Федеральное государственное автономное образовательное учреждение высшего образования «Национальный Исследовательский Университет ИТМО»

ЛАБОРАТОРНАЯ РАБОТА №5
ПРЕДМЕТ «ЧАСТОТНЫЕ МЕТОДЫ»
ТЕМА «СВЯЗЬ НЕПРЕРЫВНОГО И ДИСКРЕТНОГО»

Лектор: Перегудин А. А.
Практик: Пашенко А. В.
Студент: Румянцев А. А.
Поток: ЧАСТ.МЕТ. 1.3

Факультет: СУиР
Группа: R3241

Санкт-Петербург
2024

Содержание

1 Задание 1. Непрерывное и дискретное преобразование Фурье

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

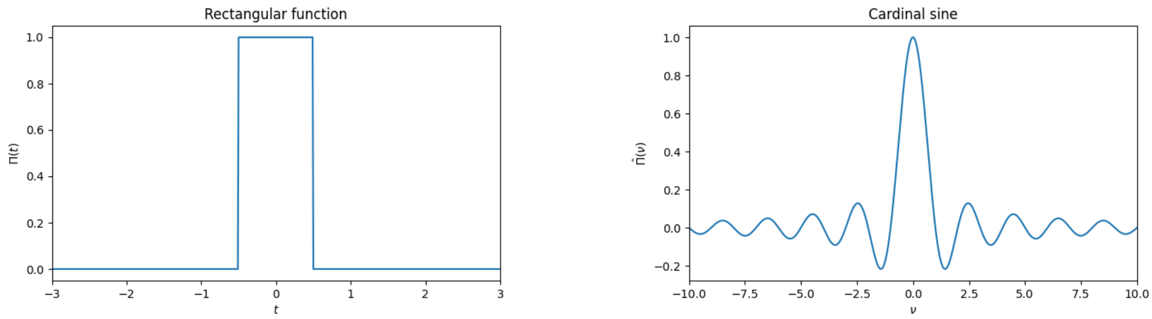
$$\Pi(t) = \begin{cases} 1, & |t| \le {}^1/_2, \\ 0, & |t| > {}^1/_2. \end{cases}$$

1.1 Истинный Фурье-образ

Найдем аналитическое выражение для Фурье-образа прямоугольной функции

$$\hat{\Pi}(\nu) = \int\limits_{-\infty}^{+\infty} \Pi(t) e^{-2\pi i \nu t}\, dt = \int\limits_{-\frac{1}{2}}^{\frac{1}{2}} e^{-2\pi i \nu t}\, dt = -\frac{e^{-\pi i \nu} - e^{\pi i \nu}}{2\pi i \nu} = \frac{\sin(\pi\nu)}{\pi\nu} = \operatorname{sinc}(\nu)$$

Построим графики $\Pi(t)$ и $\hat{\Pi}(\nu)$



(a) Прямоугольная функция (b) Кардинальный синус

Рис. 1: Исходный сигнал и его Фурье-образ

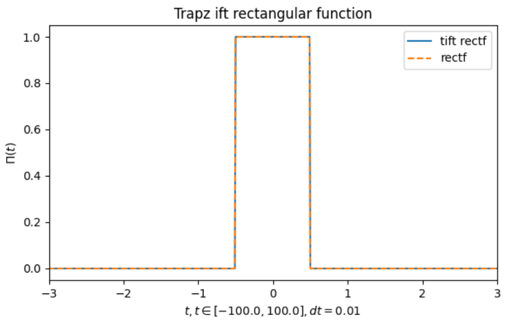
1.2 Численное интегрирование

Зададим функцию $\Pi(t)$ в `Python`. Найдем ее Фурье-образ с помощью численного интегрирования (функция `trapz`). Вновь используя численное интегрирование, выполним обратное преобразование Фурье от найденного Фурье-образа с целью восстановить исходную функцию. Схематично наши действия будут выглядеть так:

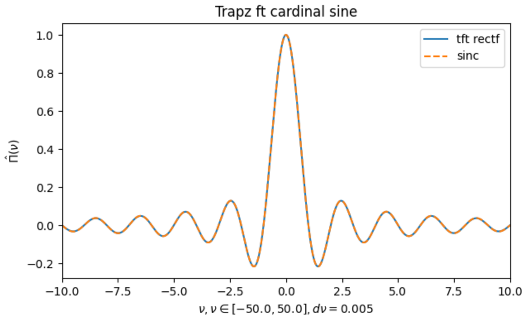
$$\Pi(t) \xrightarrow{\texttt{trapz}} \hat{\Pi}(\nu) \xrightarrow{\texttt{trapz}} \Pi(t)$$

Построим график найденной функции $\hat{\Pi}(\nu)$ и *восстановленной* функции $\Pi(t)$. Сравним результат с истинной функцией и Фурье-образом. Исследуем влияние величины шага интегрирования и размера промежутка, по которому вычисляется интеграл, на результат. Сделаем выводы о точности и быстродействии метода.

