МИНОБРНАУКИ РОССИИ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ «ЛЭТИ» ИМ. УЛЬЯНОВА (ЛЕНИНА)

Кафедра математического обеспечения и применения ЭВМ

ОТЧЕТ

по лабораторной работе №2

по дисциплине «Цифровая обработка сигналов»

Тема: Частотный анализ полиномиальных приближений

Студенты гр. 8383		Ларин А.
		Бобенко Н. С.
Преподаватель		Середа А. И.

Санкт-Петербург

2021

Цель работы.

Анализ частотных характеристик известных формул полиномиального сглаживания временных рядов и формул численного интегрирования.

Основные теоретические положения.

В качестве временного ряда рассматривается дискретный сигнал с шагом дискретизации, равным единице.

Под полиномиальным сглаживанием понимается аппроксимация в смысле МНК значений конечного (нечетного) числа элементов сглаживаемого ряда полиномом заданного порядка с присвоением среднему из этих элементов значения сглаживающего полинома в центре выбранного временного отрезка. Такой подход соответствует так называемому сглаживанию в скользящем окне.

В качестве исследуемых формул численного интегрирования используются квадратурные формулы Ньютона-Котеса.

Постановка задачи.

Получить формулы для передаточных функций нерекурсивных фильтров, соответствующих полиномиальному сглаживанию дискретного сигнала для полиномов различного порядка и построить графики $\widetilde{H}(f)$. Проинтерпретировать частотные свойства передаточных функций. Провести сопоставительный анализ частотных характеристик передаточных функций для различных степеней полиномов. Получить формулы для передаточных функций рекурсивных фильтров, соответствующих квадратурным формулам Ньютона-Котеса различного порядка. Проинтерпретировать частотные свойства передаточных функций. Провести сопоставительный анализ частотных характеристик передаточных функций для различных квадратурных формул.

Порядок выполнения работы.

1. Вывести формулы для передаточной функции нерекурсивного фильтра, соответствующего сглаживанию прямой линией по 3, 5, 7 и 9 точкам. Построить графики $\widetilde{H}(f)$. Проинтерпретировать частотные свойства передаточных функций для различного количества точек.
2. Вывести формулы для передаточной функции нерекурсивного фильтра, соответствующего сглаживанию полиномом второй степени по 7, 9, 11 и 13 точкам. Построить графики $\widetilde{H}(f)$. Проинтерпретировать частотные свойства передаточных функций для различного количества точек.
3. Вывести формулы для передаточной функции нерекурсивного фильтра, соответствующего сглаживанию полиномом четвёртой степени по 9, 11, 13 и 15 точкам. Построить графики $\widetilde{H}(f)$. Проинтерпретировать частотные свойства передаточных функций для различного количества точек.
4. Вывести формулы для передаточной функции нерекурсивного фильтра, соответствующего сглаживанию по формулам Спенсера по 13, 15 и 21 точкам. Построить графики $\widetilde{H}(f)$. Проинтерпретировать частотные свойства передаточных функций для различного количества точек.
5. Построить графики из предыдущих пунктов в логарифмической шкале (Дб). Объясните, чем отличаются данные графики от полученных ранее и объясните их смысл.
6. Провести сопоставительный анализ свойств передаточных функций, полученных при выполнении пп. 1-4.
7. Вывести формулы передаточных функций рекурсивных фильтров, соответствующих квадратурным формулам прямоугольников, трапеций и Симпсона. Построить графики передаточных функций и графики отношения вычисляемого в результате фильтрации значения к

истинному. Проинтерпретировать частотные свойства полученных передаточных функций.

8. Вывести формулу передаточной функции рекурсивного фильтра, соответствующего квадратурной формуле:

$$y_{n+2}=y_{n-1}+\frac{1}{8}(x_{n+2}+3x_{n+1}+3x_n+x_{n-1}).$$

9. Построить график передаточной функции и график отношения вычисляемого в результате фильтрации значения к истинному. Проинтерпретировать частотные свойства передаточной функции.
10. Провести сопоставительный анализ частотных характеристик передаточных функций, полученных при выполнении пп. 7 и 8.
11. Сделать выводы.

Выполнение работы.

1. Вывод формул

   Входной сигнал: $s(t)$.

   Выходной сигнал: $y(t)=A+Bt$.

   Приближение (в смысле МНК) прямой линией по 2m+1 точкам:

$$F(A,B)=\sum_{k=-m}^{m}(s_k-y_k)^2=\sum_{k=-m}^{m}(s_k-A-Bk)^2 \Longrightarrow min$$

$$\frac{\delta F(A,B)}{\delta A}=0\Leftrightarrow -2\sum_{k=-m}^{m}(s_k-A-Bk)=0\Leftrightarrow -\sum_{k=-m}^{m}s_k+\sum_{k=-m}^{m}A+\sum_{k=-m}^{m}Bk=0$$

$$A=\frac{1}{2m+1}\sum_{k=-m}^{m}s_k=y_0=\frac{1}{2m+1}(s_{-m}+s_{-m+1}+\ldots+s_{m-1}+s_m)$$

$$y_0=A=\frac{1}{2m+1}\sum_{k=-m}^{m}s_k=\frac{1}{2m+1}(s_{-m}+s_{-m+1}+\ldots+s_{m-1}+s_m)$$

   В общем случае:

$$y_n=\frac{1}{2m+1}\sum_{k=-m}^{m}s_k=\frac{1}{2m+1}(s_{-m+n}+s_{-m+1+n}+\ldots+s_{m-1+n}+s_{m+n})$$

$$s_n=e^{i\omega n}$$

$$y_n=\frac{1}{2m+1}(e^{-mi\omega}+e^{(-m+1)i\omega}+\ldots+e^{(m-1)i\omega}+e^{mi\omega})=H(\omega)$$

$$H(\omega)=\frac{1}{2m+1}(1+2\cos(\omega)+2\cos(2\omega)+\ldots+2\cos(m\omega))$$

$$H(\omega)=\frac{\sin(\frac{(2m+1)\omega}{2})}{(2m+1)\sin(\frac{\omega}{2})}$$

$$H(\omega)=H(2\pi f)=\widetilde{H}(f)$$

Формула для передаточной функции нерекурсивного фильтра, соответствующего сглаживанию прямой линией для m=1 (по 3-м точкам):

$$H_3(f)=\frac{\sin(3\pi f)}{3\sin(\pi f)}$$

По 5-ти точкам:

$$H_5(f)=\frac{\sin(5\pi f)}{5\sin(\pi f)}$$

По 7-ми точкам:

$$H_7(f)=\frac{\sin(7\pi f)}{7\sin(\pi f)}$$

По 9-ти точкам:

$$H_9(f)=\frac{\sin(9\pi f)}{9\sin(\pi f)}$$

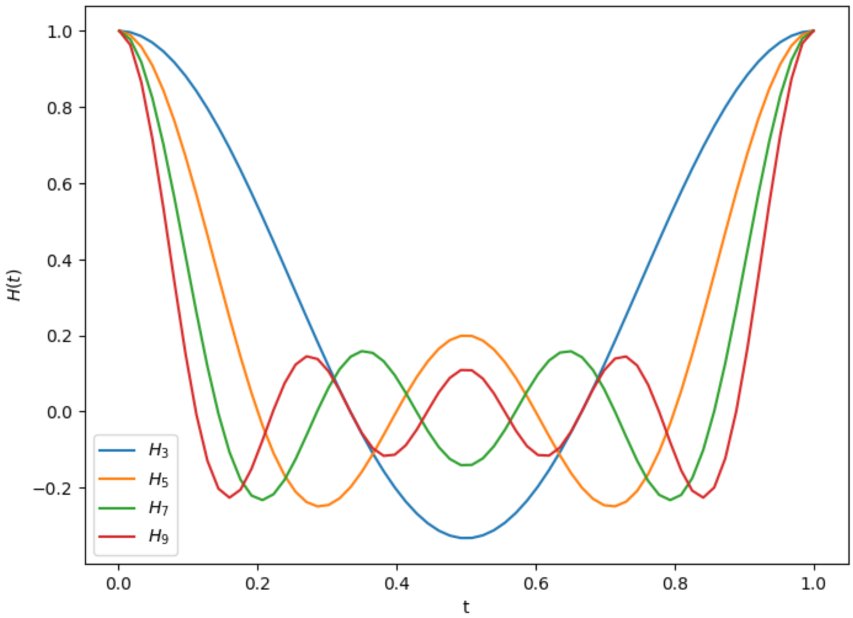

Рисунок 1 – Передаточная функция для сглаживания прямой по МНК

Видно, что количество точек, использованных для аппроксимации равно количеству экстремумов

2. Вывод формул

Входной сигнал: $s(t)$.

Выходной сигнал: $y(t)=A+Bt+Ct^2$.

