Цель работы:

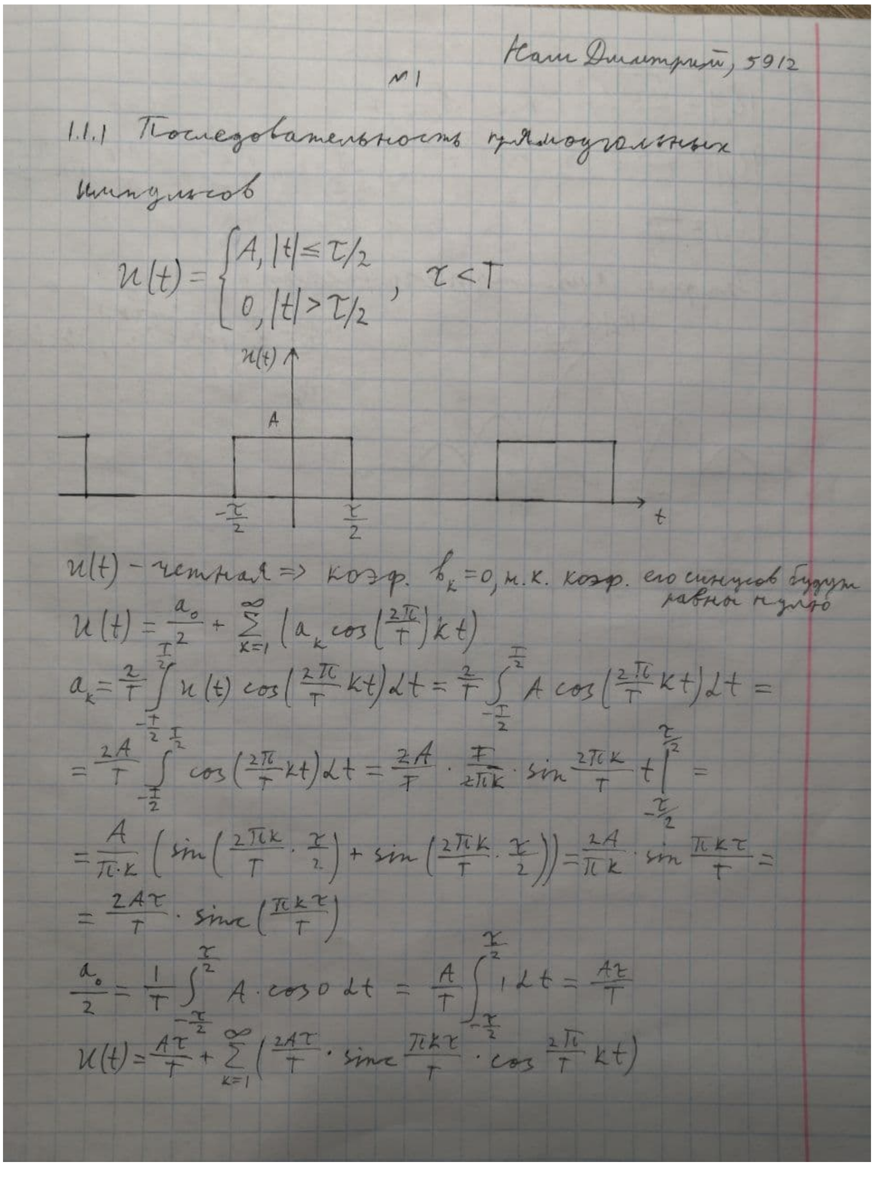
1. Изучение свойств дискретизации аналоговых сигналов;

2. Изучение методов Фурье-анализа аналоговых сигналов.

1. Ход работы

1. Выполнить аналитический расчет спектров для двух классов сигналов (периодических и непериодических). В отчете представить графики исходных сигналов и полученных спектров, а также описать их характерные особенности.

1.2 Последовательность прямоугольных импульсов


Нам Дмитрий, 5912

№ 1

1.1.1 Последовательность прямоугольных импульсов

$$u(t) = \begin{cases} A, & |t| \le \tau/2 \\ 0, & |t| > \tau/2 \end{cases}, \quad \tau < T$$

$u(t)$ – четная => коэф. $b_k = 0$, т.к. коэф. его синусов будут равны нулю

$$u(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} \left(a_k \cos\left(\frac{2\pi}{T} k t\right)\right)$$

$$a_k = \frac{2}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} u(t)\cos\left(\frac{2\pi}{T}kt\right)dt = \frac{2}{T}\int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} A\cos\left(\frac{2\pi}{T}kt\right)dt =$$

$$= \frac{2A}{T}\int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} \cos\left(\frac{2\pi}{T}kt\right)dt = \frac{2A}{T}\cdot\frac{T}{2\pi k}\cdot \sin\frac{2\pi k}{T}t\Big|_{-\frac{\tau}{2}}^{\frac{\tau}{2}} =$$

$$= \frac{A}{\pi \cdot k}\left(\sin\left(\frac{2\pi k}{T}\cdot\frac{\tau}{2}\right) + \sin\left(\frac{2\pi k}{T}\cdot\frac{\tau}{2}\right)\right) = \frac{2A}{\pi k}\cdot\sin\frac{\pi k \tau}{T} =$$

$$= \frac{2A\tau}{T}\cdot \mathrm{sinc}\left(\frac{\pi k \tau}{T}\right)$$

$$\frac{a_0}{2} = \frac{1}{T}\int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} A\cdot\cos 0\, dt = \frac{A}{T}\int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} 1\, dt = \frac{A\tau}{T}$$

$$u(t) = \frac{A\tau}{T} + \sum_{k=1}^{\infty}\left(\frac{2A\tau}{T}\cdot \mathrm{sinc}\frac{\pi k\tau}{T}\cdot\cos\frac{2\pi}{T}kt\right)$$

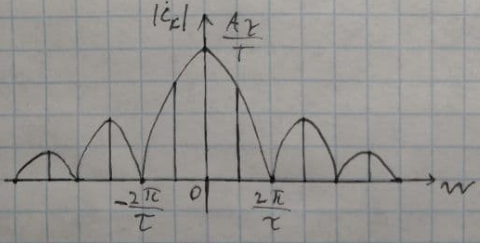
Рисунок 1 - Последовательность прямоугольных импульсов

$|\dot{C}_k| = \frac{\sqrt{a_k^2+b_k^2}}{2}$, т.к. $b_k=0$, то

$|\dot{C}_k| = \frac{a_k}{2} = \frac{\frac{2A}{\pi k}\cdot\sin\frac{\pi k\tau}{T}}{2} = \frac{A}{\pi k}\cdot\sin\frac{\pi k\tau}{T} = \frac{A\tau}{T}\cdot\sin\frac{\pi k\tau}{T}$

