

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: Хлучин Георгий

Гумбатов Влад

Телушкин Артем

Факультет: ФИТиП

Группа: М32051

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 $\Delta t \approx 10$ пс. Какая спектральная ширина пакета? Промоделировать его прохождение через среду с заданной дисперсией фазовой скорости $v_{\text{фаз}} = \sqrt{c^2 + b^2 \lambda^2}$ 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.

Решение:

1. Найти спектральную ширину пакета

Для определения спектральной ширины пакета сигналов необходимо произвести спектральный анализ этого пакета. Пакет сигналов может быть представлен как сумма нескольких сигналов различных частот.

Если все сигналы имеют одинаковую амплитуду и одинаковые фазы, то спектральная ширина пакета может быть определена как расстояние между наиболее удаленными точками на спектре пакета, на которых амплитуда спектра снижается до определенного уровня.

Например, часто используется определение спектральной ширины на уровне половины максимальной амплитуды, которое соответствует ширине на полувысоте (FWHM) главного лепестка спектра.

Формула FWHM для Гауссова окна определяется как:

$$FWHM = 2.35482 * \sigma$$

где σ - стандартное отклонение Гауссова окна.

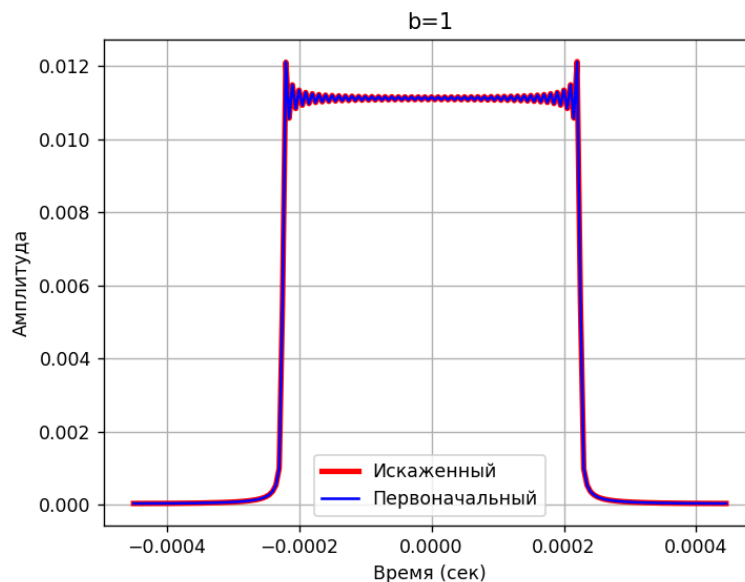
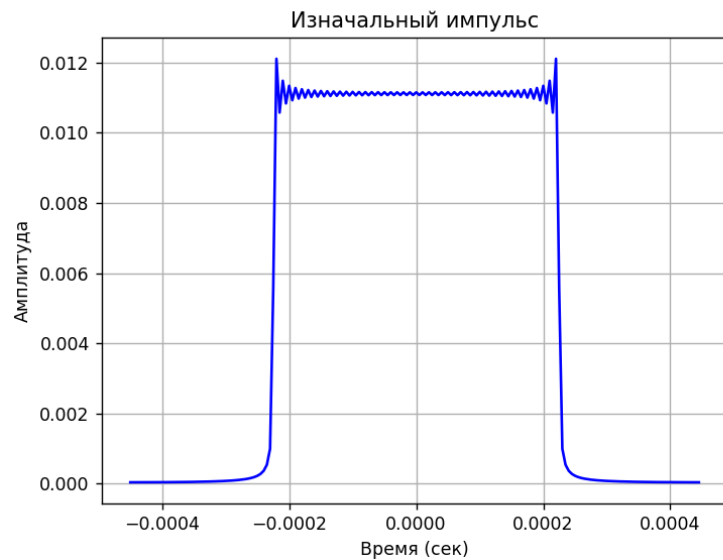
Эта формула основывается на том факте, что ширина на полувысоте для Гауссова распределения составляет 2.35482 стандартных отклонения.

