Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное образовательное учреждение высшего образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

по задаче от лектора

по дисциплине «Дополнительные главы физики»

Авторы: Новиков Георгий

Факультет: ФИТиП

Группа: М32101

Преподаватель: Ефремова Екатерина Александровна

Санкт-Петербург 2023

Условие

Сформировать импульс, близкий к прямоугольному, из спектра плоских гармонических волн с несущей длиной волны $\lambda_0 = 1.5$ мкм и длительностью $\Delta t \approx 10$ мкс. Какая спектральная ширина пакета? Промоделировать его прохождение через среду с заданной дисперсией фазовой скорости $v_{фаз} = \sqrt{c^2 + b^2\lambda^2}$ для $b = \{1, 10, 100\}$. Определить характерное время расплывания пакета.

Закодировать сообщение (короткое слово) такими импульсами и промоделировать передачу такого сообщения до времен порядка характерного времени расплывания пакета.

Поэтапное решение задачи

- **Найти спектральную ширину пакета:**

  Для определения спектральной ширины пакета будем использовать определение спектральной ширины на уровне половины максимальной амплитуды, которое соответствует ширине на полувысоте(FWHM) графика амплитудного спектра:

$$\mathrm{FWHM} = 2\sqrt{2\ln 2}\,\sigma \approx 2{,}355\,\sigma\,.$$

  где σ – полуширина пика спектра.

  Реализация на **Python** (ф-ция spectral_width) с комментариями:

```python
import numpy as np

2 usages
def spectral_width(pulse, delta_t, lambda_0):
    # pulse - одномерный массив значений сигнала во временной области
    # delta_t - длительность импульса
    # lambda_0 - несущая длина волны

    # расчет и получение амплитудного спектра
    spectrum = np.fft.fft(pulse)
    amplitude_spectrum = np.abs(spectrum)

    # нахождений максимального и половины от максимального значения амплитуды
    max_amplitude = np.max(amplitude_spectrum)
    half_max_amplitude = max_amplitude / 2

    idx = np.argwhere(amplitude_spectrum >= half_max_amplitude)

    # вычисления FWHM в единицах частоты и длины волны
    fwhm = (idx[-1] - idx[0]) * (1 / delta_t) / len(pulse)
    spectral_width = fwhm / (2 * np.sqrt(2 * np.log(2))) * lambda_0**2

    return spectral_width
```

По результату работы кода нами будет вычислено следующее число спектральной ширины пакета:

```
Спектральная ширина пакета: [0.00191097] мкм
```

- **Промоделировать прохождение данного импульса, черезсреду с заданной дисперсией фазовой скорости v_{фаз} =sqrt(c^2 + b^2 lambda^2) для b = {1, 10, 100}.**

Для моделирования прохождения импульса через среду с заданной дисперсией фазовой скорости можно использовать преобразование Фурье и умножение на функцию фазовой скорости.

Для этого предстоит выполнить следующую последовательность шагов:

– *Рассчитаем дисперсионную задержку для каждой длины волны в уже созданном спектре, используя формулу заданной дисперсии фазовой скорости.*