$|\dot{C}_0| = \frac{A\tau}{T}$

График амплитудного спектра

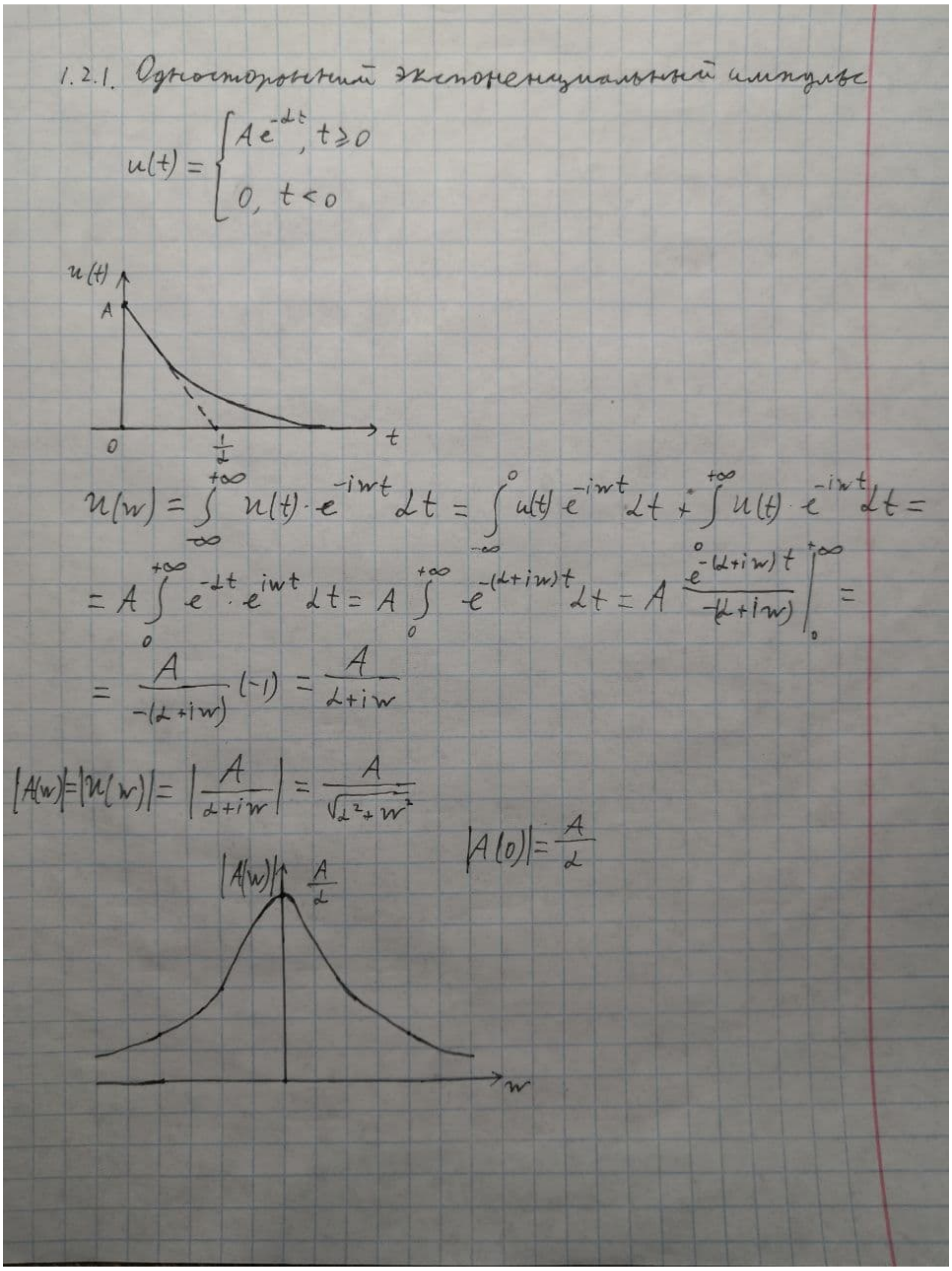


$A_k = \frac{A}{q}\cdot\left|\mathrm{sinc}\left(\frac{\pi k}{q}\right)\right|$, где $q=\frac{T}{\tau}$

$w_k = \frac{2\pi k}{\tau}$

Рисунок 2 – График спектра

1.3 Односторонний экспоненциальный импульс



1.2.1. Односторонний экспоненциальный импульс

$$u(t) = \begin{cases} Ae^{-\alpha t}, & t \geq 0 \\ 0, & t < 0 \end{cases}$$

$$U(w) = \int_{-\infty}^{+\infty} u(t) \cdot e^{-iwt} d t = \int_{-\infty}^{0} u(t) e^{-iwt} d t + \int_{0}^{+\infty} u(t) e^{-iwt} d t =$$

$$= A \int_{0}^{+\infty} e^{-\alpha t} \cdot e^{iwt} d t = A \int_{0}^{+\infty} e^{-(\alpha+iw)t} d t = A \left. \frac{e^{-(\alpha+iw)t}}{-(\alpha+iw)} \right|_{0}^{+\infty} =$$

$$= \frac{A}{-(\alpha+iw)} (-1) = \frac{A}{\alpha+iw}$$

$$|A(w)| = |U(w)| = \left| \frac{A}{\alpha+iw} \right| = \frac{A}{\sqrt{\alpha^2+w^2}}$$

$$|A(0)| = \frac{A}{\alpha}$$

Рисунок 3 - Односторонний экспоненциальный импульс

2. Проанализировать свойства двух функций $\boldsymbol{u_1(t) = sin(2\pi f_1 t)}$ и $\boldsymbol{u_2(t) = sin(2\pi f_2 t)}$ в интервале [−T/2; T/2].

Пусть f_1= 2,5 Гц, f_2= 3 Гц, T = 2 с.

2.1 Вычислить все значения функций $u_1(t)$ и $u_2(t)$ на заданном интервале с шагом 10^{-3}.

На интервале [-1; 1] построены графики функций исходных сигналов с частотами f_1= 2,5 Гц и f_2= 3 Гц (рис. 6).