Далее приведены соответствующие графики. Оранжевым цветом выделены оригинальные функции, синим – найденные через преобразования. Каждый график подписан сверху. Под временной шкалой также указаны рассматриваемый промежуток времени или частот и шаг дискретизации во временной или частотной областях.

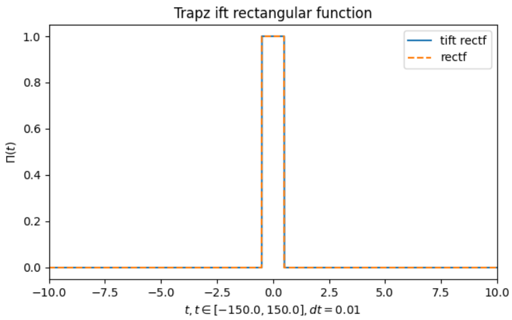


(a) $\Pi(t)$, восстановленная `trapz`

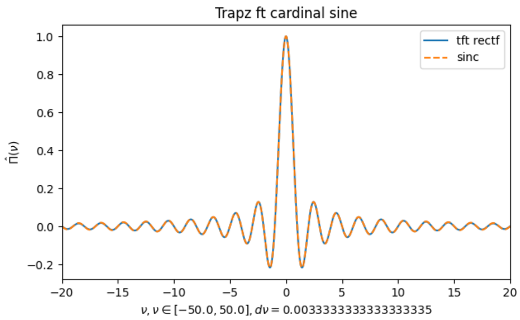


(b) $\hat{\Pi}(t)$ восстановленной `trapz` $\Pi(t)$

Рис. 2: Интеграл по всей области определения функции от -100 до 100

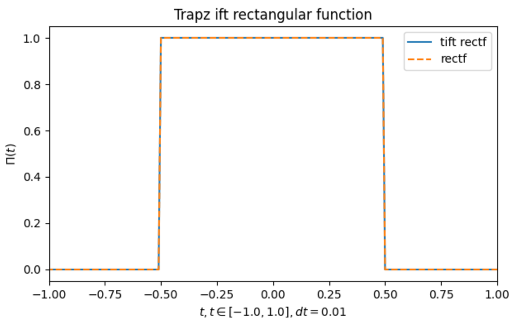


(a) $\Pi(t)$, восстановленная `trapz`

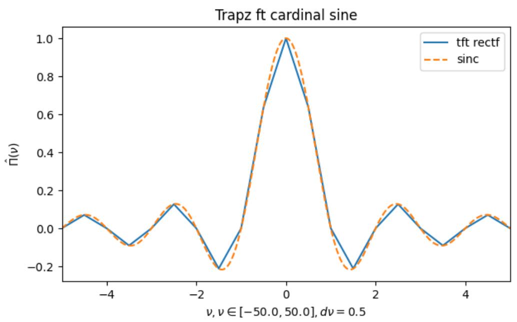


(b) $\hat{\Pi}(t)$ восстановленной `trapz` $\Pi(t)$

Рис. 3: Интеграл на увеличенном промежутке от -150 до 150

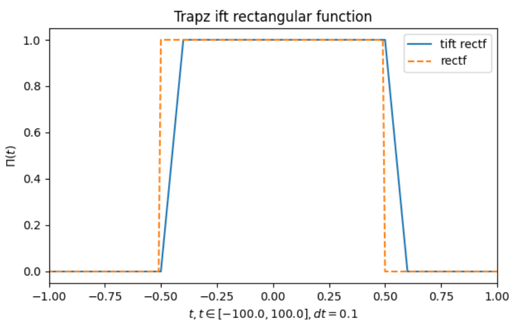


(a) $\Pi(t)$, восстановленная `trapz`

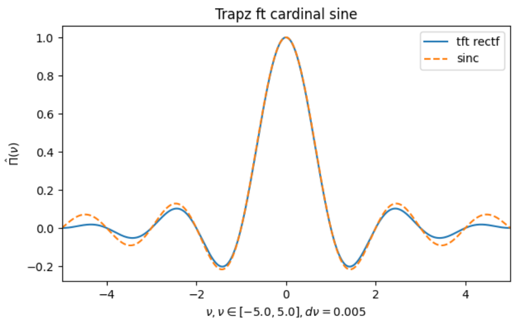


(b) $\hat{\Pi}(t)$ восстановленной `trapz` $\Pi(t)$

Рис. 4: Интеграл на уменьшенном промежутке от -1 до 1

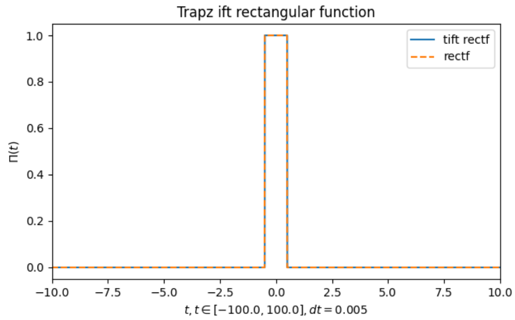


(a) $\Pi(t)$, восстановленная `trapz`

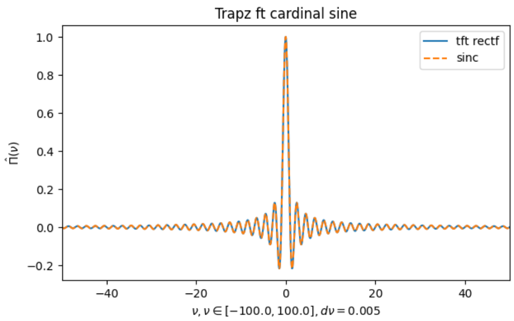


(b) $\hat{\Pi}(t)$ восстановленной `trapz` $\Pi(t)$

Рис. 5: Увеличение шага интегрирования $dt = 0.1$, интеграл аналогично рис. 2



(a) $\Pi(t)$, восстановленная `trapz`



(b) $\hat{\Pi}(t)$ восстановленной `trapz` $\Pi(t)$

