

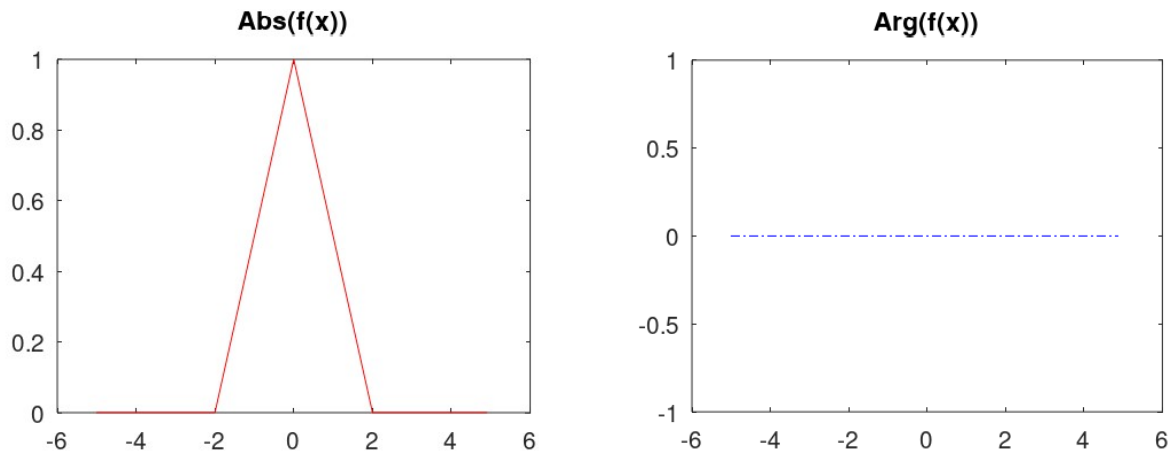
Отчет по лабораторной работе №2 по оптической информатике

Выполнил: Чичикин Артем

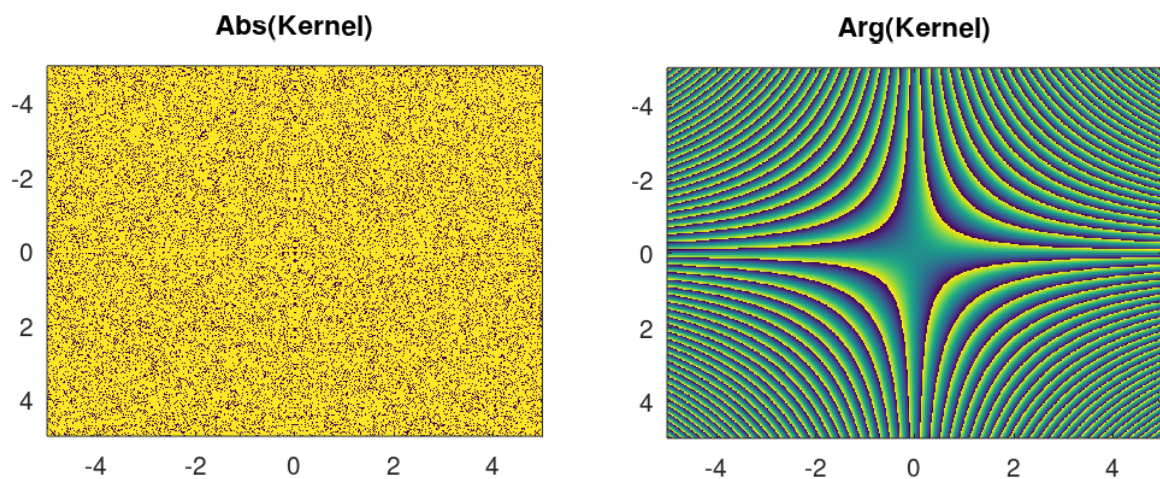
Проверил: Кириленко Михаил Сергеевич

Исходный код: <https://github.com/ArtyomStebenev/OI>

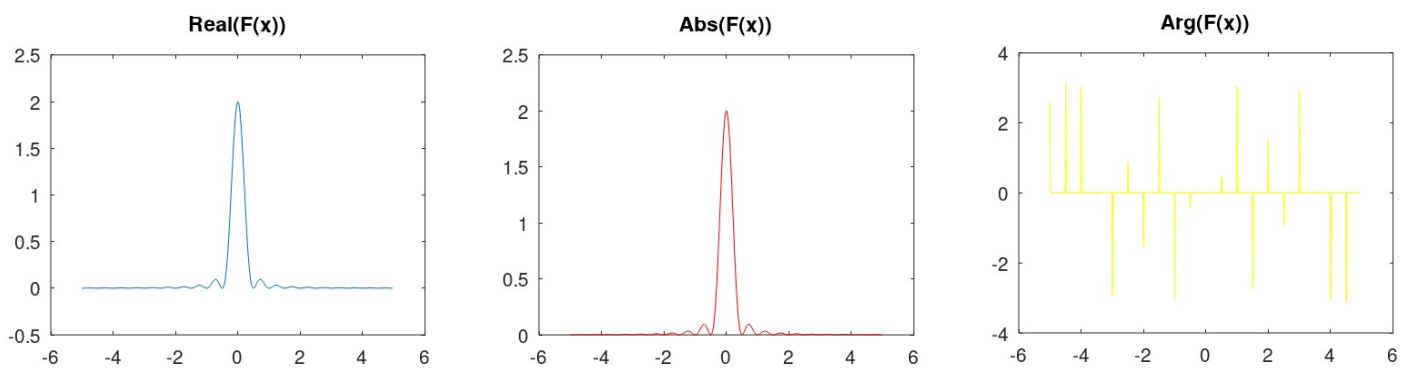
Первоначально создаю файл **main.m**, вписываю в него основные константы и формулы относительно варианта, которые будут использоваться в других файлах, и инициализирую переменные. Интеграл считаю с помощью функции **sum**, который оказался равен двум.