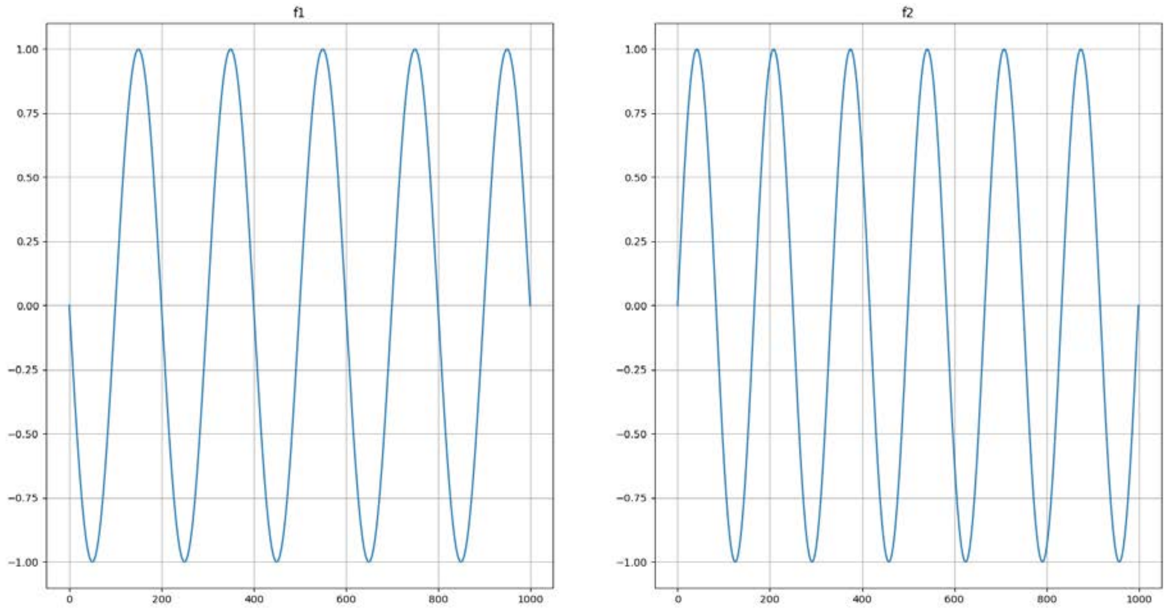


Рисунок 4 - Графики функций исходных сигналов

2.2 Вычислить приближенное значение скалярного произведения двух функций $(\boldsymbol{u_1(t)}, \boldsymbol{u_2(t)})$.

Ниже приведена формула скалярного произведения (1).

$$\left(u_1(t), u_2(t)\right) = \frac{1}{T}\int_a^b u_1(t)\cdot u_2(t)dt = \tag{1}$$

$$\left(u_1(t), u_2(t)\right) = \frac{1}{T}\int_a^b u_1(t)\cdot u_2(t)dt = \frac{1}{T}\int_a^b sin(2\pi f_1 t)\cdot sin(2\pi f_2 t)dt =$$

$$= \frac{1}{2T}\left(\int_a^b cos(2\pi t(f_1 - f_2))dt - \int_a^b cos(2\pi t(f_1 + f_2))dt\right)$$

Так как интервал (a, b) равен [−T/2; T/2] и периоды функций $u_1(t) = sin(2\pi f_1 t)$ и $u_2(t) = sin(2\pi f_2 t)$ входят в данный интервал целым числом раз, скалярное произведение этих функций равно нулю. Вычисления приведены ниже.

$$T = \frac{k}{f_1} = \frac{n}{f_2}, \qquad k, n \in N$$

$$\left(u_1(t), u_2(t)\right) = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} sin(2\pi t f_1)\cdot sin(2\pi t f_2)dt =$$

$$= \frac{1}{2T}\left(\int_{-\frac{T}{2}}^{\frac{T}{2}} cos(2\pi t(f_1 - f_2))dt - \int_{-\frac{T}{2}}^{\frac{T}{2}} cos(2\pi t(f_1 + f_2))dt\right) = 0$$

Значение скалярного произведения было рассчитано в программе:

$$\big(u_1(t), u_2(t)\big) = 1.4013948354085718e^{-17}$$

Такое расхождение вызвано тем, что программно были вычислены суммы отсчётов.

2.3 Вычислить нормы обеих функций.

Ниже приведены формула вычисления нормы функции (2).

$$\|u(t)\| = \sqrt[2]{\frac{1}{T} \cdot \int_a^b u^2(t)dt} \qquad (2)$$

Интервал [−T/2; T/2] и периоды функций $u_1(t) = sin(2\pi f_1 t)$ и $u_2(t) = sin(2\pi f_2 t)$ входят в данный интервал целым числом раз.

$$T = \frac{k}{f_1} = \frac{n}{f_2}, \qquad k, n \in N$$

$$\|u(t)\| = \sqrt[2]{\frac{1}{T} \cdot \int_{-T/2}^{T/2} \sin(2\pi f t)^2 \, dt}$$

Значения нормы были рассчитаны в программе:

$$\|u_1\| = 0.7071068$$

$$\|u_2\| = 0.7071068$$

2.4 Определить, являются ли исходные функции ортогональными друг к другу.

Ниже приведено вычисление скалярного произведения.

$$\big(u_1(t), u_2(t)\big) = \frac{1}{T}\int_a^b sin(2\pi t f_1) \cdot sin(2\pi t f_2)dt =$$

$$= \frac{1}{2T}\left(\int_a^b cos(2\pi t(f_1 - f_2))dt - \int_a^b cos(2\pi t(f_1 + f_2))dt\right)$$

Так как интервал (a, b) равен [−T/2; T/2] и периоды функций $u_1(t) = sin(2\pi f_1 t)$ и $u_2(t) = sin(2\pi f_2 t)$ входят в данный интервал целым числом раз, скалярное произведение этих функций равно нулю. Вычисления приведены ниже.

$$T = \frac{k}{f_1} = \frac{n}{f_2}, \qquad k, n \in N$$

$$\left(u_1(t), u_2(t)\right) = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} sin(2\pi t f_1) \cdot sin(2\pi t f_2) dt =$$

$$= \frac{1}{2T}\left(\int_{-\frac{T}{2}}^{\frac{T}{2}} cos(2\pi t(f_1 - f_2))dt - \int_{-\frac{T}{2}}^{\frac{T}{2}} cos(2\pi t(f_1 + f_2))dt\right) = 0$$

Так как значение скалярного произведения равно нулю, функции $u_1(t) = sin(2\pi f_1 t)$ и $u_2(t) = sin(2\pi f_2 t)$ являются ортогональными друг к другу.

Значение скалярного произведения было рассчитано в программе:

$$\left(u_1(t), u_2(t)\right) = 1.4013948354085718e^{-17}$$

Такое расхождение вызвано тем, что программно вычисляются суммы отсчётов.

Таким образом, было выяснено, что функции $u_1(t) = sin(5\pi t)$ и $u_2(t) = sin(6\pi t)$ являются ортогональными друг к другу.

2.5 Как нужно изменить исходные функции, что они могли являться элементами ортонормированного базиса? Выполните данную модификацию и продемонстрируйте результат.