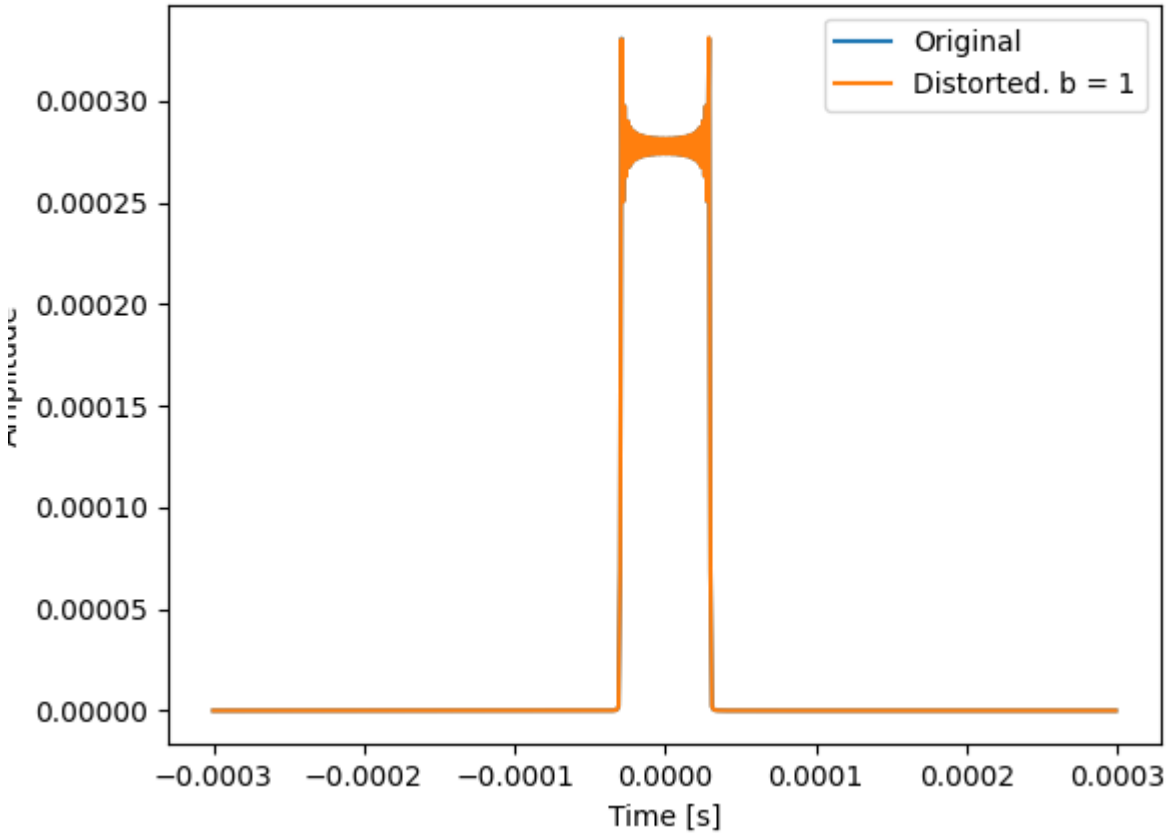
– *Выполнить преобразование Фурье для искаженного спектра с учетом дисперсионной задержки.*

