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, математики и электроники

Факультет информатики  
Кафедра технической кибернетики

ЛАБОРАТОРНАЯ РАБОТА № 2

по курсу   
Оптическая информатика

Группа 6410-010302D

Студент Борисов Д. С.

(*подпись*)

Преподаватель,

к.ф.-м.н. А.П. Порфирьев

(*подпись*)

Самара 2020

**ЗАДАНИЕ**

1. Реализовать одномерное финитное преобразование Фурье с помощью применения алгоритма БПФ.
2. Построить график гауссова пучка . Здесь и далее для каждого графика следует строить отдельно графики амплитуды и фазы.
3. Убедиться в правильности реализации преобразования, подав на вход гауссов пучок – собственную функцию преобразования Фурье. На выходе тоже должен получиться гауссов пучок (построить график на правильной области определения ). Рекомендуемая входная область:
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.
8. Выполнить пункты 1-3 и 6-7 для двумерного случая. Графики изменятся на двумерные изображения, одномерные функции следует заменить на двумерные, равные произведению соответствующих одномерных функций. Например, гауссов пучок поменяется на .

Таблица 1 – Исходные данные на ЛР № 1

|  |  |  |
| --- | --- | --- |
| 4 |  | Для аналитики применить **не** финитное преобразование. |

Далее будут использоваться следующие параметры: .

**РЕФЕРАТ**

**Отчет по лабораторной работе:** 23 c., 14 рисунков, 1 таблица, 8 источников, 1 приложение.

ОПТИЧЕСКАЯ ИНФОРМАТИКА, ИНТЕГРАЛЬНОЕ ПРЕОБРАЗОВАНИЕ

Данная лабораторная работа содержит сведения о применении быстрого преобразования Фурье (БПФ) и его адаптации для оптических систем.

Необходимо реализовать оптическое преобразование Фурье, используя алгоритм БПФ и стандартные методы численного интегрирования, сравнить результаты и убедиться, что они совпадают.

Также необходимо изучить некоторые свойства преобразования Фурье с помощью аналитических выводов и численного моделирования.**СОДЕРЖАНИЕ**

[**ЗАДАНИЕ** 2](#_Toc56339889)

[**РЕФЕРАТ** 3](#_Toc56339890)

[**1 ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ** 5](#_Toc56339891)

[**2 ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ** 10](#_Toc56339892)

[**2.1 Решение одномерной задачи** 10](#_Toc56339893)

[**2.3 Решение двумерной задачи** 15](#_Toc56339894)

[**ЗАКЛЮЧЕНИЕ** 18](#_Toc56339895)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 19](#_Toc56339896)

[**ПРИЛОЖЕНИЕ А** 20](#_Toc56339897)

**1 ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

Рассматривается оптическая система, изображённая на рисунке 1.

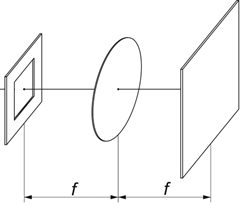


Рисунок 1 – Схема оптической системы

Будем считать, что на вход системе падает плоская волна, тождественно равная единице. Квадратная апертура ограничивает входной пучок, так что мы считаем его равным нулю за пределами апертуры. В отверстии апертуры находится дифракционный оптический элемент (ДОЭ). Линза расположена между апертурой и экраном на фокусном расстоянии от них.

Для упрощения будем рассматривать только те случаи, когда , где – длина волны, – фокусное расстояние линзы. Тогда оператор распространения запишется в виде преобразования Фурье в конечных пределах (или финитного преобразования Фурье):

, (1)

где – финитная функция (), задающая вид оптического распределения после прохождения ДОЭ, – квадратная область внутри апертуры, – спектр, – оператор преобразования Фурье в конечных пределах.

Обычное преобразование Фурье отличается от финитного лишь тем, что интегрирование в последнем случае осуществляется по конечной области.

Так как входное распределение перед апертурой равно единице, а при прохождении через ДОЭ эффекты дифракции упрощаются до обыкновенного умножения, то входное распределение будет совпадать с функцией пропускания ДОЭ.

Важно: мы считаем толщину апертуры и ДОЭ в приближении бесконечно малой величиной.

Прежде чем приступать к двумерной задаче (т.к. функции зависят от двух переменных), рассмотрим более простую – одномерную:

. (2)

Для расчёта такого преобразования можно воспользоваться алгоритмом быстрого преобразование Фурье (БПФ). Он не рассматривается в данном пособии, поэтому можно использовать готовую реализацию. Однако, при реализации финитного преобразования Фурье по формуле (2) через БПФ следует учитывать нижеописанные замечания.

Предположим, что после дискретизации функции получается вектор размерности :

(3)

где – символ транспонирования. Здесь и далее подразумевается, что количество отсчётов чётно.

Классически дискретное преобразования Фурье записывается для периодических функций в виде:

(4)

где – спектр дискретного преобразования Фурье. Формула (4) аппроксимирует следующий интеграл:

. (5)

Иными словами, классическое преобразование Фурье подразумевает, что пределы интегрирования начинаются с нуля, в то время как наше преобразование Фурье центрировано.

Если продолжить функцию с периодом на всю числовую прямую, то для того чтобы дискретное преобразование (4) аппроксимировало интеграл (2), а не (5), необходимо поменять местами первую и вторую половины компонентов вектора (3).

Замечание: классическое прямое преобразование Фурье (4) не учитывает шаг дискретизации , поэтому после применения операции БПФ необходимо

умножить результат на .

После выполнения БПФ мы получаем вектор значений, но он будет определён для функции, заданной на промежутке , где предстоит определить. Поскольку нас интересует центрированная система, необходимо снова поменять местами первую и вторую половины компонентов полученного вектора, получив итоговый вектор . Тогда область задания функции изменится на .

При использовании БПФ будет выполняться соотношение: , где – шаг дискретизации по оси . Отсюда видно, что:

. (6)

Из формулы (6) следует: чем больше точек дискретизации взять, тем больше будет область задания функции При малых аппроксимация будет плохой, а при больших промежуток может быть настолько большим, что важных деталей функции мы просто не увидим.

В этом случае поступают следующим образом: исходный вектор и слева, и справа дополняют одинаковым количеством нулей, зачастую много большим, чем *N*. Будем считать, что после дополнения нулями вектор стал иметь размерность . После выполнения алгоритма будет также иметь размерность , а функция по-прежнему определена на промежутке

Если же теперь «вырезать» центральную часть вектора , оставив элементов, то область задания функции станет равной , где

. (7)

Таким образом, мы получаем алгоритм реализации оптического преобразования Фурье в конечных пределах через использование БПФ:

1. Провести дискретизацию входной функции в вектор с размерностью .
2. Дополнить вектор и слева, и справа необходимым числом нулей до размерности .
3. Разбить вектор на две половины и поменять их местами.
4. Выполнить БПФ от и умножить результат на шаг , получив вектор .
5. Разбить вектор на две половины и поменять их местами.
6. «Вырезать» центральную часть вектора , оставив центральные элементов.
7. Пересчитать область задания функции по формуле (7).

Если область оказалась слишком большой (полезная часть спектра плохо видна) или слишком маленькой (спектр не умещается), можно соответственно изменить число дополняемых нулей на шаге 2.

Замечание: некоторые реализации БПФ не требуют, чтобы число было целой степенью двойки, а сами добавляют дополнительные нули, нарушая симметрию. Это может привести к появлению в результатах расчёта неправильного фазового набега. Так что **необходимо удостовериться**, что число *M* является степенью двойки.

Вернёмся теперь к формуле (1). Поскольку в формуле имеется двумерная входная функция и двумерное преобразование Фурье, то после дискретизации функций мы будем получать матрицу . Алгоритм нахождения преобразования Фурье от неё можно свести к одномерному случаю: необходимо применить вышеописанный алгоритм к каждой строке этой матрицы, получив новую матрицу, а затем применить его к каждому столбцу полученной матрицы.

