

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего
образования

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИТМО

Факультет систем управления и робототехники

Лабораторная работа № 5
"Связь непрерывного и дискретного "
по дисциплине Частотные методы

Выполнила: студентка гр. **R3238**

Нечаева А. А.

Преподаватели: *Перегудин Алексей Алексеевич,*
Пашенко Артём Витальевич

Санкт-Петербург, 2024

1 Задание. Непрерывное и дискретное преобразование Фурье.

Рассмотрим прямоугольную функцию $\Pi : \mathbb{R} \rightarrow \mathbb{R}$:

$$\Pi(t) = \begin{cases} 1, & |t| \leq 1/2, \\ 0, & |t| > 1/2 \end{cases} \quad (1)$$

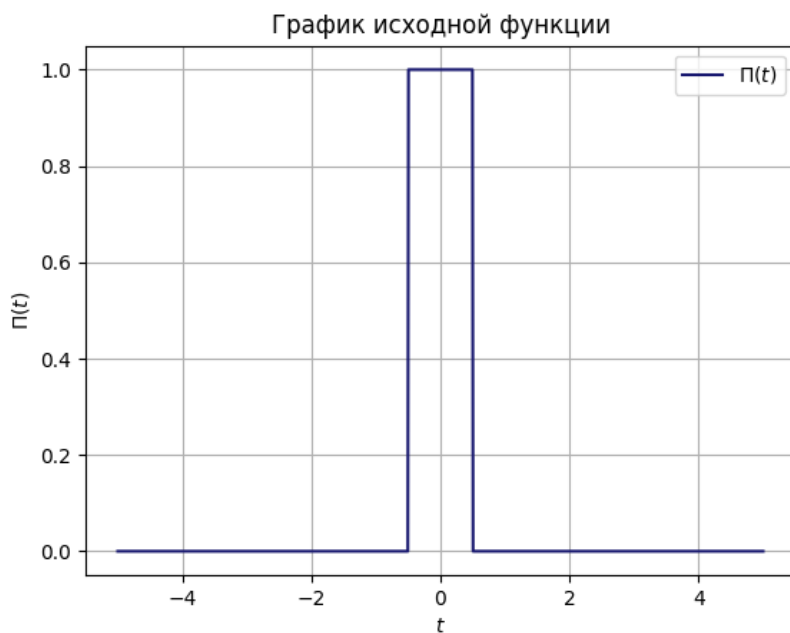


Рис. 1. График исходной функции $\Pi(t)$.

1.1 Истинный Фурье-образ.

Найдем аналитическое выражение для Фурье-образа

$$\begin{aligned}\hat{\Pi}(\nu) &= \int_{-\infty}^{+\infty} \Pi(t) e^{-2\pi i \nu t} dt = \int_{-0.5}^{0.5} e^{-2\pi i \nu t} dt = -\frac{e^{-2\pi i \nu t}}{2\pi i \nu} \Big|_{-0.5}^{0.5} = \\ &= -\frac{e^{-\pi i \nu} - e^{\pi i \nu}}{2\pi i \nu} = \frac{e^{\pi i \nu} - e^{-\pi i \nu}}{2\pi i \nu} = \frac{\sin(\pi \nu)}{\pi \nu} = \text{sinc}(\nu) \quad (2)\end{aligned}$$

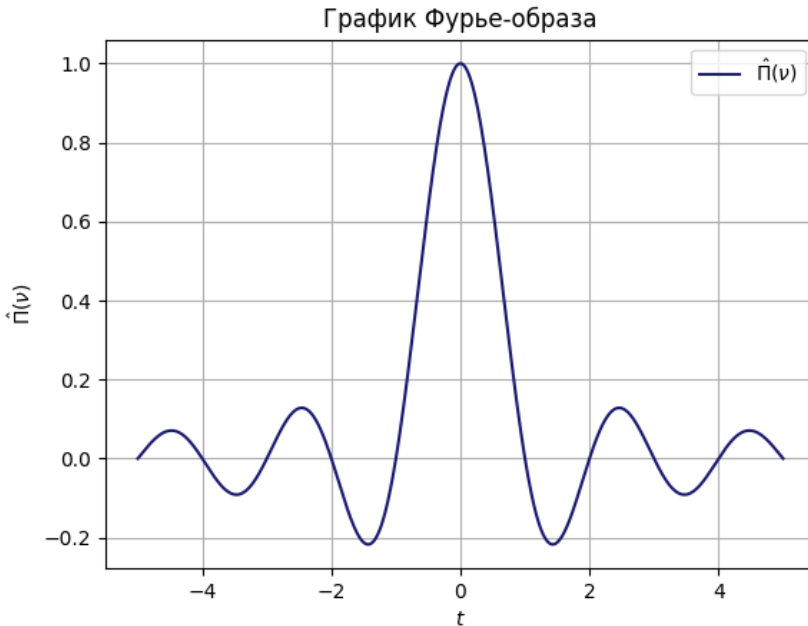
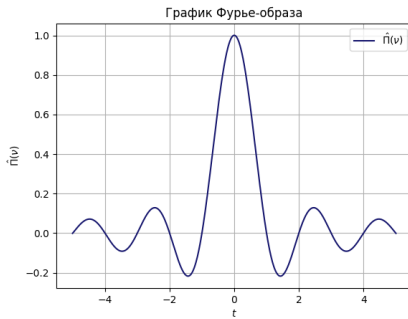