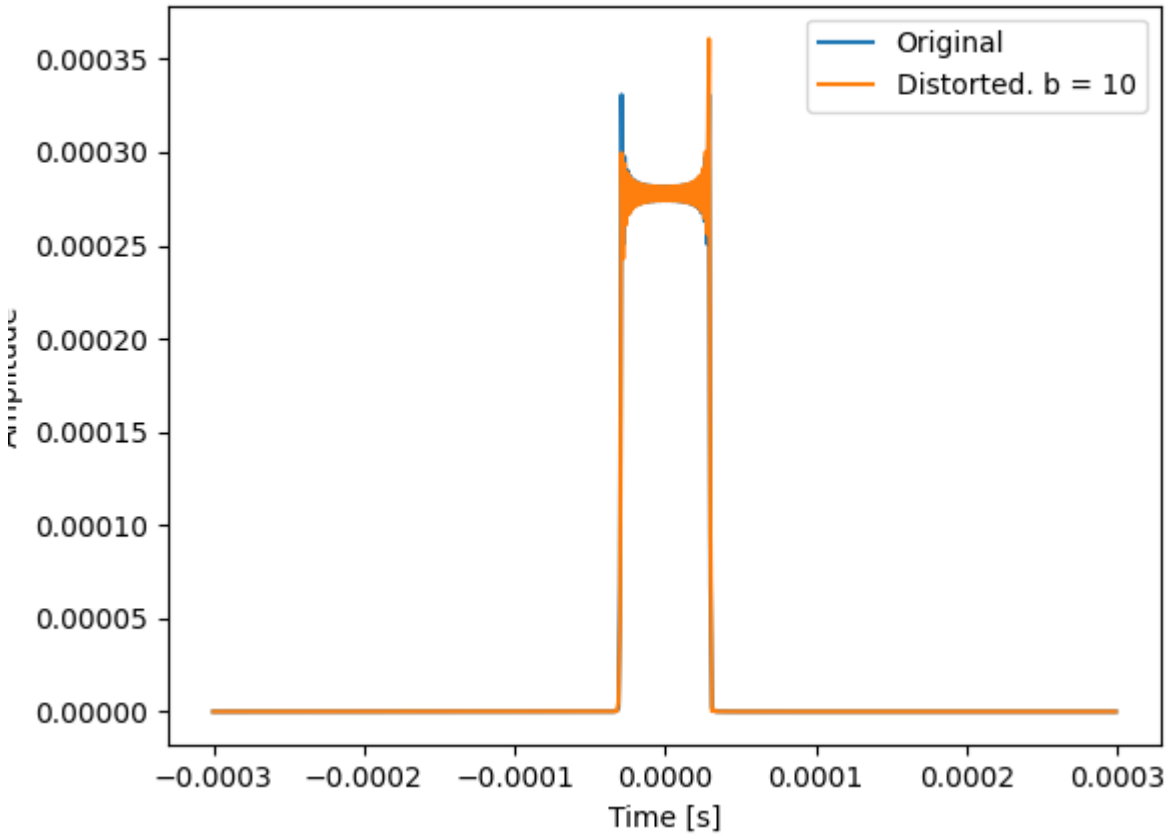
– *Выполнить обратное преобразование Фурье для получения временного сигнала.*

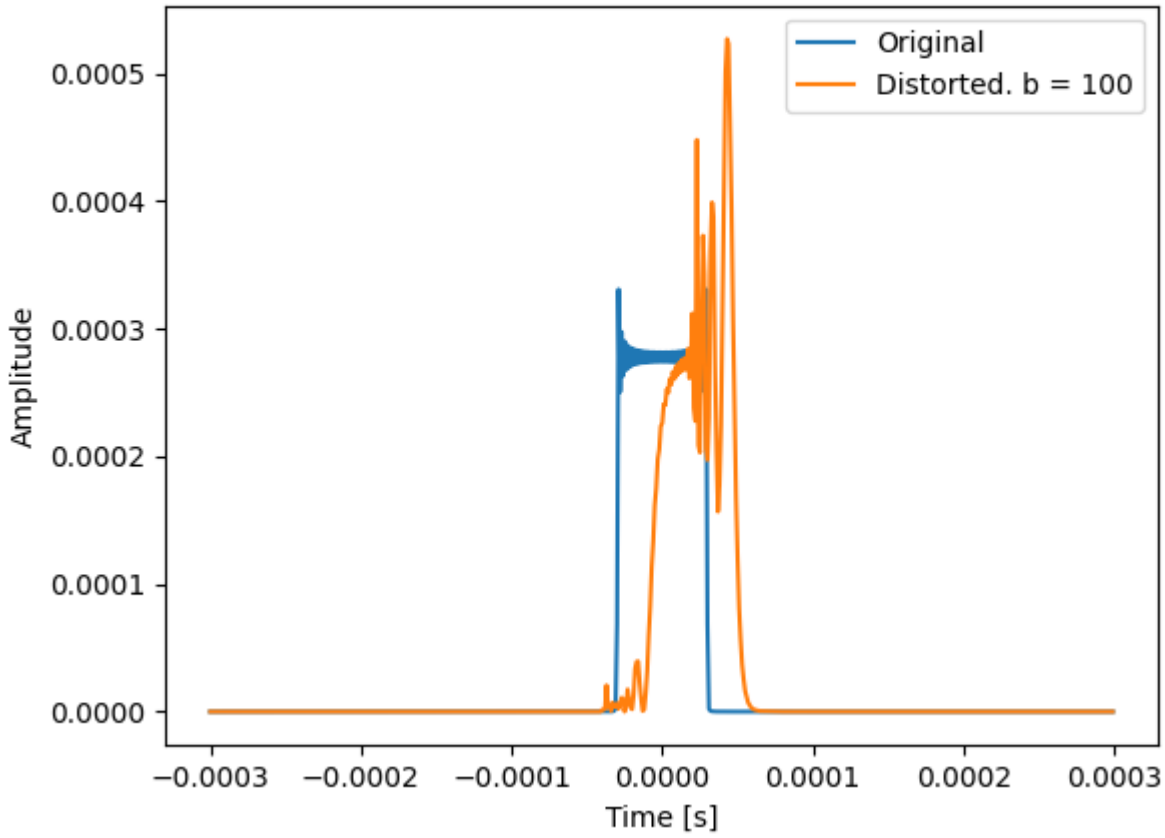
Реализация на **Python** (ф-ция dispersion) с комментариями:

```python
import numpy as np


2 usages
def dispersion(spectrum, delta_t, t, pulse, b, c):
    # pulse - импульс, который необходимо пропустить через среду
    # b - дисперсия
    # c - скорость света в вакууме

    # расчет частоты в спектре и фазовой скорости
    lambda_ = np.fft.fftfreq(len(pulse), d=delta_t)
    v_phase = np.sqrt(c ** 2 + (b ** 2 * (lambda_) ** 2))

    # умножение на функцию фазовой скорости
    distorted_spectrum = spectrum * np.exp(-1j * 2 * np.pi * v_phase * t)

    # обратное преобразование Фурье
    distorted_pulse = np.fft.ifftshift(np.fft.ifft(distorted_spectrum))

    return distorted_pulse
```

