

Санкт-Петербургский государственный политехнический  
университет Петра Великого

**Высшая школа интеллектуальных систем и  
суперкомпьютерных технологий**

Лабораторная работа

# Дифференцирование и интегрирование

Выполнил студент гр. 3530901/80201

И.С. Иванов

Преподаватель:

Н.В. Богач

Санкт-Петербург  
2021

# Содержание

<b>1</b>	<b>Упражнение №1</b>	<b>5</b>
<b>2</b>	<b>Упражнение №2</b>	<b>10</b>
<b>3</b>	<b>Упражнение №3</b>	<b>12</b>
<b>4</b>	<b>Упражнение №4</b>	<b>16</b>
<b>5</b>	<b>Упражнение №5</b>	<b>18</b>
<b>6</b>	<b>Выводы</b>	<b>21</b>

## Список иллюстраций

1	Сигнал Facebook . . . . .	5
2	Спектр сигнала Facebook . . . . .	6
3	Выходной сигнал . . . . .	7
4	Спектр выходного сигнала . . . . .	7
5	Отношения входного и выходного сигнала . . . . .	8
6	Фильтр нарастающей суммы и интегрирования . . . . .	8
7	Сравнение отношения и фильтра . . . . .	9
8	Сравнение суммирования и фильтрации . . . . .	9
9	Треугольный сигнал . . . . .	10
10	Результат применения diff к сигналу . . . . .	11
11	Результат применения differentiate к спектру сигнала . . . . .	11
12	Прямоугольный сигнал . . . . .	12
13	Результат cumsum . . . . .	13
14	Результат integrate . . . . .	14
15	Сравнение функций . . . . .	15
16	Пилообразный сигнал . . . . .	16
17	Сигнал после двойного интегрирования . . . . .	17
18	Спектр сигнала после интегрирования . . . . .	17
19	Кубический сигнал . . . . .	18
20	Вторая разность кубического сигнала . . . . .	19
21	Вторая производная кубического сигнала . . . . .	19
22	Сравнение фильтров . . . . .	20

## Листинги

1	Создание сигнала facebook . . . . .	5
2	Получение выходного сигнала . . . . .	6
3	Создание треугольного сигнала . . . . .	10
4	Создание прямоугольного сигнала . . . . .	12
5	Применение integrate к спектру сигнала . . . . .	13
6	Сравнение cumsum и integrate . . . . .	14
7	Создание пилообразного сигнала . . . . .	16
8	Применение двойного интегрирования . . . . .	16
9	Создание кубического сигнала . . . . .	18

# 1 Упражнение №1

В первом упражнении необходимо изучить примеры из файла `chap09.ipynb`. Далее необходимо заменить пилообразный сигнал на непериодические данные Facebook в примере нарастающей суммы.

Создадим сигнал.

```
1 df = pd.read_csv('Res/FB_2.csv', header=0, parse_dates=[0])
2 ys = df['Close']
3 in_wave = Wave(ys, framerate=1)
4 in_wave.plot()
5 decorate(xlabel='Time (s)')
6
```

Листинг 1: Создание сигнала facebook

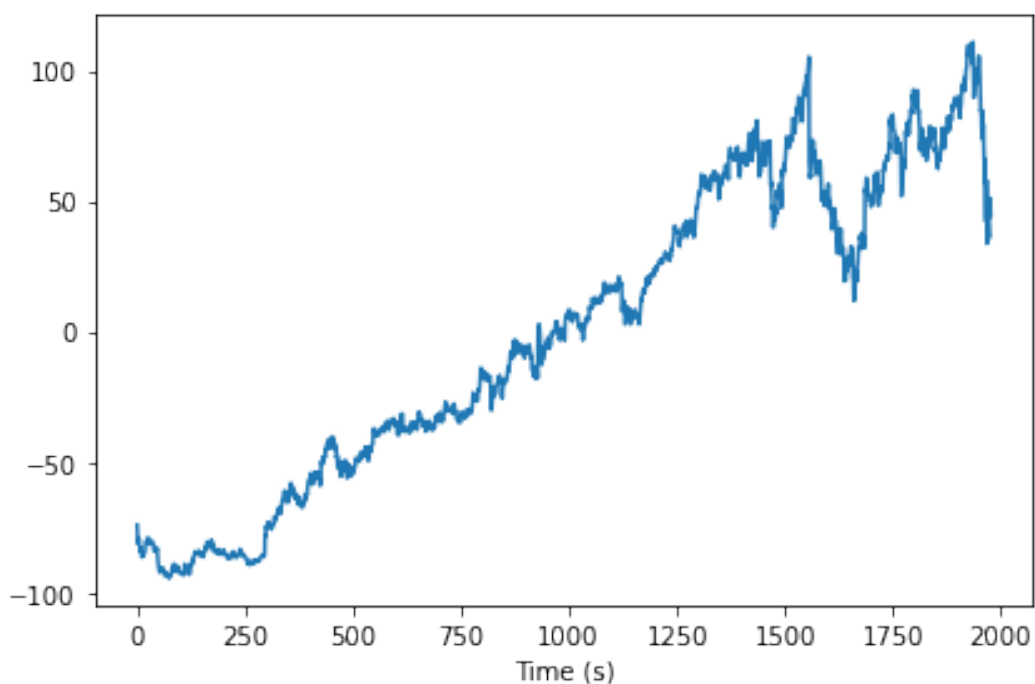


Рис. 1: Сигнал Facebook

Построим спектр.

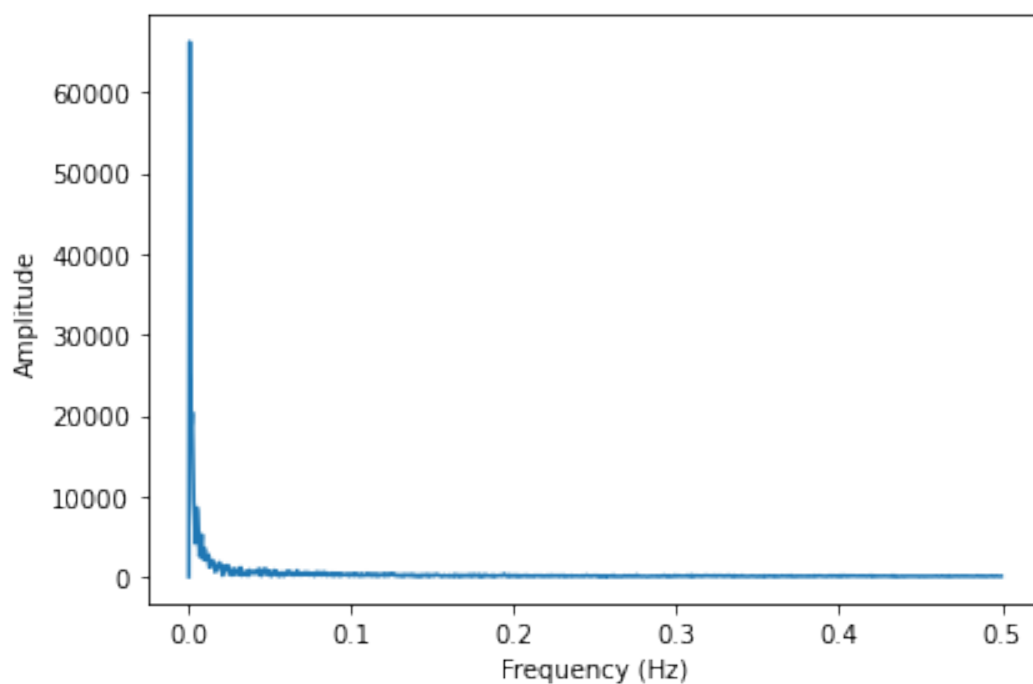


Рис. 2: Спектр сигнала Facebook

Получим выходной сигнал, который является совокупной суммой входных сигналов, и его спектр

```
1 out_wave = in_wave.cumsum()  
2 out_wave.unbias()  
3 out_wave.plot()  
4 decorate(xlabel='Time (s)')  
5
```

