

Лабораторная работа 9

Чернышев Ярослав

1 июня 2021 г.

Оглавление

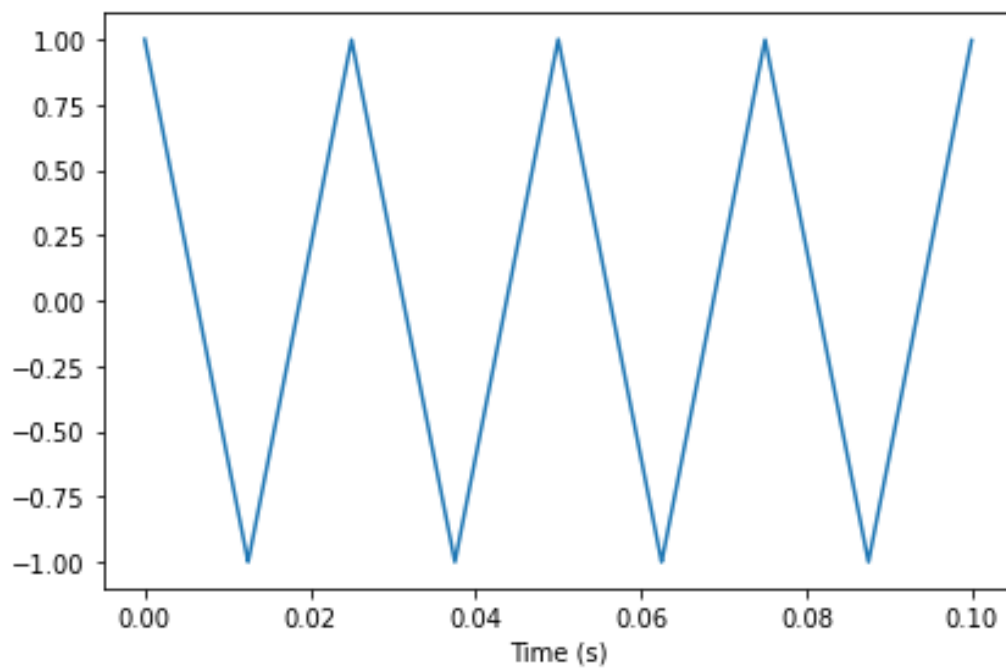
1	Задание 9.1	2
2	Задание 9.2	5
3	Задание 9.3	9
4	Задание 9.4	12

Глава 1

Задание 9.1

В данном задании требуется оценить разницу между `diff` и `differentiate` на сигнале. Для этого создадим треугольный сигнал:

```
1 in_wave = TriangleSignal(freq=40).make_wave(duration=0.1,  
    framerate=44100)  
2 in_wave.plot()  
3 decorate(xlabel='Time (s)')
```