Приближение (в смысле МНК) полиномом 2-й степени по 2m+1 точкам:

$$F(A,B,C)=\sum_{k=-m}^{m}(s_k-y_k)^2=\sum_{k=-m}^{m}(s_k-A-Bk-Ck^2)^2 \Longrightarrow min$$

Найдем частные пргоизводные по A, C

$$\begin{cases}\dfrac{\delta F(A,B,C)}{\delta A}=0\\ \dfrac{\delta F(A,B,C)}{\delta C}=0\end{cases} \Leftrightarrow \begin{cases}-2\sum_{k=-m}^{m}(s_k-A-Bk-Ck^2)=0\\ -2\sum_{k=-m}^{m}(k^2 s_k-k^2A-Bk^3-Ck^4)=0\end{cases} \Leftrightarrow$$

$$\begin{cases}-\sum_{k=-m}^{m}s_k+\sum_{k=-m}^{m}A+\sum_{k=-m}^{m}Bk+\sum_{k=-m}^{m}Ck^2=0\\ -\sum_{k=-m}^{m}k^2 s_k+\sum_{k=-m}^{m}k^2A+\sum_{k=-m}^{m}Bk^3+\sum_{k=-m}^{m}Ck^4=0\end{cases}$$

Система нормальных уравнений:

$$\begin{cases}(2m+1)A+\dfrac{m(m+1)(2m+1)}{3}C=\sum_{k=-m}^{m}s_k\\ \dfrac{m(m+1)(2m+1)}{3}A+\dfrac{m(m+1)(2m+1)(3m^2+3m-1)}{15}C=\sum_{k=-m}^{m}k^2 s_k\end{cases}$$

Подставим $\dfrac{m(m+1)(2m+1)}{3}C$ во второе уравнение:

$$\frac{m(m+1)(2m+1)}{3}A+\frac{3m^2+3m-1}{5}\left(\sum_{k=-m}^{m}s_k-(2m+1)A\right)=\sum_{k=-m}^{m}k^2 s_k \Longleftrightarrow$$

$$\left(\frac{m(m+1)(2m+1)}{3}-\frac{(3m^2+3m-1)(2m+1)}{5}\right)A=\sum_{k=-m}^{m}k^2 s_k-\frac{3m^2+3m-1}{5}\sum_{k=-m}^{m}s_k$$

Тогда:

$$A=\frac{\sum_{k=-m}^{m}k^2 s_k-\frac{3m^2+3m-1}{5}\sum_{k=-m}^{m}s_k}{\frac{m(m+1)(2m+1)}{3}-\frac{(3m^2+3m-1)(2m+1)}{5}}$$

В итоге получаем:

$$y_0=A=\frac{\sum_{k=-m}^{m}k^2 s_k-\frac{3m^2+3m-1}{5}\sum_{k=-m}^{m}s_k}{\frac{m(m+1)(2m+1)}{3}-\frac{(3m^2+3m-1)(2m+1)}{5}}$$

Для 7 точек:

$$y_7=A=\frac{\sum_{k=-3}^{3}k^2 s_k-7\sum_{k=-3}^{3}s_k}{28-49}=\frac{1}{21}\left(7\sum_{k=-3}^{3}s_k-\sum_{k=-3}^{3}k^2 s_k\right)=$$

$$=\frac{1}{21}(7s_{-3}+7s_{-2}+7s_{-1}+7s_0+7s_1+7s_2+7s_3-9s_{-3}-4s_{-2}-s_1-4s_2-9s_3)=$$

$$=\frac{1}{21}(-2s_{-3}+3s_{-2}++s_{-1}+7s_0+6s_1+3s_2-2s_3)$$

$$y_7=\frac{1}{21}(-2s_{-3}+3s_{-2}++s_{-1}+7s_0+6s_1+3s_2-2s_3)$$

В общем случае:

$$y_n=\frac{1}{21}(-2s_{n-3}+3s_{n-2}++s_{n-1}+7s_n+6s_{n+1}+3s_{n+2}-2s_{n+3})$$

$$s_n=e^{i\omega n}$$

$$y_n=\frac{1}{21}(-2e^{-3i\omega}+3e^{-2i\omega}+6e^{-i\omega}+7+6e^{i\omega}+3e^{2i\omega}-2e^{3i\omega})e^{i\omega n}=H(\omega)e^{i\omega n}$$

$$H(\omega)=\frac{1}{21}(-2e^{-3i\omega}+3e^{-2i\omega}+6e^{-i\omega}+7+6e^{i\omega}+3e^{2i\omega}-2e^{3i\omega})$$

$$H(\omega)=\frac{1}{21}(7+12\cos\omega+6\cos 2\omega-4\cos 3\omega)$$

$$H(\omega)=H(2\pi f)=\widetilde{H}(f)$$

Формула для передаточной функции нерекурсивного фильтра, соответствующего сглаживанию полиномом второй степени для по 7-ми точкам:

$$H_7(f)=\frac{1}{21}(7+12\cos(2\pi f)+6\cos(4\pi f)-4\cos(6\pi f))$$

Аналогично, по 9 точкам:

$$H_9(f)=\frac{1}{231}(59+108\cos(2\pi f)+78\cos(4\pi f)+28\cos(6\pi f)-42\cos(8\pi f))$$

По 11 точкам:

$$H_{11}(f)=\frac{1}{429}(89+168\cos(2\pi f)+138\cos(4\pi f)+88\cos(6\pi f)+18\cos(8\pi f)-72\cos(10\pi f))$$

По 13-ти точкам

$$H_{13}(f)=\frac{1}{143}(25+48\cos(2\pi f)+42\cos(4\pi f)+32\cos(6\pi f)+18\cos(8\pi f)-22\cos(10\pi f))$$

Графики для передаточных функций на интервале $f\in[0;1]$ представлены рис. 2.

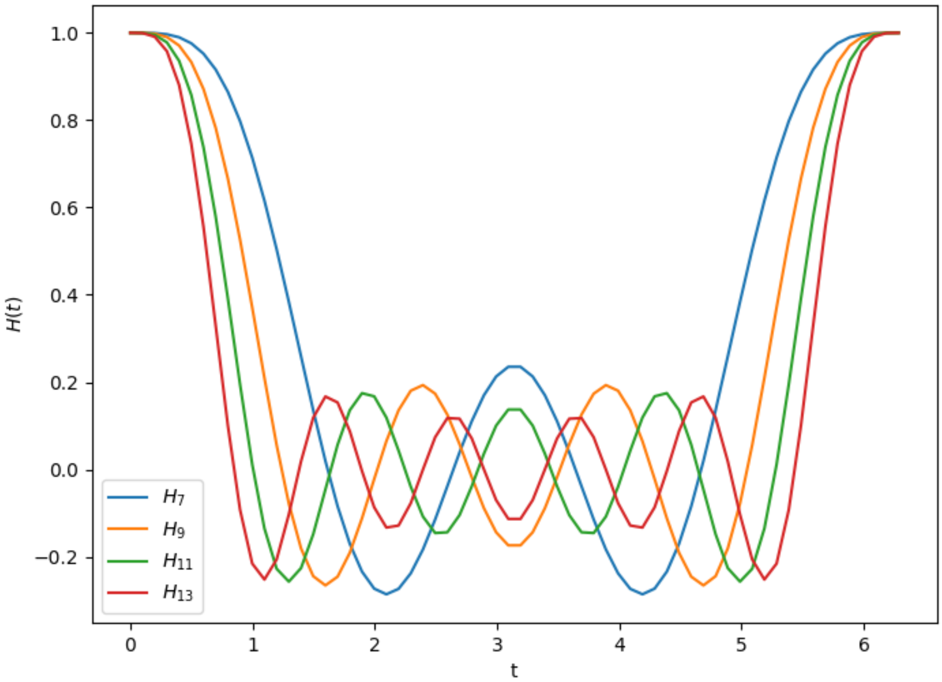

Рисунок 2 – Передаточная функция для сглаживания полиномом 2-й степени по МНК

3. Входной сигнал: $s(t)$.

Выходной сигнал: $y(t)=A+Bt+Ct^2+Dt^3+Et^4$.

Приближение (в смысле МНК) полиномом 4-й степени по 2m+1 точкам:

$$F(A,B,C)=\sum_{k=-m}^{m}(s_k-y_k)^2=\sum_{k=-m}^{m}(s_k-A-Bk-Ck^2+Dk^3+Ek^4)^2\Longrightarrow min$$

Посчитаем частные производные по A, C и E, получим:

$$\begin{cases} \dfrac{\delta F(A,B,C,D,E)}{\delta A}=0 \\ \dfrac{\delta F(A,B,C,D,E)}{\delta C}=0 \\ \dfrac{\delta F(A,B,C,D,E)}{\delta E}=0 \end{cases} \Leftrightarrow \begin{cases} -2\sum_{k=-m}^{m}(s_k-A-Bk-Ck^2-Dk^3-Ek^4)=0 \\ -2\sum_{k=-m}^{m}\left(k^2 s_k-k^2A-Bk^3-Ck^4-Dk^5-Ek^6\right)=0 \\ -2\sum_{k=-m}^{m}\left(k^4 s_k-k^4A-Bk^5-Ck^6-Dk^7-Ek^8\right)=0 \end{cases} \Leftrightarrow$$

$$\Leftrightarrow \begin{cases} -\sum_{k=-m}^{m} s_k+\sum_{k=-m}^{m} A+\sum_{k=-m}^{m} Bk+\sum_{k=-m}^{m} Ck^2+\sum_{k=-m}^{m} Dk^3+\sum_{k=-m}^{m} Ek^4=0 \\ -\sum_{k=-m}^{m} k^2 s_k+\sum_{k=-m}^{m} k^2A+\sum_{k=-m}^{m} Bk^3+\sum_{k=-m}^{m} Ck^4+\sum_{k=-m}^{m} Dk^5+\sum_{k=-m}^{m} Ek^6=0 \\ -\sum_{k=-m}^{m} k^4 s_k+\sum_{k=-m}^{m} k^4A+\sum_{k=-m}^{m} Bk^5+\sum_{k=-m}^{m} Ck^6+\sum_{k=-m}^{m} Dk^7+\sum_{k=-m}^{m} Ek^8=0 \end{cases}$$

$$\begin{cases} -\sum_{k=-m}^{m} s_k+\sum_{k=-m}^{m} A+\sum_{k=-m}^{m} Ck^2+\sum_{k=-m}^{m} Ek^4=0 \\ -\sum_{k=-m}^{m} k^2 s_k+\sum_{k=-m}^{m} k^2A+\sum_{k=-m}^{m} Ck^4+\sum_{k=-m}^{m} Ek^6=0 \\ -\sum_{k=-m}^{m} k^4 s_k+\sum_{k=-m}^{m} k^4A+\sum_{k=-m}^{m} Ck^6+\sum_{k=-m}^{m} Ek^8=0 \end{cases}$$

Система нормальных уравнений:

$$\begin{cases} (2m+1)A+\dfrac{m(m+1)(2m+1)}{3}C+ \\ +\dfrac{m(m+1)(2m+1)(3m^2+3m-1)}{15}F=\sum_{k=-m}^{m} s_k \\ \dfrac{m(m+1)(2m+1)}{3}A+\dfrac{m(m+1)(2m+1)(3m^2+3m-1)}{15}C+ \\ +\dfrac{m(m+1)(2m+1)(1-3m+6m^3+3m^4)}{21}F=\sum_{k=-m}^{m} k^2 s_k \\ \dfrac{m(m+1)(2m+1)(3m^2+3m-1)}{15}A+\dfrac{m(m+1)(2m+1)(1-3m+6m^3+3m^4)}{21}C+ \\ +\dfrac{m(m+1)(2m+1)(-3+9m-m^2-15m^3+5m^4+15m^5+5m^6)}{45}=\sum_{k=-m}^{m} k^4 s_k \end{cases}$$