Смоделированные графики исходного и искаженного импульсов:







- **Определить характерное время расплывания пакета.**

Для определения времени расплывания пакета используется метод, состоящий из нескольких этапов:

– *Найдем максимальное значение амплитуды пакета и его половинную амплитуду.*

– *Произведем поиск первой точки слева и справа от максимума, где амплитуда пакета уменьшается до уровня половинной амплитуды.*

– *Вычислим разница между индексами первых точек слева и справа, что и дает нам характерное время расплывания пакета.*

Реализация на **Python** (ф-ция compute_fwhm) с комментариями:

```python
import numpy as np

2 usages
def compute_fwhm(distorted_spectrum, delta_freq):
    # t - массив временных значений
    # pulse - импульс, для которого необходимо вычислить FWHM

    # Вычисляем корректированный спектр
    corrected_spectrum = np.fft.fftshift(distorted_spectrum)
    corrected_spectrum = np.abs(corrected_spectrum)

    # Находим максимальное значение
    max_value = np.max(corrected_spectrum)

    # Находим индексы точек ниже половины максимального значения
    lower_half_indices = corrected_spectrum < max_value / 2.0
    first_index = np.min(np.where(lower_half_indices)[0])
    last_index = np.max(np.where(lower_half_indices)[0])

    # Находим соответствующие значения времени
    first_time = first_index * delta_freq
    last_time = last_index * delta_freq

    # Находим разницу между временными значениями
    fwhm = last_time - first_time

    return fwhm
```

По результату работы кода нами будет вычислено следующие три характерных времени расплывания пакета для трех различных дисперсий:

```
b = 1, FWHM = 1.614e+00 s
b = 10, FWHM = 1.611e+00 s
b = 100, FWHM = 1.623e+00 s
```

- **Закодировать сообщение («exam») такими импульсами и промоделировать передачу такого сообщения до времен порядка характерного времени расплывания пакета.**