Рис. 6: Уменьшение шага интегрирования $dt = 0.005$, интеграл аналогично рис. 2

Исходя из графиков можно сделать вывод, что `trapz` имеет высокую точность аппроксимации исходного сигнала и его Фурье-образа, однако ее мы достигаем при маленьком шаге dt и большом промежутке T (см. рис. 2, 3, 6), что увеличивает вычислительную сложность метода. В итоге приходится долго ждать получения результата. При уменьшении промежутка интегрирования `trapz` выдает точный результат прямоугольной функции (см. рис. 4), однако Фурье-образ становится негладким. При большом шаге интегрирования подсчеты происходят быстро, но теряется точность функций во временной и частотных областях (см. рис. 5). Таким образом, метод точный, но затрачивает много времени на вычисления.

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. Выполним обратное преобразование от найденного Фурье-образа с помощью обратного дискретного преобразования (конструкция `ifft(ifftshift())`). Схематично наши действия можно представить так:

$$\Pi(t) \xrightarrow{\texttt{fftshift(fft())}} \hat{\Pi}(\nu) \xrightarrow{\texttt{ifft(ifftshift())}} \Pi(t)$$

Для того, чтобы преобразование было унитарным, необходимо домножить ряд дискретного преобразования Фурье на коэффициент

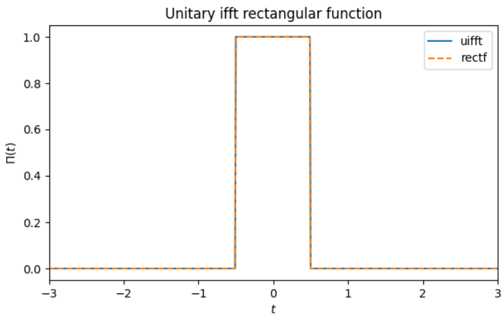
$$\frac{1}{\sqrt{N}}, \quad N \text{ -- количество членов ряда.}$$

Аналогично для обратного преобразования Фурье. Таким образом, формулы DFT и IDFT будут иметь вид:

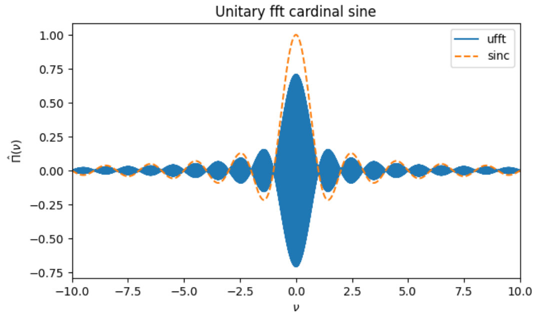
$$\mathcal{F}_m = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} f_n e^{-2\pi i \frac{mn}{N}}, \quad f_n = \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} \mathcal{F}_m e^{2\pi i \frac{mn}{N}}$$

Далее приведены сравнительные графики найденной $\hat{\Pi}(\nu)$ и *восстановленной* $\Pi(t)$ функций с исходными. Цвета и обозначения аналогичны предыдущему пункту.

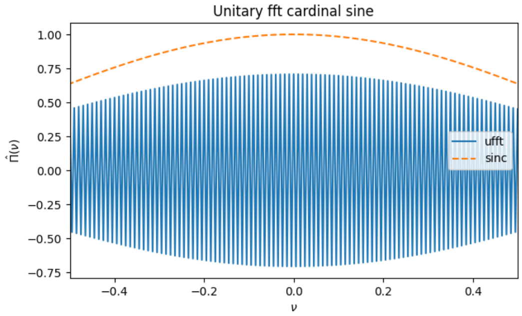
Исходя из графиков можно сделать вывод, что унитарное быстрое преобразование Фурье точно восстанавливает исходный сигнал, затрачивая малое количество времени на вычисления. Однако Фурье-образ не совпадает с истинным (кардинальным синусом). Функция приобрела лишние амплитуды на всей частотной области (см. рис. 7).



(a) $\Pi(t)$, восстановленная `ufft`



(b) $\hat{\Pi}(t)$ восстановленной `ufft` $\Pi(t)$



(c) Рис. (b) в приближении

Рис. 7: Унитарное быстрое преобразование Фурье `ufft`

1.4 Выводы о trapz и fft

Метод `trapz` работает медленно, но точно – он смог приблизиться к истинному Фурье-образу прямоугольной функции. `trapz` выполняет численное интегрирование с помощью трапециевидного метода – он аппроксимирует интегрирование на интервале путем разламывания области на трапецоиды с более легко вычислимыми областями. Для интеграции с $N+1$ равномерно распределенными точками приближение будет иметь вид:

$$\int\limits_a^b f(x)\,dx \approx \frac{b-a}{2N}\sum_{n=1}^{N}\left(f(x_n)+f(x_{n+1})\right), \quad \frac{b-a}{N} \text{ – интервал между каждой точкой.}$$

К выражению выше добавляется умножение на экспоненту, в степени которой находится переменная, по которой вычисляется интеграл. Обратное преобразование находится аналогично.

Метод `fft` работает быстро, но Фурье-образ не похож на кардинальный синус. В предыдущем пункте были приведены формулы, которые используются для дискретного преобразования Фурье – через них работает быстрое преобразование Фурье. `fft` основывается на «симметриях» корней из единицы матрицы `dft`, которую можно составить из элементов одноименного ряда. Эта матрица квадратная и обратимая, иначе мы не смогли бы найти `idft`. Более того, она унитарная – обратная матрица является транспонированной комплексно-сопряженной – воздействует как «поворот». Симметрия корней из 1 связана с тем, что значения $\omega_N^k = e^{2\pi i k \div N}$ делятся на две группы: те, у которых k четное, и те, у которых k нечетное. При этом $\omega_N^{N\div 2}$ является корнем из -1.

Фурье-образ получается лучше у метода трапеций, так как он проходит по каждому значению времени и вычисляет в приближении сложный интеграл. Однако у `fft` по времени лучше получается восстановленный сигнал несмотря на то, что Фурье-образ не совпадает с истинным. Причина в тех симметриях, о которых написано в абзаце выше – алгоритмы `fft` и `ifft` взаимно обратны. Ошибки округления и другие численные погрешности, возникающие при прямом преобразовании, компенсируются при обратном преобразовании.