Для того, чтобы функции стали элементами ортонормированного базиса, нужно разделить их на собственные нормы:

$$g(t) = \frac{u(t)}{\|u(t)\|} \tag{3}$$

Значения норм функций были пересчитаны с помощью языка Python (см. приложение А):

$$\|s(t)\| = \sqrt{\frac{1}{b-a}\int_a^b \left(\frac{u(t)}{\|u(t)\|}\right)^2 dt} \tag{4}$$

$$\|s_1\| = 1$$

$$\|s_2\| = 1$$

2.6 Останутся ли исследуемые функции элементами ортонормированного базиса, если:

2.6.1. Частоты $f1$ и $f2$ удвоятся:

Функции $s_1(t) = sin(2\pi f_1 t)$ и $s_2(t) = sin(2\pi f_2 t)$ останутся элементами ортонормированного базиса, т.к. при увеличении частоты в 2 раза периоды функций уменьшаются в 2 раза, но все также будут целым числом раз входить в интервал [−T/2; T/2].

$$T = \frac{k}{2f_1} = \frac{n}{2f_2}, \qquad k, n \in N$$

$$\left(\grave{s_1}(t), \grave{s_2}(t)\right) = \frac{2}{2\,T}\int_{-\frac{T}{2}}^{\frac{T}{2}} sin(4\pi t f_1) \cdot sin(4\pi t f_2) dt =$$

$$= \frac{1}{T}\left(\int_{-\frac{T}{2}}^{\frac{T}{2}} cos(4\pi t(f_1 - f_2))dt - \int_{-\frac{T}{2}}^{\frac{T}{2}} cos(4\pi t(f_1 + f_2))dt\right) = 0$$

Таким образом, было выяснено, что функции $u_1(t) = sin(5\pi t)$ и $u_2(t) = sin(6\pi t)$ останутся элементами ортонормированного базиса, если частоты этих функций увеличатся вдвое.

2.6.2. Интервал [−T/2; T/2] увеличится вдвое:

Функции $s_1(t) = sin(2\pi f_1 t)$ и $s_2(t) = sin(2\pi f_2 t)$ останутся элементами ортонормированного базиса, т.к. при увеличении интервала [−T/2; T/2] в 2 раза периоды функций также будут целым числом раз входить в интервал [−T/2;T/2].

$$T = \frac{k}{f_1} = \frac{n}{f_2}, \qquad k, n \in N$$

$$\left(\grave{s_1}(t), \grave{s_2}(t)\right) = \frac{1}{T}\int_{-T}^{T} sin(2\pi t f_1) \cdot sin(2\pi t f_2) dt =$$

$$= \frac{1}{2T}\left(\int_{-T}^{T} cos(2\pi t(f_1 - f_2))dt - \int_{-T}^{T} cos(2\pi t(f_1 + f_2))dt\right) = 0$$

Таким образом, было выяснено, что функции $u_1(t) = sin(5\pi t)$ и $u_2(t) = sin(6\pi t)$ останутся элементами ортонормированного базиса, если интервал [−T/2; T/2] увеличится вдвое.

2.6.3. Интервал [−T/2; T/2] уменьшится вдвое:

Функции $s_1(t) = sin(2\pi f_1 t)$ и $s_2(t) = sin(2\pi f_2 t)$ останутся элементами ортонормированного базиса, только в случае, если их периоды будут целым числом раз входить в интервал [−T/4; T/4].

$$\frac{T}{2} = \frac{k}{f_1} = \frac{n}{f_2}, \qquad k, n \in N$$

$$\left(s_1^{`}(t), s_2^{`}(t)\right) = \frac{4}{T}\int_{-\frac{T}{4}}^{\frac{T}{4}} sin(2\pi t f_1) \cdot sin(2\pi t f_2) dt =$$

$$= \frac{2}{T}\left(\int_{-\frac{T}{4}}^{\frac{T}{4}} cos(2\pi t(f_1 - f_2))dt - \int_{-\frac{T}{4}}^{\frac{T}{4}} cos(2\pi t(f_1 + f_2))dt\right) = 0$$

Таким образом, было выяснено, что функция $u_2(t) = sin(6\pi t)$ останется элементом ортонормированного базиса, если интервал [−T/2; T/2] уменьшится вдвое. Однако, функция $u_2(t) = sin(5\pi t)$ не будет элементом ортонормированного базиса, если интервал [−T/2; T/2] уменьшится вдвое, так как ее период не будет целым числом раз входить в интервал [−T/4; T/4].

3. Исследовать процедуру дискретизации синусоидального сигнала u(t) с частотой F Гц. Длительность наблюдения сигнала 10/F секунд. Написать программу, которая позволит:

3.1 Сформировать выборки отсчетов $u^{(i)}[n]$ (результаты дискретизации) исследуемого сигнала с частотами дискретизации $f_d^{(i)}$ равными 1.5F, 1.75F, 2F, 3F, 1000F. Применить ряд Котельникова для восстановления исходного сигнала по его дискретным отсчетам с помощью формулы. Вывести графики исходного сигнала и восстановленных сигналов $u^{(i)}(t)$, где i— индекс частоты дискретизации.

Восстановленный сигнал получен по формуле интерполяционного ряда Котельникова.

$$u(t) = \sum_{k=-\infty}^{\infty} \mathrm{u}(\mathrm{nT_d})\mathrm{sinc}(\pi\frac{\mathrm{t} - \mathrm{nT_d}}{\mathrm{T_d}}) \quad (5)$$

3.1.1. 1.5F Гц

Произведена дискретизация исходного сигнала с частотой 1.5F, который затем был восстановлен. Результаты приведены на графике ниже (рис.7).

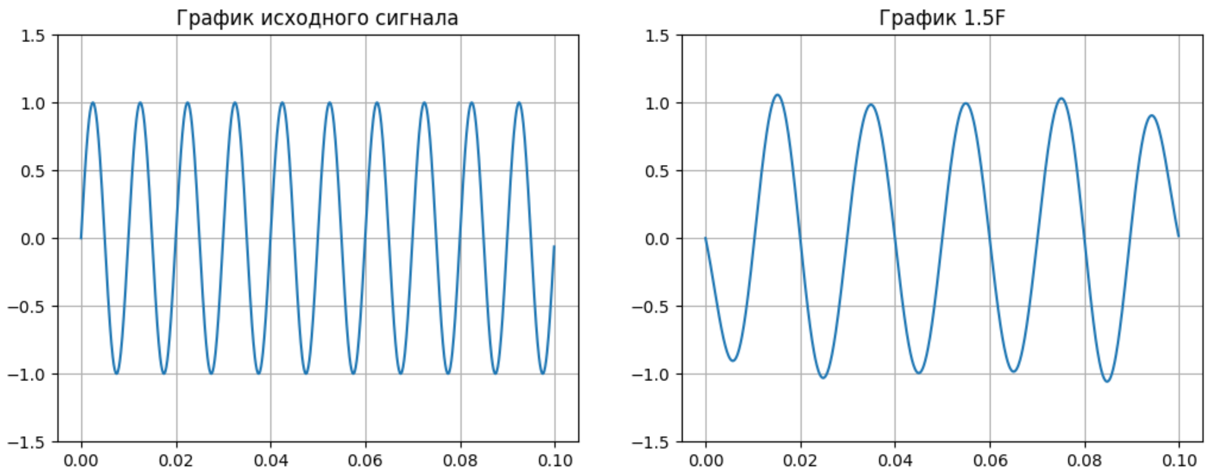


Рисунок 7 - Графики функций исходного и восстановленного сигналов

3.1.2. 1.75F Гц

Произведена дискретизация исходного сигнала с частотой 1.75F, который затем был восстановлен. Результаты приведены ниже (рис. 8).