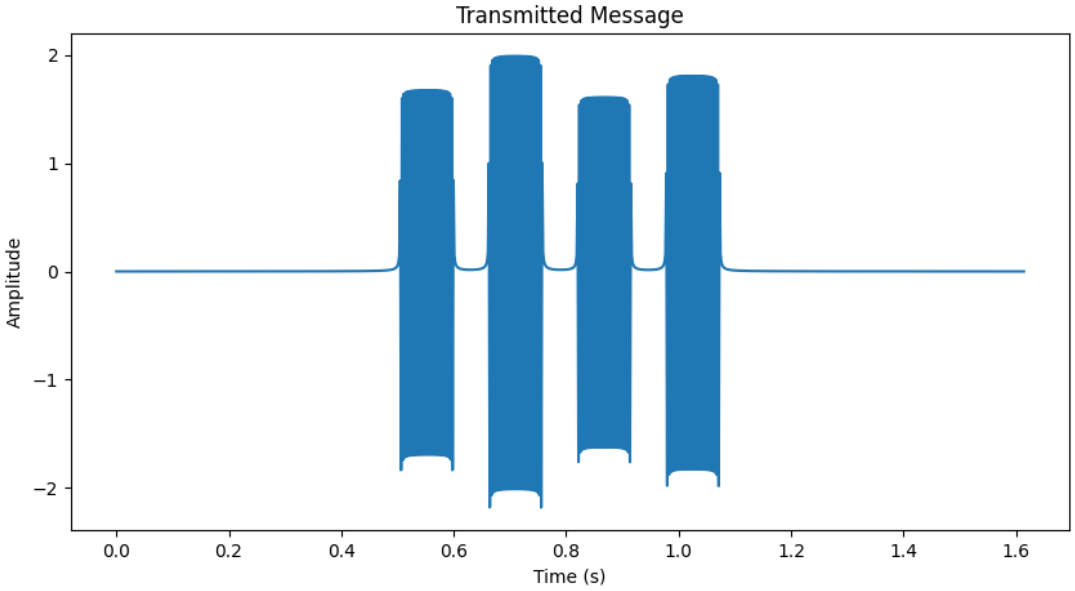
При выполнении данного задания воспользуемся уже имеющимся искаженному импульсу *distored_pulse* и его характерному времени расплывания *t_distorted*.

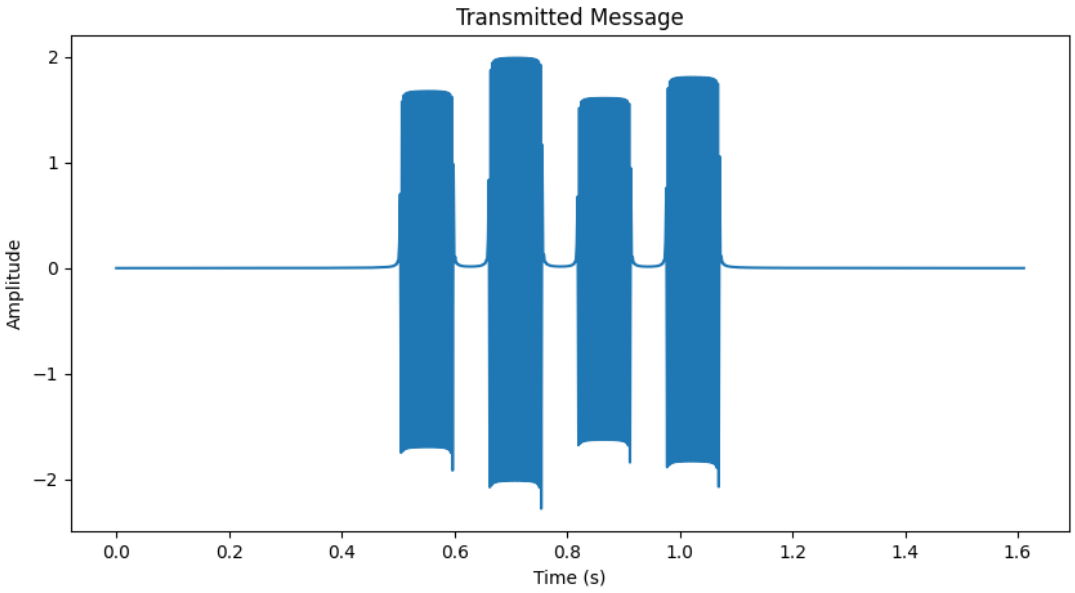
В результате выполнения кода-реализации мы получим 3 графика, показывающих, как переданное сообщение выглядит после прохождения различных искаженных импульсов до времен порядка характерного их времен расплывания пакета.