Выразим C из первого уравнения системы нормальных уравнении:

$$C=\frac{-15A-30Am+Em-10Em^3-15Em^4-6Em^5-15\sum_{k=-m}^{m} s_k}{5m(m+1)(2m+1)}$$

Подставим теперь C во второе и третье уравнение:

$$\begin{cases} \dfrac{\left(-3-2m+12m^2+8m^3\right)\left(-35A+3Em\left(-2-m+2m^2+m^3\right)\right)}{525}+ \\ +\ \dfrac{3m^2+3m-1}{5}\sum_{k=-m}^{m} s_k=\sum_{k=-m}^{m} k^2 s_k \\ \dfrac{-\left(-3-2m+12m^2+8m^3\right)}{315}* \\ *\left(3A\left(6m^2+6m-5\right)+Em\left(-6+m+12m^2+m^3-6m^4-2m^5\right)\right)+ \\ +\ \dfrac{1-3m+6m^3+3m^4}{7}\sum_{k=-m}^{m} s_k=\sum_{k=-m}^{m} k^4 s_k \end{cases}$$

Выразим из 2 уравнения E и подставим в 3 уравнение:

$$\begin{cases} E=35\cdot\dfrac{A\left(-3-2m+12m^2+8m^3\right)+\left(3-9m-9m^2\right)\sum_{k=-m}^{m} s_k+15\sum_{k=-m}^{m} k^2 s_k}{3m\left(-2-m+2m^2+m^3\right)\left(-3-2m+12m^2+8m^3\right)} \\ 4A\left(45+18m-200m^2-80m^3+80m^4+32m^5\right)- \\ -15\cdot\left(12-50m-35m^2+30m^3+15m^4\right)\sum_{k=-m}^{m} s_k+ \\ +525\cdot\left(-3+2m+2m^2\right)\sum_{k=-m}^{m} k^2 s_k=945\sum_{k=-m}^{m} k^4 s_k \end{cases}$$

И выразим из 3 уравнения A:

$$A=15\left(\frac{\left(12+5m(1+m)(-10+3m(1+m))\right)\sum_{k=-m}^{m} s_k}{4(-3+2m)(-1+2m)(1+2m)(3+2m)(5+2m)}-\right.$$
$$\left.-\frac{35\left(-3+2m(1+m)\right)\sum_{k=-m}^{m} k^2 s_k-63\sum_{k=-m}^{m} k^4 s_k}{4(-3+2m)(-1+2m)(1+2m)(3+2m)(5+2m)}\right)$$

В итоге получаем:

$$y_0=A=15\left(\frac{\left(12+5m(1+m)(-10+3m(1+m))\right)\sum_{k=-m}^{m} s_k}{4(-3+2m)(-1+2m)(1+2m)(3+2m)(5+2m)}-\right.$$
$$\left.-\frac{35\left(-3+2m(1+m)\right)\sum_{k=-m}^{m} k^2 s_k-63\sum_{k=-m}^{m} k^4 s_k}{4(-3+2m)(-1+2m)(1+2m)(3+2m)(5+2m)}\right)$$

Для 9 точек:

$$y_9=\frac{1}{429}\left(179\sum_{k=-4}^{4}s_k-\frac{1}{4}\left(185\sum_{k=-4}^{4}k^2s_k-9\sum_{k=-4}^{4}k^4s_k\right)\right)$$

$$y_9=\frac{1}{429}(15s_{-4}-55s_{-3}+30s_{-2}+135s_{-1}+179s_0+135s_1+30s_2-55s_3+15s_4)$$

В общем случае:

$$y_n=\frac{1}{429}(15s_{n-4}-55s_{n-3}+30s_{n-2}+135s_{n-1}+179s_n+135s_{n+1}+30s_{n+2}-55s_{n+3}+15s_{n+4})$$

$$s_n=e^{i\omega n}$$

$$y_n=\frac{1}{429}(15e^{-4i\omega}-55e^{-3i\omega}+30e^{-2i\omega}+135e^{-i\omega}+179+135e^{i\omega}+30e^{2i\omega}-55e^{3i\omega}+15e^{4i\omega})e^{i\omega n}=$$

$$=H(\omega)e^{i\omega n}$$

$$H(\omega)=\frac{1}{429}(15e^{-4i\omega}-55e^{-3i\omega}+30e^{-2i\omega}+135e^{-i\omega}+179+135e^{i\omega}+30e^{2i\omega}-55e^{3i\omega}+15e^{4i\omega})$$

$$H(\omega)=\frac{1}{429}(179+270\cos\omega+60\cos2\omega-110\cos3\omega+30\cos4\omega)$$

$$H(\omega)=H(2\pi f)=\widetilde{H}(f)$$

Формула для передаточной функции нерекурсивного фильтра, соответствующего сглаживанию полиномом четвертой степени по 9 точкам:

$$H_9(f)=\frac{1}{429}(179+270\cos(2\pi f)+60\cos(4\pi f)-110\cos(6\pi f)+30\cos(8\pi f))$$

Аналогично по 11-ти точкам

$$H_{11}(f)=\frac{1}{429}(143+240\cos(2\pi f)+120\cos(4\pi f)-20\cos(6\pi f)-90\cos(8\pi f)+36\cos(10\pi f))$$

По 13 точкам:

$$H_{13}(f)=\frac{1}{2431}(677+1200\cos(2\pi f)+780\cos(4\pi f)+220\cos(6\pi f)-$$

$$-270\cos(8\pi f)-396\cos(10\pi f)+220\cos(12\pi f))$$

По 15 точкам:

$$H_{15}(f)=\frac{1}{46189}(11\,063+20\,250\cos(2\pi f)+15\,000\cos(4\pi f)+7\,510\cos(6\pi f)-$$

$$-330\cos(8\pi f)-5\,874\cos(10\pi f)+5\,720\cos(12\pi f)+4\,290\cos(14\pi f))$$

Графики для передаточных функций на интервале $f\in[0;1]$ представлены рис. 3.

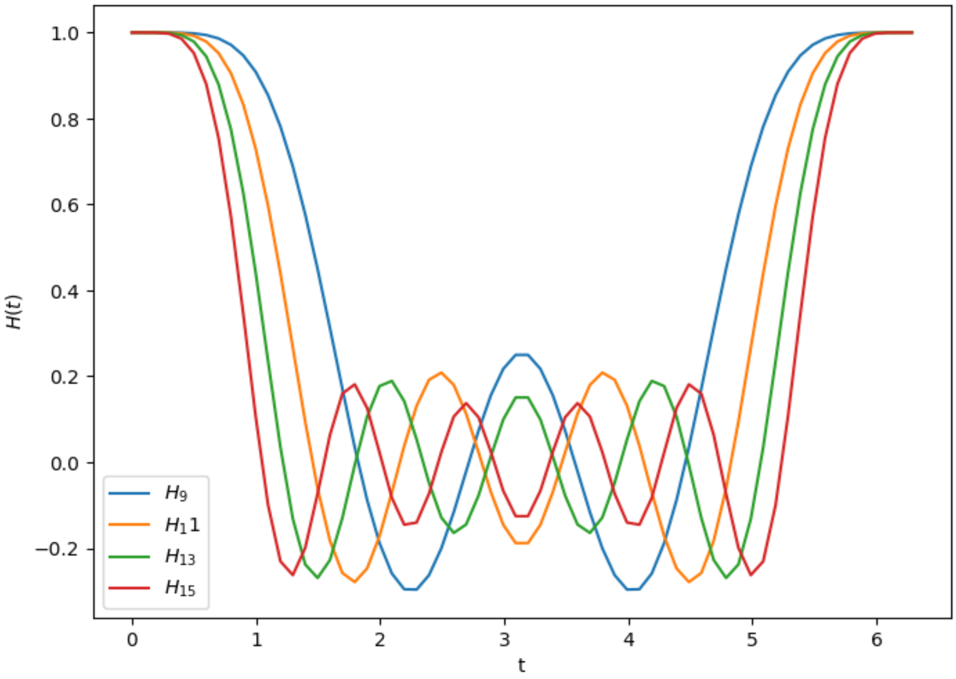

Рисунок 3 – Графики передаточной функции при сглаживании полиномом 4-й степени по МНК

4. Графики для передаточных функций на интервале $f\epsilon[0;1]$ представлены рис. 4.

$$y_n=\frac{1}{320}\left(-3\,s_{n-7}-6\,s_{n-6}-5\,s_{n-5}+3\,s_{n-4}+21\,s_{n-3}+46\,s_{n-2}-67\,s_{n-1}+74\,s_n+\right.$$

$$\left.+67\,s_{n+1}+46\,s_{n+2}+21\,s_{n+3}+3\,s_{n+4}-5\,s_{n+5}-6\,s_{n+6}-3\,s_{n+7}\right)$$

$$y_n=\frac{1}{350}\left(-s_{n-10}-3\,s_{n-6}-5\,s_{n-8}-5\,s_{n-7}-2\,s_{n-6}+6\,s_{n-5}+18\,s_{n-4}+33\,s_{n-3}+47\,s_{n-2}+\right.$$

$$\left.+57\,s_{n-1}+60\,s_n+57\,s_{n+1}+47\,s_{n+2}+33\,s_{n+3}+18\,s_{n+4}+6\,s_{n+5}-2\,s_{n+6}-5\,s_{n+7}-5\,s_{n+8}-3\,s_{n+9}-s_{n+10}\right)$$

Соответствующие передаточные функции:

$$H_{15}(f)=\frac{1}{320}\left(74+134\cos(2\pi f)+92\cos(4\pi f)+42\cos(6\pi f)+6\cos(8\pi f)-\right.$$

$$\left.-10\cos(10\pi f)-12\cos(12\pi f)-6\cos(14\pi f)\right)$$

$$H_{21}(f)=\frac{1}{350}\left(60+114\cos(2\pi f)+94\cos(4\pi f)+66\cos(6\pi f)+36\cos(8\pi f)+12\cos(10\pi f)-\right.$$

$$\left.-4\cos(12\pi f)-10\cos(14\pi f)-10\cos(16\pi f)-6\cos(18\pi f)-2\cos(20\pi f)\right)$$

