

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования «Самарский национальный исследовательский
университет имени академика С.П. Королева»
(Самарский университет)

Институт информатики и кибернетики
Кафедра технической кибернетики

ОТЧЕТ

по Лабораторной работе № 2

**Численная реализация оптического преобразования Фурье на
основе быстрого преобразования Фурье**

Вариант 21

Выполнил студент
группы 6402-010302D
Янкин И.Ю.

Самара 2024

1 Задание на лабораторную работу

№	Входное поле	Примечание
21	$\text{rect}((x + 2)/2)$	Для аналитики применить не финитное преобразование.

1. Реализовать одномерное финитное преобразование Фурье с помощью применения алгоритма БПФ.

2. Построить график гауссова пучка e^{-x^2} . Здесь и далее для каждого графика следует строить отдельно графики амплитуды и фазы.

3. Убедиться в правильности реализации преобразования, подав на вход гауссов пучок e^{-x^2} – собственную функцию преобразования Фурье. На выходе тоже должен получиться гауссов пучок (построить график на правильной области определения $[-\tilde{b}, \tilde{b}]$). Рекомендуемая входная область: $[-a, a] = [-5, 5]$.

4. Реализовать финитное преобразование Фурье стандартным методом численного интегрирования (например, методом прямоугольников). Важно: необходимо вычислить интеграл для каждого дискретного значения u , чтобы получить результат в виде вектора. На вход преобразования вновь следует подавать гауссов пучок.

5. Построить результаты двух разных реализаций преобразования на одном изображении (одно для амплитуды, одно для фазы) и убедиться, что они совпадают.

6. Используя первую реализацию преобразования, подать на вход световое поле, отличное от гауссова пучка, в соответствии со своим вариантом. Построить графики самого пучка и результата преобразования.

7. Рассчитать аналитически результат преобразования своего варианта поля и построить график на одной системе координат с результатом, полученным в предыдущем пункте. Аналитический расчёт должен быть помещён в отчёт. Прикрепление в отчёт фотографии или скана вывода формулы не является допустимым. В некоторых вариантах после

преобразования могут появиться особые точки. Значения в них нужно вычислять отдельно, в том числе в коде программы.

8. Исследовать параметры N и M алгоритма БПФ. Для этого сначала необходимо варьировать N , задавая при этом $M = N$. Что происходит при увеличении и уменьшении N ? После этого следует зафиксировать N и изменять $M > N$. Что происходит при увеличении M ? Выводы следует поместить в таблицу 2. Подкрепить выводы графиками. Важно: число M должно оставаться степенью двойки, чтобы график фазы содержал минимальное число ошибок.

9. Выполнить пункты 1-3 и 6 для двумерного случая. Графики изменятся на двумерные изображения, одномерные функции следует заменить на двумерные, равные произведению соответствующих одномерных функций. Например, гауссов пучок поменяется на $e^{-x^2-y^2}$. Изображение следует строить не в виде 3D-графиков, а в виде цветовой схемы (см. примеры двумерных пучков в лекциях). Рассчитать аналитически результат двумерного преобразования и нарисовать изображения его амплитуды и фазы. Двумерное преобразование следует реализовывать по предложенному выше алгоритму, а именно: применить одномерный алгоритм к строкам и столбцам матрицы. Если реализовывать алгоритм через двумерное БПФ, то добавление огромного числа нулей к матрице не будет оптимальным, сведёт на нет все преимущества БПФ, а потому будет поводом для снижения оценки за работу.

2 Результаты работы программы

2.1 Графики пучка Гаусса

Изначально на входе имеем одномерную функцию Гаусса: $f(x) = e^{-x^2}$.
Графики амплитуды и фазы Гауссова пучка приведены на рисунке 1.

Амплитуда и фаза Гауссов пучок

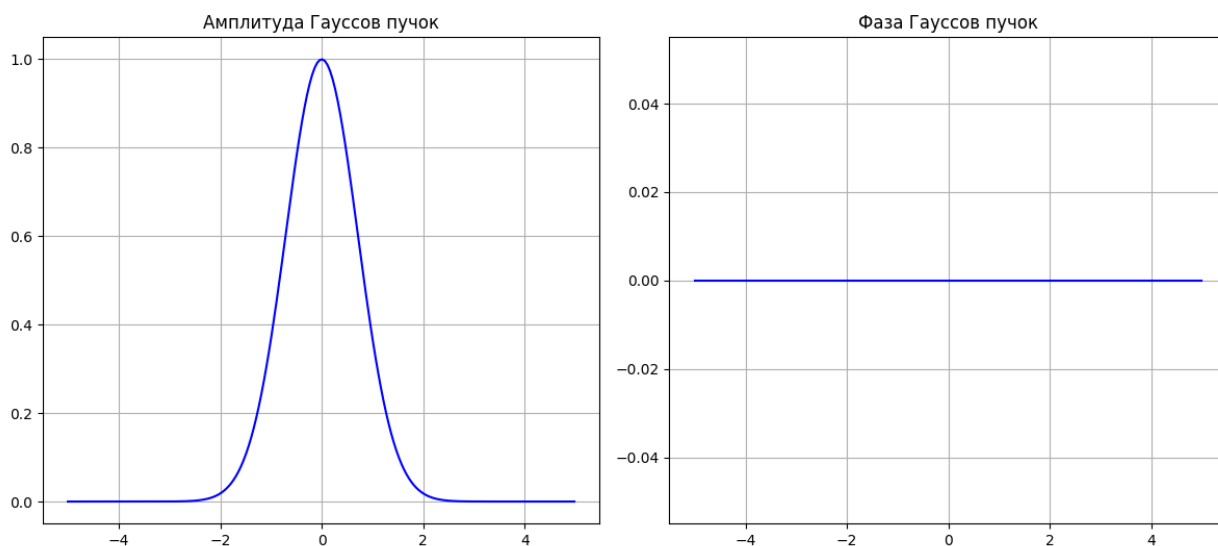


Рисунок 1 – Графики амплитуды и фазы пучка Гаусса

На рисунке 2 представлены графики амплитуды и фазы результата финитного быстрого преобразования Фурье Гауссова пучка.