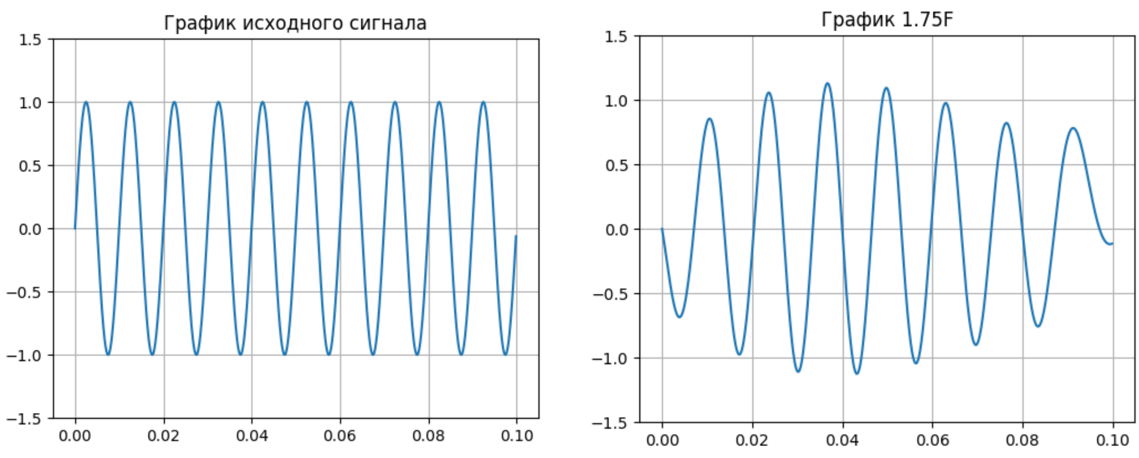


Рисунок 8 - Графики функций исходного и восстановленного сигналов

3.1.3. 2F Гц

Произведена дискретизация исходного сигнала с частотой 2F, который затем был восстановлен. Результаты приведены ниже (рис. 9).

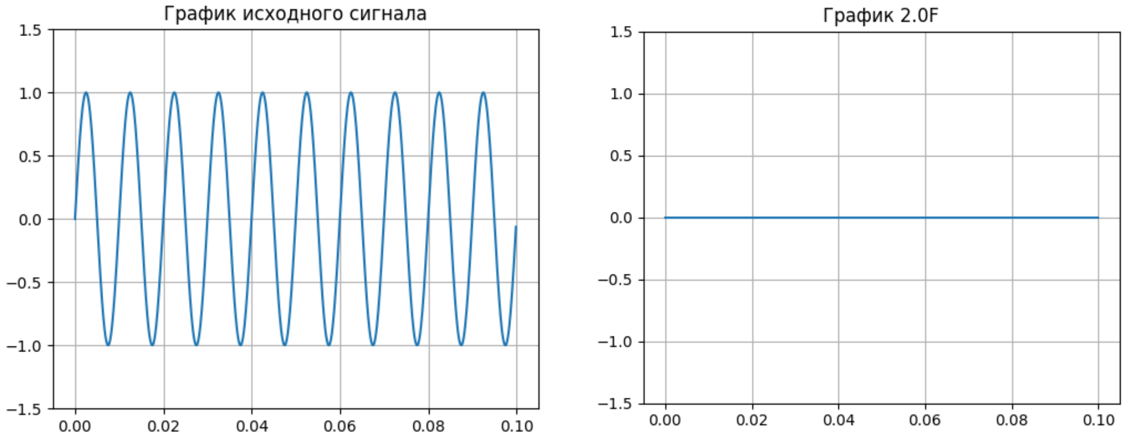


Рисунок 9 - Графики функций исходного и восстановленного сигналов

Таким образом, по рисунку 9 видно, что восстановление сигнала, дискретизированного с частотой 2F, не происходит.

3.1.4. 3F Гц

Произведена дискретизация исходного сигнала с частотой 3F, который затем был восстановлен. Результаты приведены ниже (рис. 10).

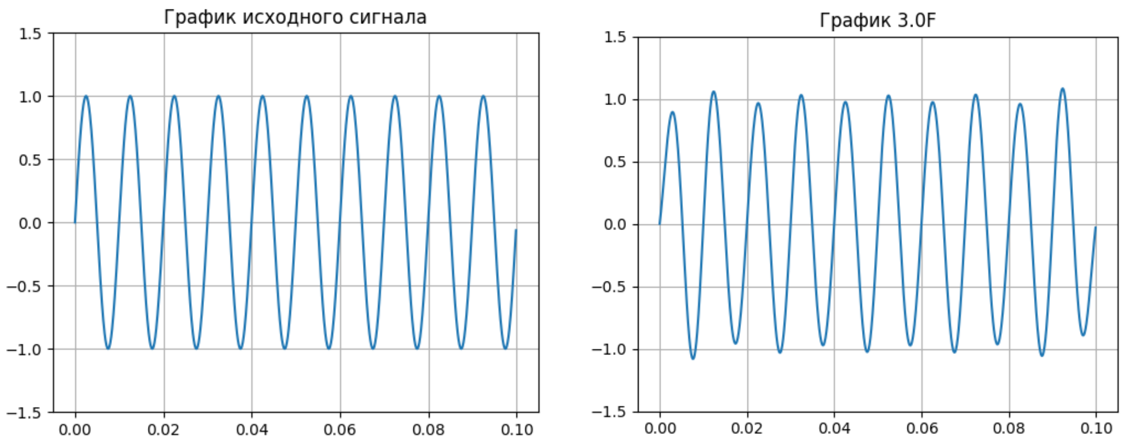


Рисунок 10 - Графики функций исходного и восстановленного сигналов

3.1.5. 1000F Гц

Произведена дискретизация исходного сигнала с частотой 1000F, который затем был восстановлен. Результаты приведены на графике ниже (рис. 11).

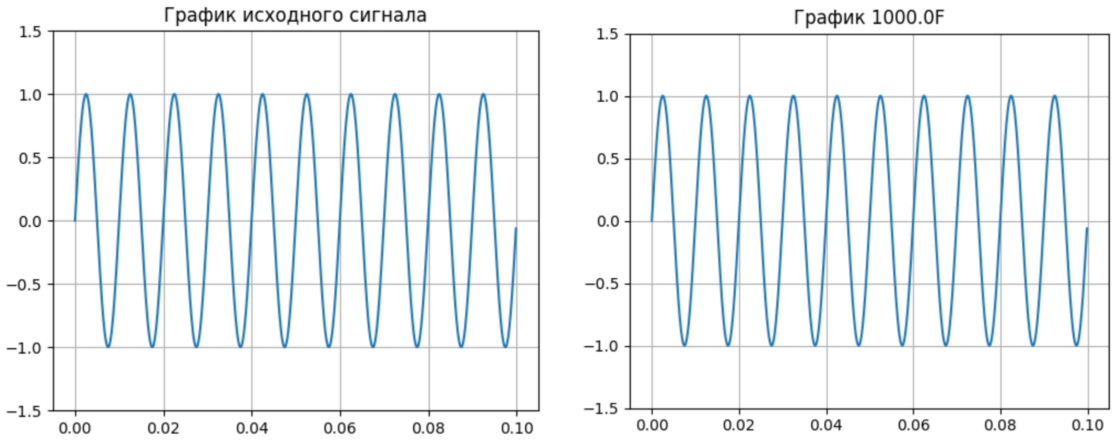


Рисунок 11 - Графики функций исходного и восстановленного сигналов

На рисунке ниже (рис. 12) продемонстрированы графики функций исходного сигнала и всех восстановленных сигналов с использованием формулы интерполяционного ряда Котельникова.