Листинг 2: Получение выходного сигнала

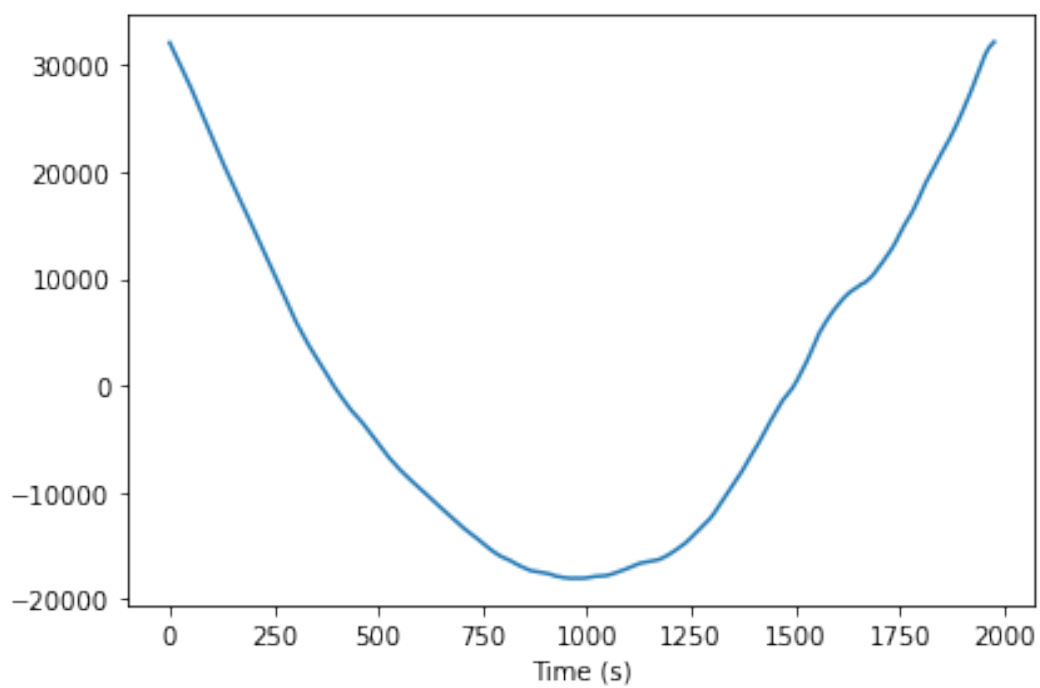


Рис. 3: Выходной сигнал

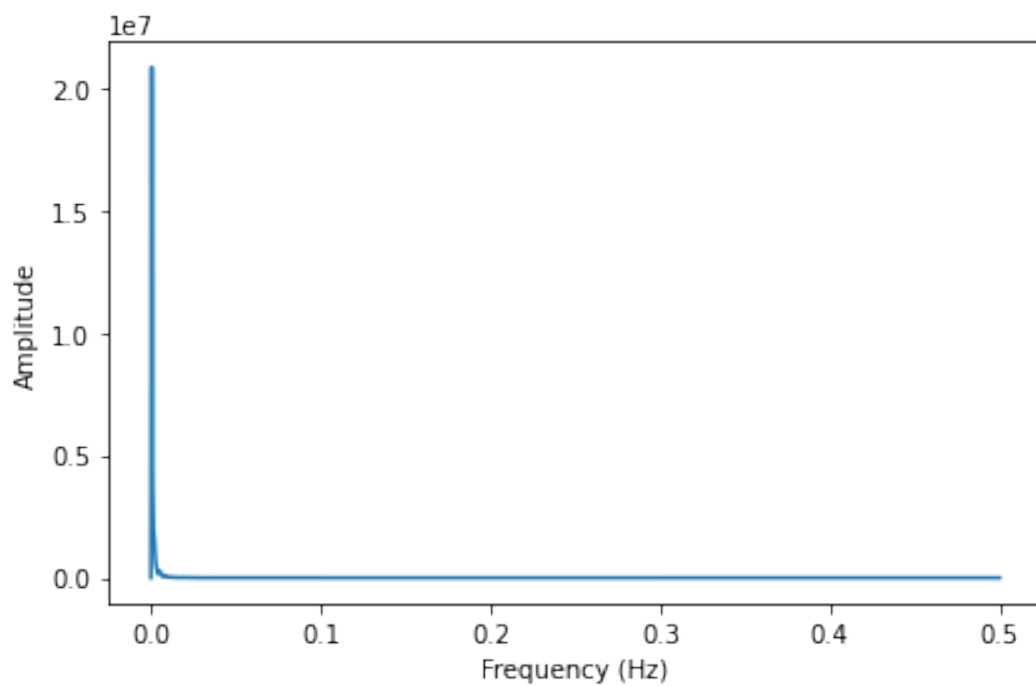


Рис. 4: Спектр выходного сигнала

Посмотрим на отношение входного и выходного сигнала.

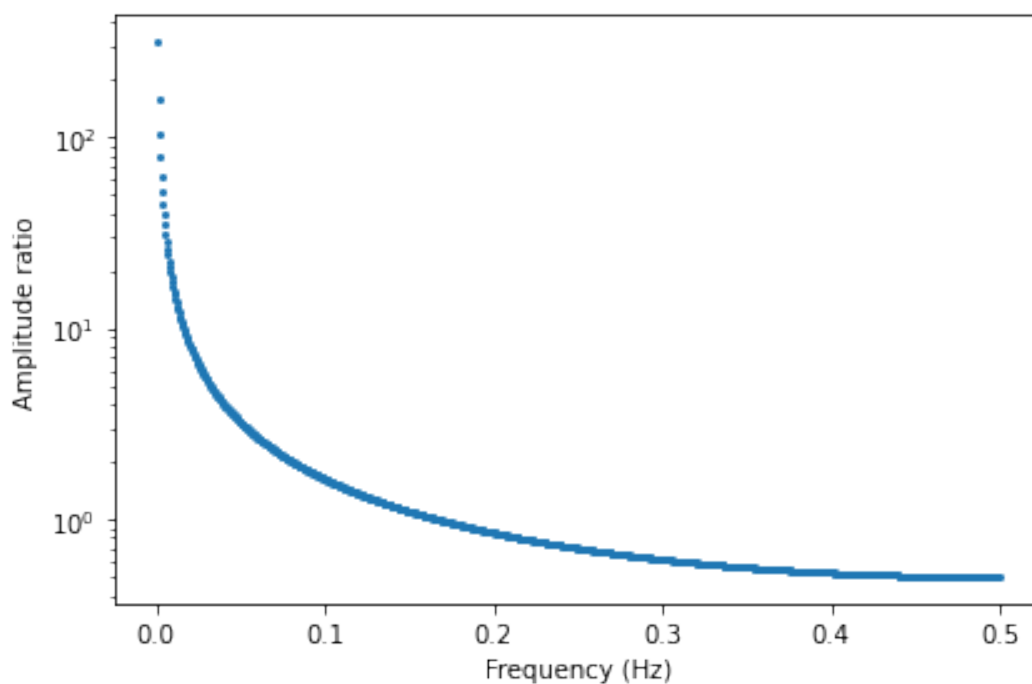


Рис. 5: Отношения входного и выходного сигнала

Построим фильтр для нарастающей суммы и сравним его с фильтром интегрирования.