Амплитуда и фаза БПФ Гауссова пучка

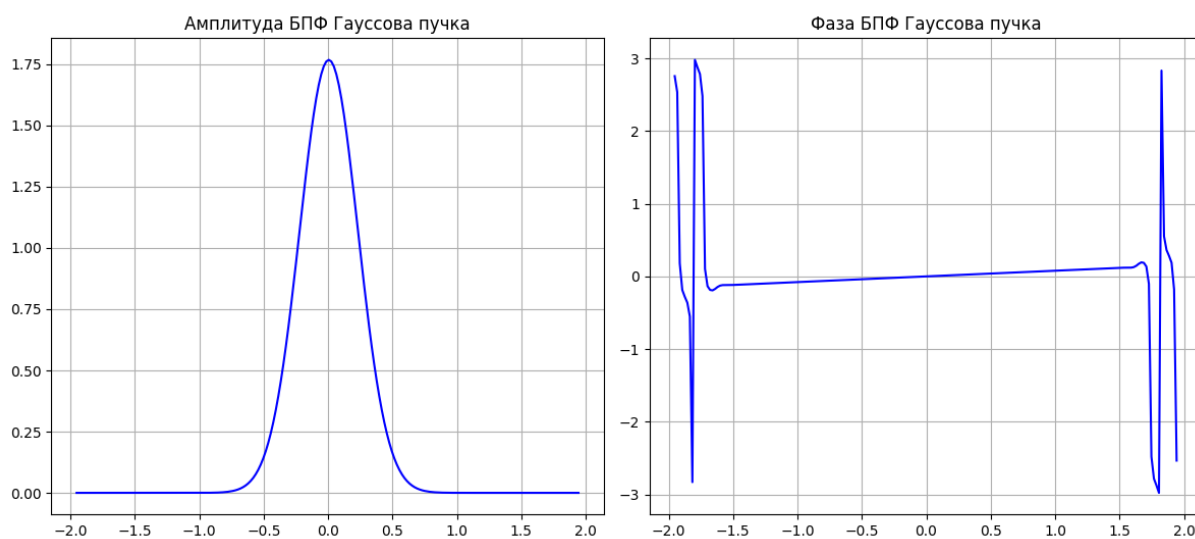


Рисунок 2 – Графики амплитуды и фазы БПФ пучка Гаусса

Чтобы убедиться в правильности реализации преобразования, необходимо выполнить преобразование Фурье методом численного интегрирования и сравнить графики результатов. Результаты преобразования представлены на рисунке 3.

Можно заметить, что графики БПФ и метода численного интегрирования полностью совпадают.

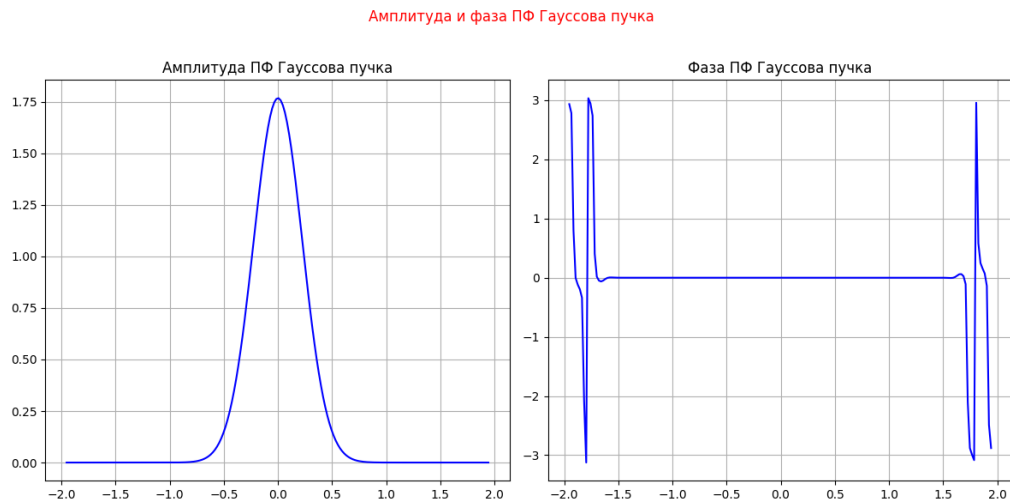


Рисунок 3 – Амплитуда и фаза пучка Гаусса в результате применения преобразования Фурье методом численного интегрирования

2.2 Функция согласно варианту

На рисунке 4 представлены амплитуда и фаза входного светового поля, согласно варианту: $f(x) = \text{rect}\left(\frac{x+2}{2}\right)$

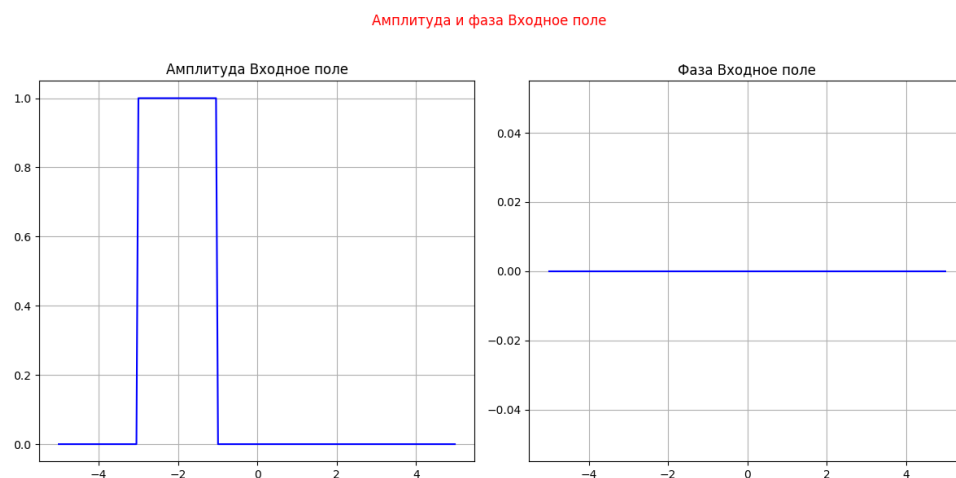


Рисунок 4 – Амплитуда и фаза входного поля

На рисунке 5 представлены графики амплитуды и фазы входного поля в результате применения быстрого преобразования Фурье.