1.5 Приближение непрерывного с помощью DFT

Чтобы исправить ситуацию, попробуем совместить достоинства обоих подходов: точность и быстродействие. Найдем способ получить правильный Фурье-образ, соответствующий непрерывному преобразованию Фурье, используя функцию `fft` и не прибегая к численному интегрированию. Найдем способ восстановить исходный сигнал по полученному Фурье-образу – тоже с помощью `fft`. Схема нашего успеха:

$$\Pi(t) \xrightarrow{\text{умное использование }\texttt{fft}} \hat{\Pi}(\nu) \xrightarrow{\text{умное использование }\texttt{ifft}} \Pi(t)$$

Ранее мы выяснили, что метод трапеций работает наиболее точно, но он ресурсозатратен. Вспомним вид интеграла Фурье для ограниченной по времени функции:

$$\mathcal{F}(\nu) = \int_{-t_0}^{t_0} f(t)e^{-2\pi i\nu t}\,dt$$

Аппроксимируем интеграл Фурье с помощью суммы Римана – бесконечного числа маленьких прямоугольников, аппроксимирующих кривую в интеграле. Представим $t = n\cdot\Delta t + t_0$ – время на n-ом шаге с масштабированием Δt и некоторым положением относительно t_0.

$$\mathcal{F}(\nu) \approx \sum_{n=0}^{N-1} f(n\cdot\Delta t + t_0)e^{-2\pi i\nu(n\cdot\Delta t+t_0)}\,\Delta t$$

Разделив временной диапазон на небольшие интервалы и задав постоянное значение сигнала для каждого интервала, мы, по сути, просто выполняем временную дискретизацию сигнала. Упростим выражение, собрав как можно больше слагаемых, не зависящих от n:

$$\mathcal{F}(\nu) \approx \Delta t\sum_{n=0}^{N-1} f(n\cdot\Delta t + t_0)e^{-2\pi i\nu n\Delta t}e^{-2\pi i\nu t_0}$$

Так как Δt и t_0 – постоянные, обозначим:

$$f[n] = f(n\cdot\Delta t + t_0)$$

Частоты ν нам известны. Оставим их как есть для константы экспоненты вне ряда, а для экспоненты со степенью, содержащей n, выразим их как

$$\nu_k = k\cdot\Delta\nu = \frac{k}{N\cdot\Delta t}$$

Δt в экспоненте сократятся, получим:

$$\mathcal{F}\left(\nu_k = \frac{k}{N\cdot\Delta t}\right) \approx \Delta t\, e^{-2\pi i\nu t_0}\sum_{n=0}^{N-1} f[n]e^{-2\pi i\frac{kn}{N}}$$

Формула дискретного преобразования Фурье может быть представлена как

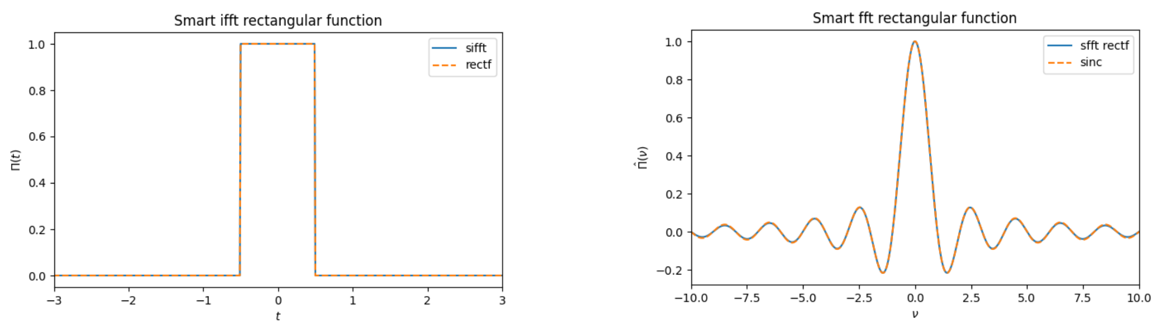
$$\mathcal{F}[k] = \sum_{n=0}^{N-1} f[n] e^{-2\pi i \frac{kn}{N}},$$

подставляя ее в формулу $\mathcal{F}(\nu)$ получим

$$\mathcal{F}\left(\nu_k = \frac{k}{N \cdot \Delta t}\right) \approx \Delta t \, e^{-2\pi i \nu t_0} \mathcal{F}[k],$$

где $\mathcal{F}[k]$ – дискретное преобразование Фурье, которое можно заменить на быстрое преобразование Фурье. То есть результат `fft` необходимо домножить на $\Delta t \, e^{-2\pi i \nu t_0}$, чтобы получить Фурье-образ, соответствующий эталону. Для обратного преобразования перед применением `ifft` просто разделим умный `fft` на то же слагаемое, так как оно является константой, а `ifft` в точности быстро восстанавливает исходный сигнал без дополнительных преобразований `fft`. Кратко будем обозначать такое преобразование как `sfft` и `sifft`.