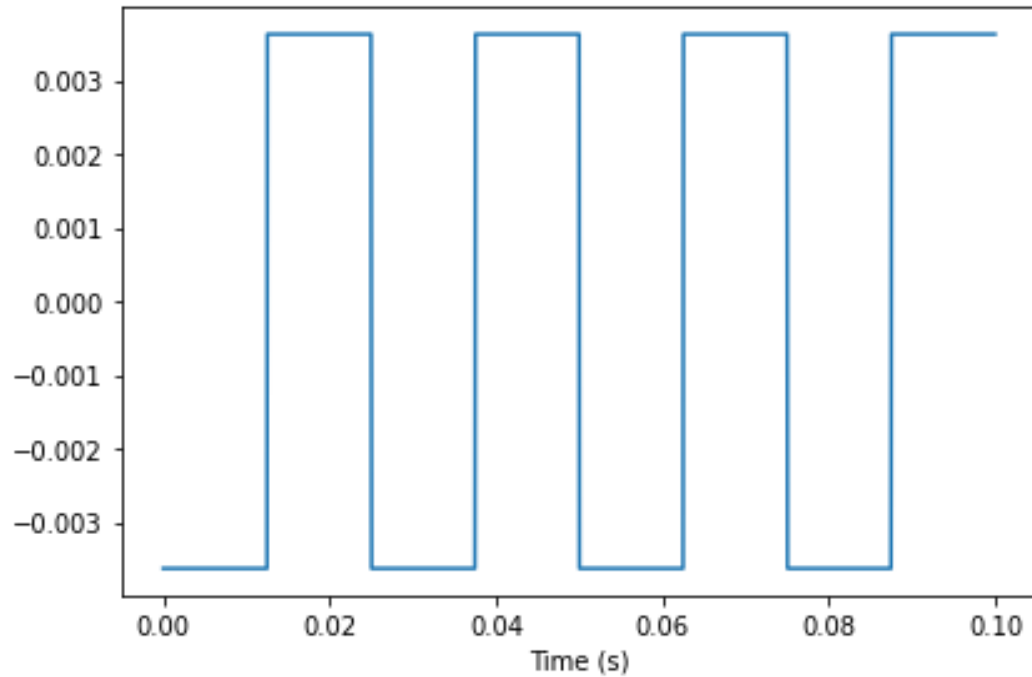


Первой применим `diff`:

```

1 out_wave = in_wave.diff()
2 out_wave.plot()
3 decorate(xlabel='Time (s)')

```



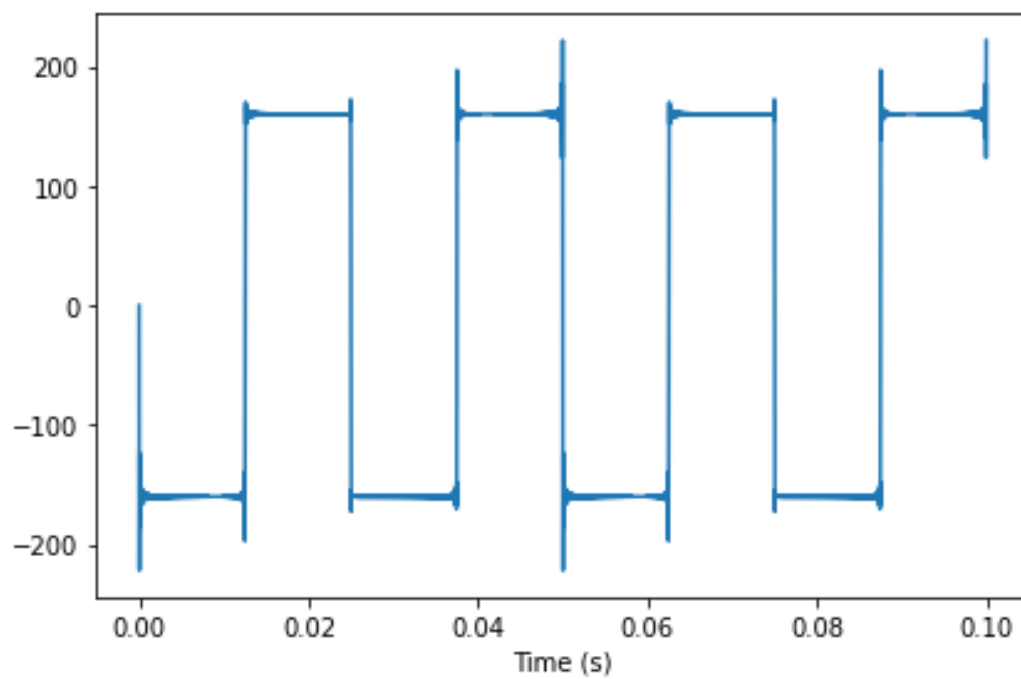
Действительно, производная от треугольной функции является прямоугольной функцией.

Для сравнения, создаем спектр сигнала и применяем к нему differentiate:

```

1 out_wave2 = in_wave.make_spectrum().differentiate().make_wave
  ()
2 out_wave2.plot()
3 decorate(xlabel='Time (s)')

```



Можно видеть, что применение `differentiate` дает более размытое изображение прямоугольной функции в местах разрывов. Это связано с неопределенностью функции в местах вершин треугольного сигнала.