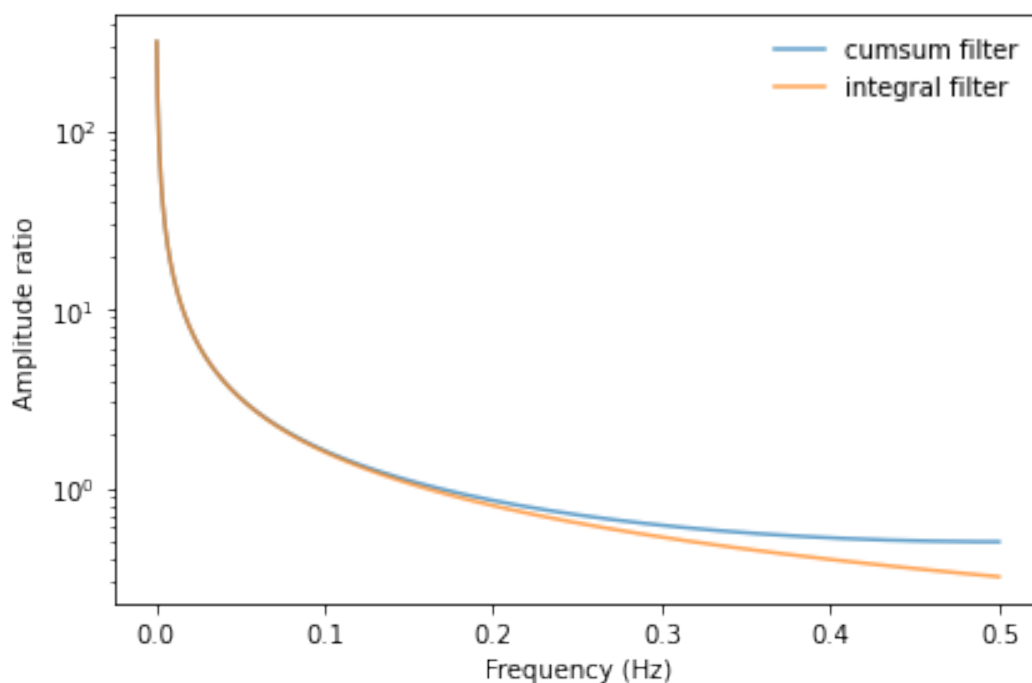


Рис. 6: Фильтр нарастающей суммы и интегрирования

На этой функции видно, что графики сначала совпадают, после расходятся.



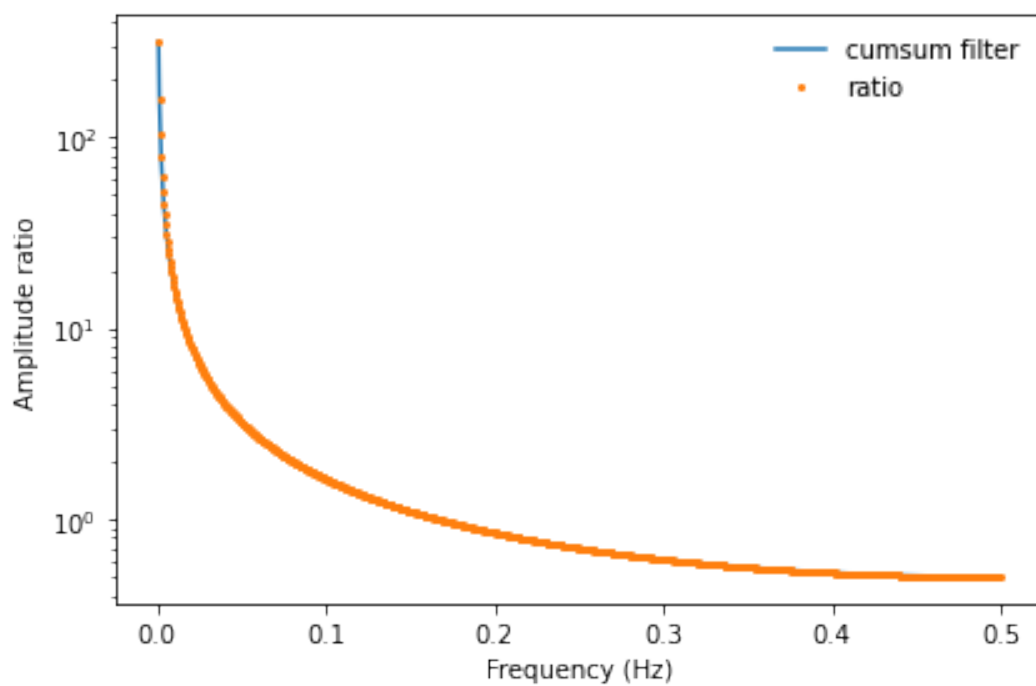


Рис. 7: Сравнение отношения и фильтра

Графики совпадают. Фильтр cumsum обратный фильтру diff.

Применим cumsum в частотной области.

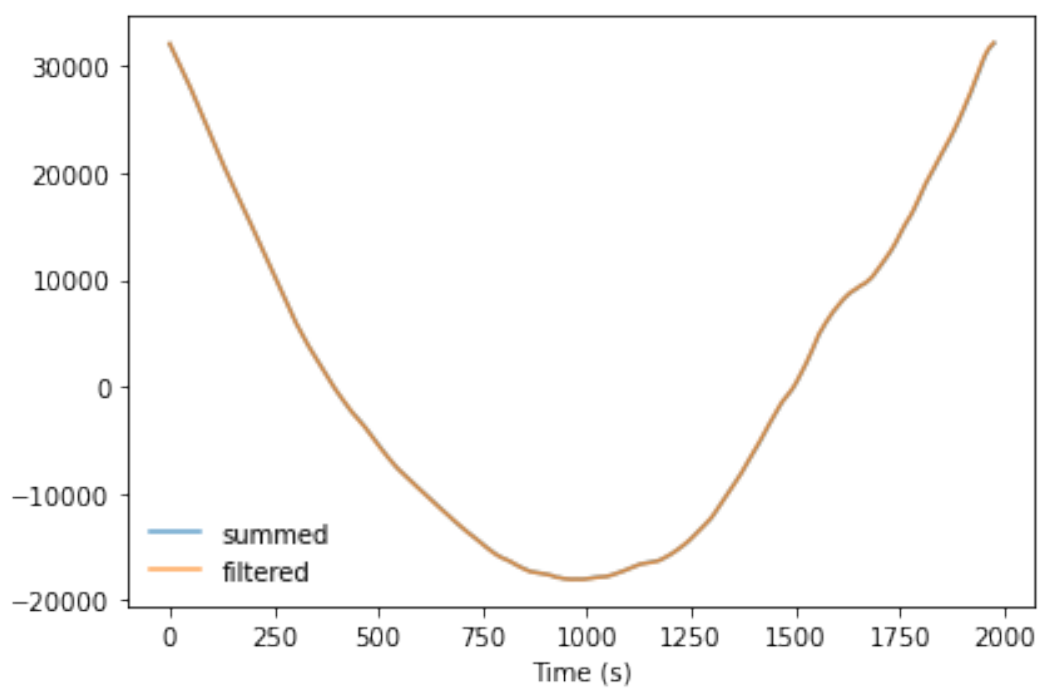


Рис. 8: Сравнение суммирования и фильтрации

Графики совпали.