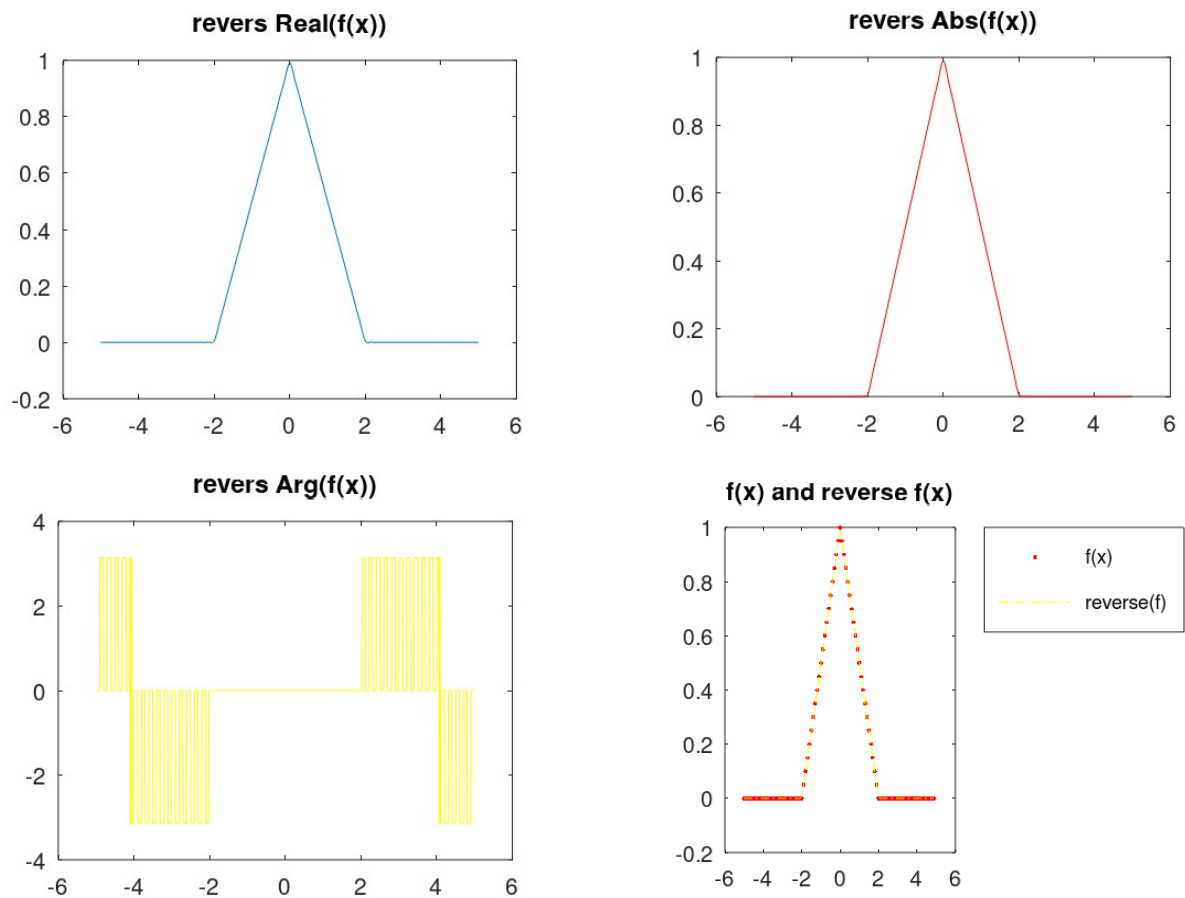
Для нахождения ядра создаю две матрицы X и Z с помощью функции **meshgrid**.



Нахожу преобразование Фурье матричным способом.