Примечание: седьмой шаг, нахождение области задания , следует выполнить только один раз.

**2 ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ**

**2.1 Решение одномерной задачи**

Рассмотрим финитное преобразование Фурье с использованием алгоритма быстрого преобразования Фурье, проведём его исследование и стандартного численного интегрирования (метод прямоугольников). В качестве проверки правильности работы БПФ в качестве входного поля будем использовать гауссов пучок , амплитуда и фаза которого изображены на рисунке 2.

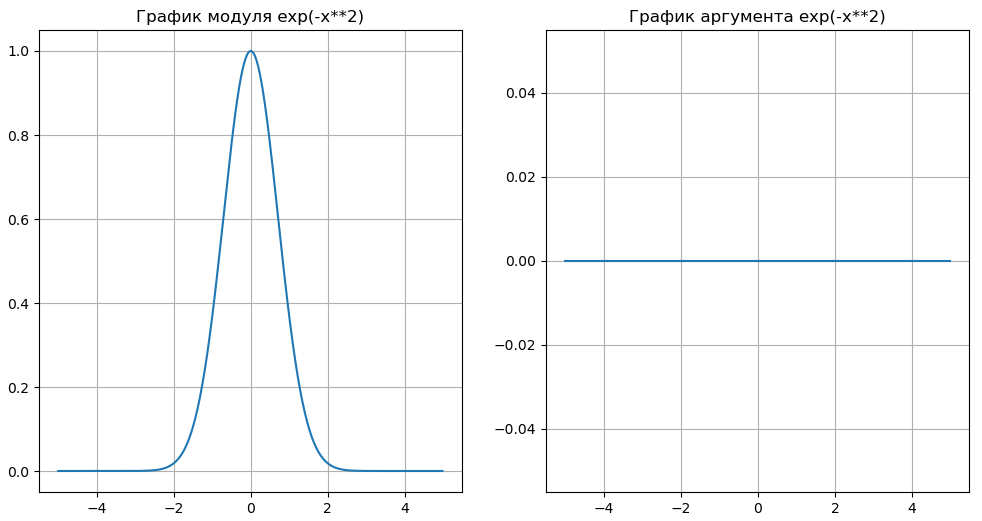


Рисунок 2 — Амплитуда и фаза гауссова пучка

Теперь изобразим на рисунке 3 фазу и амплитуду финитного преобразования Фурье гауссова пучка с использованием БПФ.

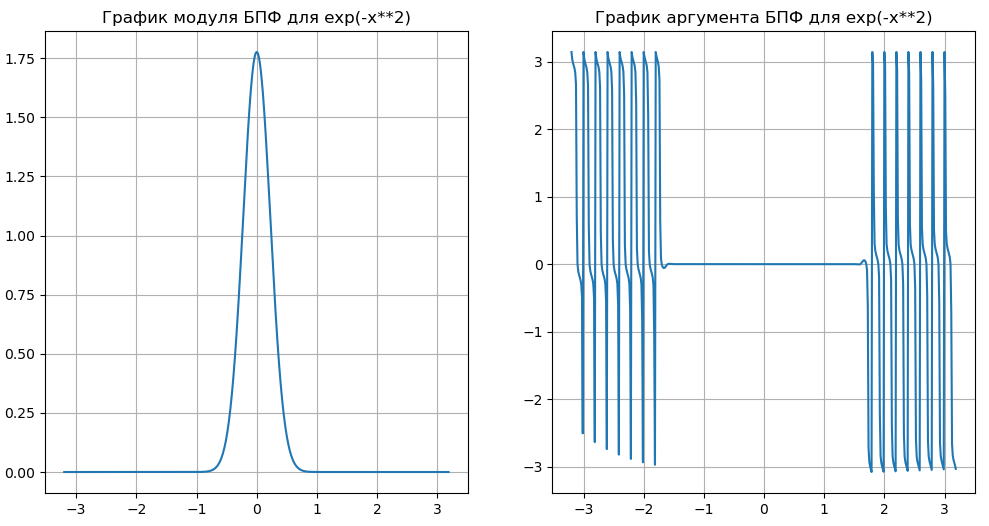


Рисунок 3 — Амплитуда и фаза гауссова пучка после БПФ

Как видно из графиков, после БПФ у гауссового пучка изменился только масштаб. Объясняется это тем, что мода Гаусса имеет свойство инвариантности относительно преобразований Фурье.

Далее рассмотрим преобразование Фурье гауссового пучка с помощью численного метода интегрирования (метод прямоугольников) и сравним с графиками, полученными ранее на рисунке 3. Сравнение методов показано на рисунке 4.

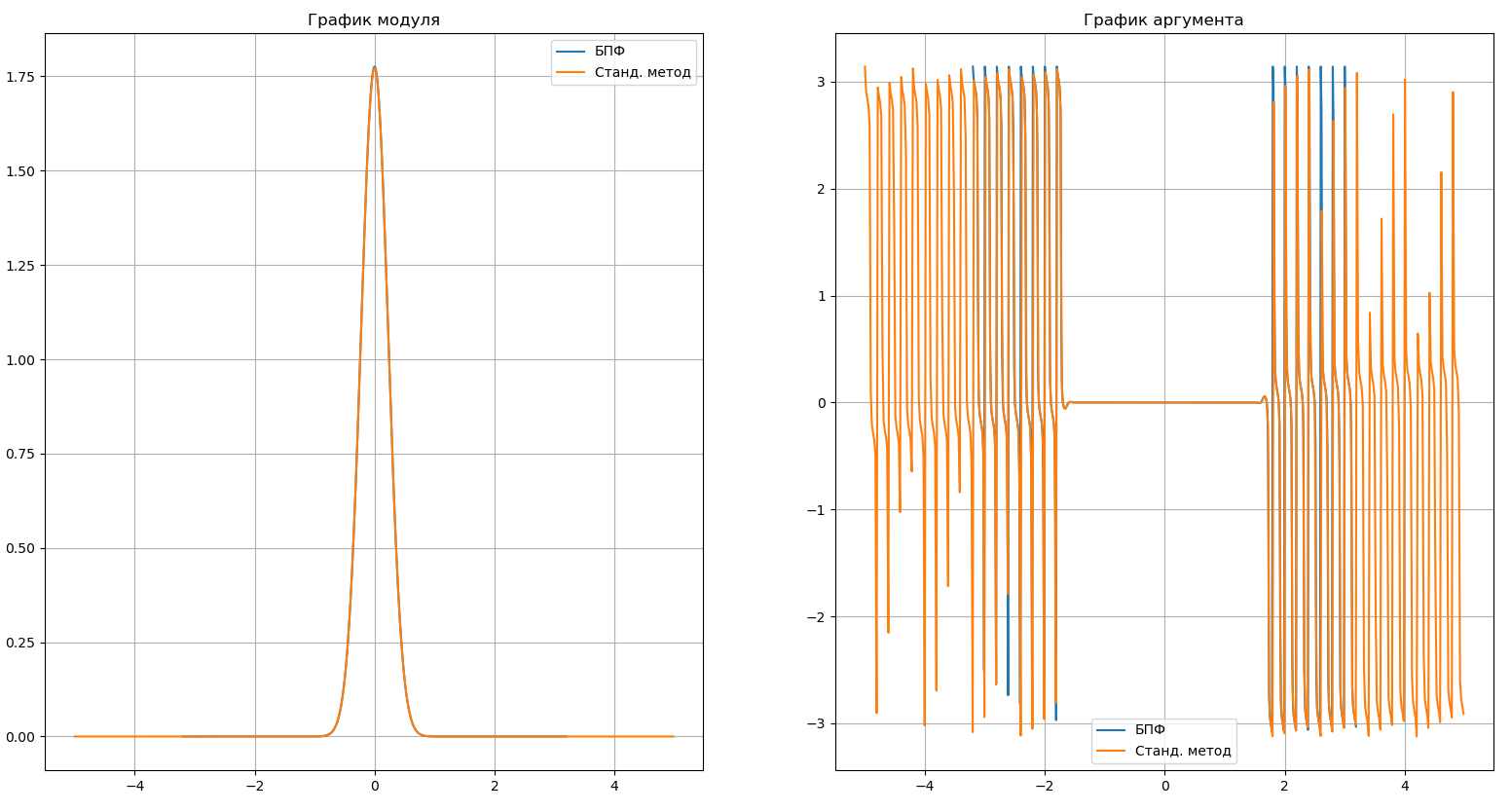


Рисунок 4 — Амплитуда и фаза гауссова пучка после ПФ с помощью ЧМ и с помощью БПФ