Рис. 2. График Фурье-образа функции $\Pi(t)$.

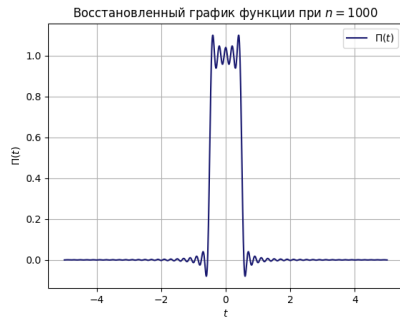
1.2 Численное интегрирование

Теперь найдем Фурье-образ $\Pi(t)$ с помощью численного интегрирования (функции `trapz` библиотеки *NumPy* языка *Python*), а затем с помощью численного интегрирования восстановим исходную функцию $\Pi(t)$.

Число шагов интегрирования будем задавать переменной n .



а)



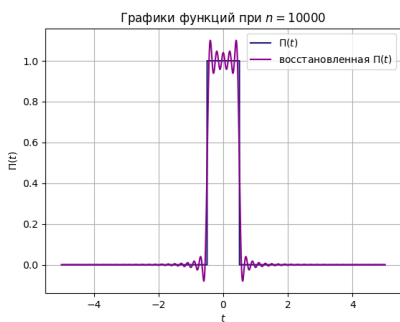
б)

Рис. 3. а) График Фурье-образа функции $\Pi(t)$, полученный с помощью численного интегрирования, б) график восстановленной функции.

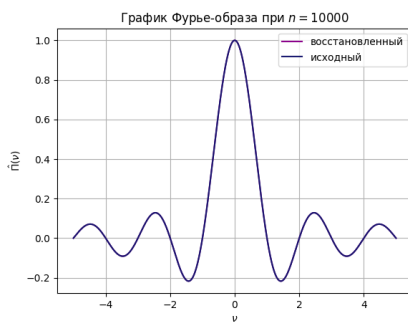
Заметим, что при $n = 1000$ график Фурье-образа (рисунок 3а) совпадает с графиком Фурье-образа, построенного с помощью аналитической формулы (рисунок 2). В то же время, график восстановленной функции (рисунок 3б) имеет заметные отличия от исходного графика функции (рисунок 1). Далее построим сравнительные графики Фурье-образов, исходной функции и восстановленной при разных значениях n , также будем фиксировать время работы программы. Промежуток интегрирования обозначим $[-d, d]$. Основные характеристики графиков представлены в таблице 1.

Таблица 1. Параметры графиков восстановленной функции на промежутке интегрирования $[-5, 5]$.

№ рисунка	объект	n	d	t, ms
4 а	функция	10000	5	10369
4 б	образ	10000	5	5341
5 а	функция	200	5	196
5 б	образ	200	5	202
5 в	функция	100	5	198
5 г	образ	100	5	276
6 а	функция	90	5	270
6 б	образ	90	5	236



а)

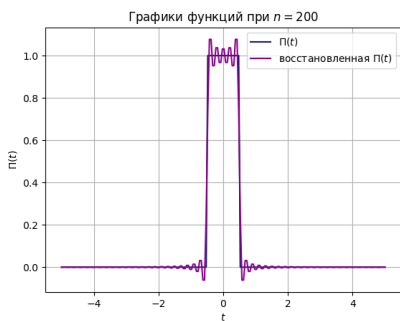


б)

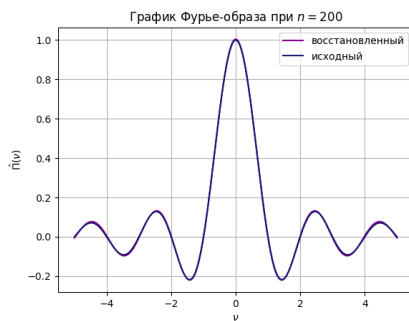
Рис. 4. Графики восстановленной и исходной функции и их Фурье-образов при $n = 10000$.