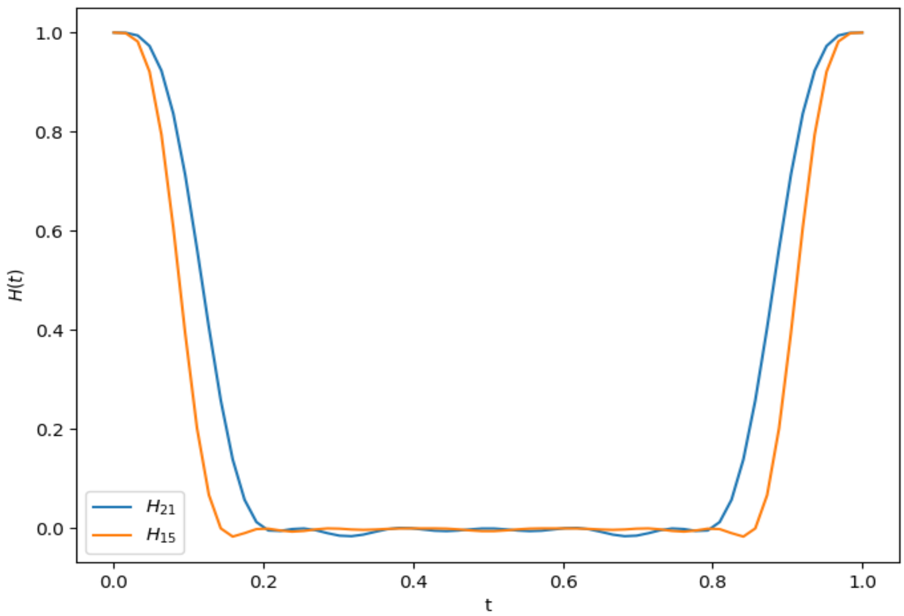

Рисунок 4 – Графики передаточной функции при сглаживании по формулам Спенсера по 15, 21 точкам

5. Построим логарифмический график (в Дб) значений, чтобы различать разницу при малых колебаниях

$$\text{Значение в Дб} = 20 \lg \left(\frac{|y_n|}{|s_n|} \right)$$

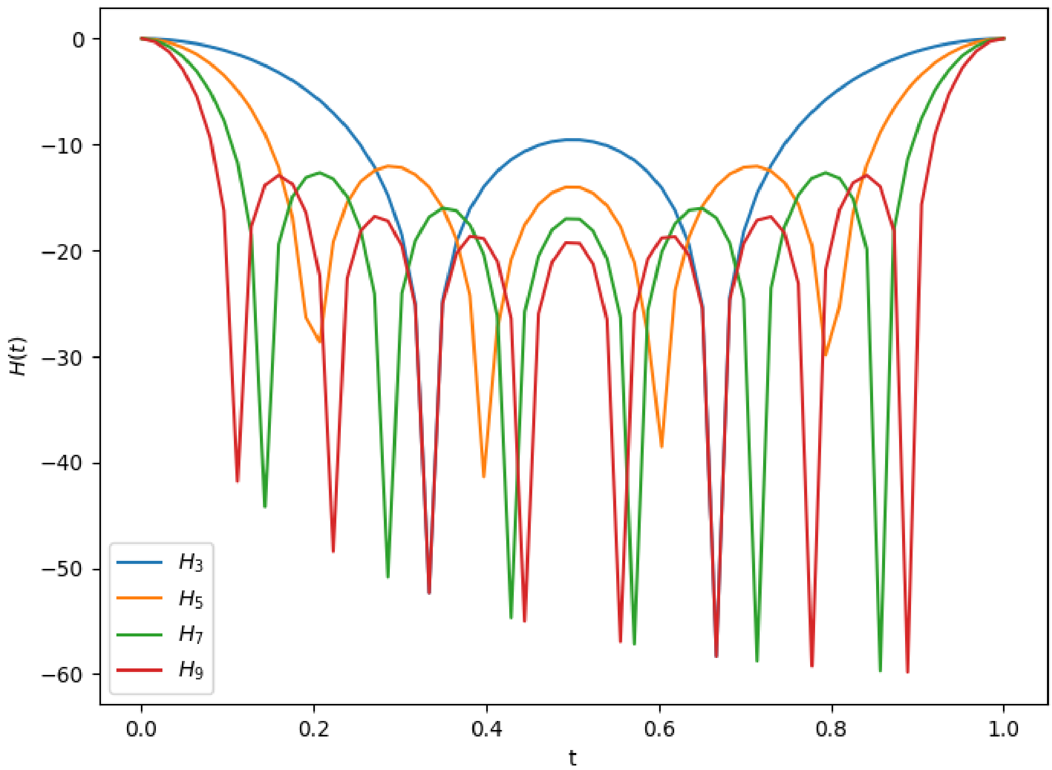

Рисунок 5.1 – Сглаживание прямой по методу наименьших квадратов в логарифмической шкале

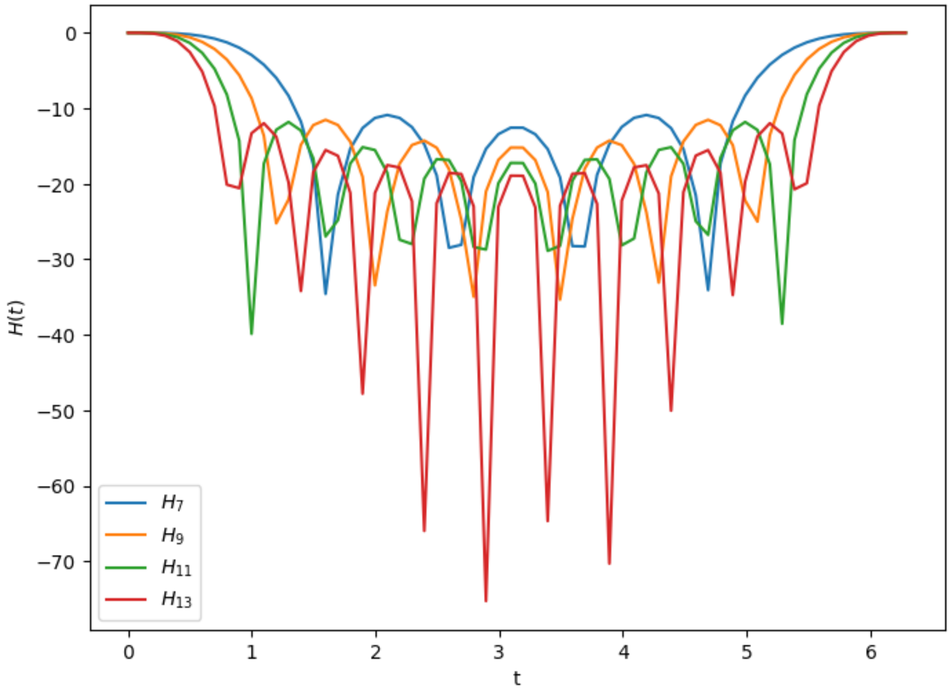

Рисунок 5.2 – Сглаживание полиномом 2-й степени по методу наименьших квадратов в логарифмической шкале.

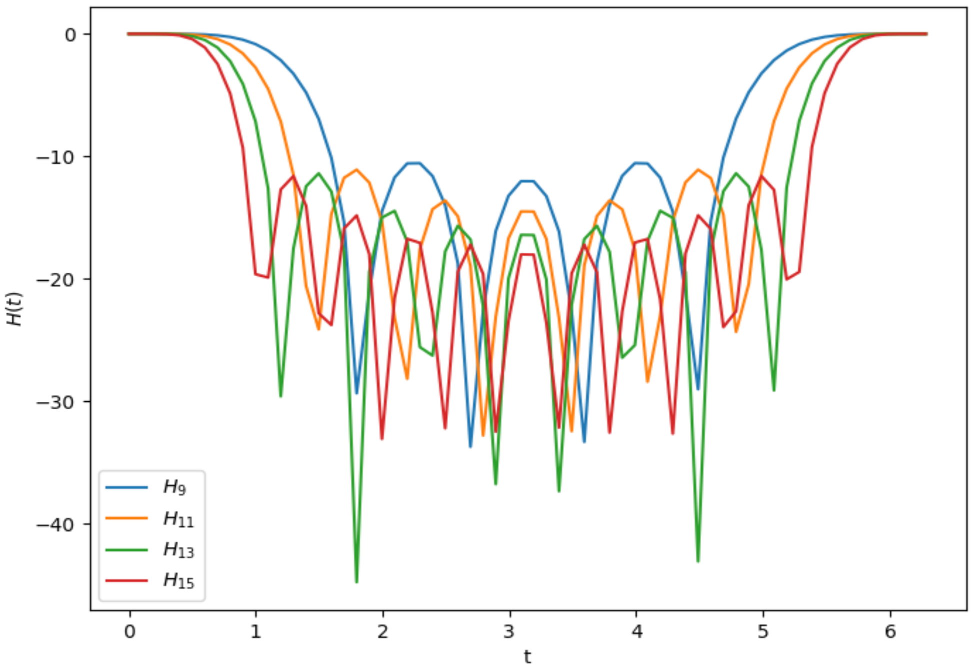

Рисунок 5.3 – Сглаживание полиномом 4-й степени по методу наименьших квадратов в логарифмической шкале.