Как видно из рисунка, мы получили снова моду Гаусса, но другого масштаба, что также подтверждает инвариантность гауссового пучка относительно преобразования Фурье.

Заметим, что амплитуды и фазы в районе спектра, полученные после БПФ и после ЧМ полностью совпали – это означает, что оба метода преобразования Фурье реализованы верно.

Теперь подадим другое световое поле, соответствующее варианту 4 и проанализируем результаты. На рисунке 5 изображена входная функция.

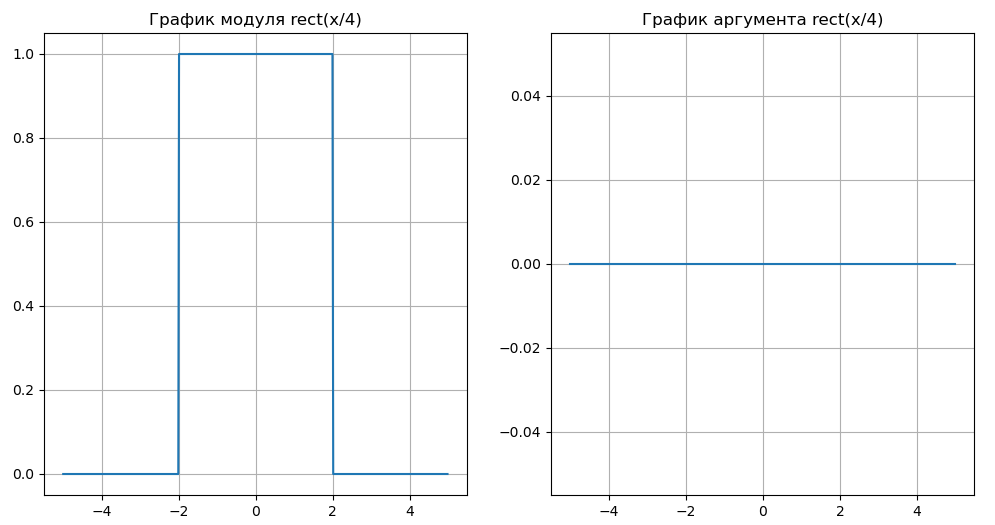


Рисунок 5 — Амплитуда и фаза входного поля

Теперь применим БПФ к данному входному полю. Результат представлен на рисунке 6.

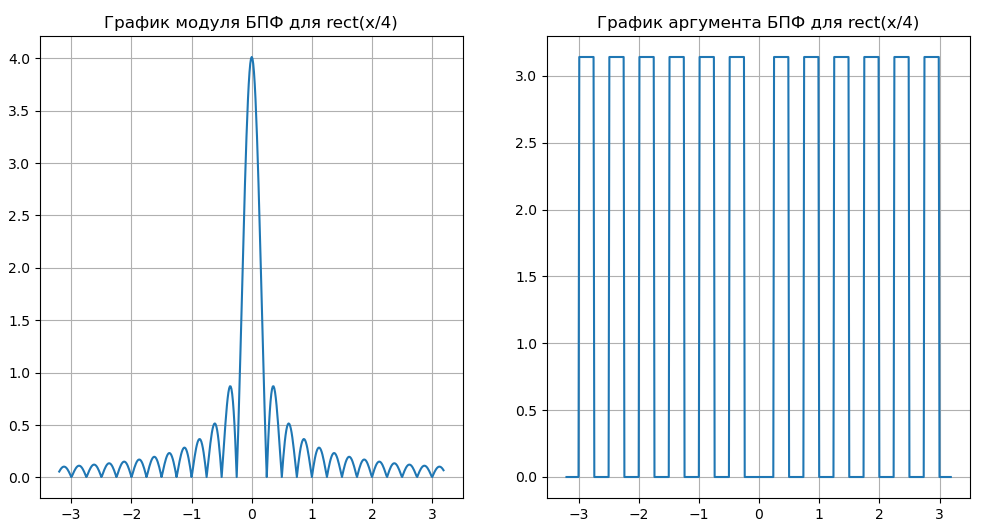


Рисунок 6 — Амплитуда и фаза выходного поля (БПФ)

На рисунке можно увидеть, что изначальный ступенчатый скачок после БПФ стал представлять собой постоянно изменяющееся синусоидальное распределение, отдалённо напоминающее график функции .

Построим теперь аналитически рассчитанное на рисунке 7 преобразование Фурье данной входной функции и сравним с результатом БПФ на рисунке 8.

