5          plt.plot(t, y)
6
7      plt.xlabel(r'$t$')
8      plt.ylabel(r'$f(t)$')
9      plt.grid(True)
10     plt.show()
11
12     f_t_clr = colors_strs[0]
13     build_audio_f_t(t, y, f_t_clr)
```

Листинг 13: Программа для построения функции $f(t)$ аудиозаписи

Результат выполнения программы для считанной в список аудиозаписи

Рис. 15: График $f(t)$ аудиозаписи

С помощью численного интегрирования `trapz` из библиотеки `numpy` найден Фурье-образ $\hat{f}(\nu)$ по частотам `freqs` и записан в список амплитуд `ampls`. Пример использования расположен на 11 строке

```

1  def find_freqs_ampls(t, y, sr):
2      freqs = linspace(0, sr / 2, len(y) // 2)
3      amps = []
4
5      for freq in freqs:
6          int = trapz(y * exp(-1j * 2 * pi * freq * t), t)
7          amps.append(abs(int))
8
9      return freqs, amps
10
11  freqs, amps = find_freqs_ampls(t, y, sr)

```

Листинг 14: Программа для нахождения Фурье-образа аудиозаписи

Программа ниже построит график $|\hat{f}(\nu)|$. Пример использования расположен на 20-27 строках

```

1  def build_audio_f_v(freqs, amps, start=None, stop=None, step=None,
2                      fz1=None, fz2=None, clr=None):
3      if ((fz1 != None) and (fz2 != None)):
4          plt.figure(figsize=(fz1, fz2))
5
6      if (clr != None):
7          plt.plot(freqs, amps, color=clr)
8      else:
9          plt.plot(freqs, amps)
10
11     plt.xlabel(r'$\nu$')
12     plt.ylabel(r'$\left|\hat{f}\right|\left(\nu\right)$')
13     plt.grid(True)
14     if ((start != None) and
15         (stop != None and stop != 0) and
16         (step != None and step != 0)):
17         plt.xticks(arange(start, stop, step))
18     plt.show()
19
20     start = 0
21     stop = 10001
22     step = 1000
23     figsize1 = 10
24     figsize2 = 6
25     f_v_clr = colors_strs[1]
26

```

```

27 build_audio_f_v(freqs, ampls, start=start, stop=stop, step=step,
28                 fz1=figsize1, fz2=figsize2, clr=f_v_clr)

```

Листинг 15: Программа для построения графика $|\hat{f}(\nu)|$ аудиозаписи

Построенный график $|\hat{f}(\nu)|$ расположен ниже

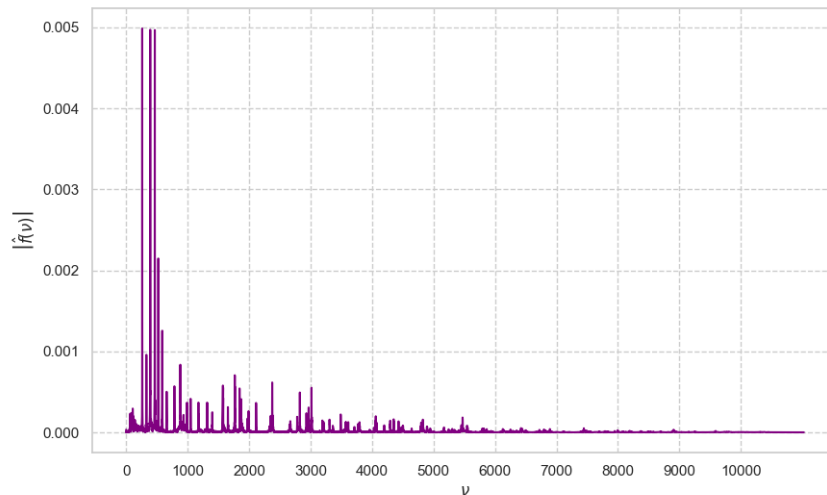


Рис. 16: График $|\hat{f}(\nu)|$ аудиозаписи

Так как на большом интервале частот трудно разобрать из каких нот состоит аккорд, нужен код для уменьшения интервала и шага без потери данных или искажения графиков