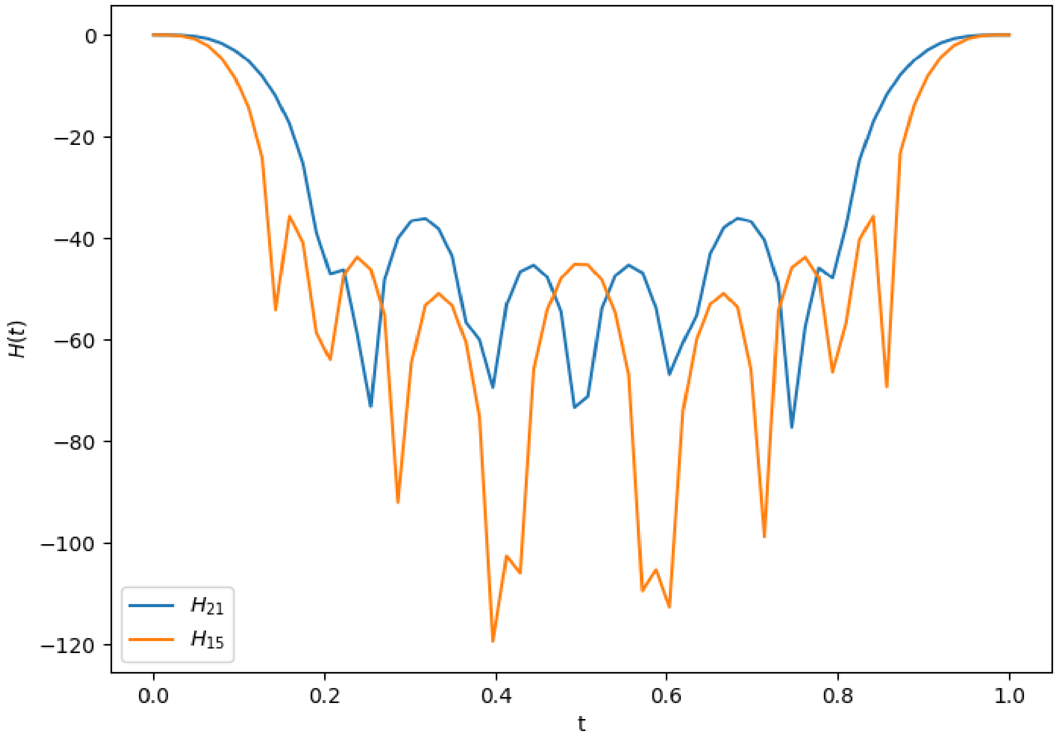

Рисунок 5.4 – Сглаживание по формулам Спенсера в логарифмической шкале.

Логарифмические шкалы позволяют лучше различать малые значения, или порядок больших, что позволяет анализировать экстремумы графиков, иначе выглядящих почти прямыми

6. При сравнении передаточных функций видно, что фильтры устраняют высокочастотных шум, из предположения, что сигнал является более низкочастотным. Разные методы сохраняют сигнал и шум с разной точностью

7. Выведем формулы передаточных функций рекурсивных фильтров, соответствующих квадратурным формулам прямоугольников, трапеций и Симпсона. Построим графики передаточных функций и графики отношения вычисляемого в результате фильтрации значения к истинному. Проинтерпретируем частотные свойства полученных передаточных функций.

   Формула прямоугольников:

   $y_{n+1} = y_n + s_{n+\frac{1}{2}}, y_0 = 0$

Пусть $s_n = e^{i\omega n}$ и $y_n = H(\omega) e^{i\omega n}$, тогда:

$$\begin{cases} y_{n+1} = H(\omega) e^{i\omega n} + e^{i\omega\left(n+\frac{1}{2}\right)} \\ y_{n+1} = H(\omega) e^{i\omega(n+1)} \end{cases}$$

$$H(\omega)\left(e^{i\omega n} e^{i\omega}\right) = H(\omega) e^{i\omega n} + e^{i\omega n} e^{\frac{1}{2} i\omega}$$

$$H(\omega)\left(e^{i\omega n} e^{i\omega} - e^{i\omega n}\right) = e^{i\omega n} e^{\frac{1}{2} i\omega}$$

$$H(\omega)\left(e^{i\omega} - 1\right) = e^{\frac{1}{2} i\omega}$$

$$H(\omega) = \frac{1}{e^{\frac{i\omega}{2}} - e^{\frac{-i\omega}{2}}} = \frac{1}{2 i sin\frac{\omega}{2}}$$

$$\widetilde{H}(f) = \frac{1}{2 i sin(\pi f)}$$

Точное значение интеграла $e^{i\omega t}$ равно $\frac{e^{i\omega t}}{i\omega}$, тогда отношение значений:

$$\gamma = \frac{\textit{Вычисленное}}{\textit{Точное}} = \frac{i\omega}{2 i sin\frac{\omega}{2}} = \frac{\frac{\omega}{2}}{\sin\frac{\omega}{2}} = 1 + \frac{x^2}{24} + \frac{7x^4}{5760} + \dots$$

$$\gamma = \frac{\pi f}{\sin(\pi f)} = 1 + \frac{\pi^2 f^2}{6} + \frac{7\pi^4 f^4}{360} + \dots$$

Формула трапеций:

$$y_{n+1} = y_n + \frac{1}{2}(s_n + s_{n+1}), y_0 = 0$$

Пусть $s_n = e^{i\omega n}$ и $y_n = H(\omega) e^{i\omega n}$, тогда:

$$\begin{cases} y_{n+1} = H(\omega) e^{i\omega n} + \frac{e^{i\omega n} + e^{i\omega(n+1)}}{2} \\ y_{n+1} = H(\omega) e^{i\omega(n+1)} \end{cases}$$

$$H(\omega)\left(e^{i\omega n} e^{i\omega}\right) = H(\omega) e^{i\omega n} + e^{i\omega n}\frac{1 + e^{i\omega}}{2}$$

$$H(\omega)\left(e^{i\omega} - 1\right) = \frac{1 + e^{i\omega}}{2}$$

$$H(\omega) = \frac{1 + e^{i\omega}}{2\left(e^{i\omega} - 1\right)} = \frac{\cos\frac{\omega}{2}}{2 i sin\frac{\omega}{2}}$$

$$\widetilde{H}(f)=\frac{\cos(\pi f)}{2\,isin(\pi f)}$$

Точное значение интеграла $e^{i\omega t}$ равно $\frac{e^{i\omega t}}{i\omega}$, тогда отношение значений:

$$\gamma=\frac{\textit{Вычисленное}}{\textit{Точное}}=\cos\frac{\omega}{2}\frac{\frac{\omega}{2}}{\sin\frac{\omega}{2}}=1-\frac{\omega^2}{12}+\frac{\omega^4}{720}+\ldots$$

$$\gamma=\cos(\pi f)\frac{\pi f}{\sin(\pi f)}=1-\frac{\pi^2\omega^2}{3}+\frac{\pi^4\omega^4}{45}+\ldots$$

Формула Симпсона:

$$y_{n+1}=y_{n-1}+\frac{1}{3}(s_{n-1}+4s_n+s_{n+1}),\,y_0=0$$

Пусть $s_n=e^{i\omega n}$ и $y_n=H(\omega)e^{i\omega n}$, тогда:

$$\begin{cases} y_{n+1}=H(\omega)e^{i\omega(n-1)}+\frac{e^{i\omega(n-1)}+4e^{i\omega n}+e^{i\omega(n+1)}}{3} \\ y_{n+1}=H(\omega)e^{i\omega(n+1)} \end{cases}$$

$$H(\omega)\left(e^{i\omega n}e^{i\omega}\right)=H(\omega)e^{i\omega n}e^{-i\omega}+e^{i\omega n}\frac{e^{-i\omega}+4+e^{i\omega}}{3}$$

$$H(\omega)\left(e^{i\omega}-e^{-i\omega}\right)=\frac{e^{-i\omega}+4+e^{i\omega}}{3}$$

$$H(\omega)=\frac{e^{-i\omega}+4+e^{i\omega}}{3(e^{i\omega}-e^{-i\omega})}=\frac{\cos\omega+2}{3\,isin\,\omega}$$

$$\widetilde{H}(f)=\frac{\cos(2\pi f)+2}{3isin(2\pi f)}$$

Точное значение интеграла $e^{i\omega t}$ равно $\frac{e^{i\omega t}}{i\omega}$, тогда отношение значений:

$$\gamma=\frac{\textit{Вычисленное}}{\textit{Точное}}=\frac{(\cos\omega+2)\,i\omega}{3\,isin\,\omega}=\frac{\cos\omega+2}{3}\cdot\frac{\omega}{\sin\omega}=1+\frac{\omega^4}{180}+\ldots$$

$$\gamma=\frac{\cos(2\pi f)+2}{3}\cdot\frac{2\pi f}{\sin(2\pi f)}=1+\frac{4\pi^4 f^4}{45}+\ldots$$

Графики передаточных функций и графики отношения вычисляемого в результате фильтрации значения к истинному для формул прямоугольников, трапеций и Симпсона представлены на рис. 7.1,7.2,7.3 соответственно. Сравнение на рис. 7.4

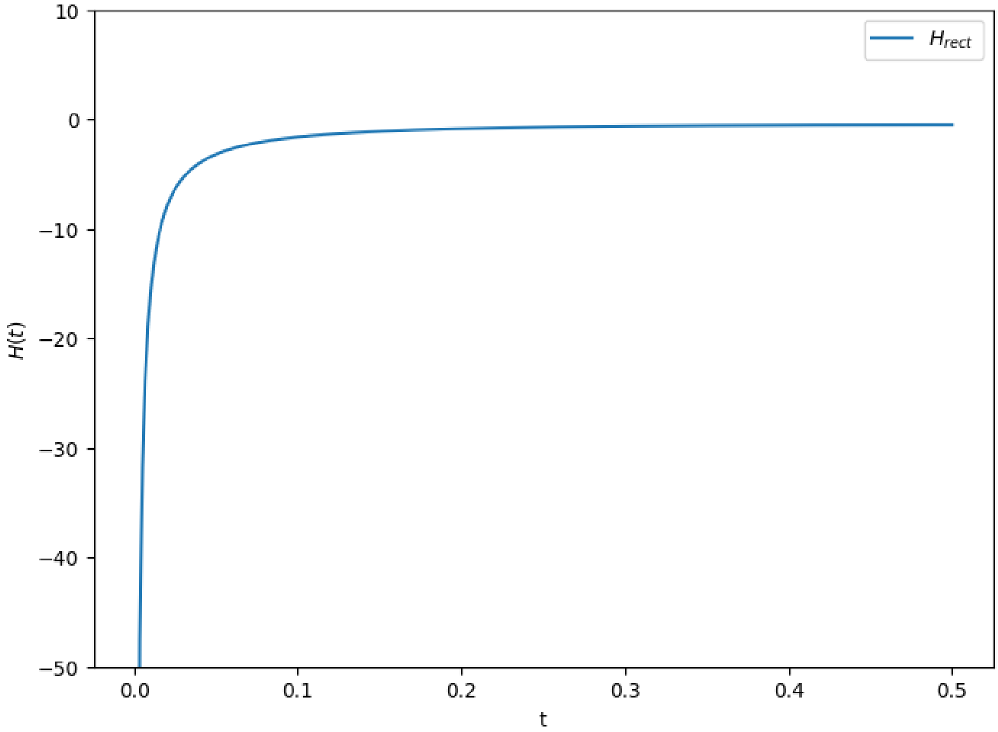

Рисунок 7.1 – График отношения вычисляемого в результате фильтрации значения к истинному для формул прямоугольников