Сравнивая графики на рисунках 4 а, 5 а, 5 в, можно сделать вывод о том, что при уменьшении числа итераций в вычислении интегралов для нахождения Фурье-образа и последующего восстановления функции, сходство графиков исходной и восстановленной функций возрастает.

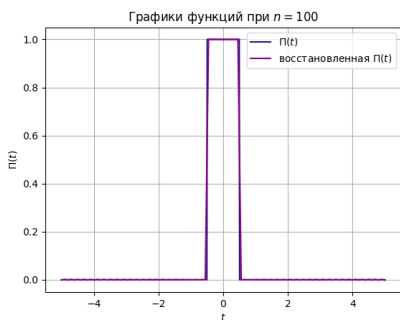
Однако при дальнейшем уменьшении n ($n < 100$) различия между графиками исходной и восстановленной функций возрастают (рисунки 5в и 5г). Время выполнения также возрастает (таблица 1).



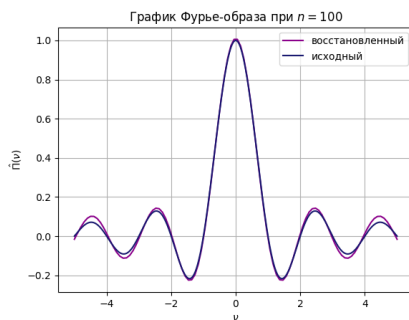
а)



б)



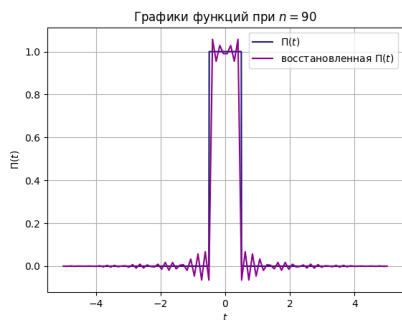
в)



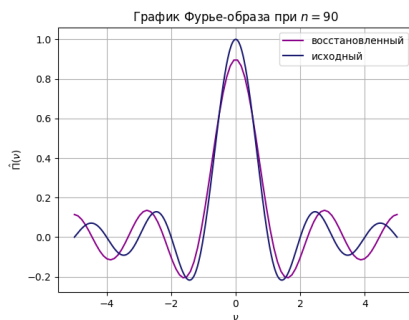
г)

Рис. 5. Графики восстановленной и исходной функции и их Фурье-образов при $n = 200, 100$.

Наиболее оптимальным по результату и времени выполнения в нашем случае является $n = 100$.



а)

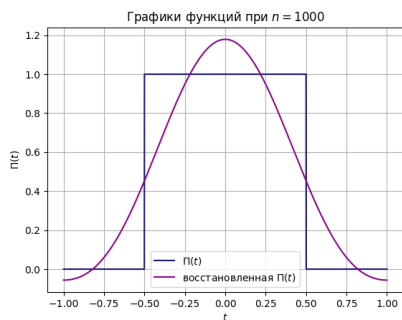


б)

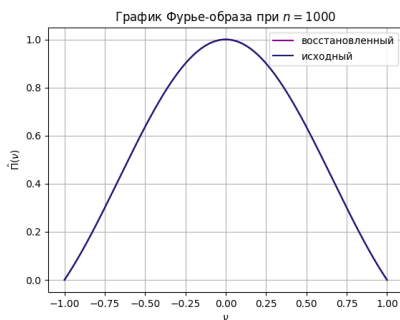
Рис. 6. Графики восстановленной и исходной функции и их Фурье-образов при $n = 90$.

Таблица 2. Параметры графиков восстановленной функции на промежутке интегрирования $[-1, 1]$.