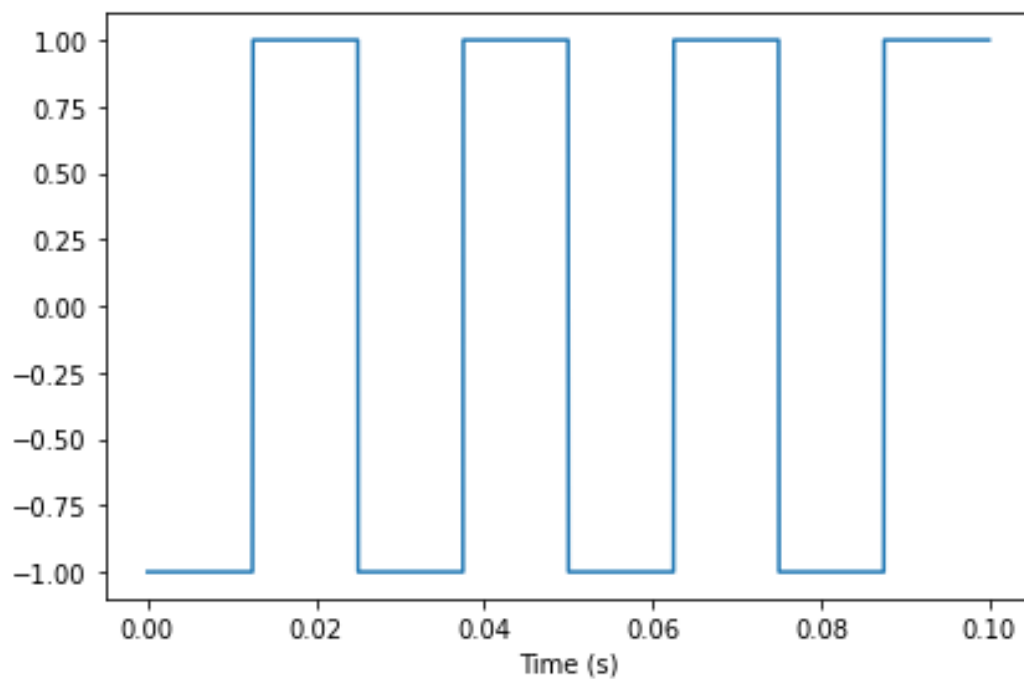
Глава 2

Задание 9.2

В данном задании требуется оценить разницу между `cumsum` and `integrate` на сигнале. Для этого создадим прямоугольный сигнал:

```
1 in_wave = SquareSignal(freq=40).make_wave(duration=0.1,  
      framerate=44100)  
2 in_wave.plot()  
3 decorate(xlabel='Time (s)')
```

Таким образом, наша функция имеет вид:

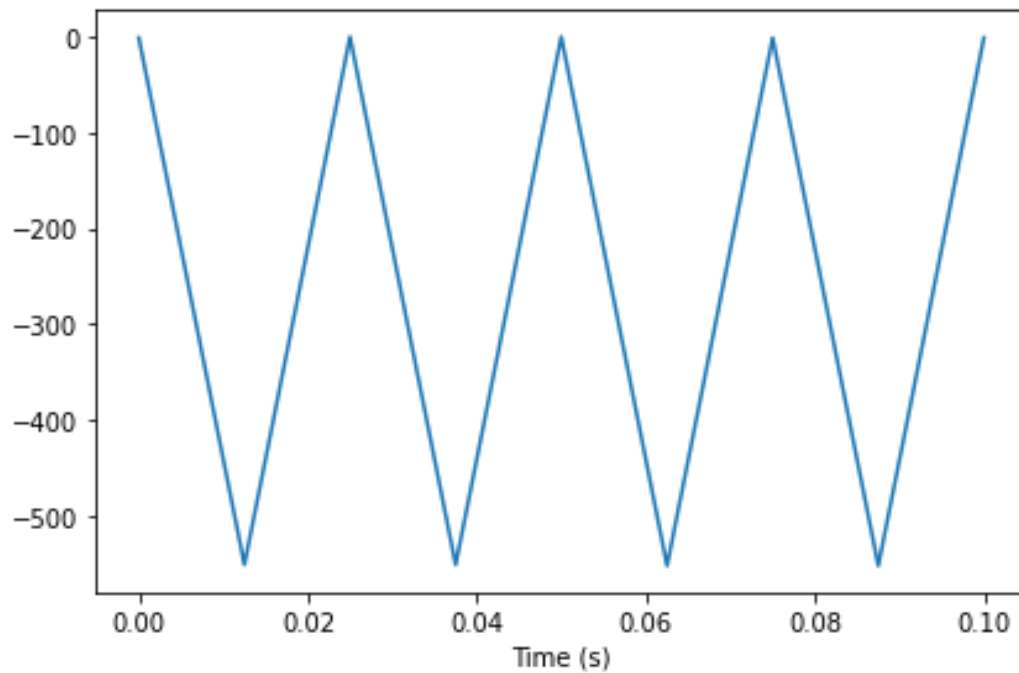


Первой применим `cumsum`:

```

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