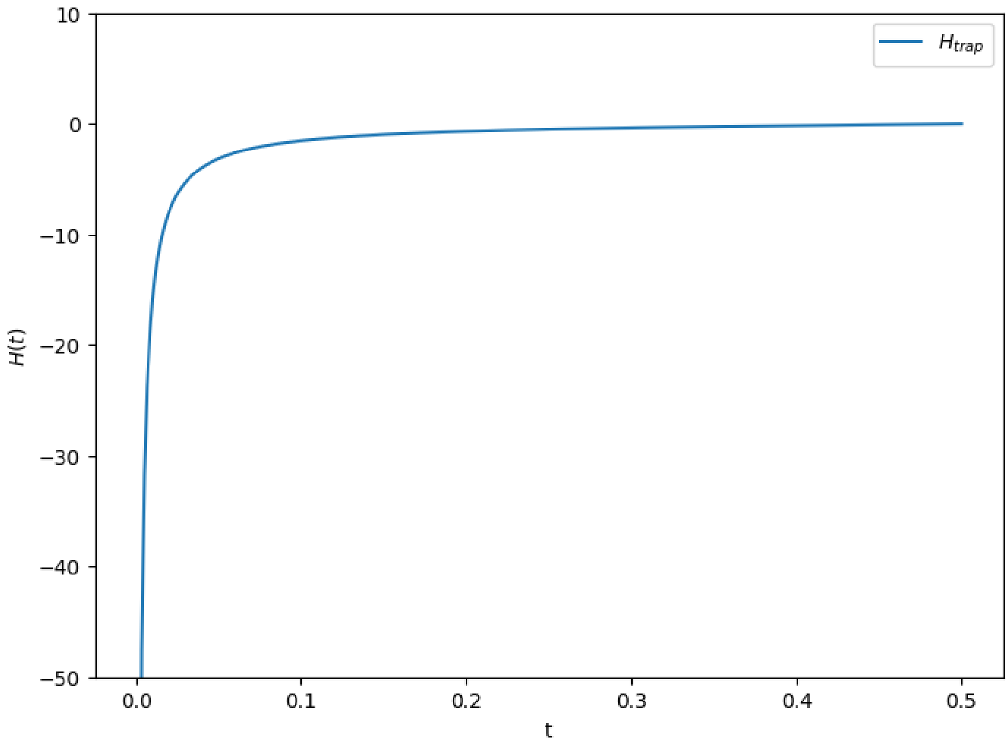

Рисунок 7.2 – График отношения вычисляемого в результате фильтрации значения, к истинному(b) для формул трапеций

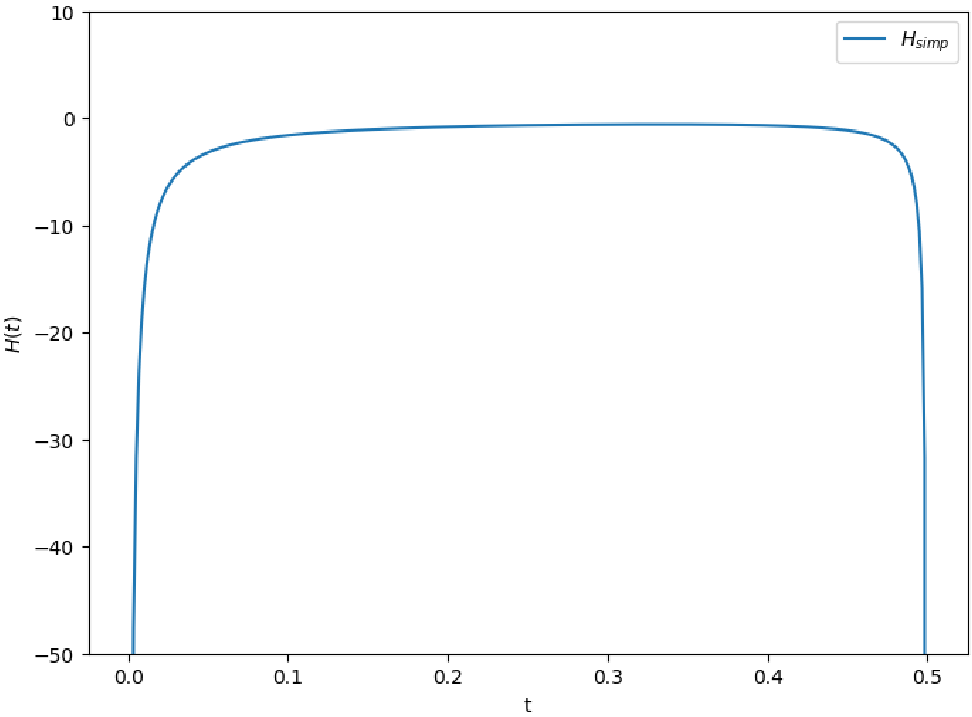

Рисунок 7.3 – График отношения вычисляемого в результате фильтрации значения к истинному(b) для формул Симпсона

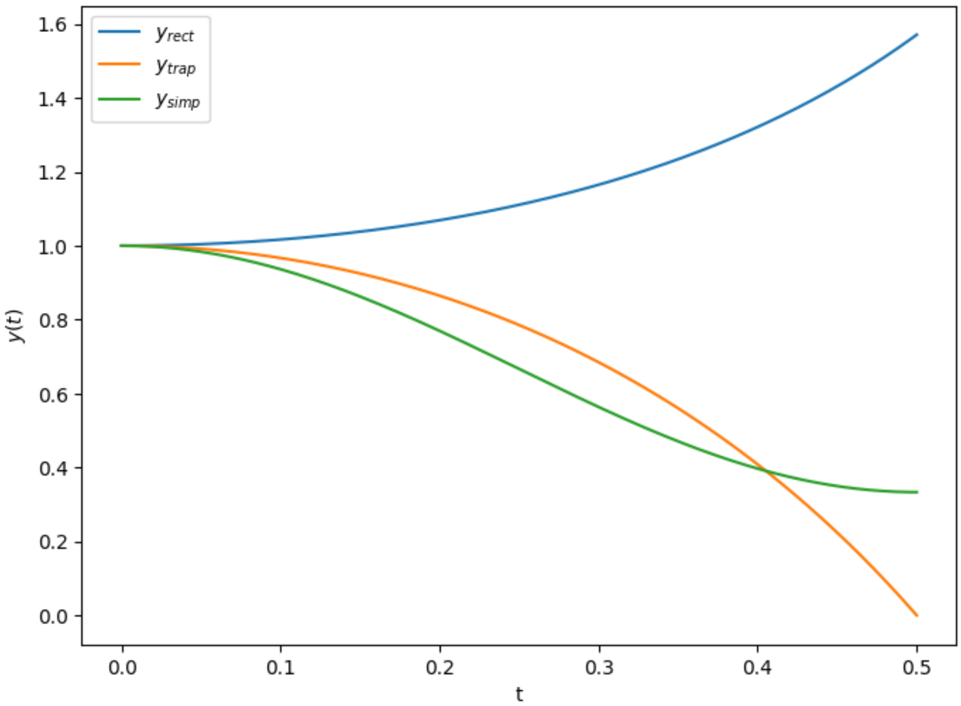

Рисунок 7.4 – Графики отношения вычисляемого в результате фильтрации значения к истинному(b) для формул прямоугольников, для формул трапеций и для формул Симпсона

8. Выведем формулу передаточной функции рекурсивного фильтра, соответствующего квадратурной формуле. Построим график передаточной функции и график отношения вычисляемого в результате фильтрации значения к истинному. Проинтерпретируем частотные свойства передаточной функции.

$y_{n+2} = y_{n-1} + \frac{1}{8}(s_{n+2} + 3s_{n+1} + 3s_n + s_{n-1})$

Пусть $s_n = e^{i\omega n}$ и тогда $H(\omega)e^{i\omega n}$, тогда:

$$\begin{cases} y_{n+2} = H(\omega)e^{i\omega(n-1)} + \dfrac{e^{i\omega(n+2)} + 3e^{i\omega(n+1)} + 3e^{i\omega n} + e^{i\omega(n-1)}}{8} \\ y_{n+2} = H(\omega)e^{i\omega(n+2)} \end{cases}$$

$$H(\omega)(e^{i\omega n}e^{2i\omega}) = H(\omega)e^{i\omega n}e^{-i\omega} + e^{i\omega n}\frac{e^{2i\omega} + 3e^{i\omega} + 3 + e^{-i\omega}}{8}$$

$$H(\omega)(e^{2i\omega} - e^{-i\omega}) = \frac{e^{2i\omega} + 3e^{i\omega} + 3 + e^{-i\omega}}{8}$$

$$H(\omega) = \frac{e^{2i\omega} + 3e^{i\omega} + 3 + e^{-i\omega}}{8(e^{2i\omega} - e^{-i\omega})} \cdot \frac{e^{\frac{-i\omega}{2}}}{e^{\frac{-i\omega}{2}}}$$

$$H(\omega) = \frac{e^{\frac{3i\omega}{2}} + 3e^{\frac{i\omega}{2}} + 3e^{\frac{-i\omega}{2}} + e^{\frac{-3i\omega}{2}}}{8(e^{\frac{3i\omega}{2}} - e^{\frac{-3i\omega}{2}})} = \frac{2\cos\frac{3\omega}{2} + 6\cos\frac{\omega}{2}}{16 i sin\frac{3\omega}{2}}$$

$$\widetilde{H}(f) = \frac{\cos(3\pi f) + 3\cos(\pi f)}{8 i sin(3\pi f)}$$

Точное значение интеграла $e^{i\omega t}$ равно $\frac{e^{i\omega t}}{i\omega}$, тогда отношение значений:

$$\gamma = \frac{Вычисленное}{Точное} = \omega\frac{\cos\frac{3\omega}{2} + 3\cos\frac{\omega}{2}}{8\sin\frac{3\omega}{2}} = \frac{1}{12}\left(\cos\frac{3\omega}{2} + 3\cos\frac{\omega}{2}\right)\cdot\frac{\frac{3\omega}{2}}{\sin\frac{3\omega}{2}} = \frac{1}{3} + \frac{\omega^4}{240} + \dots$$

$$\gamma = \frac{1}{12}(\cos(3\pi f) + 3\cos(\pi f))\frac{3\pi f}{\sin(3\pi f)} = \frac{1}{3} + \frac{\pi^4 f^4}{15} + \dots$$

9. Графики передаточных функций и график отношения вычисляемого в результате фильтрации значения к истинному представлены на рис. 9.1, 9.2 соответственно.