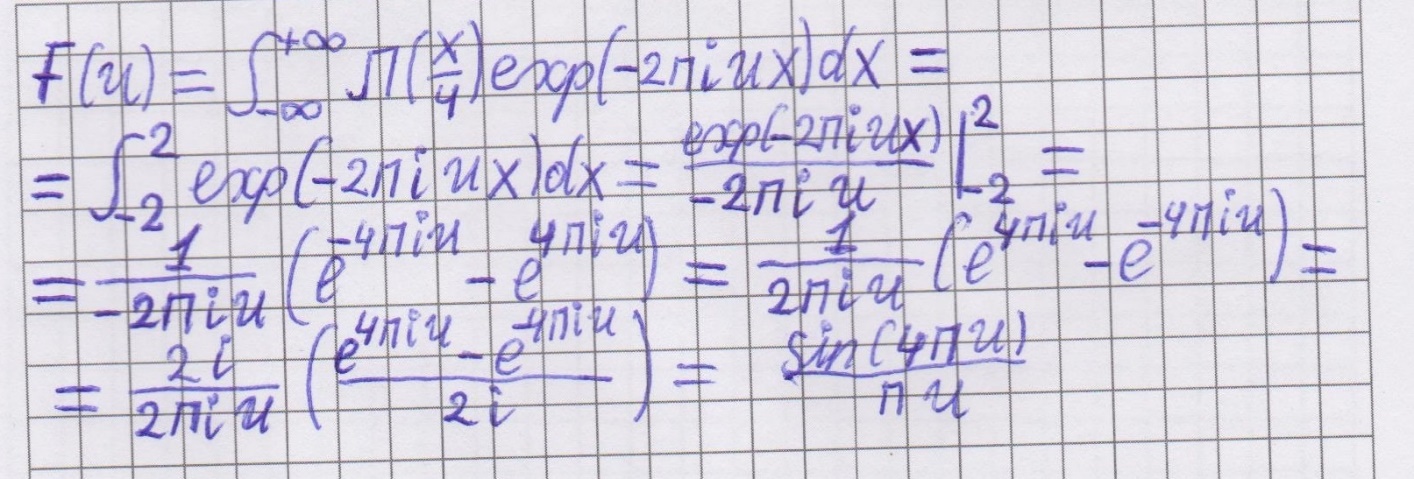


Рисунок 7 – Аналитическое решение преобразования Фурье

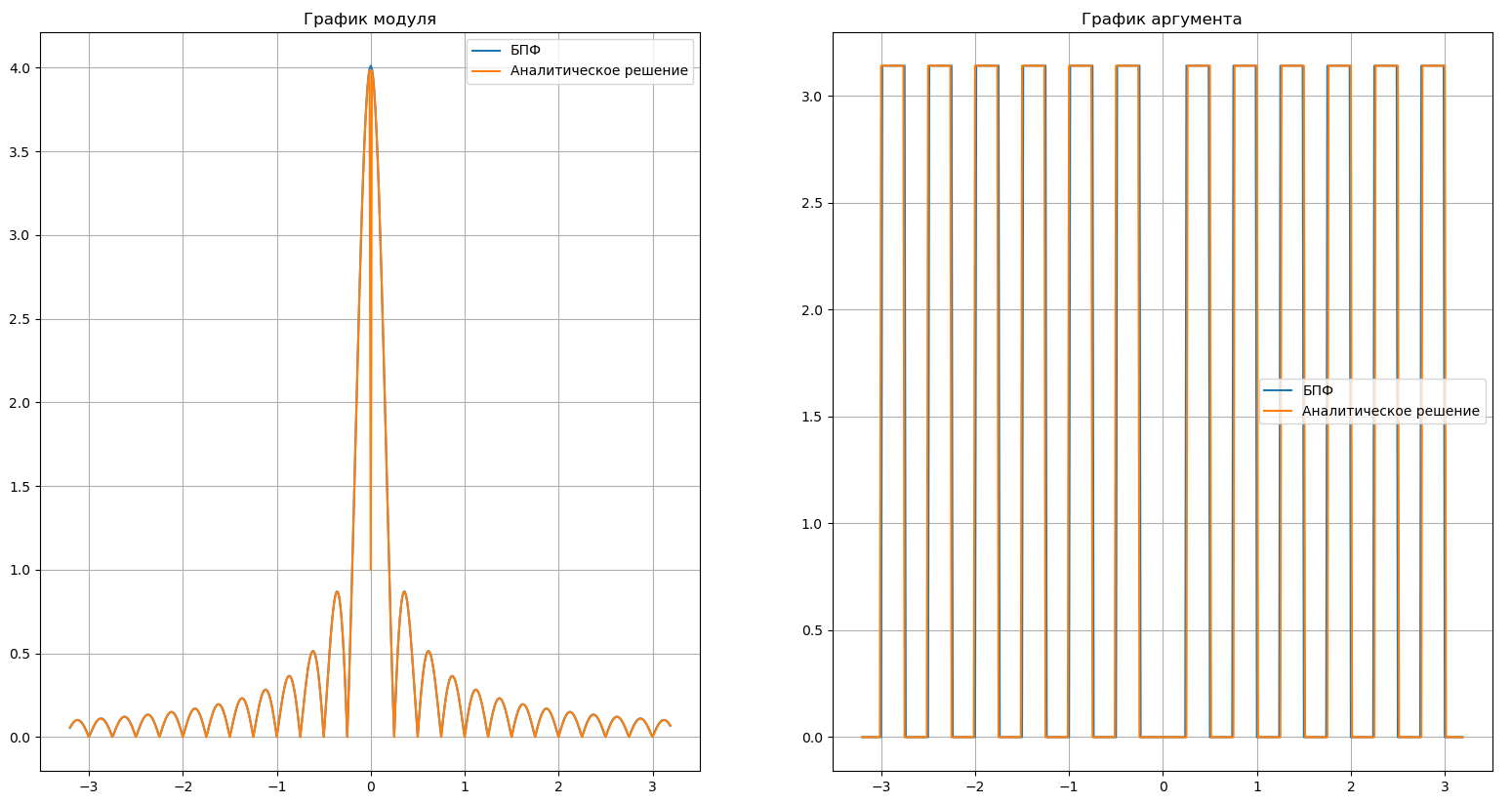


Рисунок 8 — Амплитуда и фаза аналитического решения и БПФ

Сравнение показало, что графики амплитуд и фаз совпадают, однако амплитуда аналитического решения в точке 0 стремится к 1. Это обусловлено тем, что при достижении значения 0 произойдёт деление нуля на ноль, поэтому было решено задать данному решению результат равный единице в точке 0.

**2.2 Решение двумерной задачи**

Проведём исследование теперь применения БПФ к двумерной моде Гаусса. Двумерный график моды и БПФ представлены на рисунке 10 и 11 соответственно.