Далее приведены сравнительные графики исходного сигнала и его Фурье-образа с результатами после умного применения `fft` и `ifft`.



(a) $\Pi(t)$, восстановленная `sfft` (b) $\hat{\Pi}(t)$ восстановленной `sfft` $\Pi(t)$

Рис. 8: Умное использование быстрого преобразования Фурье `sfft` и `sifft`

2 Задание 2. Сэмплирование

В этом задании будем исследовать теорему Найквиста-Шеннона-Котельникова на двух примерах.

2.1 Сэмплирование синусов

Зададим параметры $a_1 = 1$, $a_2 = 2$, $\omega_1 = 3$, $\omega_2 = 4$, $\varphi_1 = \pi \div 5$, $\varphi_2 = \pi \div 6$ и рассмотрим функцию

$$y(t) = a_1 \sin(\omega_1 t + \varphi_1) + a_2 \sin(\omega_2 t + \varphi_2)$$

Зададим в `Python` массивы времени `t` и значений `y`. Выберем малый шаг дискретизации $dt = 10^{-4}$ для имитирования непрерывной функции. Построим график полученной функции.

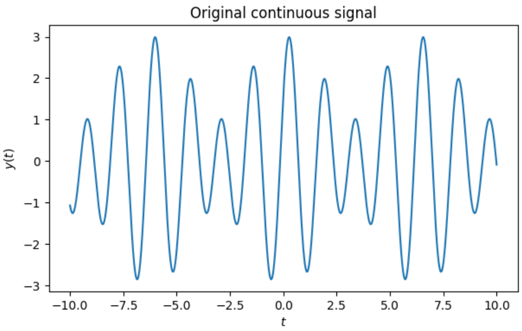


Рис. 9: График непрерывного сигнала

Теперь зададим сэмплированный вариант указанной функции: рассмотрим разреженный вариант массива времени и соответствующий ему массив значений. Сначала рассмотрим большой шаг $dt = 0.2$. Построим дискретный график поверх непрерывного.

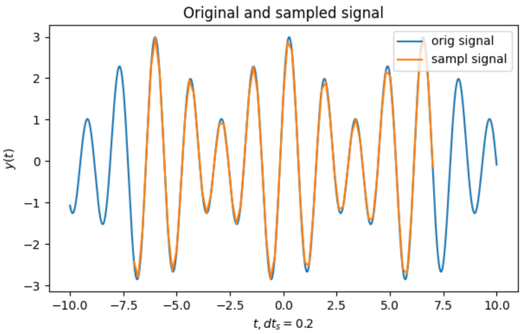


Рис. 10: График сэмплированного сигнала поверх непрерывного

Применим интерполяционную формулу из лекции к сэмплированным данным с целью восстановить непрерывную функцию.

$$f(t) = \sum_{n=-\infty}^{\infty} f(t_n) \cdot \operatorname{sinc}\left(2B(t - t_n)\right), \quad t_n = \frac{n}{2B}$$

В результате получатся новые массивы времени и значений – той же размерности, что и исходные.

Построим график восстановленной функции поверх исходной.

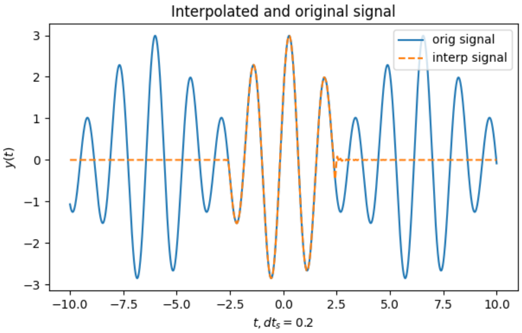
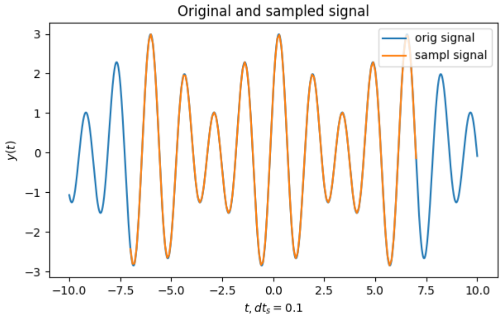
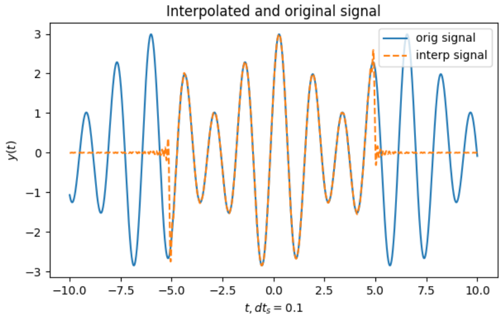