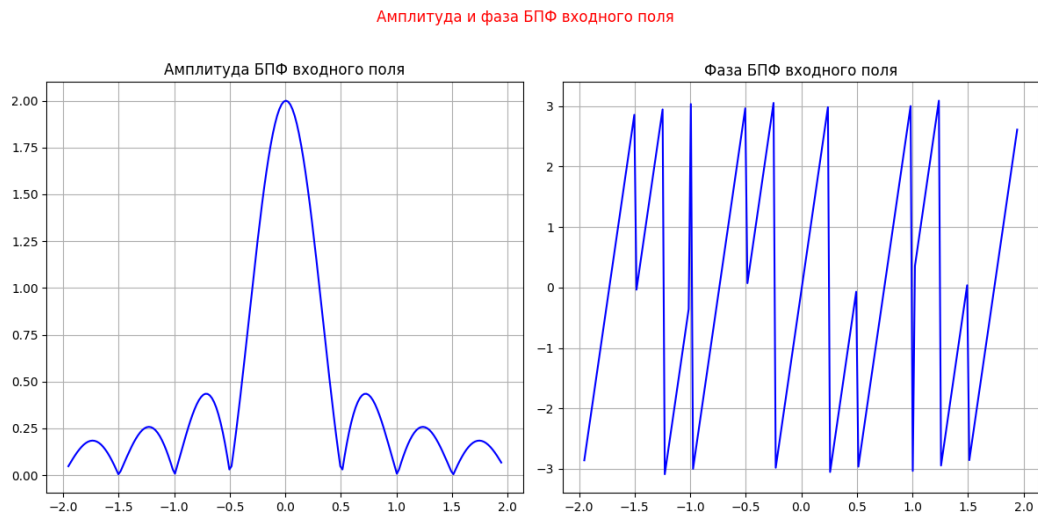


Рисунок 5 – Амплитуда и фаза входного поля в результате применения БПФ

Далее следует рассчитать аналитически результат преобразования своего варианта поля и построить график. Аналитически данное поле рассчитывается следующим образом:

$$\begin{aligned}
 \operatorname{rect}\left(\frac{x+2}{2}\right) &= \begin{cases} 1, & |x+2| \leq 1; \\ 0, & |x+2| > 1; \end{cases} \\
 \int_{-\infty}^{+\infty} \operatorname{rect}\left(\frac{x+2}{2}\right) e^{-i2\pi x\xi} dx &= \int_{-3}^{-1} e^{-i2\pi x\xi} dx = -\frac{1}{i2\pi\xi} \int_{-3}^{-1} e^{-i2\pi x\xi} d(i2\pi x\xi) = \\
 &= -\frac{1}{i2\pi\xi} (e^{-i2\pi\xi(-1)} - e^{-i2\pi\xi(-3)}) = \frac{4e^{i2\pi\xi} \sin(4\pi\xi)}{4\pi\xi} = 4e^{i2\pi\xi} \operatorname{sinc}(4\xi).
 \end{aligned}$$

Сравним графики аналитического решения и БПФ для входного сигнала $f(x) = \operatorname{rect}\left(\frac{x+2}{2}\right)$. Результаты представлены на рисунке 6.

Сравнение амплитуд и фаз

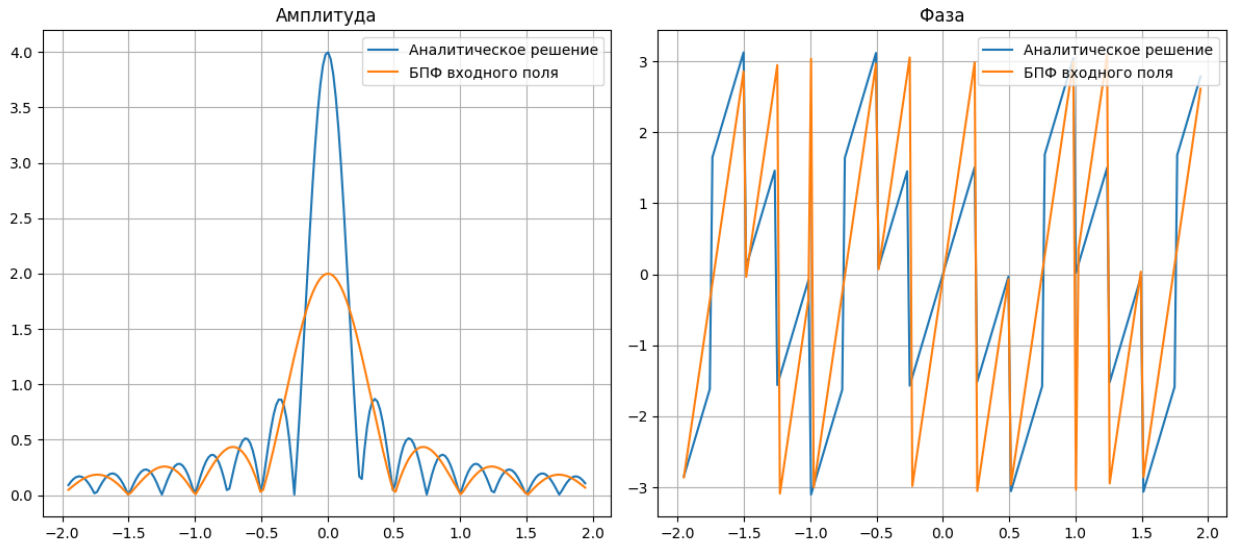


Рисунок 6 – Сравнение графиков амплитуды и фазы входного поля для БПФ и аналитического решений

2.3 Исследование параметров N и M

Перейдем к исследованию параметров N и M алгоритма БПФ, для этого будем варьировать значения N и M . На рисунках 7-12 представлены графики БПФ пучка Гаусса при различных значениях этих параметров, а выводы о влиянии параметров на графики представлены в таблице 1.