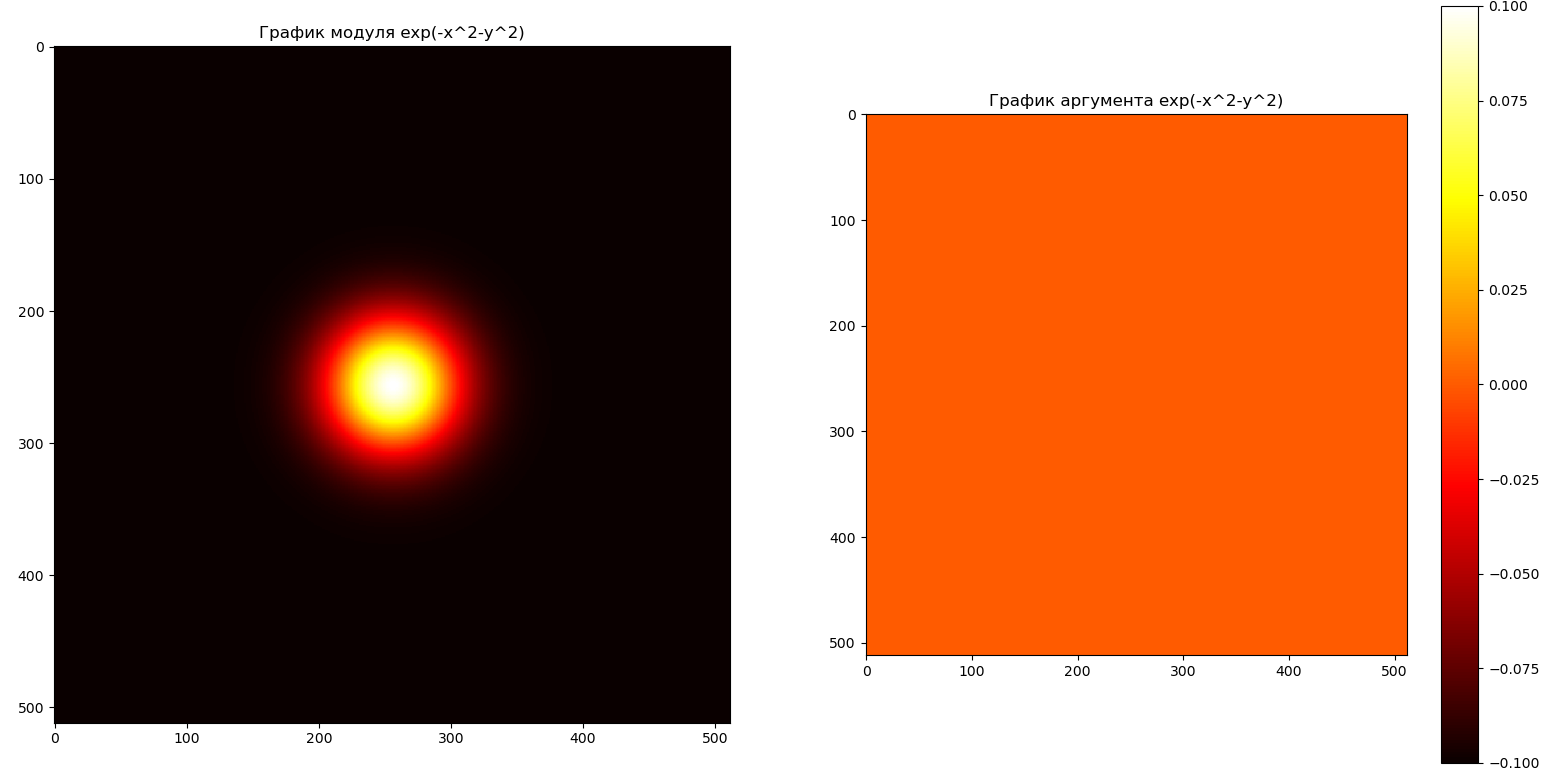


Рисунок 9 — Амплитуда и фаза гауссова пучка

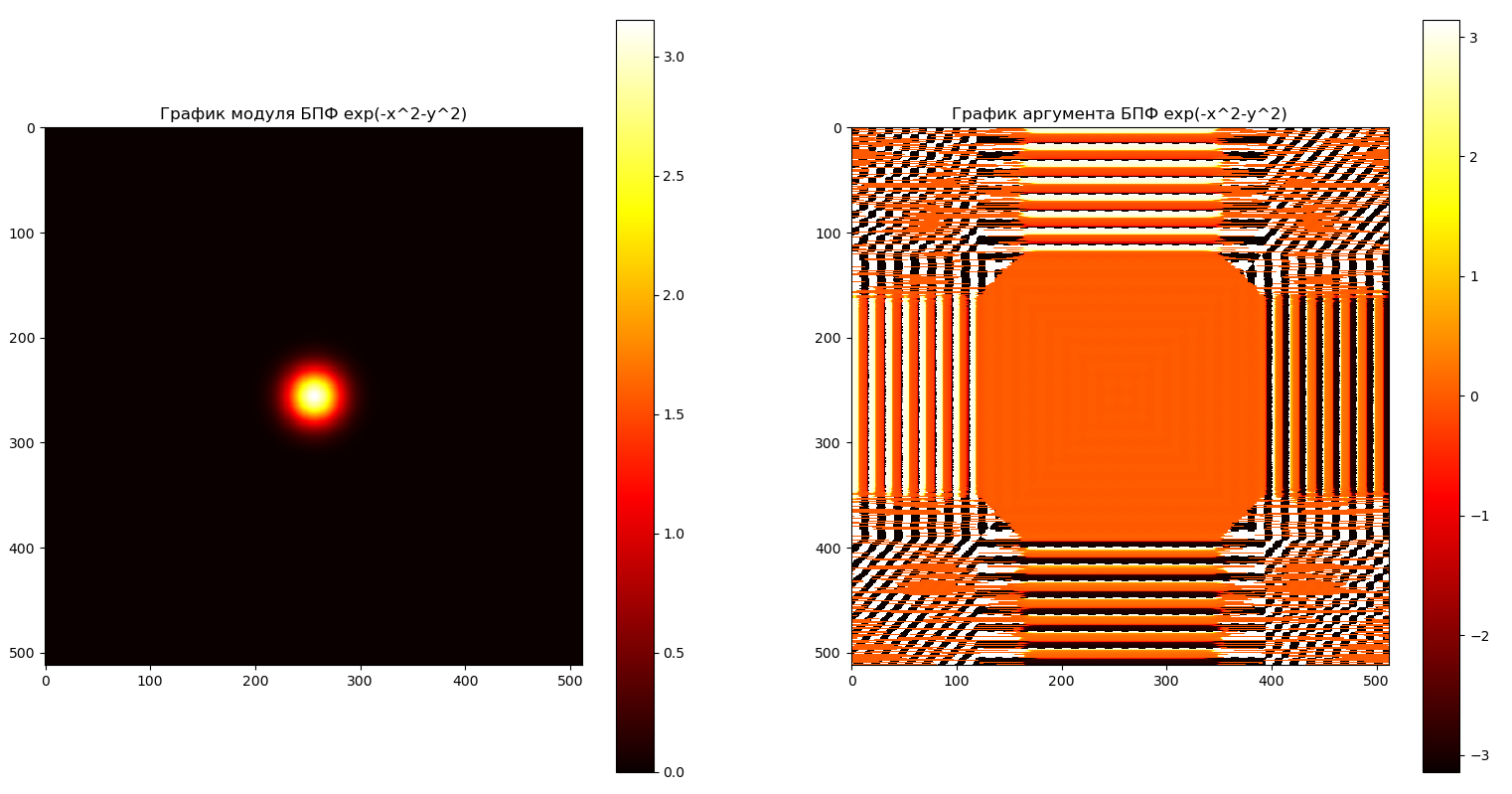


Рисунок 10 — Амплитуда и фаза БПФ гауссова пучка

На данных графиках подтверждается теория инвариантности моды Гаусса относительно преобразования Фурье, так как пучок сохранил форму, изменился только масштаб светового поля.

Теперь рассмотрим двумерный случай заданного, согласно варианту, светового поля. Аналитическое решение заданного двумерного поля изображено на рисунке 11. На рисунках 12-14 представлены амплитуды и фазы входного поля, БПФ и аналитического решения соответственно.