Рис. 11: График интерполированного сигнала поверх непрерывного

Далее исследуем влияние шага дискретизации на вид восстановленной функции и соотнесем свои результаты с теоремой Найквиста-Шеннона-Котельникова. Синим цветом выделены исходные сигналы, оранжевым – сэмплированные и восстановленные. Под шкалой времени на каждом графике указан шаг дискретизации сэмплирования dt_s.

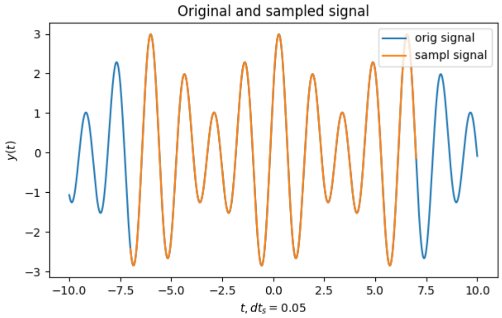


(a) Сэмплированный поверх непрерывного

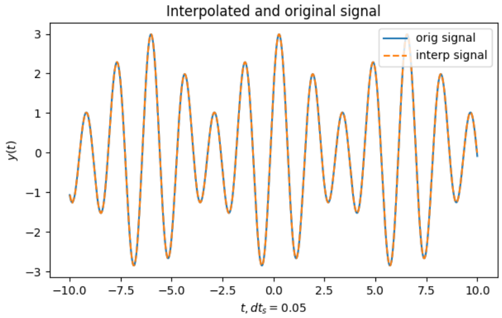


(b) Интерполированный поверх непр-ого

Рис. 12: Уменьшение шага дискретизации $dt = 0.1$



(a) Сэмплированный поверх непр-ого



(b) Интерполированный поверх непр-ого

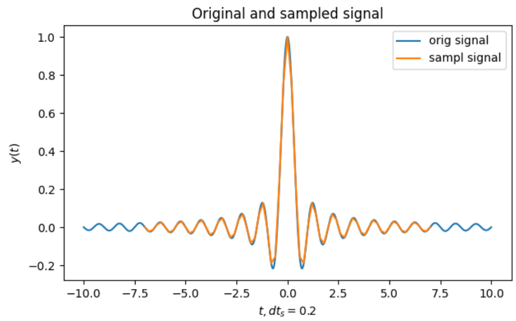
Рис. 13: Уменьшение шага дискретизации $dt = 0.05$

2.2 Сэмплирование sinus cardinalis

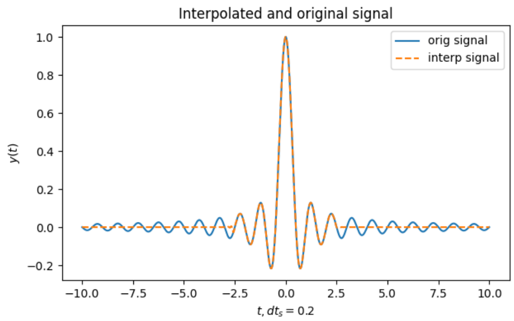
Зададим параметр $b = 2$ и рассмотрим функцию

$$y(t) = \operatorname{sinc}(bt) = \operatorname{sinc}(2t)$$

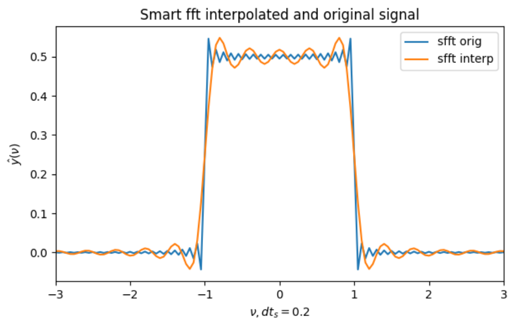
Выполним все шаги из предыдущего пункта. Дополнительно для каждой величины шага дискретизации построим Фурье-образ исходного и восстановленного сигналов. Дадим объяснение увиденному, сделаем выводы.