Исследование влияния $N=128$, $M=128$

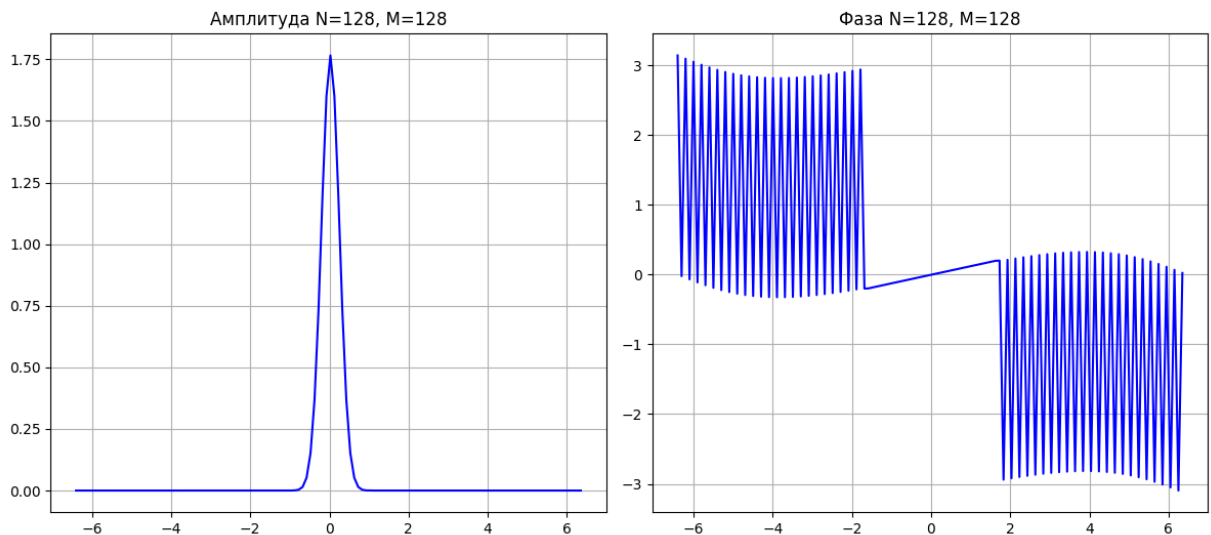


Рисунок 7 – Графики амплитуды и фазы пучка Гаусса $N = M = 128$

Исследование влияния $N=256, M=256$

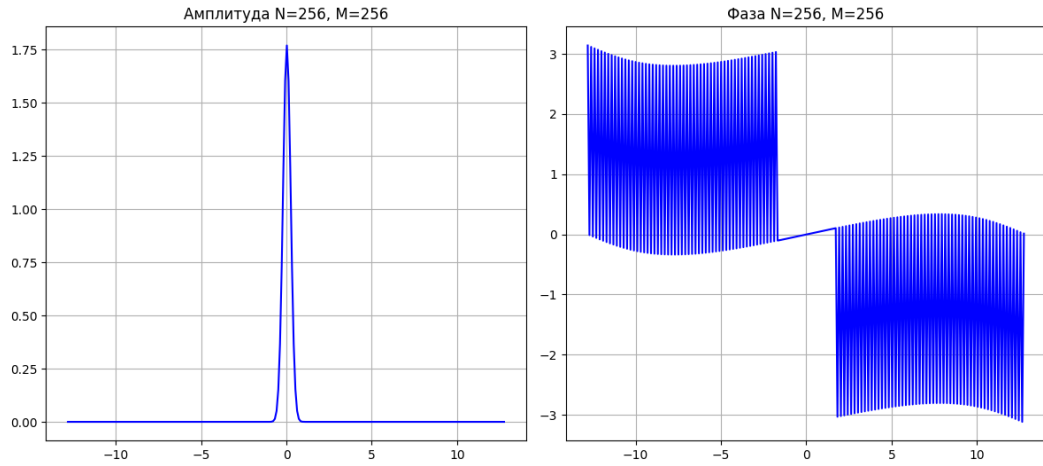


Рисунок 8 – Графики амплитуды и фазы пучка Гаусса $N = M = 256$

Исследование влияния $N=512, M=512$

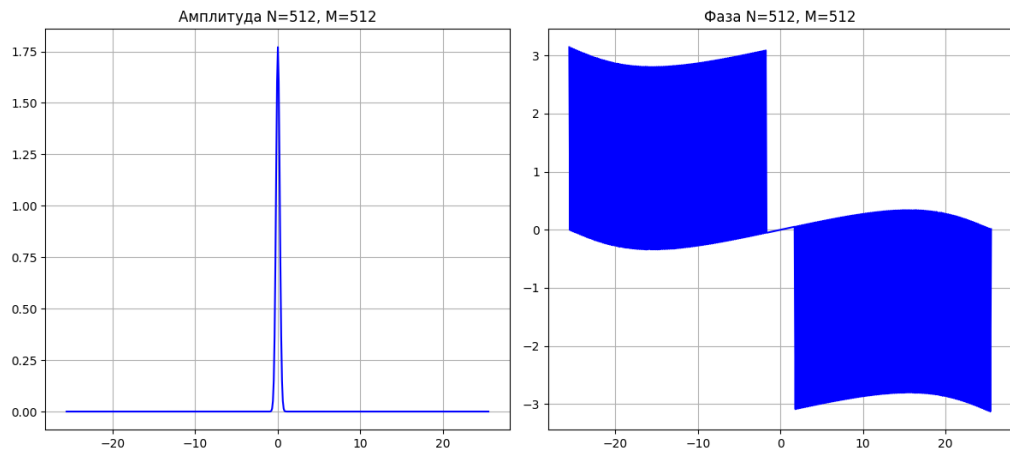


Рисунок 9 – Графики амплитуды и фазы пучка Гаусса $N = M = 512$

Исследование влияния $N=256, M=512$

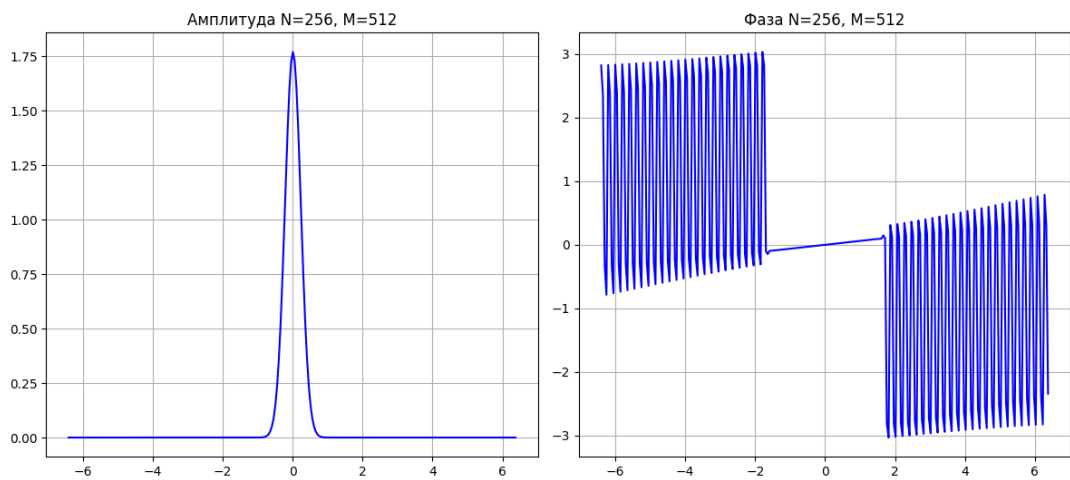


Рисунок 10 – Графики амплитуды и фазы пучка Гаусса $N = 256, M = 512$

Исследование влияния $N=256$, $M=1024$

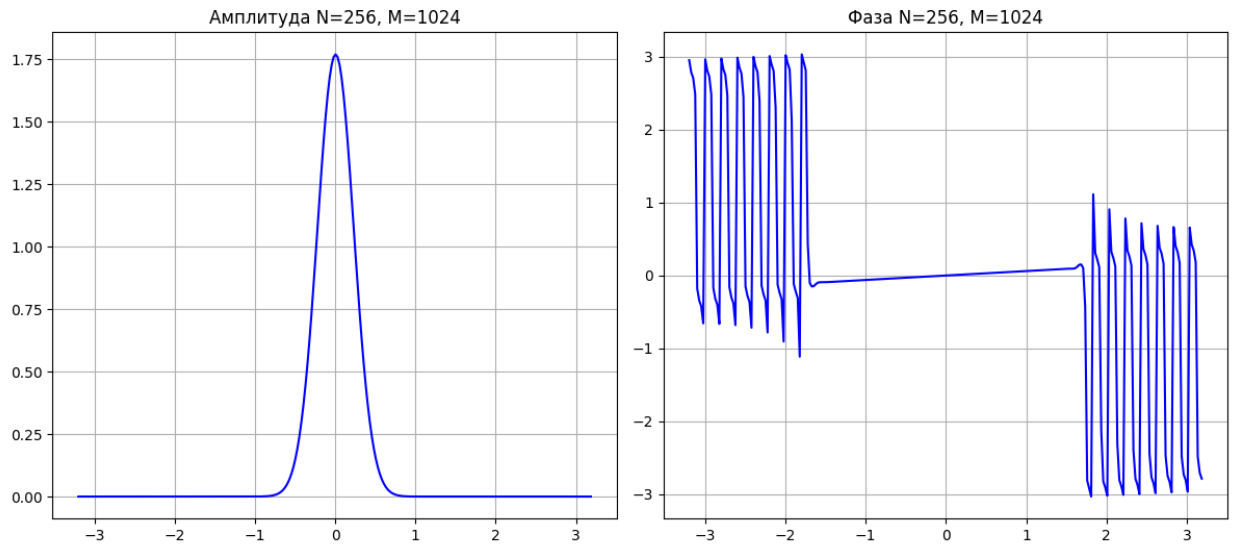


Рисунок 11 – Графики амплитуды и фазы пучка Гаусса $N = 256, M = 1024$

Исследование влияния $N=256$, $M=2048$

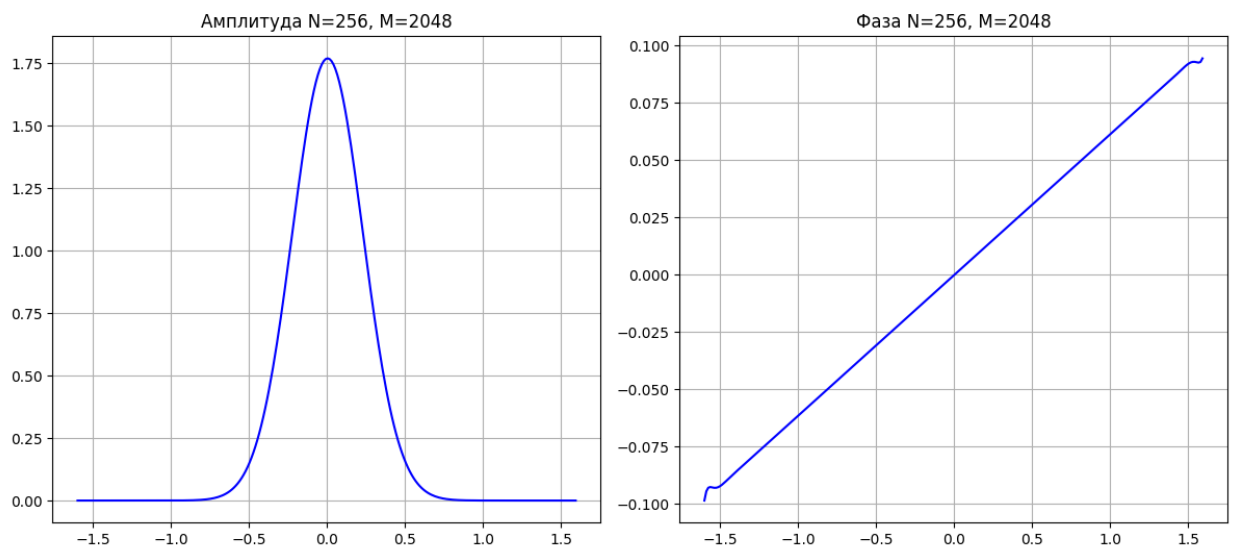


Рисунок 12 – Графики амплитуды и фазы пучка Гаусса $N = 256, M = 2048$

Таблица 1 – Влияние параметров на график результата преобразования

N	<p>При увеличении параметра на фиксированном промежутке увеличивается количество пиков, график становится более ломаным. Промежуток $[-b, b]$ увеличивается, чем больше N, тем меньше деталей будет видно на графике, при уменьшении параметра N происходит обратная ситуация – график приближается, промежуток $[-b, b]$ уменьшается.</p>
M	<p>При увеличении параметра M график приближается, промежуток $[-b, b]$ уменьшается, а при уменьшении – отдаляется, промежуток $[-b, b]$ увеличивается. Чем меньше M, тем меньше деталей будет видно на графике.</p>

3 Двумерный случай

3.1 Двумерный пучок Гаусса

Рассмотрим теперь двумерный Гауссов пучок, который примет вид $f(x) = e^{-x^2-y^2}$. Графики амплитуды и фазы двумерного Гауссова пучка представлены на рисунке 13.