## 2 Упражнение №2

В втором упражнении необходимо изучить влияние `diff` и `differentiate` на сигнал. Для этого необходимо создать треугольный сигнал, и применить у нему `diff`. Далее необходимо вычислить спектр сигнала, применить `differentiate` и посмотреть на результат.

Создадим треугольный сигнал.

```
1 from thinkdsp import TriangleSignal
2
3 in_wave = TriangleSignal(freq=50).make_wave(duration=0.1, framerate=44100)
4 in_wave.plot()
5 decorate(xlabel='Time (s)')
6
```

Листинг 3: Создание треугольного сигнала

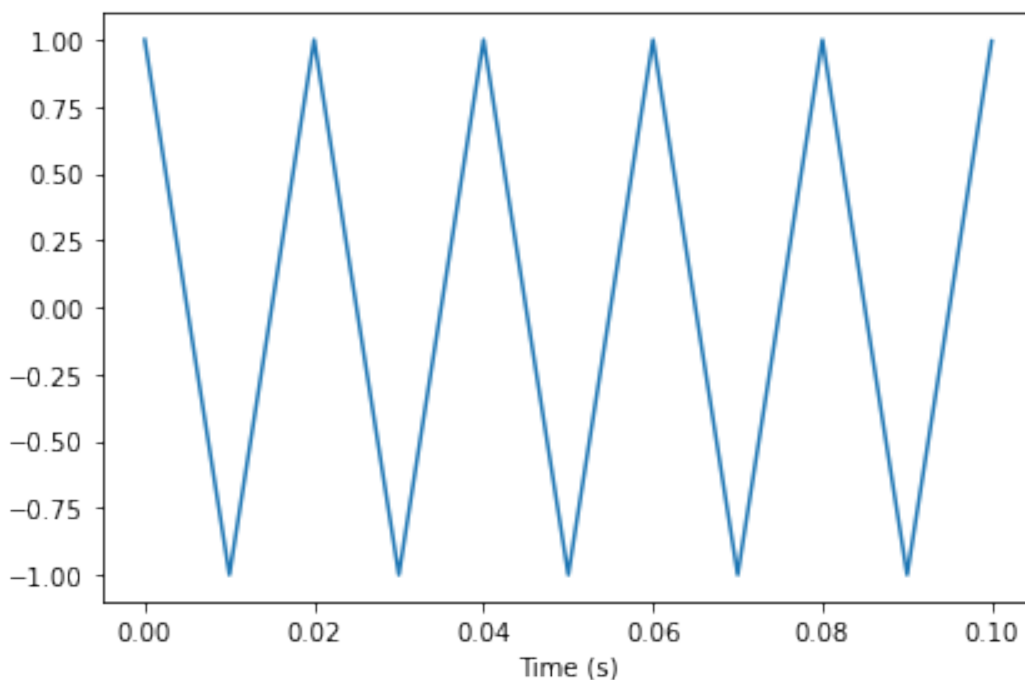


Рис. 9: Треугольный сигнал

Применим `diff` к сигналу.

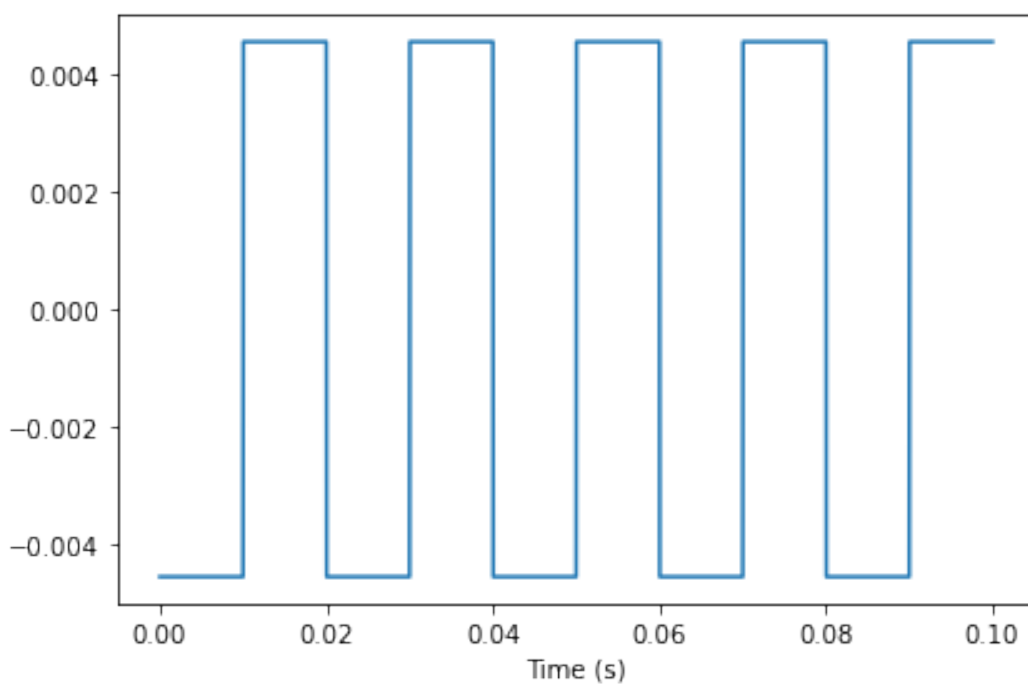


Рис. 10: Результат применения diff к сигналу

Вычислим спектр сигнала и применим к нему differentiate

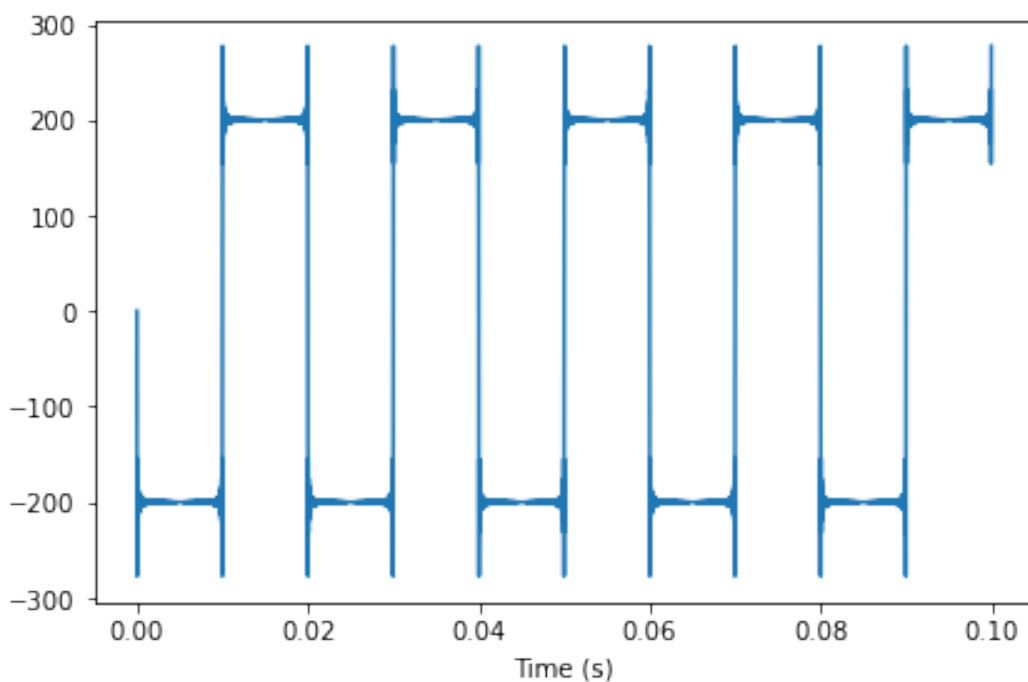


Рис. 11: Результат применения differentiate к спектру сигнала

На графике виден необычный эффект. Он вызван тем, что производная треугольного сигнала не определена в вершинах треугольника.