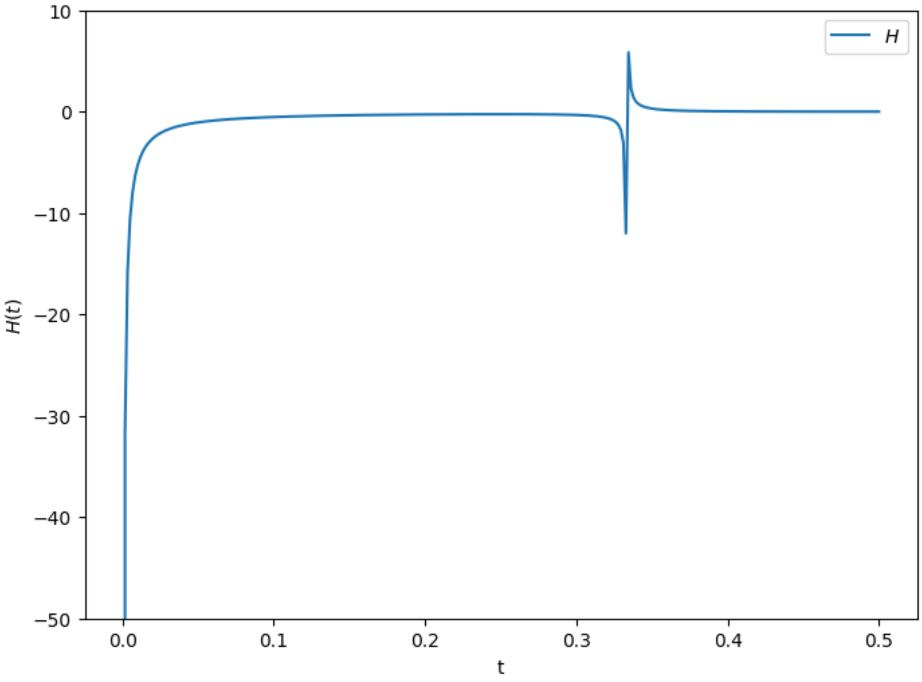

Рисунок 9.1 – График передаточной функции

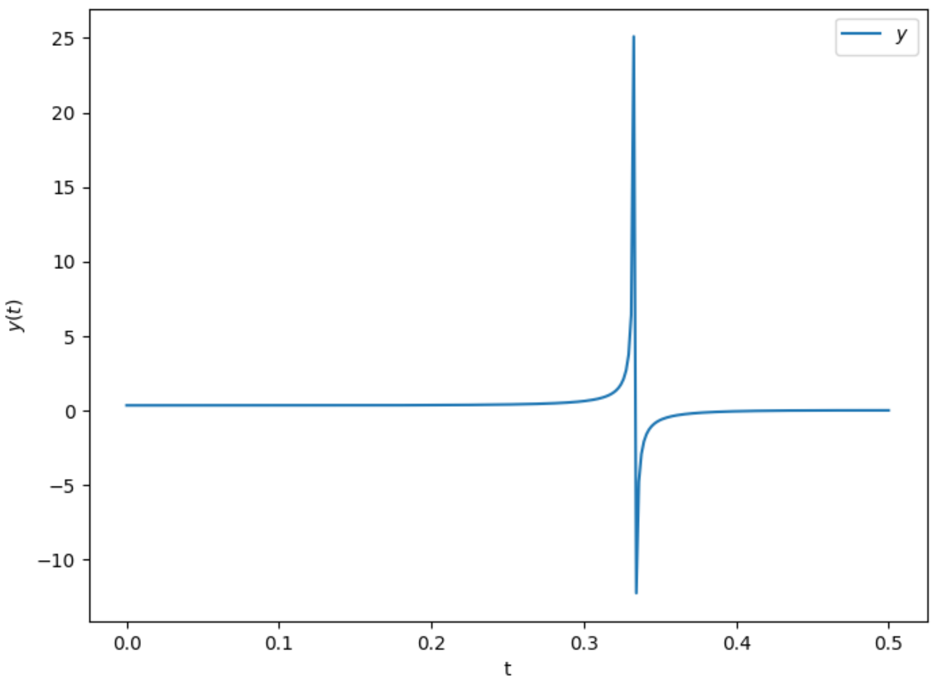

Рисунок 9.2 – График отношения вычисляемого в результате фильтрации значения к истинному

График отношения придерживается единицы, после чего делает скачёк с разрывом второго рода.

10. Были получены графики передаточных функций рекурсивных фильтров, соответствующих квадратурным формулам

прямоугольников, трапеций, Симпсона, а также квадратурной формуле, полученной в п.8.

У формулы прямоугольников значение отношения значения после фильтрации к истиному возрастает, в отличие от формул трапеций и Симпсона.

Выводы.

Были выведены и исследованы формулы передаточных функций фильтров по формулам апроксимации. Были проанализированы отношения значений после фильтра к истиным, построены графики. Проанализированы частотные характеристики передаточных функций.