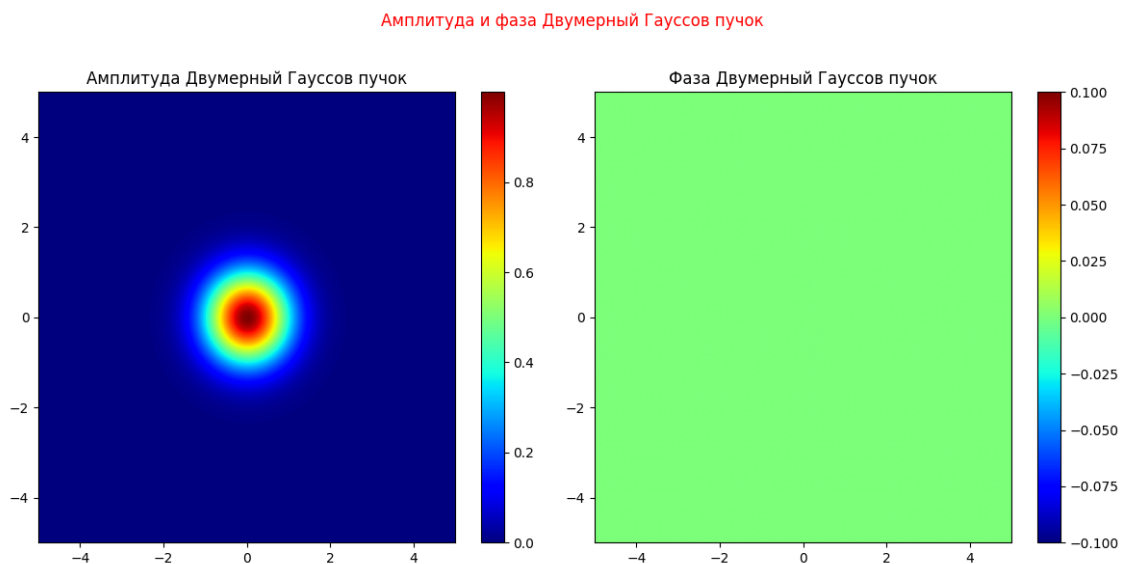


Рисунок 13 – Амплитуда и фаза двумерного Гауссова пучка

Теперь применим алгоритм быстрого преобразования Фурье для двумерного Гауссова пучка. Графики амплитуды и фазы представлены на рисунке 14.

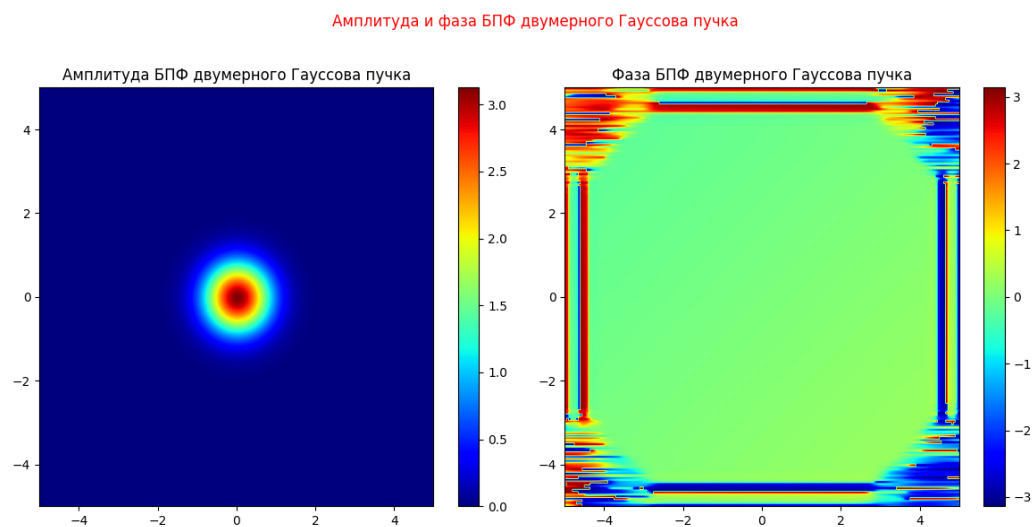


Рисунок 14 – Амплитуда и фаза БПФ двумерного Гауссова пучка

3.2 Функция согласно варианту

Также рассмотрим и двумерное входное поле, графики амплитуды и фазы которого представлены на рисунке 15. В результате применения БПФ были получены графики для амплитуды и фазы, которые представлены на рисунке 16.