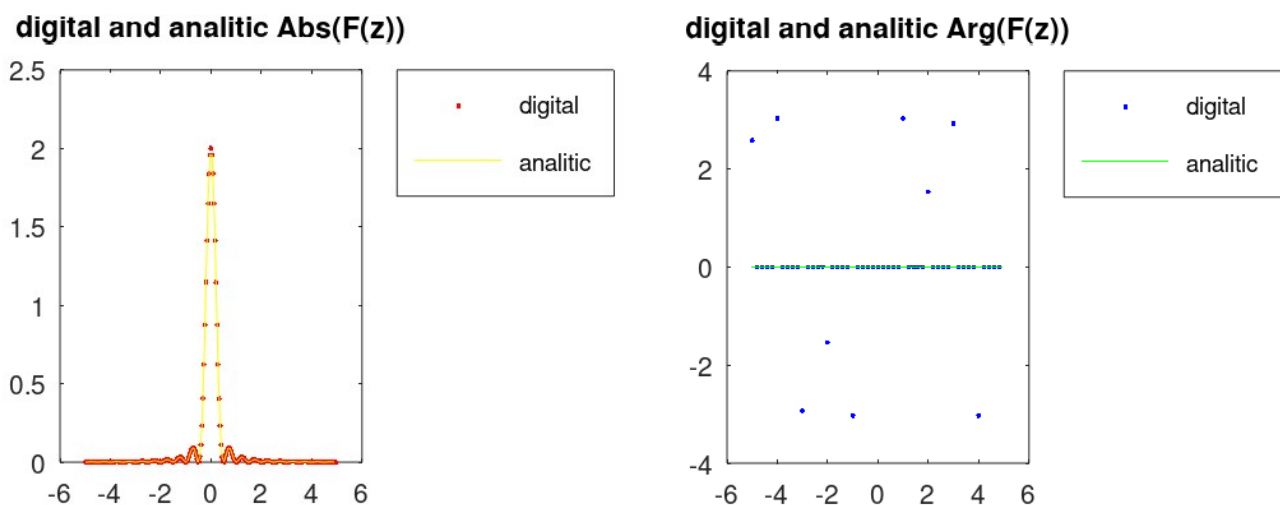


С помощью обратного преобразования Фурье восстанавливаю функцию. Результат получился достаточно точным, однако аргумент имеет «дефекты».



Далее считаю преобразование Фурье аналитически и вывожу его на экран. Исходя из графика можно сказать, что численно и аналитически подсчитанные интегралы похожи друг на друга с хорошей точностью, однако численно подсчитанный аргумент имеет «дефектные» точки.

$$\mathcal{F} [tri(x/2)] = \mathcal{F} [rect(x/2) * rect(x/2)] = \mathcal{F} [rect(x/2)] * \mathcal{F} [rect(x/2)] = 2 sinc^2(2z)$$



ЛИСТИНГ 1

```
n = 1000;
left_border = -5;
right_border = 5;
step = (right_border - left_border)/n;

x = left_border:step:(right_border - step/2);

f = zeros(size(x));
f(x > -2 & x < 2) = 1-abs(x(x > -2 & x < 2) / 2);

plot(x(1:10:end), abs(f)(1:10:end), "-r");
title("Abs(f(x))");
figure;
plot(x(1:10:end), arg(f)(1:10:end), "-.b");
title("Arg(f(x))");

integral = sum(step * f(1:n))
```

ЛИСТИНГ 2

```
#8
m = 500;
p = -5;
q = 5;
height = (q - p)/m;

z = p:height:(q - height);

[X, Z] = meshgrid(x, z);
Kernel = exp(-i * 2 * pi * X .* Z);

imagesc(x, z, abs(Kernel));
title("Abs(Kernel)");
figure;
imagesc(x, z, arg(Kernel));
title("Arg(Kernel)");
```

ЛИСТИНГ 3

```
#10
F = Kernel * f.' * step;

plot(z, real(F));
title("Real(F(x))");

figure;
plot(z, abs(F), "-r");
title("Abs(F(x))");

figure;
plot(z, arg(F), "-y");
title("Arg(F(x))");
```

ЛИСТИНГ 4

```
#13
R_Kernel = Kernel';

Revers_f = R_Kernel * F * height;

plot(x, real(Revers_f));
title("revers Real(f(x))");

figure;
plot(x, abs(Revers_f), "-r");
title("revers Abs(f(x))");

figure;
plot(x, arg(Revers_f), "-y");
title("revers Arg(f(x))");
```

ЛИСТИНГ 5

```
#14

plot(x(1:10:end), (f)(1:10:end), ".r",
      x(1:10:end), (Revers_F)(1:10:end), "-.y");
title("f(x) and reverse f(x)");
legend('f(x)', 'reverse(f)', 'Location', 'bestoutside')
```

ЛИСТИНГ 6

```
analitic_F = 2 * sinc(2*z).^2;

plot(z(1:4:end), abs(F)(1:4:end), ".r; abs(F(z));",
      z(1:4:end), abs(analitic_F(1:4:end)), "-y; analitic abs(F(z));");

figure;
plot(z(1:2:end), arg(F)(1:2:end), ".b; arg(F(z));",
      z(1:2:end), arg(analitic_F(1:2:end)), "-g; analitic arg(F(z));");
```