### 3 Упражнение №3

В третьем упражнении необходимо изучить влияние `cumsum` и `integrate` на сигнал. Для этого необходимо создать прямоугольный сигнал и применить к нему `cumsum`. Затем вычислить спектр сигнала и применить `integrate`.

Создадим сигнал для дальнейшей работы.

```
1 from thinkdsp import SquareSignal
2
3 in_wave = SquareSignal(freq=50).make_wave(duration=0.1, framerate=44100)
4 in_wave.plot()
5 decorate(xlabel='Time (s)')
6
```

Листинг 4: Создание прямоугольного сигнала

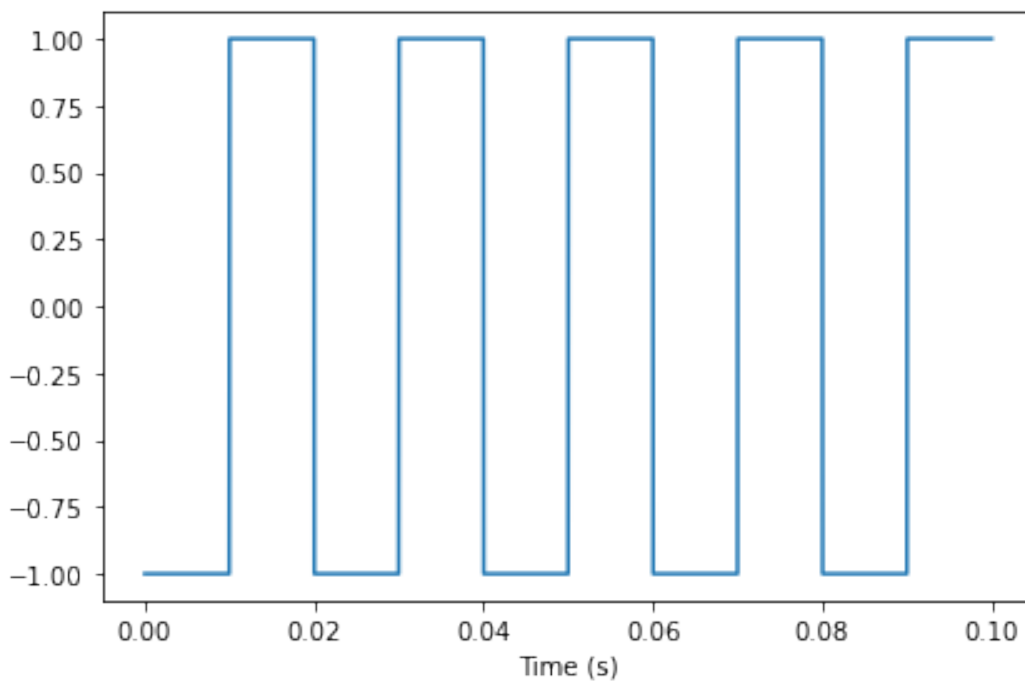


Рис. 12: Прямоугольный сигнал

Применим в полученному сигналу `cumsum`.

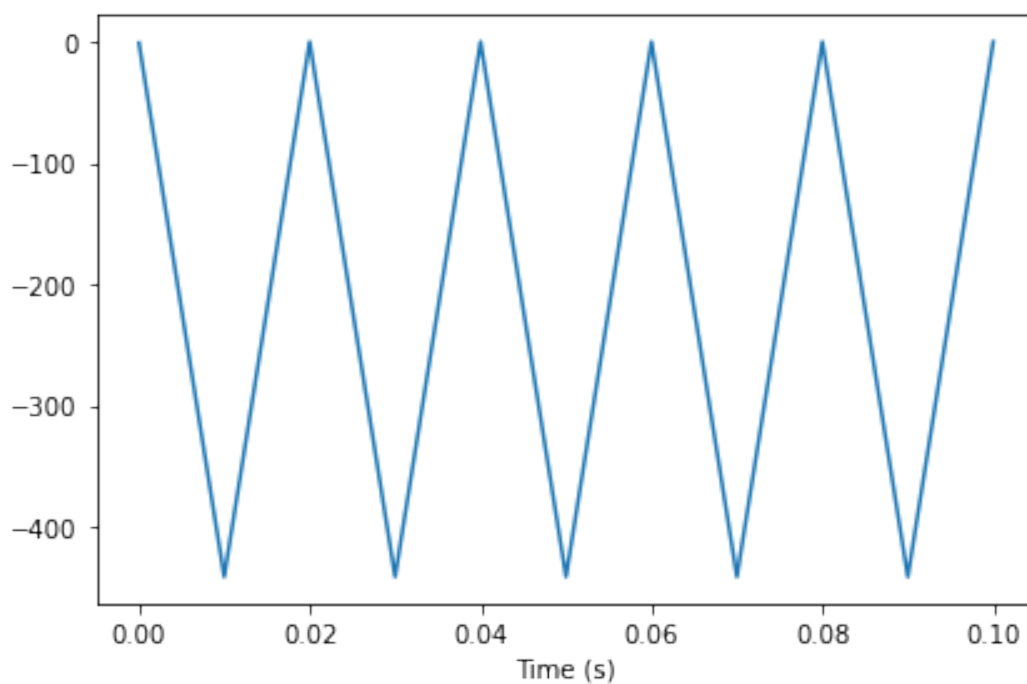


Рис. 13: Результат cumsum

Применим к спектру сигнала `integrate`

```

1 spectrum = in_wave.make_spectrum().integrate()
2 spectrum.hs[0] = 0
3 out_wave2 = spectrum.make_wave()
4 out_wave2.plot()
5 decorate(xlabel='Time (s)')
6