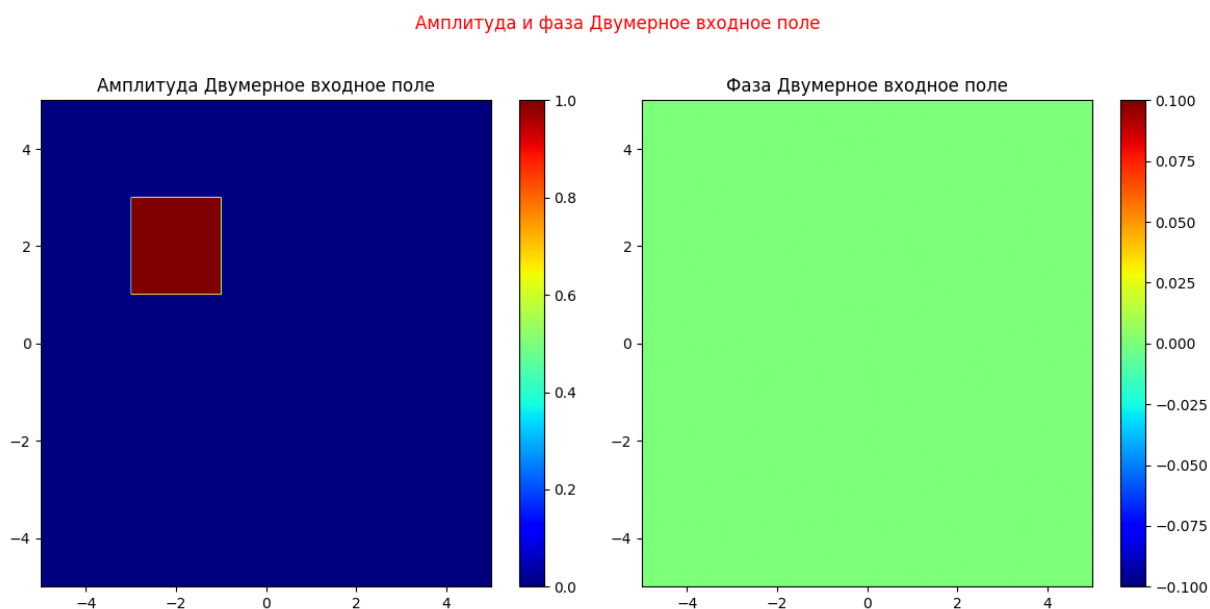


Рисунок 15 – Амплитуда и фаза двумерного входного поля

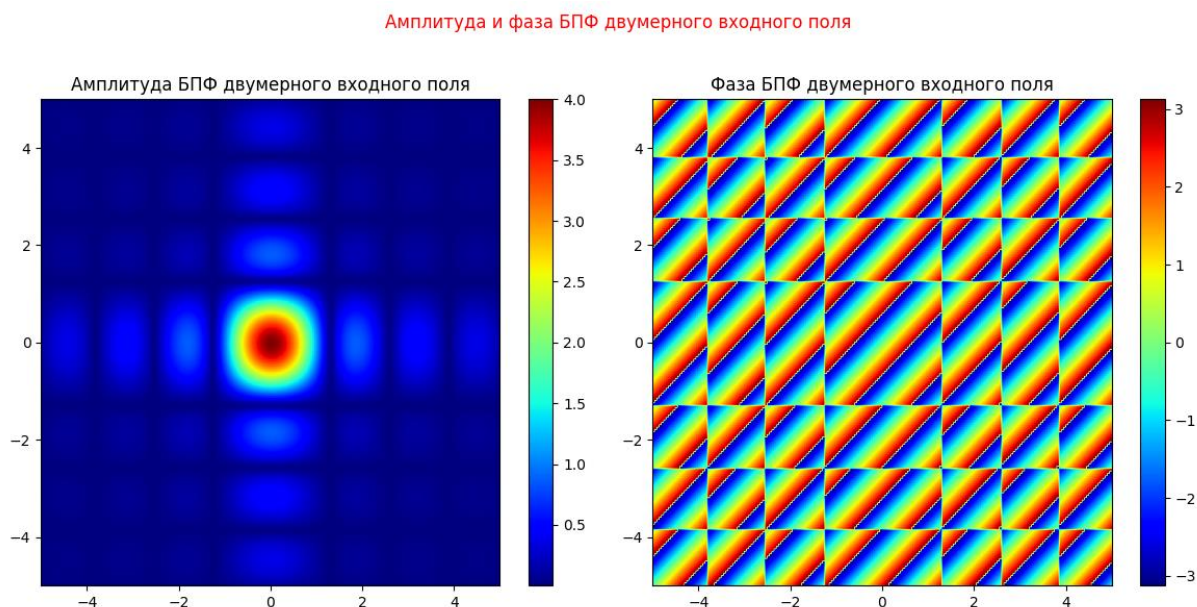


Рисунок 16 – Амплитуда и фаза БПФ двумерного входного поля

Также был рассчитан аналитически результат двумерного преобразования, который представлен на рисунке 17.

Амплитуда и фаза Аналитическое двумерное решение

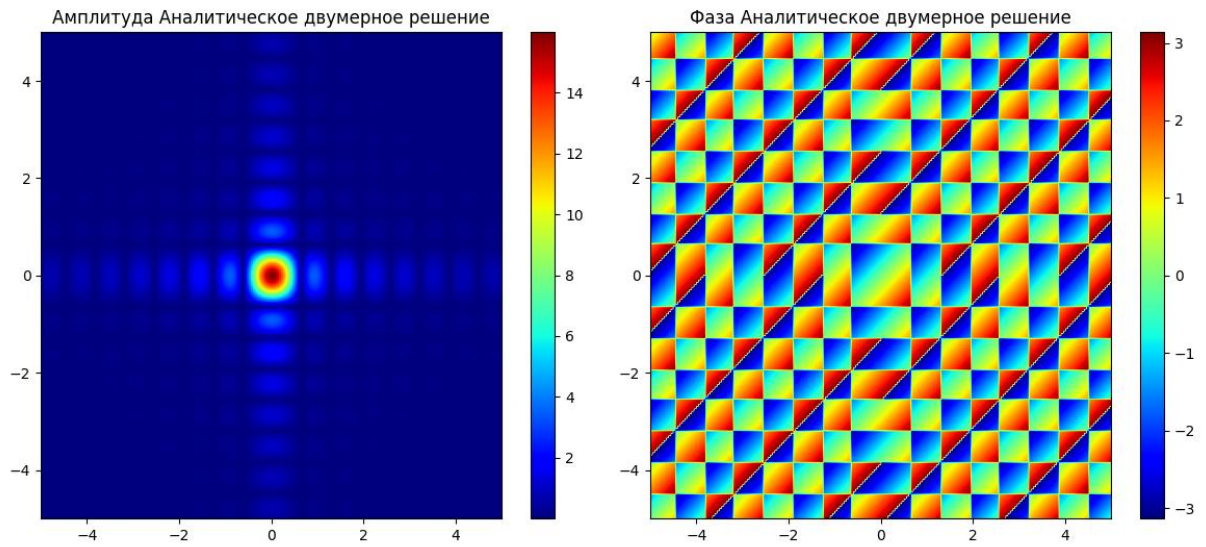


Рисунок 17 – Амплитуда и фаза двумерного входного поля (аналитическое)