№ рисунка	объект	n	d	t, ms
7 а	функция	1000	1	486
7 б	образ	1000	1	287
8 а	функция	30	1	241
8 б	образ	30	1	211
8 в	функция	10	1	191
8 г	образ	10	1	274
9 а	функция	5	1	161
9 б	образ	5	1	188



а)



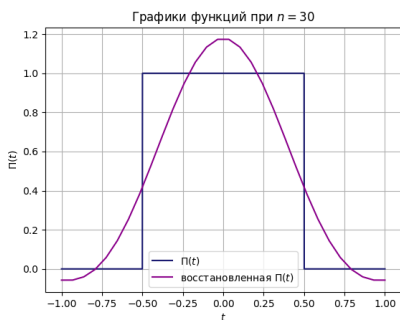
б)

Рис. 7. Графики восстановленной и исходной функции и их Фурье-образов при $n = 1000$.

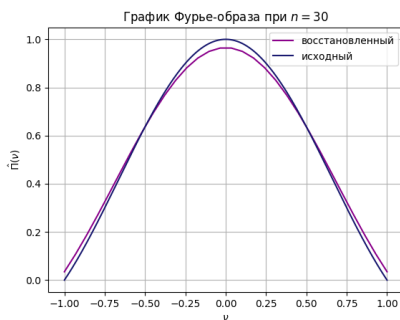
Заметим, что при $n = 1000$ и при $n = 30$ существенной разницы между восстановленными функциями нет (рисунок 7 а и 8 а), кроме того, что в восстановленной функции на рисунке 8 а отчетливее видны ломанная, из которой она состоит. При дальнейшем уменьшении n точность восстановленной функции снижается. Время работы программы также снижается и достигает минимума при $n = 5$, далее опять увеличивается.

Наиболее оптимальными с точки зрения точности приближения исходного графика функции и времени выполнения в данном случае являются $n = 5$ и $n = 6$.

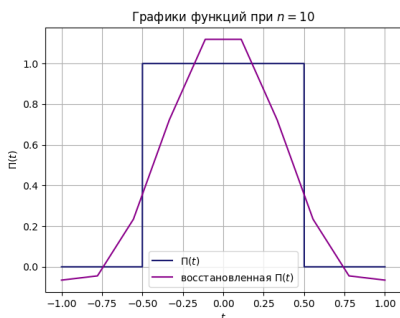
Теперь рассмотрим интегрирование на промежутке $[-10, 10]$, то есть при $d = 10$.



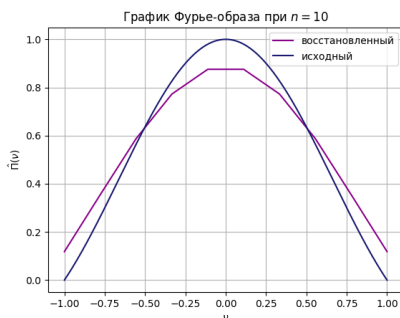
а)



б)



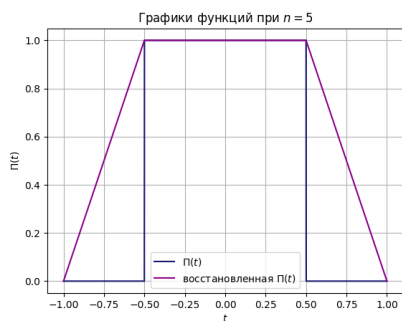
в)



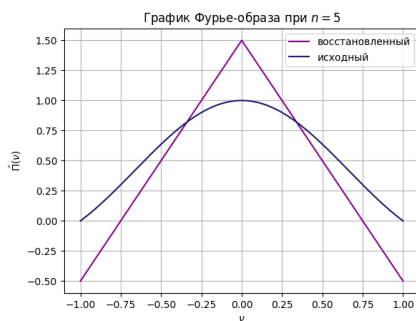
г)

Рис. 8. Графики восстановленной и исходной функции и их Фурье-образов при $n = 30, 10$.

Заметим, что при $n = 10000$ и $n = 1000$ для случая $d = 10$ результат на графике практически идентичен, но время, затраченное на вычисление последнего примерно в 30 раз меньше. С точки зрения точности приближения в обоих случаях график исходной функции узнаваем, наибольшие различия замечены в точках скачков функции.



а)



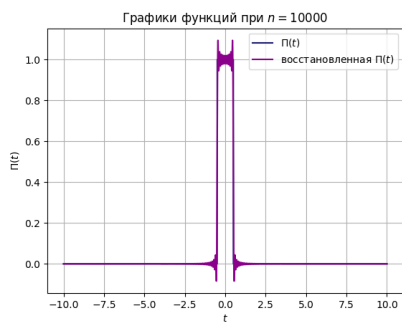
б)

Рис. 9. Графики восстановленной и исходной функции и их Фурье-образов при $n = 5$.