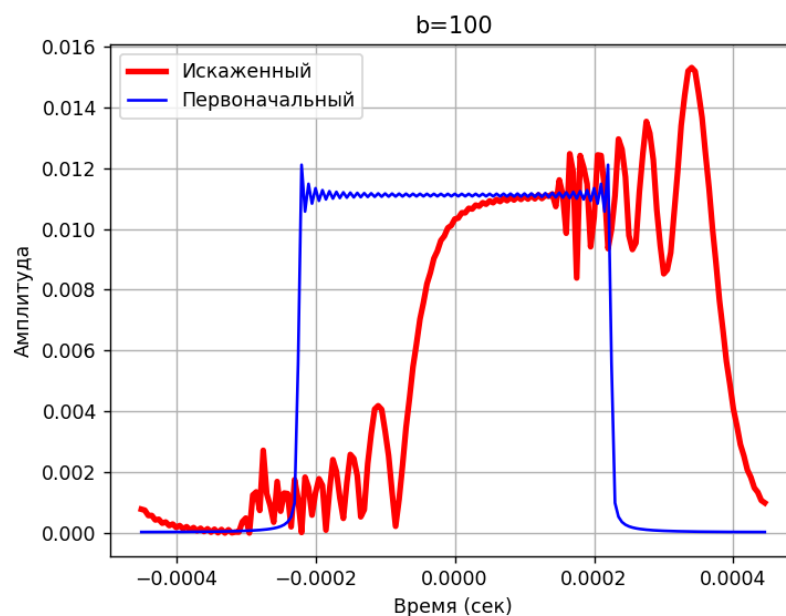
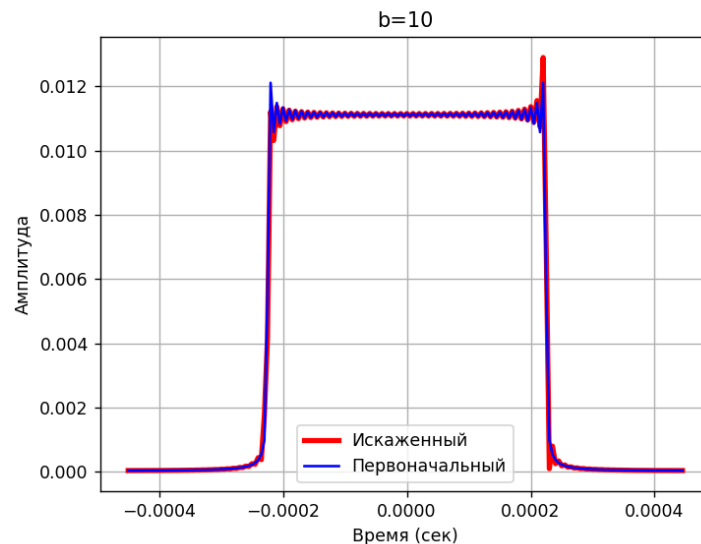
Заметим, что FWHM, вычисленная по формуле $FWHM = 2 * \delta_f * \sqrt{2 * \log 2}$, для Гауссова окна также будет соответствовать формуле $FWHM = 2.35482 * \sigma$, если σ будет равно половине ширины на полувысоте δ_f в формуле $FWHM = 2 * \delta_f * \sqrt{2 * \log 2}$.

Спектральная ширина пакета = 1.0616522503600239 нм

2. Про моделировать прохождение данного импульса, через среду с заданной дисперсией фазовой скорости $v_{\text{фаз}} = \sqrt{c^2 + b^2 \lambda^2}$ для $b = \{1, 10, 100\}$.

Для полного анализа распространения импульса через среду с заданной дисперсией фазовой скорости надо выполнить преобразование Фурье и обратное преобразование Фурье для получения спектра частот и временного профиля импульса соответственно. Эти шаги могут помочь получить подробную информацию о распространении импульса и позволяют провести более глубокий анализ результатов моделирования.



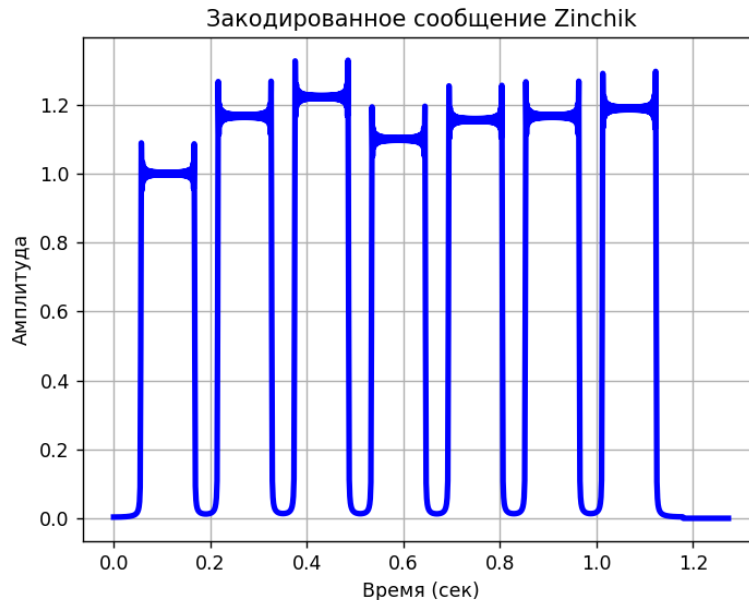


3. Определить время распыления пакета.