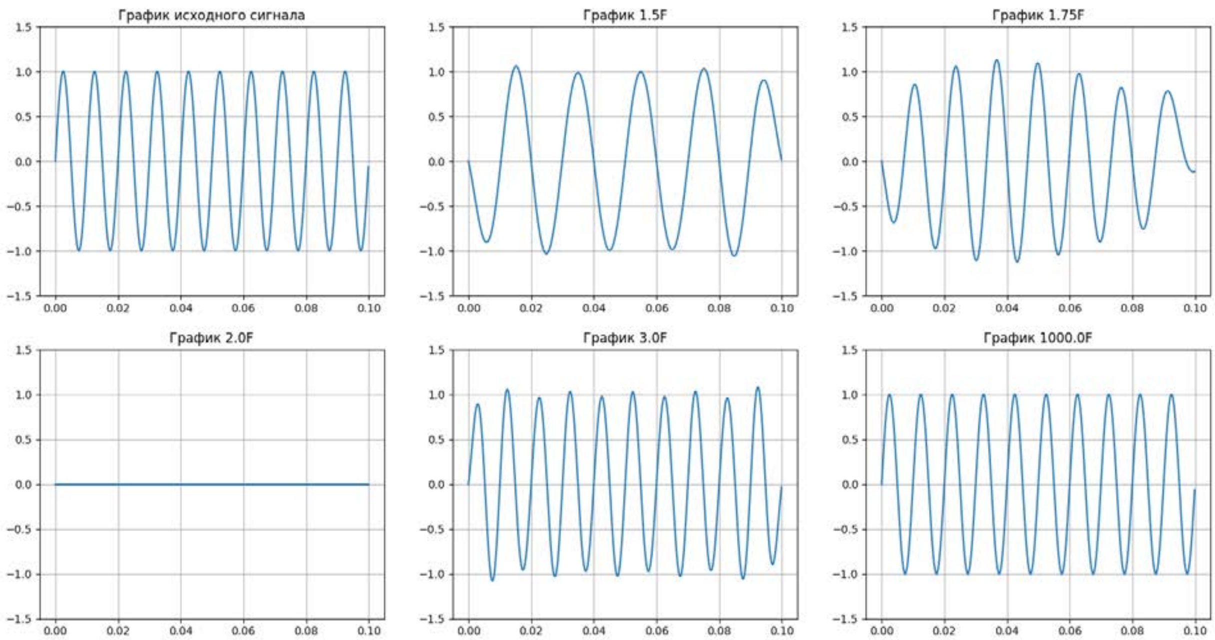


Рисунок 12 - Графики функций исходного и всех восстановленных сигналов

Было изучено, насколько точно можно восстановить исходный сигнал с помощью интерполяционного ряда Котельникова по формуле (5), дискретизированный с равными частотами. По теореме Котельникова восстановление исходного аналогового сигнала возможно, если равномерная дискретизация выполняется с частотой f_d, минимум вдвое превышающую частоту f_c: $f_d > 2\ f_c$.

Исходный синусоидальный сигнал с малой точностью восстановлен при малых частотах (1.5F, 1.75F), с большей точностью восстановлен при выполнении теоремы Котельникова (3F > 2F, 1000F > 2F), причем, чем больше частота дискретизации, тем точнее выполняется восстановление. Стоит отметить, что восстановление сигнала, дискретизированного с частотой 2F, не происходит.

4. Реализовать процедуру передискретизации изображения с помощью интерполяционного ряда Котельникова. Формат исходного изображения – BMP24, разрешение исходного изображения WxH пикселей. Результатом передискретизации будет изображение размером nWxmH. Выполните передискретизацию с различными комбинациями значений m и n:

Описание алгоритма передискретизации

Задаются коэффициенты передискретизации m и n. Изначально каждый пиксель изображения считывается из BMP файла в массив размером $[h \times w]$. Создается новая матрица размером $[n \cdot h \times w]$, которая заполняется пикселями исходной матрицы. Таким образом, полученная матрица представляет из себя массив отсчетов исходной матрицы, впоследствии в матрице отсчетов по каждой строке проводится восстановление сигнала по формуле интерполяционного ряда Котельникова.

Исходное изображение m = 1, n = 1 (рис. 13):