Таблица 3. Параметры графиков восстановленной функции на промежутке интегрирования $[-10, 10]$.

№ рисунка	объект	n	d	t, mc
10 а	функция	10000	10	10766
10 б	образ	10000	10	5302
10 в	функция	1000	10	390
10 г	образ	1000	10	281
11 а	функция	500	10	271
11 б	образ	500	10	192
11 в	функция	250	10	223
11 г	образ	250	10	188

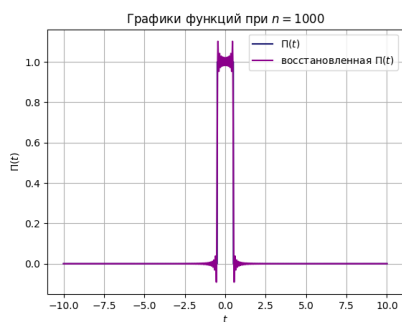
При $n = 500$ получаем еще удовлетворительный результат: исходная функция угадывается по графику восстановленной, в ходе дальнейшего уменьшения $n < 500$ график восстановленной функции все меньше похож на исходную.



а)



б)



в)



г)

Рис. 10. Графики восстановленной и исходной функции и их Фурье-образов при $n = 10000, 1000$.

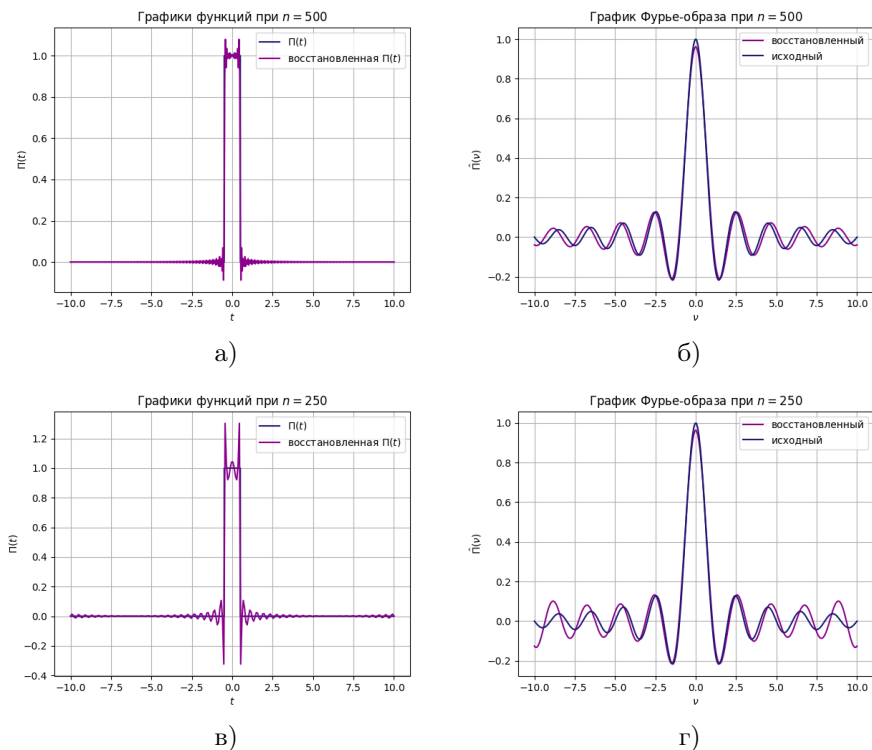


Рис. 11. Графики восстановленной и исходной функции и их Фурье-образов при $n = 500, 250$.

Вывод. Для получения оптимального результата требуется достаточно большое количество времени: 200 – 300 мс. Точность позволяет узнать в восстановленной функции исходную, пусть и с небольшими помехами в областях скачков функции.

1.3 Использование DFT

Найдем Фурье-образ функции $\Pi(t)$ с помощью дискретного преобразования Фурье (конструкция `fftshift(fft())`), используя его так, чтобы преобразование было *унитарным*. Затем выполним обратное преобразование от найденного Фурье-образа с помощью обратного дискретного преобразования. Сначала запишем, как работают преобразования:

`numpy.fft.fft :`

$$y(j) = \sum_{k=0}^{n-1} x(k) \cdot \exp \left\{ -\frac{2\pi i j k}{n} \right\} \quad (3)$$

`numpy.fft.ifft :`

$$x(k) = \frac{1}{n} \sum_{j=0}^{n-1} y(j) \cdot \exp \left\{ \frac{2\pi i j k}{n} \right\} \quad (4)$$

Для того, чтобы преобразования стали *унитарными*:

$$F(j) = \frac{y(j)}{\sqrt{n}}, \quad f(k) = x(k)\sqrt{n}.$$

Таблица 4. Параметры графиков восстановленной с помощью DFT функции и ее Фурье-образа на промежутке интегрирования $[-5, 5]$.