```

Листинг 5: Применение `integrate` к спектру сигнала

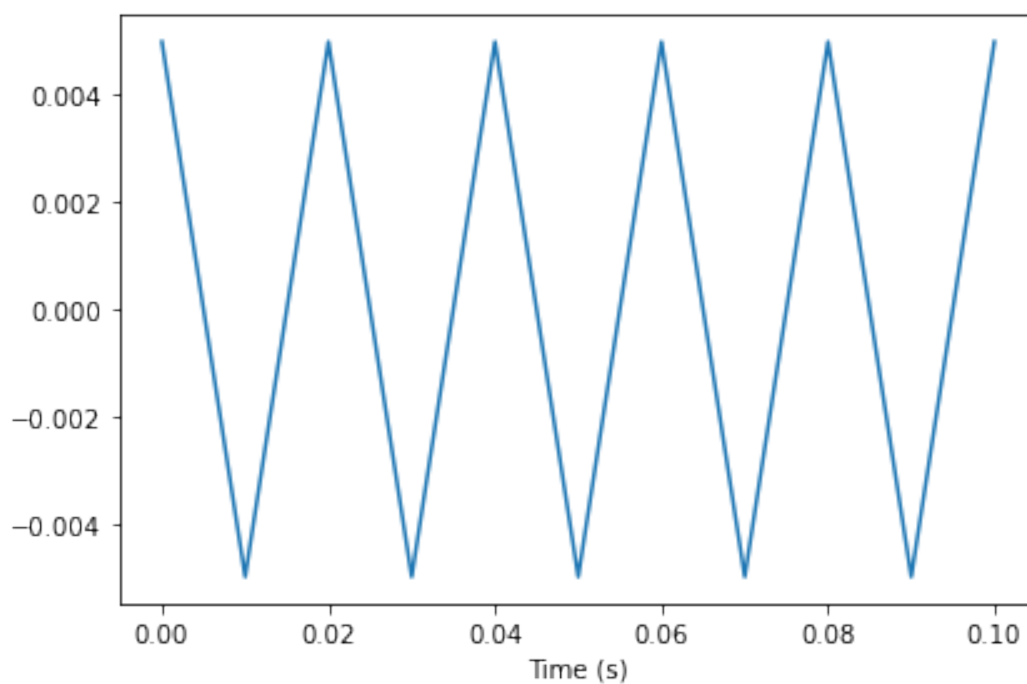


Рис. 14: Результат integrate

Результаты `cumsum` и `integrate` на вид получились одинаковыми. Проверим их схожесть.

```
1 out_wave.unbias()  
2 out_wave.normalize()  
3 out_wave2.normalize()  
4 out_wave.plot()  
5 out_wave2.plot()  
6
```

Листинг 6: Сравнение `cumsum` и `integrate`

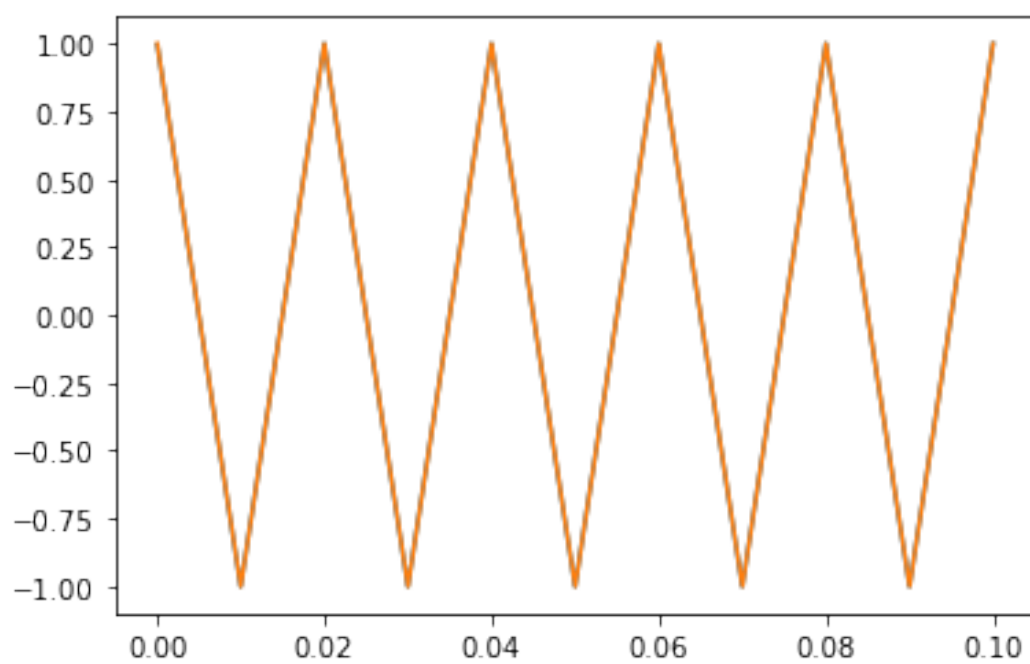


Рис. 15: Сравнение функций

По результатам видно, что различия в сигналах почти неразличимы.

## 4 Упражнение №4

В четвертом упражнении необходимо изучить влияние двойного интегрирования. Для этого надо создать пилообразный сигнал, вычислить его спектр и дважды применить `integrate`.

Создадим пилообразный сигнал.

```
1 from thinkdsp import SawtoothSignal
2
3 in_wave = SawtoothSignal(freq=50).make_wave(duration=0.1, framerate=44100)
4 in_wave.plot()
5 decorate(xlabel='Time (s)')
6
```

Листинг 7: Создание пилообразного сигнала

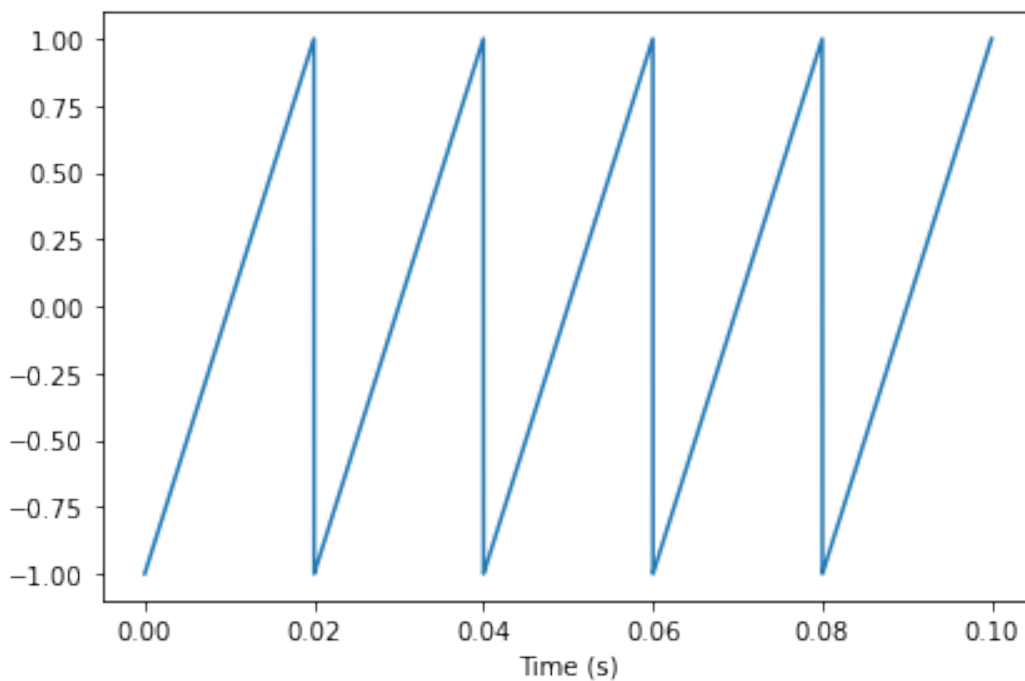


Рис. 16: Пилообразный сигнал

Применим двойное интегрирование к созданному сигналу.

```
1 spectrum = in_wave.make_spectrum().integrate().integrate()
2 spectrum.hs[0] = 0
3 out_wave2 = spectrum.make_wave()
4 out_wave2.plot()
5 decorate(xlabel='Time (s)')
6
```