(a) Сэмплированный поверх непр-ого

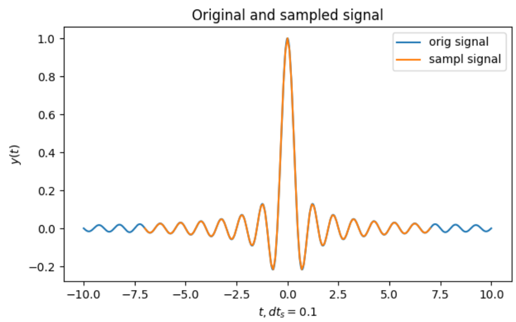


(b) Интерполированный поверх непр-ого

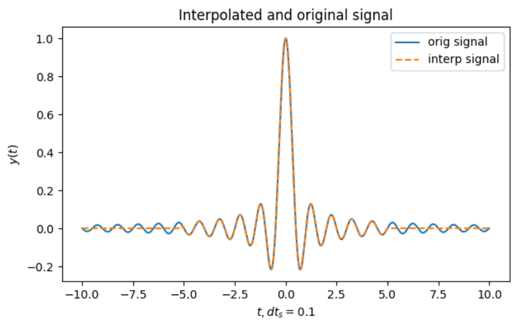


(c) Фурье-образ

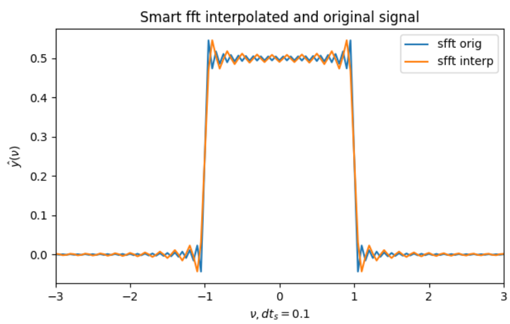
Рис. 14: Подпись



(a) Сэмплированный поверх непр-ого

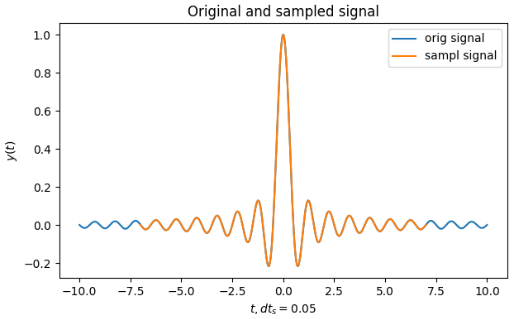


(b) Интерполированный поверх непр-ого

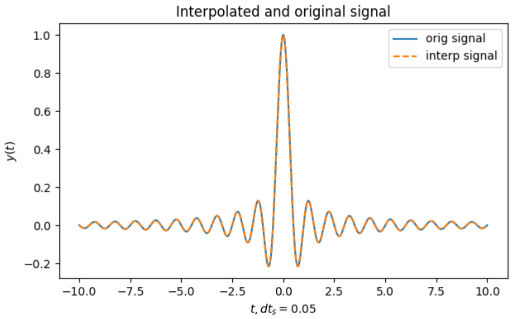


(c) Фурье-образ

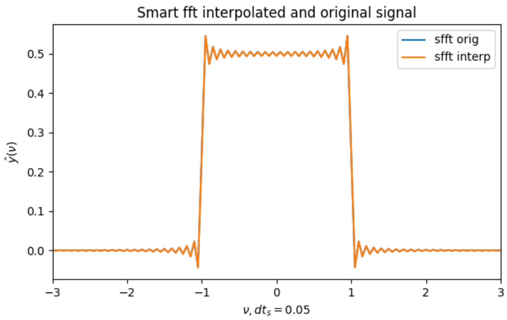
Рис. 15: Подпись



(a) Сэмплированный поверх непр-ого



(b) Интерполированный поверх непр-ого



(c) Фурье-образ

Рис. 16: Подпись