4 Код программы на языке Python

```
from typing import List, Optional

import matplotlib.pyplot as plt
import numpy as np
from numpy.typing import NDArray

def gauss_beam(x: NDArray[np.float64]) -> NDArray[np.float64]:
    """Гауссов пучок"""
    return np.exp(-x ** 2)

def plot_signal(x: NDArray[np.float64], signal:
    NDArray[np.complex128], title: str, phase: bool = False) -> None:
    """График сигнала"""
    plt.figure()
    data = np.angle(signal) if phase else np.abs(signal)
    plt.plot(x, data)
    plt.title(title)
    plt.grid(True)
    plt.tight_layout()
    plt.show()

def fft_process(signal: NDArray[np.float64], M: int, N: int, hx:
    float) -> NDArray[np.complex128]:
    """Обработка сигнала через БПФ"""
    padded_signal = np.pad(signal, pad_width=(M - N) // 2,
    mode='constant')
    FFT_signal =
    np.fft.fftshift(np.fft.fft(np.fft.fftshift(padded_signal))) * hx
    return FFT_signal[M // 2 - N // 2:M // 2 + N // 2]

def fourier_integral(x: NDArray[np.float64], u:
    NDArray[np.float64], signal: NDArray[np.float64], hx: float) ->
    NDArray[
        np.complex128]:
    """Преобразование Фурье через интеграл"""
    X, U = np.meshgrid(x, u)
    Kernel = np.exp(-2j * np.pi * X * U)
    return Kernel @ signal * hx

def finite_fourier_integral(f_x, x, u, hx):
    F_u = np.zeros(len(u), dtype=complex)
    for i in range(len(u)):
        F_u[i] = np.sum(f_x * np.exp(-2j * np.pi * u[i] * x)) * hx
    # Метод прямоугольников
    return F_u

def input_field(x: NDArray[np.float64]) -> NDArray[np.float64]:
    """Входное поле rect((x + 2) / 2)"""
    return np.where(np.abs(x + 2) < 1, 1, 0)

def analytical_solution(u: NDArray[np.float64]) ->
    NDArray[np.float64]:
    """Аналитическое решение"""
    return 4 * np.exp(2 * 1j * np.pi * u) * np.sinc(4 * u)
```

```

def fft_2d_process(field: NDArray[np.float64], M: int, N: int, hx:
float) -> NDArray[np.complex128]:
    """Обработка двумерного поля через БПФ"""
    field = np.apply_along_axis(fft_process, axis=0, arr=field,
M=M, N=N, hx=hx)
    field = np.apply_along_axis(fft_process, axis=1, arr=field,
M=M, N=N, hx=hx)
    return field

def plot_2d_results(field: NDArray[np.complex128], title: str,
extent: Optional[List[float]] = None) -> None:
    """Построение двумерных графиков амплитуды и фазы"""
    extent = extent if extent else (-5, 5, -5, 5)
    fig, axes = plt.subplots(1, 2, figsize=(12, 6))

    axes[0].imshow(np.abs(field), extent=extent, cmap='jet',
aspect='auto')
    axes[0].set_title(f'Амплитуда {title}')
    axes[0].figure.colorbar(axes[0].images[0], ax=axes[0])

    axes[1].imshow(np.angle(field), extent=extent, cmap='jet',
aspect='auto')
    axes[1].set_title(f'Фаза {title}')
    axes[1].figure.colorbar(axes[1].images[0], ax=axes[1])

    fig.suptitle(f'Амплитуда и фаза {title}', color='r')
    plt.tight_layout(rect=(0, 0, 1, 0.96))
    plt.show()

def plot_results(x: np.ndarray, signal: np.ndarray, title: str) ->
None:
    """Построение графиков амплитуды и фазы"""
    fig, axes = plt.subplots(1, 2, figsize=(12, 6))

    axes[0].plot(x, np.abs(signal), color='b')
    axes[0].set_title(f'Амплитуда {title}')
    axes[0].grid(True)

    axes[1].plot(x, np.angle(signal), color='b')
    axes[1].set_title(f'Фаза {title}')
    axes[1].grid(True)

    fig.suptitle(f'Амплитуда и фаза {title}', color='r')
    plt.tight_layout(rect=(0, 0, 1, 0.96))
    plt.show()

def plot_results_multiple(x: np.ndarray, signals:
List[NDArray[np.complex128]], titles: List[str]) -> None:
    """Построение графиков амплитуды и фазы для нескольких
сигналов"""
    fig, axes = plt.subplots(1, 2, figsize=(12, 6))

    for signal, label in zip(signals, titles):
        axes[0].plot(x, np.abs(signal), label=label)
        axes[1].plot(x, np.angle(signal), label=label)

    axes[0].set_title('Амплитуда')
    axes[1].set_title('Фаза')

```

```

for ax in axes:
    ax.grid(True)
    ax.legend()

fig.suptitle('Сравнение амплитуд и фаз', color='r')
plt.tight_layout(rect=(0, 0, 1, 0.96))
plt.show()

def experiment_change_N_M(signal_func, a: float) -> None:
    N_values = np.array([128, 256, 512])
    M_values = [512, 1024, 2048]

    for N in N_values:
        b = N / (4 * a)
        hx = (2 * a) / N
        hu = (2 * b) / N
        x = np.linspace(-a, a - hx / 2, N)
        u = np.linspace(-b, b - hu / 2, N)

        signal = signal_func(x)
        FFT_signal = fft_process(signal, N, N, hx)

        plot_fft_results(u, FFT_signal, N, N)

    N = 256
    for M in M_values:
        b = (N ** 2) / (4 * a * M)
        hx = (2 * a) / N
        hu = (2 * b) / N
        x = np.linspace(-a, a - hx / 2, N)
        u = np.linspace(-b, b - hu / 2, N)

        signal = signal_func(x)
        FFT_signal = fft_process(signal, M, N, hx)

        plot_fft_results(u, FFT_signal, N, M)

def plot_fft_results(u, FFT_signal, N, M):
    plt.figure(figsize=(12, 6))

    plt.subplot(1, 2, 1)
    plt.title(f'Амплитуда N={N}, M={M}')
    plt.plot(u, np.abs(FFT_signal), color='b')
    plt.grid(True)

    plt.subplot(1, 2, 2)
    plt.title(f'Фаза N={N}, M={M}')
    plt.plot(u, np.angle(FFT_signal), color='b')
    plt.grid(True)

    plt.suptitle(f'Исследование влияния N={N}, M={M}', color='r')
    plt.tight_layout(rect=(0, 0, 1, 0.96))
    plt.show()

def main() -> None:
    M, N = 1024, 200
    a = 5
    b = (N ** 2) / (4 * a * M)
    hx = (2 * a) / N
    hu = (2 * b) / N
    x = np.linspace(-a, a - hx / 2, N)

```



```

u = np.linspace(-b, b - hu / 2, N)

# Гауссов пучок
gauss = gauss_beam(x)
plot_results(x, gauss, 'Гауссов пучок')

FFT_gauss = fft_process(gauss, M, N, hx)
plot_results(u, FFT_gauss, 'БПФ Гауссова пучка')

G = fourier_integral(x, u, gauss, hx)
plot_results(u, G, 'ПФ Гауссова пучка')

# Входное поле
f = input_field(x)
plot_results(x, f, 'Входное поле')

# БПФ входного поля
FFT2 = fft_process(f, M, N, hx)
plot_results(u, FFT2, 'БПФ входного поля')

# Входное поле - аналитическое решение
FA = analytical_solution(u)
# plot_results_multiple(u, [FA, FFT2], ['Аналитическое
решение', 'БПФ входного поля'])

# Двумерный гауссов пучок
X, Y = np.meshgrid(x, x)
gauss2 = np.exp(-(X ** 2 + Y ** 2))
plot_2d_results(gauss2, 'Двумерный Гауссов пучок')

# БПФ двумерного гауссова пучка
gauss2 = fft_2d_process(gauss2, M, N, hx)
plot_2d_results(gauss2, 'БПФ двумерного Гауссова пучка')

# Двумерное входное поле
f2 = np.zeros_like(X)
f2[(np.abs(X + 2) < 1) & (np.abs(Y + 2) < 1)] = 1
plot_2d_results(f2, 'Двумерное входное поле')

# БПФ двумерного входного поля
f2 = fft_2d_process(f2, M, N, hx)
plot_2d_results(f2, 'БПФ двумерного входного поля')

# Аналитическое двумерное решение
FA2 = np.outer(FA, FA)
plot_2d_results(FA2, 'Аналитическое двумерное решение')

experiment_change_N_M(gauss_beam, a)

if __name__ == '__main__':
    main()

```