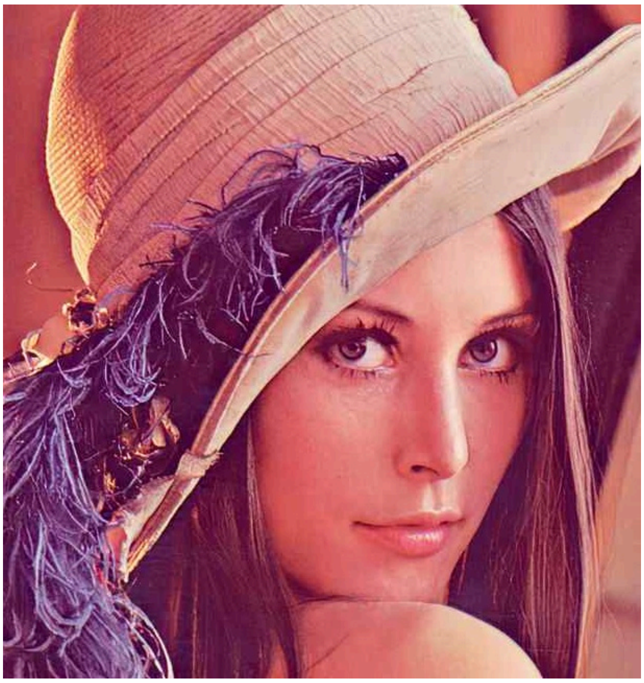


Рисунок 13 - Исходное изображение

4.1 m > 1, n > 1

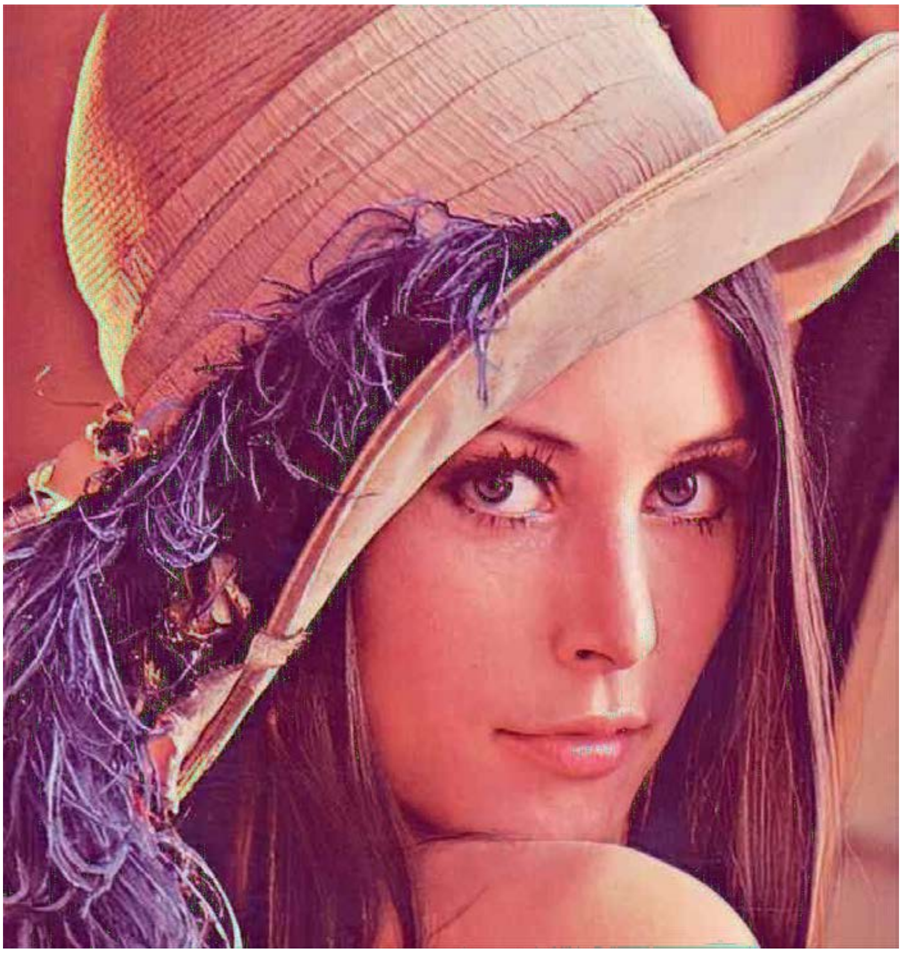


Рисунок 14 - Изображение при $m = 2$, $n = 2$

4.2 m < 1, n < 1

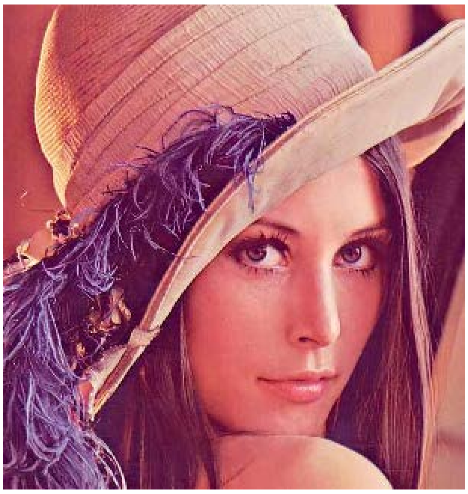


Рисунок 15 - Изображение при m = 0.5, n = 0.5

4.3 m > 1, n < 1

Рисунок 16 - Изображение при m = 2, n = 0.5

4.3 m < 1, n > 1

Рисунок 17 - Изображение при $m = 0.5$, $n = 2$

Можно заметить, что при передискретизации качество изображений снижается. Это связано с тем, что при уменьшении разрешения часть пикселей теряется, а при увеличении разрешения некоторые пиксели копируются, что приводит к размытию, потери качества изображения.

Вывод

В ходе выполнения пункта 1 лабораторной работы был проведен анализ Фурье для периодического и непериодического сигналов. Также были построены графики сигналов, были посчитаны фазовые и амплитудные спектры рассмотренных сигналов, построены графики спектров.

В пункте 2 был проведен анализ двух функций $u_1(t) = sin(5\pi t)$ и $u_2(t) = sin(6\pi t)$, по формулам (1) и (2) были соответственно определены их скалярное произведение и нормы. Было выяснено, что после преобразования данные функции являются элементами ортонормированного базиса, если разделить функции на собственные нормы. Также было выяснено, что функции $u_1(t) = sin(5\pi t)$ и $u_2(t) = sin(6\pi t)$ являются ортогональными друг к другу. Также можно сделать вывод, что при увеличении интервала вдвое и увеличении частот вдвое, функции остаются элементами ортонормированного базиса. А при уменьшении интервала вдвое функции останутся элементами ортонормированного базиса только в случае, если их периоды будут целым числом раз входить в рассматриваемый интервал, деленный пополам. В ходе выполнения задания было выяснено, что функция $u_2(t) = sin(6\pi t)$ останется элементом ортонормированного базиса, если интервал [−T/2; T/2] уменьшится вдвое. Однако, функция $u_2(t) = sin(5\pi t)$ не будет элементом ортонормированного базиса, если интервал [−T/2; T/2] уменьшится вдвое, так как ее период не будет целым числом раз входить в интервал [−T/4; T/4].

При выполнении пункта 3 было изучено, насколько точно можно восстановить с помощью интерполяционного ряда Котельникова по формуле (5) исходный сигнал, ранее дискретизированный с равными частотами. По теореме Котельникова восстановление исходного аналогового сигнала возможно, если равномерная дискретизация выполняется с частотой f_d, минимум вдвое превышающую частоту f_c:

$$f_d > 2\ f_c$$

Исходный синусоидальный сигнал с малой точностью восстановлен при малых частотах (1.5F, 1.75F), с большей точностью восстановлен при выполнении теоремы Котельникова (3F > 2F, 1000F > 2F), причем, чем больше частота дискретизации, тем точнее выполняется восстановление. Однако в ходе работы было выяснено, что восстановление сигнала, дискретизированного с частотой 2F, не происходит.

На построенных восстановленных графиках можно заметить искажения. Это связано с тем, что графики строятся по отсчетам, а при наименьшей частоте дискретизации недостаточно отсчетов для построения точного графика функции.

В пункте 4 лабораторной работы передискретизировано изображение формата BMP с помощью интерполяционного ряда Котельникова.

Формат BMP хранит изображение в виде трехмерного массива пикселей [h, w, c], где h – высота изображения, w – ширина изображения, c – количество цветовых компонент.

С помощью интерполяционного ряда Котельникова получены новые отсчеты [m*h, n*w, c] и заполнена матрица восстанавливаемого изображения. Результаты передискретизации представлены в пункте 4. Можно заметить, что качество изображения снижается. Это связано с тем, что при уменьшении разрешения часть пикселей теряется, а при увеличении разрешения некоторые пиксели копируются, что приводит к размытию, потере качества изображения.