```



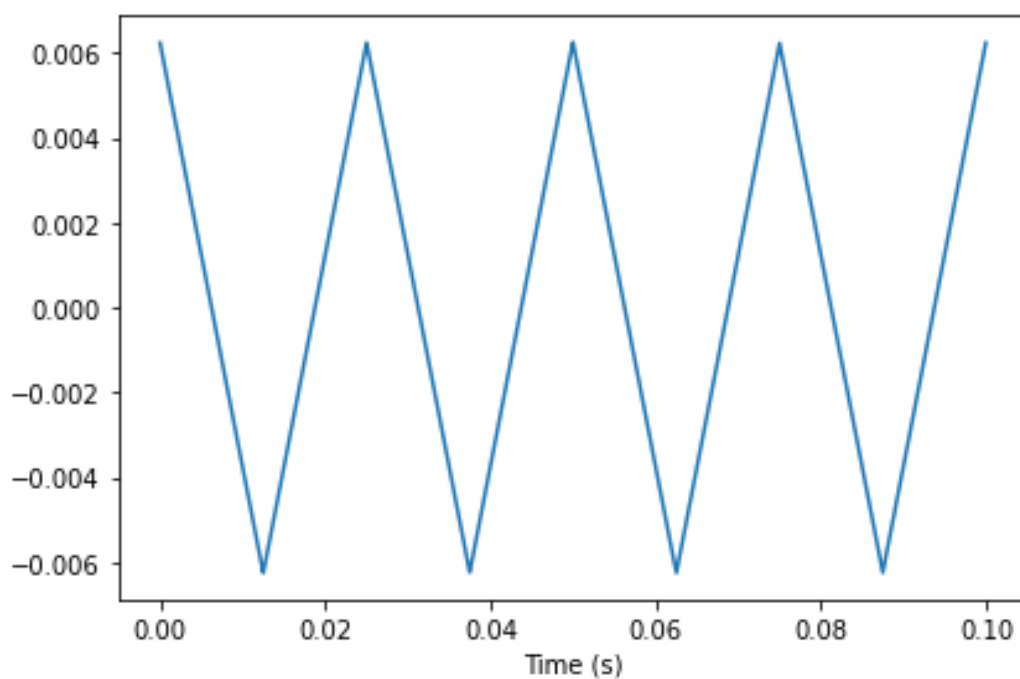
Как и предполагалось, производная от прямоугольной функции является треугольной функцией. Значит, результат соответствует ожидаемому.

Для сравнения, создаем спектр сигнала и применяем к нему `differentiate`:

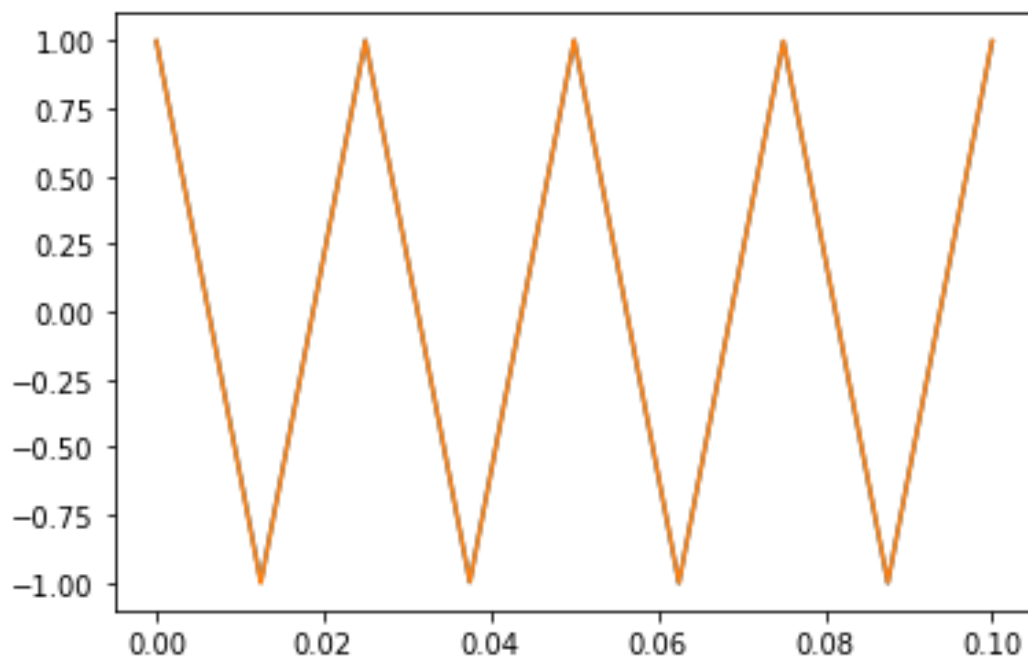
```

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)')