Листинг 8: Применение двойного интегрирования



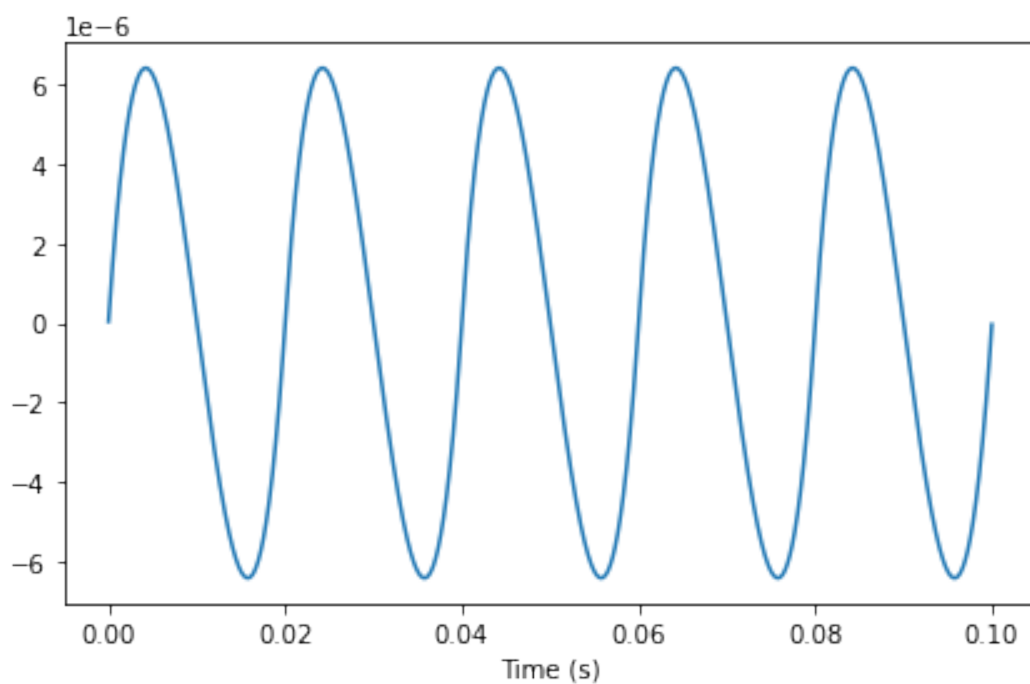


Рис. 17: Сигнал после двойного интегрирования

Создадим спектр сигнала после интегрирования.

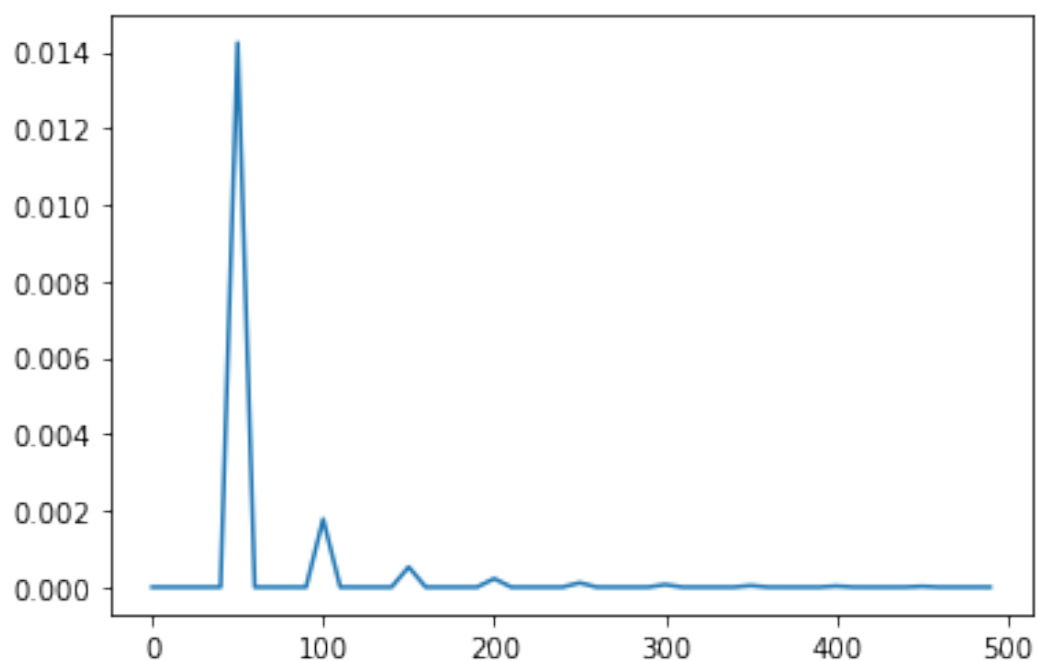


Рис. 18: Спектр сигнала после интегрирования

На графике видно, что сигнал напоминает синусоиду. Происходит это из-за того, что `integrate` действует как фильтр НЧ. На спектре видно, что отфильтровано все, кроме НЧ.

## 5 Упражнение №5

В четвертом упражнении необходимо изучить влияние второй разности и второй производной. Для этого надо создать CubicSignal, вычислить двойную разность применив diff и проанализировать результат. Далее вычислить вторую производную, дважды применив differentiate к спектру.

Начнем с создания сигнала.

```
1 from thinkdsp import CubicSignal
2
3 in_wave = CubicSignal(freq=0.0005).make_wave(duration=10000, framerate=1)
4 in_wave.plot()
5
```

Листинг 9: Создание кубического сигнала

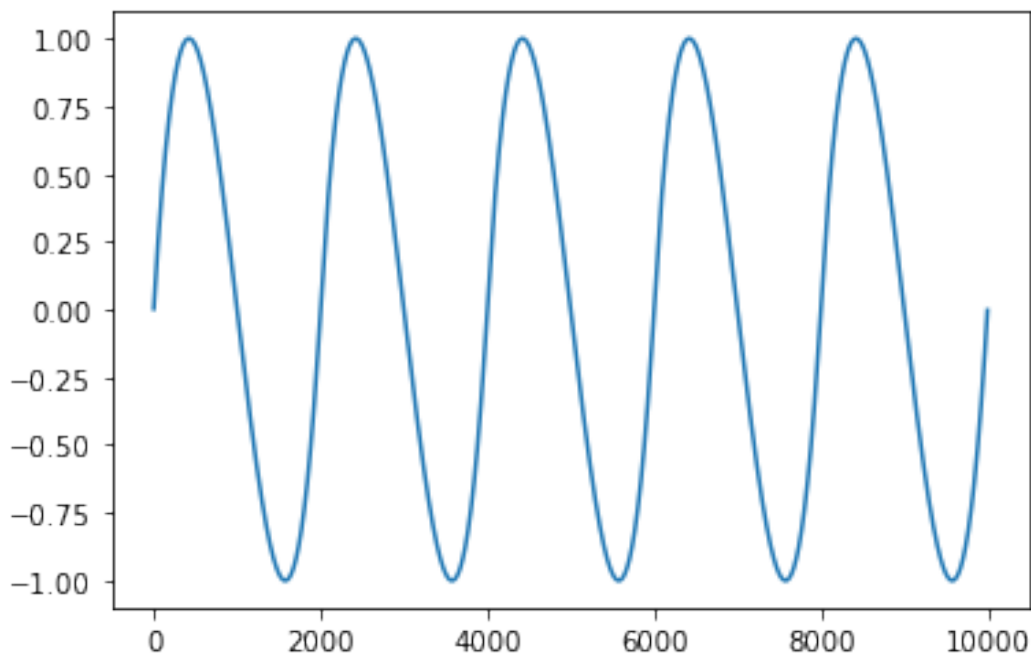


Рис. 19: Кубический сигнал

Вычислим двойную разность, дважды применив diff.