Приложение А

Листинг программы

```
#файл 1_2DSP.py

import math as math
import matplotlib.pyplot as plt


def integral(u, tn):
    integ = 0
    for i in range(1, len(tn)):
        integ += (tn[i] - tn[i - 1]) * ((u[i] + u[i - 1]) / 2)
    return integ


def scalar(u1, u2, tn):
    a = min(tn)
    b = max(tn)
    u = [u1[i] * u2[i] for i in range(len(tn))]
    return integral(u, tn) / (b - a)


def norma(u, tn):
    a = min(tn)
    b = max(tn)
    u = [i ** 2 for i in u]
    return math.sqrt(integral(u, tn) / (b - a))


def func(f, t):
    tmp = []
    for i in t:
        tmp.append(math.sin(2 * math.pi * f * i))
    return tmp


def calculate(u1, u2, tn):
    sc = scalar(u1, u2, tn)
```

```
    norm1 = norma(u1, tn)
    norm2 = norma(u2, tn)

    print(f"Скалярное произведение: {sc}")
    print(f"Норма u1: {norm1}")
    print(f"Норма u2: {norm2}")
    if -1e-6 < sc < 1e-6:  # если скалярное произведение ненулевых
векторов = 0, то
        print("Ортогональные")
    else:
        print("Не ортогональные")

    if (-1e-6 < sc < 1e-6) & (norm1 < 1 + 1e-6) & (norm1 > 1 - 1e-6) &
\
            (norm2 < 1 + 1e-6) & (norm2 > 1 - 1e-6):
        print("Ортонормированный базис\n")
    else:
        print("Не ортонормированный базис\n")

    return norm1, norm2


def linspace(start, end, num):
    step = (end - start) / (num - 1)
    tn = [0] * num
    tmp = start
    for i in range(num):  # 0-1000
        tn[i] = tmp
        tmp = tmp + step
    return tn


def dsp2():
    f1 = 2.5
    f2 = 3
    T = 2
    dt = 1000
    tn = linspace(-T / 2, T / 2, dt)  # массив отсчетов
```

```
    u1 = func(f1, tn)  # вычисляем все значения u1 и u2
    u2 = func(f2, tn)

    fig, axes = plt.subplots(nrows=1, ncols=2)
    axes[0].set(title='f1')
    axes[1].set(title='f2')

    fig.axes[0].grid()  # график исходного сигнала
    fig.axes[0].plot(u1)

    fig.axes[1].grid()  # график исходного сигнала
    fig.axes[1].plot(u2)
    #plt.show()

    norm1, norm2 = calculate(u1, u2, tn)  # вычисляем скалярное
произведение, нормы 1 и 2

    print("Векторы базиса должны быть попарно ортогональными и
единичными:")
    u1 = [i / norm1 for i in u1]
    u2 = [i / norm2 for i in u2]
    calculate(u1, u2, tn)

    print("Удвоим частоты f1 и f2:")
    u1 = [i / norm1 for i in func(2 * f1, tn)]
    u2 = [i / norm2 for i in func(2 * f2, tn)]
    calculate(u1, u2, tn)

    print("Увеличим интервал вдвое:")
    tn = linspace(-T, T, dt)
    u1 = [i / norm1 for i in func(f1, tn)]
    u2 = [i / norm2 for i in func(f2, tn)]
    calculate(u1, u2, tn)

    print("Уменьшим интервал вдвое:")
    tn = linspace(-T / 4, T / 4, dt)
    u1 = [i / norm1 for i in func(f1, tn)]
```

```
    u2 = [i / norm2 for i in func(f2, tn)]
    calculate(u1, u2, tn)


dsp2()


#файл 1_3DSP.py
import math as math
import matplotlib.pyplot as plt


def func(f, t):
    tmp = []
    for i in t:
        tmp.append(math.sin(2 * math.pi * f * i))
    return tmp


def Sinc(x):
    if x == 0:
        wh = 1.0e-20
    else:
        wh = x
    y = math.pi * wh  # если x==0, то 1
    return math.sin(y) / y


def interpolation(un, fn, t):
    T = 1 / fn
    inter = [0 for _ in t]
    for i in range(len(un)):
        for j in range(len(t)):
            x = ((t[j] / T) - i)
            inter[j] += un[i] * Sinc(x)
    return inter
```

```
def arange(start, end, step):
    num = int((end - start) / step)
    tn = [0] * num
    tmp = start
    for i in range(num):
        tn[i] = tmp
        tmp = tmp + step
    return tn


def dsp3():
    f = 100  # частота исходного сигнала
    T = 10 / f  # длительность наблюдения сигнала
    t = arange(0, T, 1 / (100 * f))

    fd = [1.5 * f, 1.75 * f, 2 * f, 3 * f, 1000 * f]  # частоты дескритизации
    td = [arange(0, T, 1 / i) for i in fd]  # периоды дескритизации
    ud = [func(f, td[i]) for i in range(len(fd))]  # выборки отсчетов

    fig, axes = plt.subplots(nrows=2, ncols=3)

    axes[0, 0].set(title='График исходного сигнала')
    axes[0, 1].set(title=f'График {fd[0] / f}F')
    axes[0, 2].set(title=f'График {fd[1] / f}F')
    axes[1, 0].set(title=f'График {fd[2] / f}F')
    axes[1, 1].set(title=f'График {fd[3] / f}F')
    axes[1, 2].set(title=f'График {fd[4] / f}F')

    fig.axes[0].grid()  # график исходного сигнала
    fig.axes[0].set_ylim(-1.5, 1.5)
    fig.axes[0].plot(t, func(f, t))

    for i in range(1, len(fd) + 1):  # графики восстановленных сигналов
        fig.axes[i].grid()
        fig.axes[i].set_ylim(-1.5, 1.5)
        fig.axes[i].plot(t, interpolation(ud[i - 1], fd[i - 1], t))
```

```
    plt.show()


dsp3()


#файл 1_4DSP.py


import numpy as np
import matplotlib.image as mpl_image


def interpolation(un, fn, t):
    T = 1 / fn
    inter = [0 for _ in t]
    for i in range(len(un)):
        inter += un[i] * np.sinc((t / T) - i)
    return inter


def dsp4(m, n, num):
    img = mpl_image.imread("Lenna.bmp", "bmp")
    height, width, channels = img.shape

    new_height = int(height * n)
    new_width = int(width * m)

    new_img1 = np.zeros((height, new_width, channels), np.uint8)
    for i in range(height):
        for k in range(channels):
            new_img1[i, :, k] = interpolation(img[i, :, k], 1 / m,
np.array(range(new_width)))

    new_img2 = np.zeros((new_height, new_width, channels), np.uint8)
    for j in range(new_width):
        for k in range(channels):
```

```
            new_img2[:, j, k] = interpolation(new_img1[:, j, k], 1 /
n, np.array(range(new_height)))

    mpl_image.imsave(f"NewLenna{num}.bmp", new_img2)


dsp4(2, 2, 1)
dsp4(0.5, 0.5, 2)
dsp4(2, 0.5, 3)
dsp4(0.5, 2, 4)
```