```



Разница между этими графиками на глаз незаметна. Поэтому, я построю оба графика ещё раз на одной плоскости:

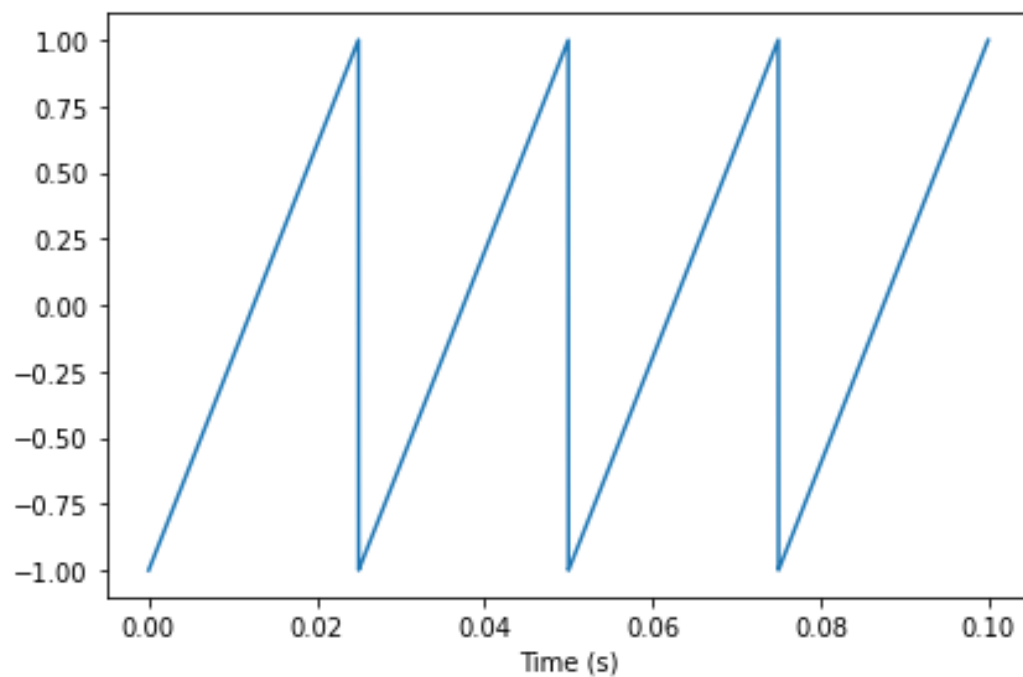


Максимальная погрешность составляет всего 0.00464956318805787.

Глава 3

Задание 9.3

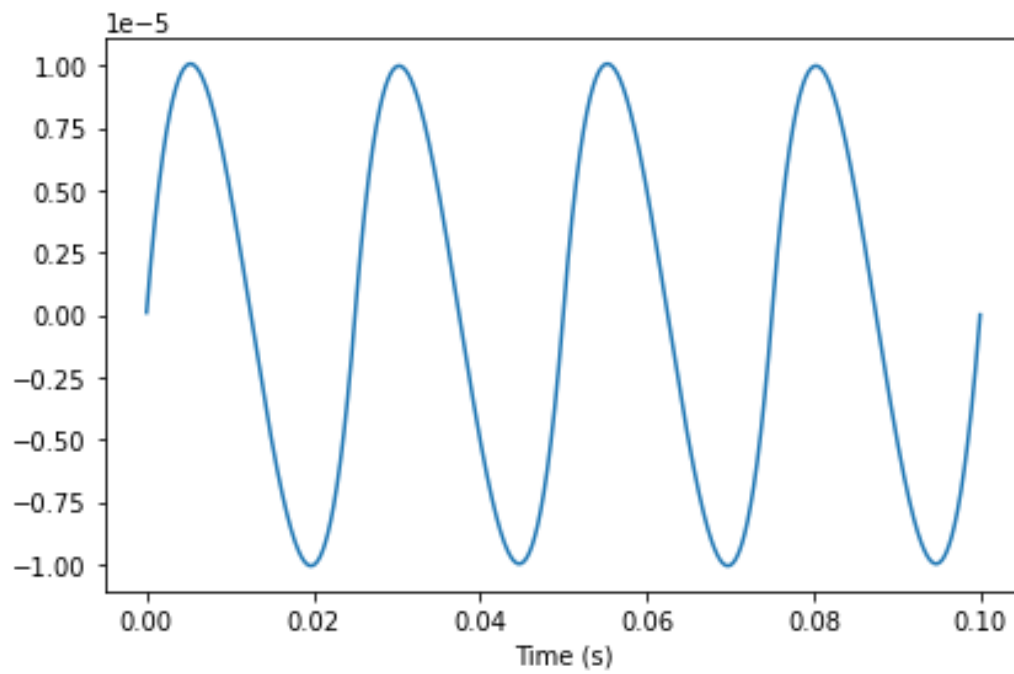
В данном задании требуется исследовать влияние двойного интегрирования на пилообразный сигнал. Создадим и выведем его:



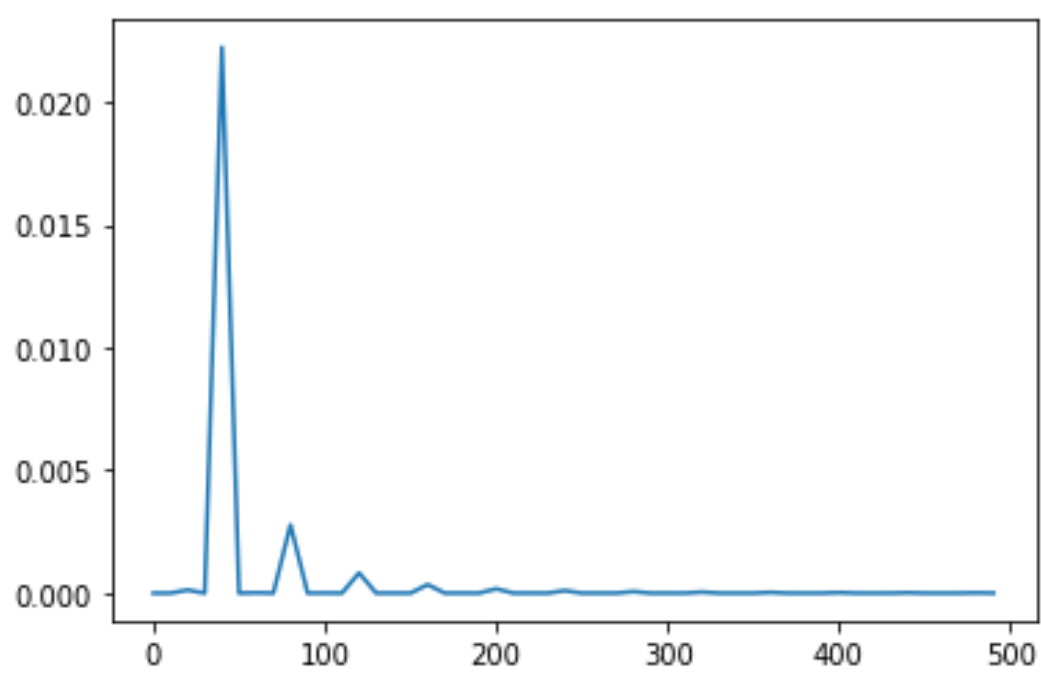
Далее, требуется вычислить его спектр и применить к нему дважды функцию `integrate`:

```
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)')
```