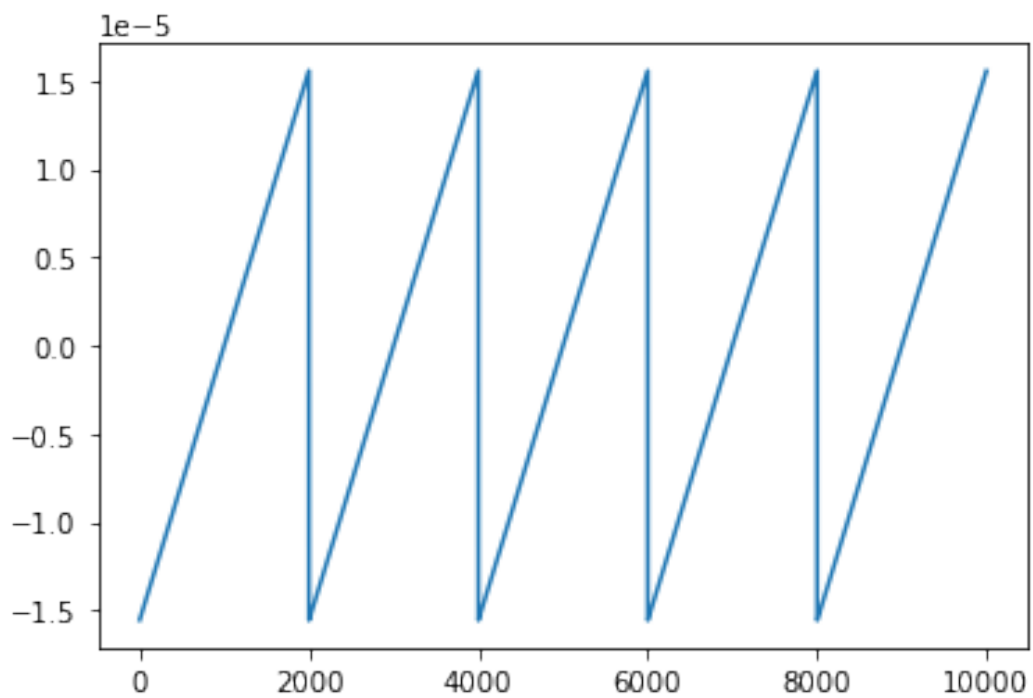


Рис. 20: Вторая разность кубического сигнала

Результат похож на пилообразный сигнал.

Вычислим двойную производную, дважды применив `differentiate`.

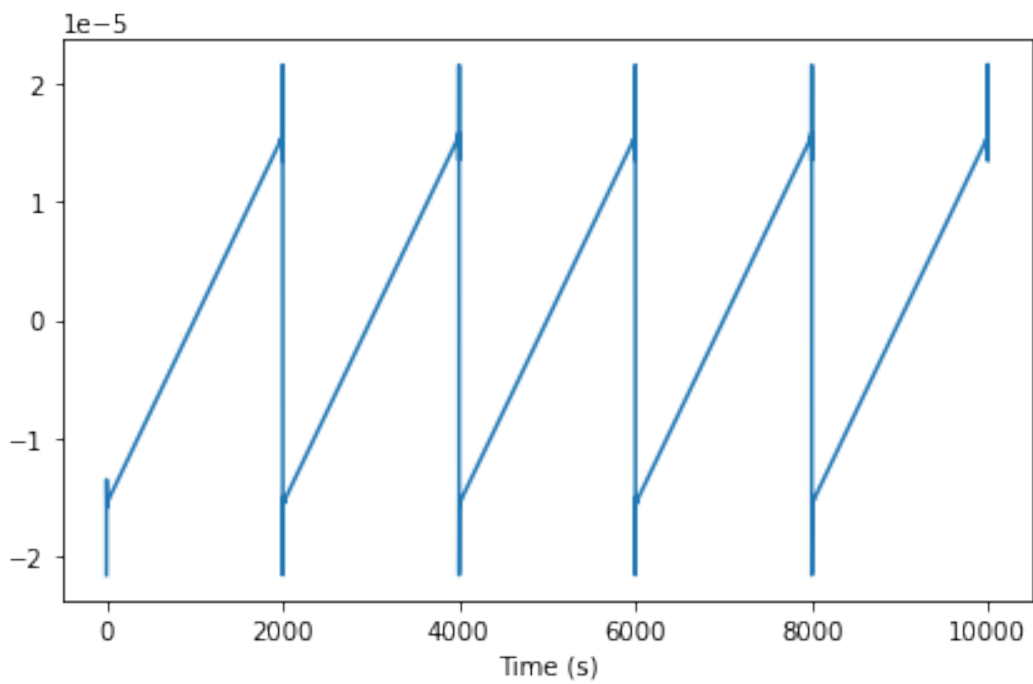


Рис. 21: Вторая производная кубического сигнала

Мы так же получили пилообразный сигнал с некоторым шумом. Связано

это с тем, что производная параболического сигнала не определена в некоторых точках.

Расчитаем фильтры, соответствующие второй разности и второй производной, и сравним их.

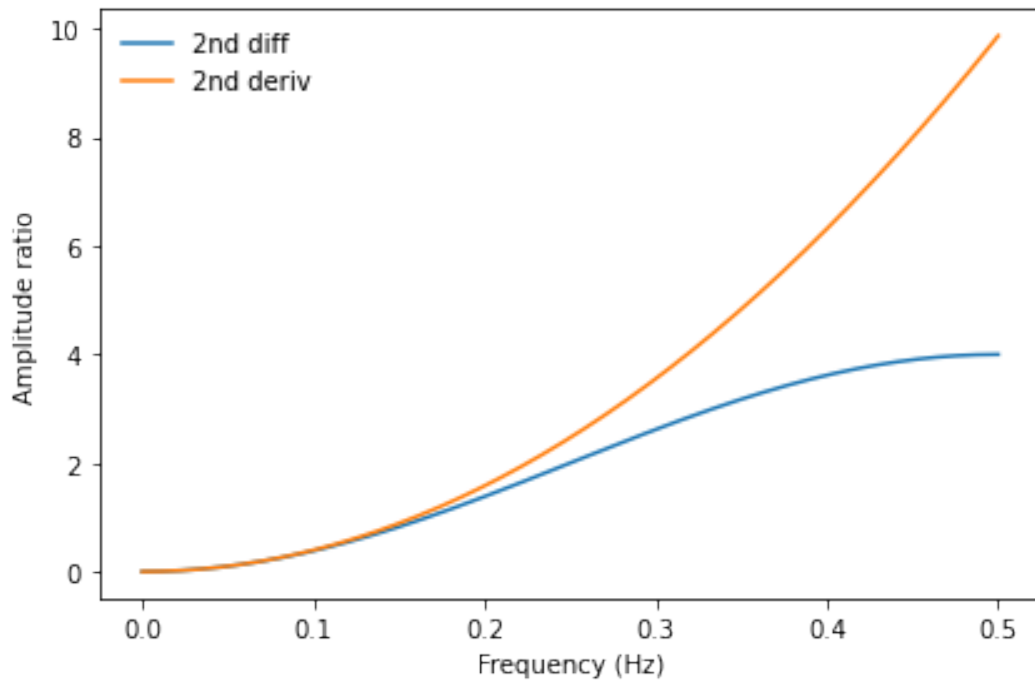


Рис. 22: Сравнение фильтров

Оба фильтра являются фильтрами ВЧ, которые усиливают компоненты высшей частоты. Поэтому на низких частотах различий нет, но на высоких становятся заметны.

## 6 Выводы

В результате выполнения данной лабораторной работы были получены знания по интегрированию и дифференцированию сигналов. Была разобрана работа `diff`, `differentiate`, `cumsum` и `integrate`. Было изучено влияние двойного интегрирования, второй производной и второй разности на сигналы.