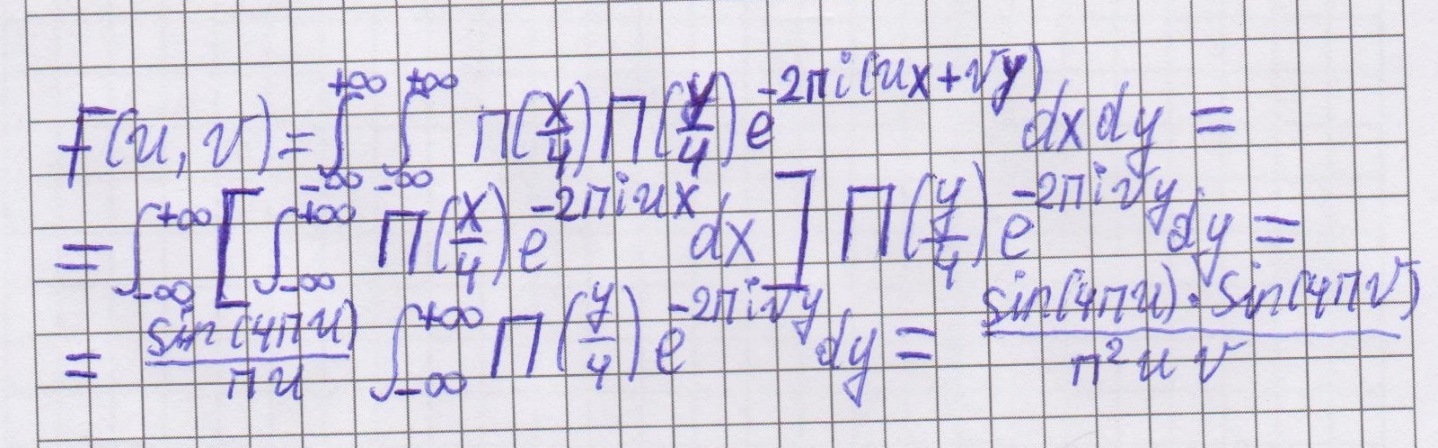


Рисунок 11 – Аналитическое решение двумерного входного поля

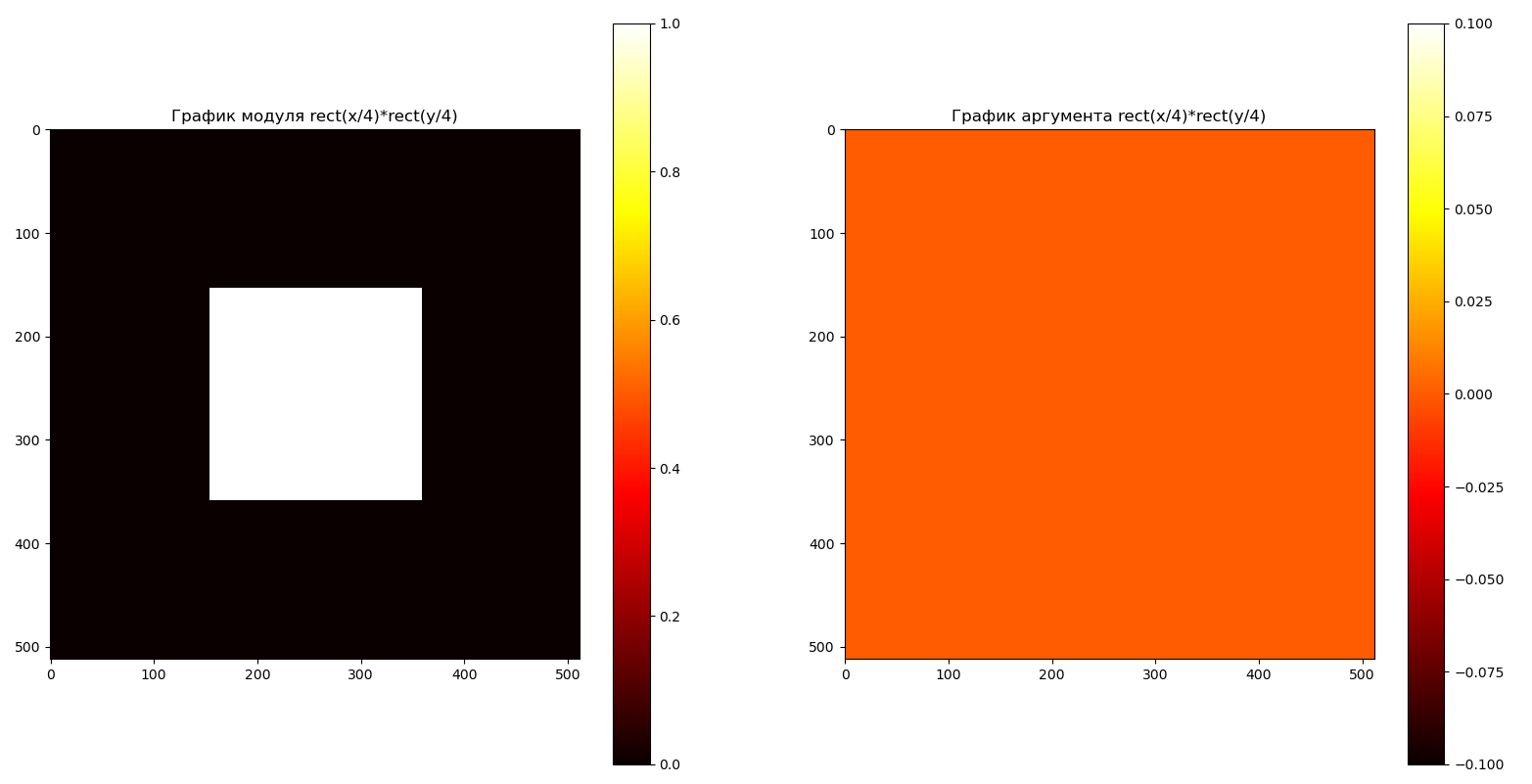


Рисунок 12 — Амплитуда и фаза входного поля

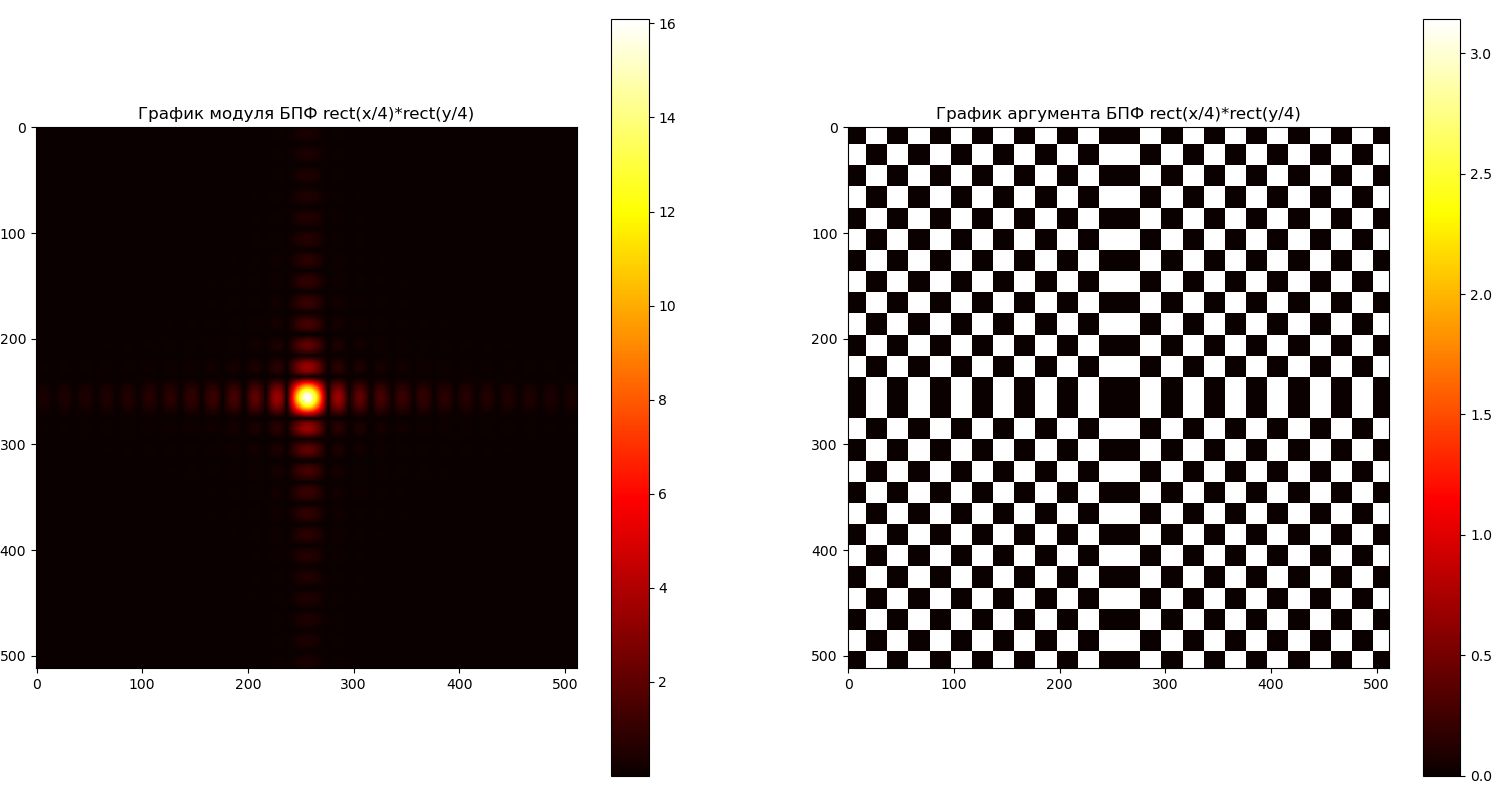


Рисунок 13 — Амплитуда и фаза выходного поля (БПФ)