```

1  r_start = 0
2  r_end = 1000
3
4  start_idx = next(idx for idx, freq in enumerate(freqs) if freq >= r_start)
5  end_idx = next(idx for idx, freq in enumerate(freqs) if freq > r_end)
6
7  r_ampls = ampls[start_idx:end_idx]
8  r_freqs = freqs[start_idx:end_idx]
9  r_step = 20
10
11 build_audio_f_v(r_freqs, r_ampls, start=r_start, stop=r_end, step=r_step, fz1=figsize1,
                  fz2=figsize2, clr=f_v_clr)

```

Листинг 16: Программа для изменения интервала частот и амплитуд

С помощью программы построен график $|\hat{f}(\nu)|$ на интервале частот от 0 до 1000 с шагом 200, а после на интервале от 200 до 500 с шагом 20 для точности определения нот

Исходя из графика, построенного на уменьшенном интервале частот, определены следующие ноты: первая До, первая Соль, первая Ля диез

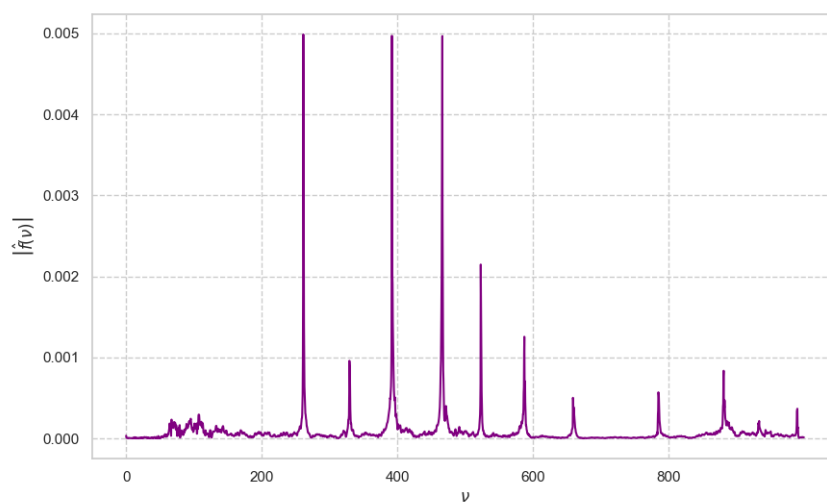


Рис. 17: График $|\hat{f}(\nu)|$ аудиозаписи на интервале частот от 0 до 1000

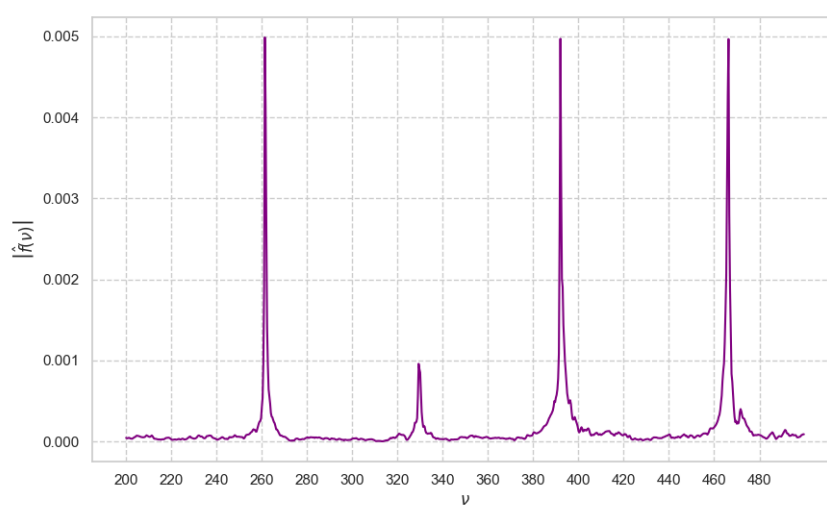


Рис. 18: График $|\hat{f}(\nu)|$ аудиозаписи на интервале частот от 200 до 500