ПРИЛОЖЕНИЕ А
ЛИСТИНГ

```
#%%

import math
import matplotlib.pyplot as plt
import numpy as np

DPI = 100

#%%

def task1():
    h_3 = lambda x: np.sin(3 * math.pi * x) / (3 * np.sin(x * math.pi))
    h_5 = lambda x: np.sin(5 * math.pi * x) / (5 * np.sin(x * math.pi))
    h_7 = lambda x: np.sin(7 * math.pi * x) / (7 * np.sin(x * math.pi))
    h_9 = lambda x: np.sin(9 * math.pi * x) / (9 * np.sin(x * math.pi))

    t = np.linspace(0.001, 1.0, 64)
    plt.figure(figsize=(8, 6), dpi=DPI)
    plt.plot(t, h_3(t), label=r'$H_3$')
    plt.plot(t, h_5(t), label=r'$H_5$')
    plt.plot(t, h_7(t), label=r'$H_7$')
    plt.plot(t, h_9(t), label=r'$H_9$')
    plt.xlabel('t')
    plt.ylabel(r'$H(t)$')
    plt.legend()
    plt.show()


def task2():
    h_7 = lambda x: 1 / 21 * (7 + 12 * np.cos(x) + 6 * np.cos(2 * x) - 4 *
np.cos(3 * x))
    h_9 = lambda x: 1 / 231 * (59 + 108 * np.cos(x) + 78 * np.cos(2 * x) + 28 *
np.cos(3 * x) - 42 * np.cos(4 * x))
    h_11 = lambda x: 1 / 429 * (
            89 + 168 * np.cos(x) + 138 * np.cos(2 * x) + 88 * np.cos(3 * x) + 18
* np.cos(4 * x) - 72 * np.cos(
            5 * x))
    h_13 = lambda x: 1 / 143 * (
            25 + 48 * np.cos(x) + 42 * np.cos(2 * x) + 32 * np.cos(3 * x) + 18 *
np.cos(4 * x) - 22 * np.cos(6 * x))

    t = np.linspace(0, 2 * math.pi, 64)
    plt.figure(figsize=(8, 6), dpi=DPI)
    plt.plot(t, h_7(t), label=r'$H_7$')
    plt.plot(t, h_9(t), label=r'$H_9$')
    plt.plot(t, h_11(t), label=r'$H_{11}$')
    plt.plot(t, h_13(t), label=r'$H_{13}$')
    plt.xlabel('t')
    plt.ylabel(r'$H(t)$')
    plt.legend()
    plt.show()


def task3():
    h_9 = lambda x: 1 / 429 * (179 + 270 * np.cos(x) + 60 * np.cos(2 * x) - 110
* np.cos(3 * x) + 30 * np.cos(4 * x))
    h_11 = lambda x: 1 / 429 * (
```

```
            143 + 240 * np.cos(x) + 120 * np.cos(2 * x) - 20 * np.cos(3 * x) -
90 * np.cos(4 * x) + 36 * np.cos(
            5 * x))
    h_13 = lambda x: 1 / 2431 * (
            677 + 1200 * np.cos(x) + 780 * np.cos(2 * x) + 220 * np.cos(3 * x) -
270 * np.cos(4 * x) - 396 * np.cos(
            5 * x) + 220 * np.cos(6 * x))
    h_15 = lambda x: 1 / 46189 * (
            11063 + 20250 * np.cos(x) + 15000 * np.cos(2 * x) + 7510 * np.cos(3
* x) - 330 * np.cos(
            4 * x) - 5874 * np.cos(5 * x) - 5720 * np.cos(6 * x) + 4290 *
np.cos(7 * x))

    t = np.linspace(0, 2 * math.pi, 64)
    plt.figure(figsize=(8, 6), dpi=DPI)
    plt.plot(t, h_9(t), label=r'$H_9$')
    plt.plot(t, h_11(t), label=r'$H_11$')
    plt.plot(t, h_13(t), label=r'$H_{13}$')
    plt.plot(t, h_15(t), label=r'$H_{15}$')
    plt.xlabel('t')
    plt.ylabel(r'$H(t)$')
    plt.legend()
    plt.show()


def task4():
    h_21 = lambda x: 1 / 320 * (74 + 134 * np.cos(2 * math.pi * x) + 92 *
np.cos(4 * math.pi * x) + 42 * np.cos(
            6 * math.pi * x) + 6 * np.cos(8 * math.pi * x) - 10 * np.cos(10 *
math.pi * x) - 12 * np.cos(
            12 * math.pi * x) - 6 * np.cos(14 * math.pi * x))
    h_15 = lambda x: 1 / 350 * (60 + 114 * np.cos(2 * math.pi * x) + 94 *
np.cos(4 * math.pi * x) + 66 * np.cos(
            6 * math.pi * x) + 36 * np.cos(8 * math.pi * x) + 12 * np.cos(10 *
math.pi * x) - 4 * np.cos(
            12 * math.pi * x) - 10 * np.cos(14 * math.pi * x) - 10 * np.cos(16 *
math.pi * x) - 6 * np.cos(
            18 * math.pi * x) - 2 * np.cos(20 * math.pi * x))

    t = np.linspace(0, 1, 64)
    plt.figure(figsize=(8, 6), dpi=DPI)
    plt.plot(t, h_21(t), label=r'$H_{21}$')
    plt.plot(t, h_15(t), label=r'$H_{15}$')
    plt.xlabel('t')
    plt.ylabel(r'$H(t)$')
    plt.legend()
    plt.show()


def task5():
    def _task1():
        h_3 = lambda x: np.sin(3 * math.pi * x) / (3 * np.sin(x * math.pi))
        h_5 = lambda x: np.sin(5 * math.pi * x) / (5 * np.sin(x * math.pi))
        h_7 = lambda x: np.sin(7 * math.pi * x) / (7 * np.sin(x * math.pi))
        h_9 = lambda x: np.sin(9 * math.pi * x) / (9 * np.sin(x * math.pi))

        t = np.linspace(0.001, 1, 64)
        plt.figure(figsize=(8, 6), dpi=DPI)
        plt.plot(t, 20 * np.log10(np.abs(h_3(t))), label=r'$H_3$')
        plt.plot(t, 20 * np.log10(np.abs(h_5(t))), label=r'$H_5$')
        plt.plot(t, 20 * np.log10(np.abs(h_7(t))), label=r'$H_7$')
```

```
        plt.plot(t, 20 * np.log10(np.abs(h_9(t))), label=r'$H_9$')
        plt.xlabel('t')
        plt.ylabel(r'$H(t)$')
        plt.legend()
        plt.show()

    def _task2():
        h_7 = lambda x: 1 / 21 * (7 + 12 * np.cos(x) + 6 * np.cos(2 * x) - 4 *
np.cos(3 * x))
        h_9 = lambda x: 1 / 231 * (59 + 108 * np.cos(x) + 78 * np.cos(2 * x) +
28 * np.cos(3 * x) - 42 * np.cos(4 * x))
        h_11 = lambda x: 1 / 429 * (
                89 + 168 * np.cos(x) + 138 * np.cos(2 * x) + 88 * np.cos(3 * x)
+ 18 * np.cos(4 * x) - 72 * np.cos(
                5 * x))
        h_13 = lambda x: 1 / 143 * (
                25 + 48 * np.cos(x) + 42 * np.cos(2 * x) + 32 * np.cos(3 * x) +
18 * np.cos(4 * x) - 22 * np.cos(6 * x))

        t = np.linspace(0.001, 2 * math.pi, 64)
        plt.figure(figsize=(8, 6), dpi=DPI)
        plt.plot(t, 20 * np.log10(np.abs(h_7(t))), label=r'$H_7$')
        plt.plot(t, 20 * np.log10(np.abs(h_9(t))), label=r'$H_9$')
        plt.plot(t, 20 * np.log10(np.abs(h_11(t))), label=r'$H_{11}$')
        plt.plot(t, 20 * np.log10(np.abs(h_13(t))), label=r'$H_{13}$')
        plt.xlabel('t')
        plt.ylabel(r'$H(t)$')
        plt.legend()
        plt.show()

    def _task3():
        h_9 = lambda x: 1 / 429 * (
                179 + 270 * np.cos(x) + 60 * np.cos(2 * x) - 110 * np.cos(3 * x)
+ 30 * np.cos(4 * x))
        h_11 = lambda x: 1 / 429 * (
                143 + 240 * np.cos(x) + 120 * np.cos(2 * x) - 20 * np.cos(3 * x)
- 90 * np.cos(4 * x) + 36 * np.cos(
                5 * x))
        h_13 = lambda x: 1 / 2431 * (677 + 1200 * np.cos(x) + 780 * np.cos(2 *
x) + 220 * np.cos(3 * x) - 270 * np.cos(
                4 * x) - 396 * np.cos(5 * x) + 220 * np.cos(6 * x))
        h_15 = lambda x: 1 / 46189 * (
                11063 + 20250 * np.cos(x) + 15000 * np.cos(2 * x) + 7510 *
np.cos(3 * x) - 330 * np.cos(
                4 * x) - 5874 * np.cos(5 * x) - 5720 * np.cos(6 * x) + 4290 *
np.cos(7 * x))

        t = np.linspace(0.001, 2 * math.pi, 64)
        plt.figure(figsize=(8, 6), dpi=DPI)
        plt.plot(t, 20 * np.log10(np.abs(h_9(t))), label=r'$H_9$')
        plt.plot(t, 20 * np.log10(np.abs(h_11(t))), label=r'$H_{11}$')
        plt.plot(t, 20 * np.log10(np.abs(h_13(t))), label=r'$H_{13}$')
        plt.plot(t, 20 * np.log10(np.abs(h_15(t))), label=r'$H_{15}$')
        plt.xlabel('t')
        plt.ylabel(r'$H(t)$')
        plt.legend()
        plt.show()

    def _task4():
        h_21 = lambda x: 1 / 320 * (74 + 134 * np.cos(2 * math.pi * x) + 92 *
np.cos(4 * math.pi * x) + 42 * np.cos(
```

```
                6 * math.pi * x) + 6 * np.cos(8 * math.pi * x) - 10 * np.cos(10
* math.pi * x) - 12 * np.cos(
                12 * math.pi * x) - 6 * np.cos(14 * math.pi * x))
        h_15 = lambda x: 1 / 350 * (60 + 114 * np.cos(2 * math.pi * x) + 94 *
np.cos(4 * math.pi * x) + 66 * np.cos(
                6 * math.pi * x) + 36 * np.cos(8 * math.pi * x) + 12 * np.cos(10
* math.pi * x) - 4 * np.cos(
                12 * math.pi * x) - 10 * np.cos(14 * math.pi * x) - 10 *
np.cos(16 * math.pi * x) - 6 * np.cos(
                18 * math.pi * x) - 2 * np.cos(20 * math.pi * x))

        t = np.linspace(0.001, 1, 64)
        plt.figure(figsize=(8, 6), dpi=DPI)
        plt.plot(t, 20 * np.log10(np.abs(h_21(t))), label=r'$H_{21}$')
        plt.plot(t, 20 * np.log10(np.abs(h_15(t))), label=r'$H_{15}$')
        plt.xlabel('t')
        plt.ylabel(r'$H(t)$')
        plt.legend()
        plt.show()

    _task1()
    _task2()
    _task3()
    _task4()


def task7():
    h_rect = lambda x: (1 / (2j * np.sin(math.pi * x))).imag
    h_trap = lambda x: (np.cos(math.pi * x) / (2j * np.sin(math.pi * x))).imag
    h_simp = lambda x: ((np.cos(2 * math.pi * x) + 2) / (3j * np.sin(2 * math.pi
* x))).imag

    y_rect = lambda x: math.pi * x / (np.sin(math.pi * x))
    y_trap = lambda x: np.cos(math.pi * x) * (math.pi * x / np.sin(x * math.pi))
    y_simp = lambda x: (np.cos(2 * math.pi * x) + 2) / 3

    t = np.linspace(1e-10, 0.5, 300)
    plt.figure(figsize=(8, 6), dpi=DPI)
    plt.plot(t, h_rect(t), label=r'$H_{rect}$')
    plt.xlabel('t')
    plt.ylabel(r'$H(t)$')
    plt.legend()
    x1, x2, y1, y2 = plt.axis()
    plt.axis((x1, x2, -50, 10.))
    plt.show()
    plt.figure(figsize=(8, 6), dpi=DPI)
    plt.plot(t, h_trap(t), label=r'$H_{trap}$')
    plt.xlabel('t')
    plt.ylabel(r'$H(t)$')
    plt.legend()
    x1, x2, y1, y2 = plt.axis()
    plt.axis((x1, x2, -50, 10.))
    plt.show()
    plt.figure(figsize=(8, 6), dpi=DPI)
    plt.plot(t, h_simp(t), label=r'$H_{simp}$')
    plt.xlabel('t')
    plt.ylabel(r'$H(t)$')
    plt.legend()
    x1, x2, y1, y2 = plt.axis()
    plt.axis((x1, x2, -50, 10.))
    plt.show()
```

```
    plt.figure(figsize=(8, 6), dpi=DPI)
    plt.plot(t, y_rect(t), label=r'$y_{rect}$')
    plt.plot(t, y_trap(t), label=r'$y_{trap}$')
    plt.plot(t, y_simp(t), label=r'$y_{simp}$')
    plt.xlabel('t')
    plt.ylabel(r'$y(t)$')
    plt.legend()
    plt.show()


def task8():
    h = lambda x: ((np.cos(3 * math.pi * x) + 3 * np.cos(math.pi * x)) / (8j *
np.sin(3 * math.pi * x))).imag
    y = lambda x: (1 / 12) * (np.cos(3 * math.pi * x) + 3 * np.cos(math.pi * x))
* (
            (3 * math.pi * x) / np.sin(3 * math.pi * x))

    t = np.linspace(1e-10, 0.5, 300)
    plt.figure(figsize=(8, 6), dpi=DPI)
    plt.plot(t, h(t), label=r'$H$', )
    plt.xlabel('t')
    plt.ylabel(r'$H(t)$')
    plt.legend()
    x1, x2, y1, y2 = plt.axis()
    plt.axis((x1, x2, -50, 10.))
    plt.show()
    plt.figure(figsize=(8, 6), dpi=DPI)
    plt.plot(t, y(t), label=r'$y$')
    plt.xlabel('t')
    plt.ylabel(r'$y(t)$')
    plt.legend()
    plt.show()


#%%

task1()

#%%

task2()

#%%

task3()

#%%

task4()

#%%

task5()

#%%

task7()

#%%

task8()
```