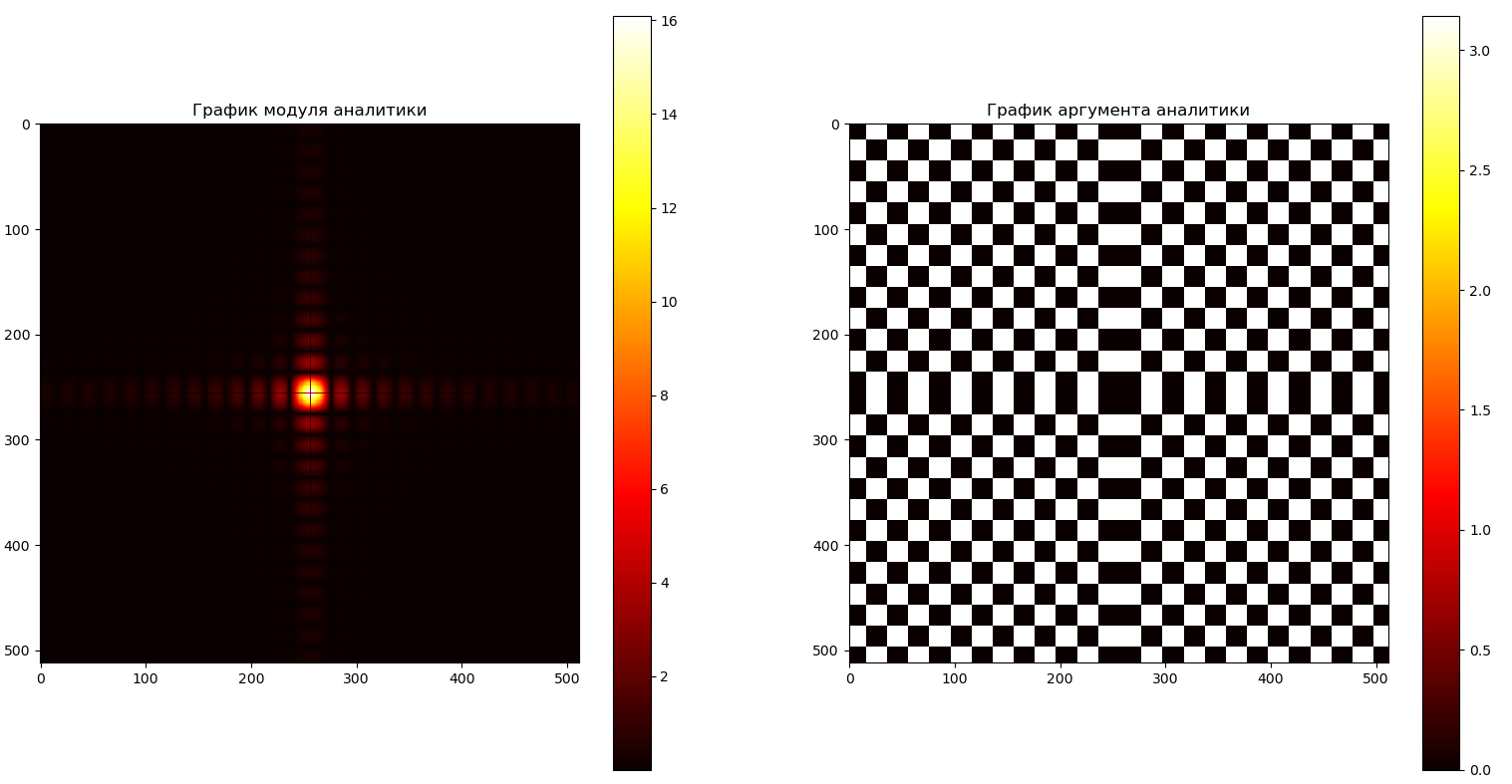


Рисунок 14 — Амплитуда и фаза аналитического двумерного решения

Можно сказать, что применение БПФ к двумерному световому полю позволяет получить результат, близкий к аналитическому двумерному решению.

**ЗАКЛЮЧЕНИЕ**

В ходе выполнения лабораторной работы были изучены теоретические сведения о применении быстрого финитного преобразования Фурье для одномерных и двумерных функций.

Были реализованы одномерные и двумерные преобразования Фурье с использованием численного метода интегрирования и метода БПФ. В качестве проверки полученных результатов программы помимо входного поля из варианта задания был подан гауссов пучок, подверженный инвариантности относительно преобразования Фурье. В результате, при применении БПФ и численного метода к заданному входному полю результаты показали очевидное сходство.

Также мною было проведено сравнение аналитического и БПФ решений. Сделан вывод, что результаты, полученные данными методами, совпадают, но в связи с делением на ноль аналитическое решение стремится к 1 в точке 0.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Фихтенгольц Г.М. Основы математического анализа, том I. – М.: Наука, 1968. – 440с.
2. Кострикин А.И., Манин Ю.И. Линейная алгебра и геометрия. – М.: Наука, 1980. – 309с.
3. Лаврентьев М.А., Шабат Б.В. Методы теории функций комплексного переменного. – М.: Наука, 1973. – 736 с.
4. Самарский А.А., Гулин А.В. Численные методы: Учебное пособие для вузов. – М.: Наука, 1989. – 432 с.
5. Люстерник Л.А., Соболев В.И. Краткий курс функционального анализа: Учебное пособие. – СПб.: Лань, 2009. 272 с.
6. Тихонов А.Н., Самарский А.А. Уравнения математической физики. – М.: Наука, 2004. – 798 с.
7. Лебедев Н.Н. Специальные функции и их приложения. – М.: Лань, 2010. 368 с.
8. Диткин В.А., Прудников А.П. Интегральные преобразования и операционное исчисление. – М.: Государственное издательство физико-математической литературы, 1961. – 524 с.

**ПРИЛОЖЕНИЕ А**

import numpy as np  
import matplotlib.pyplot as plt  
  
  
def gaussian\_beam(x):  
 return np.exp(-np.power(x, 2))  
  
  
def gaussian\_beam\_2d(x, y):  
 return np.exp(-x\*\*2 - y\*\*2)  
  
  
def rect(t):  
 if abs(t) > 0.5:  
 return 0  
 elif abs(t) < 0.5:  
 return 1  
 else:  
 return 0.5  
  
  
def rect\_2d(t1, t2):  
 return rect(t1) \* rect(t2)  
  
  
def fast\_fourier\_transform(y, b, a, N, M):  
 h = (b - a) / (N - 1)  
 # Добавление нулей  
 zeros = np.zeros(int((M - N) / 2))  
 y = np.concatenate((zeros, y, zeros), axis=None)  
 # Смена частей вектора  
 middle = int(len(y) / 2)  
 y = np.concatenate((y[middle:], y[:middle]))  
 # БПФ  
 Y = np.fft.fft(y, axis=-1) \* h  
 # Смена частей вектора  
 middle = int(len(Y) / 2)  
 Y = np.concatenate((Y[middle:], Y[:middle]))  
 # Выделение центральных N элементов  
 Y = Y[int((M - N) / 2): int((M - N) / 2 + N)]  
 # Пересчет области задания функции  
 interval = abs(N \*\* 2 / (4 \* a \* M))  
 return Y, interval  
  
  
def plots\_f(xs, f\_xs, title):  
 fig, axes = plt.subplots(1, 2, figsize=(12, 6))  
 axes[0].set\_title(f'График модуля {title}')  
 axes[0].plot(xs, np.absolute(f\_xs))  
 axes[0].grid()  
 axes[1].set\_title(f'График аргумента {title}')  
 axes[1].plot(xs, np.angle(f\_xs))  
 axes[1].grid()  
 plt.show()  
  