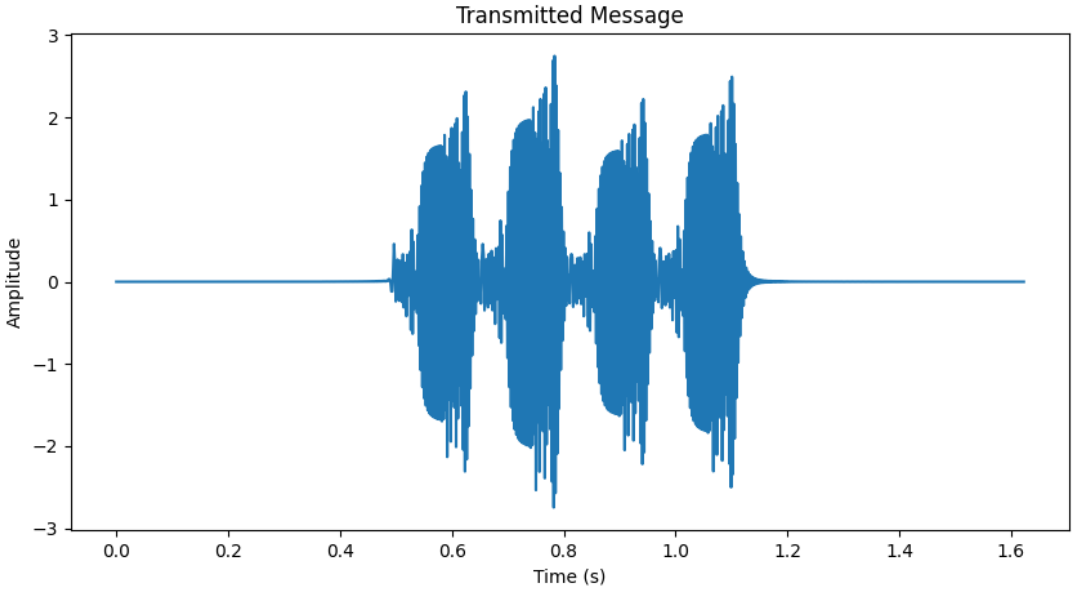
Реализация на **Python** (ф-ция signal) с комментариями:

```python
import numpy as np

2 usages
def signal(distorted_pulse, t_distorted):

    # Создание сообщения
    message = "exam"

    # Кодирование сообщения
    coded_message = np.zeros(1024)
    for i, char in enumerate(message):
        coded_message[100*i+50] = ord(char)

    # Моделирование передачи сообщения до времен порядка характерного времени расплывания пакета
    t = np.linspace(0, t_distorted, 1024)
    transmitted_message = np.convolve(coded_message, distorted_pulse)[:1024]

    # Визуализация результатов
    import matplotlib.pyplot as plt

    plt.figure(figsize=(10,5))
    plt.plot(t, transmitted_message)
    plt.xlabel('Time (s)')
    plt.ylabel('Amplitude')
    plt.title('Transmitted Message')
    plt.show()
```

Смоделированные графики передающих искаженного импульсов:







Вывод:

- В ходе выполненной работы был сформирован импульс, близкий к прямоугольному из спектра плоских гармонических волн с несущей длиной волны лямбда = 1,5 мкм и длительностью дельта t ≈ 10 мкс.

- Была рассчитана спектральная ширина пакета, которая составила [0.00191097] мкм.

- Далее было проведено моделирование прохождения пакета через фазовую скорость для нескольких значений b = {1, 10, 100}.

- Было отмечено характерное время расплывания пакета, которое было замечено на графиках.

- Также, было проведено кодирование сообщения данными импульсами и получен график передачи сообщения до времени порядка характерного времени расплывания пакета.