№ рисунка	объект	n	d	t, ms
12 а	функция	10000	5	174
12 б	образ	10000	5	174

Заметим, что относительно аналогичного случая применения `trapz`, то есть при $n = 10000$ и $d = 5$, скорость выполнения с помощью DFT возрасла примерно в 100 раз, для сравнения для получения восстановленной функции с помощью численного интегрирования понадобилось **10369** мс, а для DFT **174** мс. График восстановленной функции визуально совпадает с графиком исходной, но Фурье-образ, полученный с помощью DFT сильно отличается от графика истинного Фурье-образа, хотя при использовании численного интегрирования (рисунок 4б) Фурье-образы восстановленной и исходной функций совпадают.

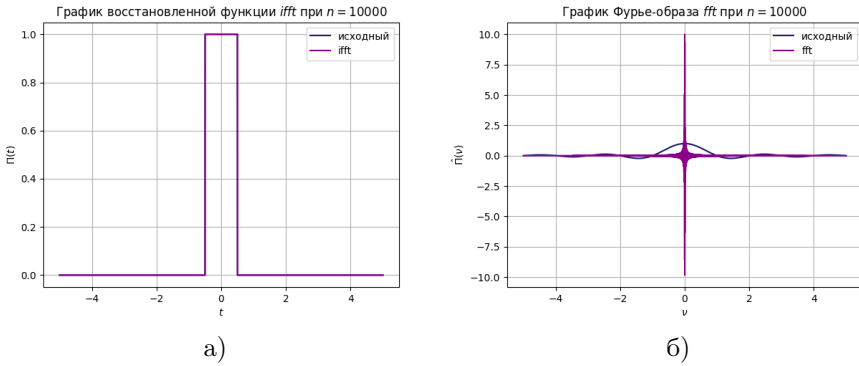


Рис. 12. Графики восстановленной и исходной функции и их Фурье-образов при $n = 10000$.

1.4 Объяснения

1.4.1 Быстрота выполнения с помощью DFT.

Скорость DFT объясняется алгоритмами, которые работают "под капотом" функций из библиотеки *NumPy*. Алгоритмы используют свойство симметрии, например, рассмотрим вычисление следующего компонента Фурье-образа:

$$\begin{aligned}
 y(j+n) &= \sum_{k=0}^{n-1} x(k) \cdot \exp \left\{ -\frac{2\pi i(n+j)k}{n} \right\} = \sum_{k=0}^{n-1} x(k) \cdot \exp \{ -2\pi i k \} \cdot \exp \left\{ -\frac{2\pi i j k}{n} \right\} = \\
 &= \sum_{k=0}^{n-1} x(k) \cdot \exp \left\{ -\frac{2\pi i j k}{n} \right\} = y(j) \quad (5)
 \end{aligned}$$

Заметим свойство симметрии: $y(j + z \cdot n) = y(j)$, где $z \in \mathbb{Z}$.

Последовательно применяя этот инструмент, рекурсивно постоянно разделяя массив, который необходимо вычислить, добиваемся того, что асимптотика алгоритма становится $O(n \log n)$ вместо $O(n^2)$, которая получается при преобразовании Фурье без использования оптимизации.

1.4.2 Отличие образа после DFT.

1.5 Приближение непрерывного с помощью DFT

...В работе...

2 Задание. Сэмплирование.

2.1 Сэмплирование синусов

Зададимся параметрами $a_1 = 1$, $a_2 = 2$, $\omega_1 = 3$, $\omega_2 = 4$, $\phi_1 = \pi$, $\phi_2 = 2\pi$ и рассмотрим функцию $y(t) = a_1 \sin(\omega_1 t + \phi_1) + a_2 \sin(\omega_2 t + \phi_2)$

$$y(t) = \sin(3t + \pi) + 2 \sin(4t + 2\pi) \quad (6)$$

2.2 Сэмплирование *sinus cardinalis*

...В работе...