  
def compare\_plots(xs1, f\_xs1, label1, xs2, f\_xs2, label2):  
 fig, axes = plt.subplots(1, 2, figsize=(12, 6))  
 axes[0].set\_title('График модуля')  
 axes[0].plot(xs1, np.absolute(f\_xs1), label=label1)  
 axes[0].plot(xs2, np.absolute(f\_xs2), label=label2)  
 axes[0].grid()  
 axes[0].legend()  
 axes[1].set\_title('График аргумента')  
 axes[1].plot(xs1, np.angle(f\_xs1), label=label1)  
 axes[1].plot(xs2, np.angle(f\_xs2), label=label2)  
 axes[1].grid()  
 axes[1].legend()  
 plt.show()  
  
  
def fast\_fourier\_transform\_2d(Z, b, a, N, M):  
 for i in range(N):  
 Z[:, i], interval = fast\_fourier\_transform(Z[:, i], b, a, N, M)  
 for i in range(N):  
 Z[i, :], interval = fast\_fourier\_transform(Z[i, :], b, a, N, M)  
 return Z, interval  
  
  
def kernel\_of\_fourier(x):  
 return np.exp(-2 \* np.pi \* 1j \* x)  
  
  
def manual\_integrate(kernel, xs, f):  
 h = xs[1] - xs[0]  
 A = kernel(np.dot(xs.reshape(-1, 1), xs.reshape(-1, 1).T))  
 return np.dot(f, A) \* h  
  
  
def rect\_analitic\_fourier(x):  
 return 1 if x == 0 else np.sin(4 \* np.pi \* x) / (np.pi \* x)  
  
  
def rect\_analitic\_fourier\_2d(us, vs):  
 Z = np.zeros((N, N))  
 for i in range(N):  
 for j in range(N):  
 Z[i, j] = rect\_analitic\_fourier(us[i, j]) \* rect\_analitic\_fourier(vs[i, j])  
 return Z  
  
  
N = 512  
a = 5  
M = 4096  
  
# ------------------------------------ ОДНОМЕРНЫЙ СЛУЧАЙ ------------------------------------  
# -------- exp(-x\*\*2) --------  
xs = np.linspace(-a, a, N, endpoint=False)  
gb\_f = gaussian\_beam(xs)  
fft\_gb\_f, interval = fast\_fourier\_transform(gb\_f, a, -a, N, M)  
xs\_interval = np.linspace(-interval, interval, N, endpoint=False)  
plots\_f(xs, gb\_f, "exp(-x\*\*2)")  
plots\_f(xs\_interval, fft\_gb\_f, "БПФ для exp(-x\*\*2)")  
  
# -------- Сравнение БПФ и стандартного метода --------  
man\_gb\_f = manual\_integrate(kernel\_of\_fourier, xs, gb\_f)  
compare\_plots(xs\_interval, fft\_gb\_f, 'БПФ', xs, man\_gb\_f, 'Станд. метод')  
  
# -------- rect(x/4) --------  
rect\_f = []  
for x in xs:  
 rect\_f.append(rect(x/4))  
fft\_rect\_f, interval = fast\_fourier\_transform(rect\_f, a, -a, N, M)  
xs\_interval = np.linspace(-interval, interval, N, endpoint=False)  
plots\_f(xs, rect\_f, "rect(x/4)")  
plots\_f(xs\_interval, fft\_rect\_f, "БПФ для rect(x/4)")  
  
# -------- Сравнение БПФ и аналитического результата --------  
analit\_rect\_f = []  
for x in xs\_interval:  
 analit\_rect\_f.append(rect\_analitic\_fourier(x))  
compare\_plots(xs\_interval, fft\_rect\_f, 'БПФ', xs\_interval, analit\_rect\_f, 'Аналитическое решение')  
  
# ------------------------------------ ДВУМЕРНЫЙ СЛУЧАЙ ------------------------------------  
# -------- exp(-x\*\*2 - y\*\*2) --------  
X, Y = np.meshgrid(xs, xs)  
Z = gaussian\_beam\_2d(X, Y)  
fig, arr = plt.subplots(1, 2, figsize=(12, 6))  
amp = arr[0].imshow(np.absolute(Z), cmap='hot', interpolation='nearest')  
arr[0].set\_title('График модуля exp(-x^2-y^2)')  
phase = arr[1].imshow(np.angle(Z), cmap='hot', interpolation='nearest')  
arr[1].set\_title('График аргумента exp(-x^2-y^2)')  
fig.colorbar(phase, ax=arr[1])  
plt.show()  
  
# -------- БПФ exp(-x\*\*2 - y\*\*2) --------  
Z = gaussian\_beam\_2d(X, Y).astype(np.complex128)  
Z\_fin\_fft, interval = fast\_fourier\_transform\_2d(Z, -a, a, N, M)  
xs\_interval = np.linspace(-interval, interval, N, endpoint=False)  
X, Y = np.meshgrid(xs\_interval, xs\_interval)  
fig, arr = plt.subplots(1, 2, figsize=(12, 6))  
amp = arr[0].imshow(np.absolute(Z\_fin\_fft), cmap='hot', interpolation='nearest')  
arr[0].set\_title('График модуля БПФ exp(-x^2-y^2)')  
fig.colorbar(amp, ax=arr[0])  
phase = arr[1].imshow(np.angle(Z\_fin\_fft), cmap='hot', interpolation='nearest')  
arr[1].set\_title('График аргумента БПФ exp(-x^2-y^2)')  
fig.colorbar(phase, ax=arr[1])  
plt.show()  
  
# -------- rect(x/4)\*rect(y/4) --------  
X, Y = np.meshgrid(xs, xs)  
Z = np.zeros((N, N))  
for i in range(N):  
 for j in range(N):  
 Z[i, j] = rect\_2d(X[i, j] / 4, Y[i, j] / 4)  
fig, arr = plt.subplots(1, 2, figsize=(12, 6))  
amp = arr[0].imshow(np.absolute(Z), cmap='hot', interpolation='nearest')  
arr[0].set\_title('График модуля rect(x/4)\*rect(y/4)')  
fig.colorbar(amp, ax=arr[0])  
phase = arr[1].imshow(np.angle(Z), cmap='hot', interpolation='nearest')  
arr[1].set\_title('График аргумента rect(x/4)\*rect(y/4)')  
fig.colorbar(phase, ax=arr[1])  
plt.show()  
  
# -------- БПФ rect(x/4)\*rect(y/4) --------  
X, Y = np.meshgrid(xs, xs)  
Z = Z.astype(np.complex128)  
Z\_fin\_fft, interval = fast\_fourier\_transform\_2d(Z, -a, a, N, M)  
xs\_interval = np.linspace(-interval, interval, N, endpoint=False)  
X, Y = np.meshgrid(xs\_interval, xs\_interval)  
fig, arr = plt.subplots(1, 2, figsize=(12, 6))  
amp = arr[0].imshow(np.absolute(Z\_fin\_fft), cmap='hot', interpolation='nearest')  
arr[0].set\_title('График модуля БПФ rect(x/4)\*rect(y/4)')  
fig.colorbar(amp, ax=arr[0])  
phase = arr[1].imshow(np.angle(Z\_fin\_fft), cmap='hot', interpolation='nearest')  
arr[1].set\_title('График аргумента БПФ rect(x/4)\*rect(y/4)')  
fig.colorbar(phase, ax=arr[1])  
plt.show()  
  
# -------- Аналитическое решение --------  
Z = rect\_analitic\_fourier\_2d(X, Y)  
fig, arr = plt.subplots(1, 2, figsize=(12, 6))  
arr[0].imshow(np.absolute(Z), cmap='hot', interpolation='nearest')  
arr[0].set\_title('График модуля аналитики')  
fig.colorbar(amp, ax=arr[0])  
phase = arr[1].imshow(np.angle(Z), cmap='hot', interpolation='nearest')  
arr[1].set\_title('График аргумента аналитики')  
fig.colorbar(phase, ax=arr[1])  
plt.show()