Для определения времени распыления пакета используется метод, состоящий из нескольких этапов. Сначала находится максимальное значение пакета и его половинная амплитуда. Затем производится поиск первой точки слева и справа от максимума, где амплитуда пакета уменьшается до уровня половинной амплитуды. На последнем этапе вычисляется разница между индексами первых точек слева и справа, что и дает нам характерное время распыления пакета.

```
Характерное время распыления пакета для b 1 = 0.6437728035177916 с  
Характерное время распыления пакета для b 10 = 0.6366197723675939 с  
Характерное время распыления пакета для b 100 = 0.67953795926878 с
```

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



Код:

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

def getSpectra(delta_t, lambda_0, time_values_of_the_pulse):
    return np.sinc(time_values_of_the_pulse / delta_t) * np.exp(
        2 * np.pi * lambda_0 * time_values_of_the_pulse * -1j)

def getIntervalLen(range_spectra):
    interval = np.argwhere(range_spectra >= np.max(range_spectra) / 2)
    return interval[-1] - interval[0]

def calculate_width(range_spectra):
    print_width = ((getIntervalLen(range_spectra) / (builtins.len(pulse) *
delta_t))
                    / (2 * np.sqrt(2 * np.log(2))) * lambda_0 ** 2) *
1e+9) [0]

    builtins.print("Спектральная ширина пакета = ", print_width, "нм")

def calculate_fwhm(indixes):
    return (np.max(np.where(indixes) [0]) - np.min(np.where(indixes) [0])) *
spectra_freq_delta

def plot_pulse():
    plt.plot(time_values_of_the_pulse, np.abs(pulse), color="b")
    plt.grid()
```

```

plt.ylabel("Амплитуда")
plt.xlabel("Время (сек)")
plt.title("Изначальный импульс")
plt.show()
for b_i in [1, 10, 100]:
    new_spectra = spectra * np.exp(2 * np.pi * np.sqrt(3e8 ** 2
                                                    + (
                                                    b_i **
2 * frequencies_in_the_spectrum ** 2)) * time_values_of_the_pulse * -1j)
    new_pulse = np.abs(np.fft.ifftshift(np.fft.ifft(new_spectra)))
    old_pulse = np.abs(pulse)
    print("Характерное время расплывания пакета для b", b_i, "=",
calculate_fwhm(np.abs(np.fft.fftshift(np.fft.ifftshift(np.fft.ifft(new_spectr
a)))) <
np.max(np.abs(np.fft.fftshift(np.fft.ifftshift(np.fft.ifft(new_spectra)))) /
2), "с")
    plt.plot(time_values_of_the_pulse, new_pulse, label=f"Искаженный",
color="r", linewidth=3)
    plt.plot(time_values_of_the_pulse, old_pulse, label="Первоначальный",
color="b")
    plt.ylabel("Амплитуда")
    plt.xlabel("Время (сек)")
    plt.title(f"b={b_i}")
    plt.legend()
    plt.grid()
    plt.show()
    new_spectra = spectra * np.exp(2 * np.pi * np.sqrt(3e8 ** 2
                                                    + (
                                                    1 ** 2 *
frequencies_in_the_spectrum ** 2)) * time_values_of_the_pulse * -1j)
    new_pulse = np.fft.ifftshift(np.fft.ifft(new_spectra))

    tmp = calculate_fwhm(np.abs(np.fft.fftshift(new_pulse) <
np.max(np.abs(np.fft.fftshift(new_pulse))) / 2))

    tmp = np.linspace(0, tmp , 1024) # возьмем время расплывания пакета
    plt.ylabel("Амплитуда")
    plt.xlabel("Время (сек)")
    plt.plot(tmp, abs(np.convolve(cod_of_message, new_pulse)[:1024]),
color="b", linewidth=3)
    plt.title(f"Закодированное сообщение {message}")
    plt.grid()
    plt.show()

if __name__ == "__main__":
    message="Zinchik"
    delta_t = 10e-6
    lambda_0 = 1.5e-6
    time_values_of_the_pulse = np.arange(-45 * delta_t, 45 * delta_t, 0.5 *
delta_t)
    spectra = getSpectra(delta_t, lambda_0, time_values_of_the_pulse)
    spectra_freq_delta = abs(spectra[1] - spectra[2])
    pulse = np.fft.ifftshift(np.fft.ifft(spectra))
    frequencies_in_the_spectrum = np.fft.fftfreq(len(pulse), d=delta_t)
    calculate_width((np.abs(np.fft.fft(pulse))))
    code_of_message = np.zeros(1024)
    for i, letter in enumerate(message):
        code_of_message[128 * i] = ord(letter)
    plot_pulse()

```