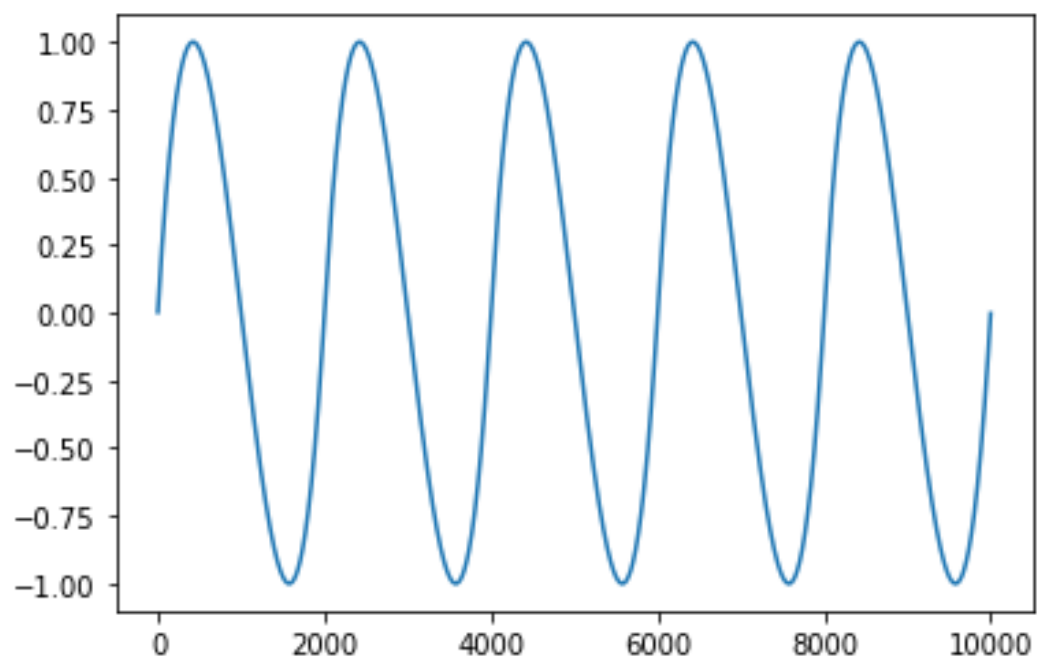
Полученная функция имеет кубический вид, близкий к синусоидальному. Это объясняется тем, что интегрирование отфильтровало большую часть высоких частот:



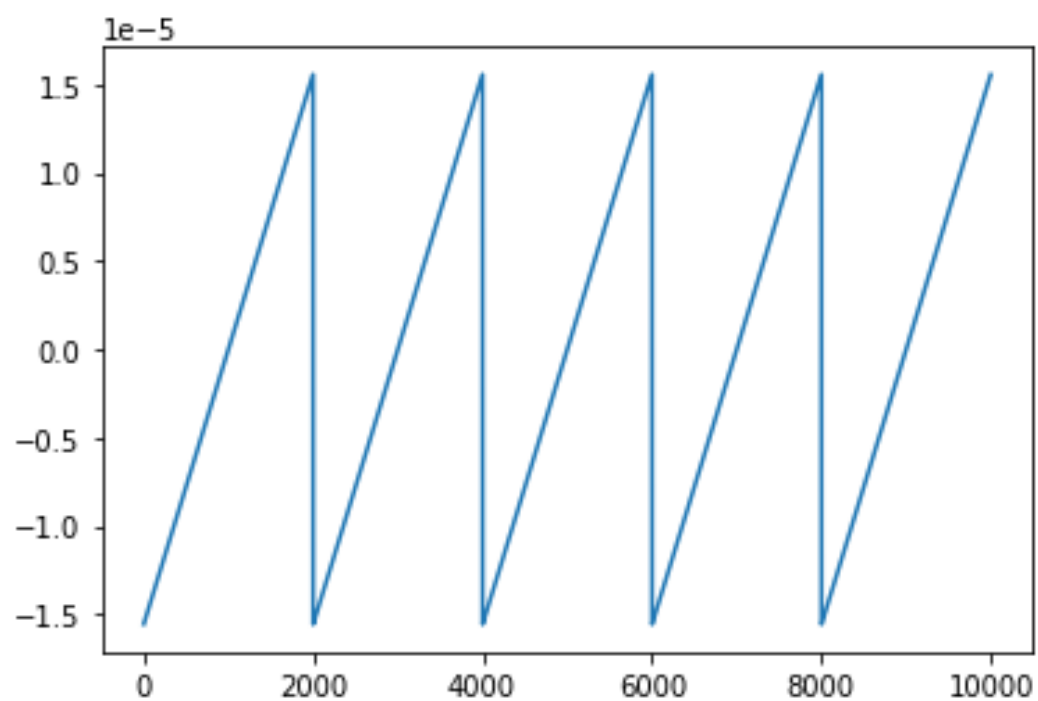
Глава 4

Задание 9.4

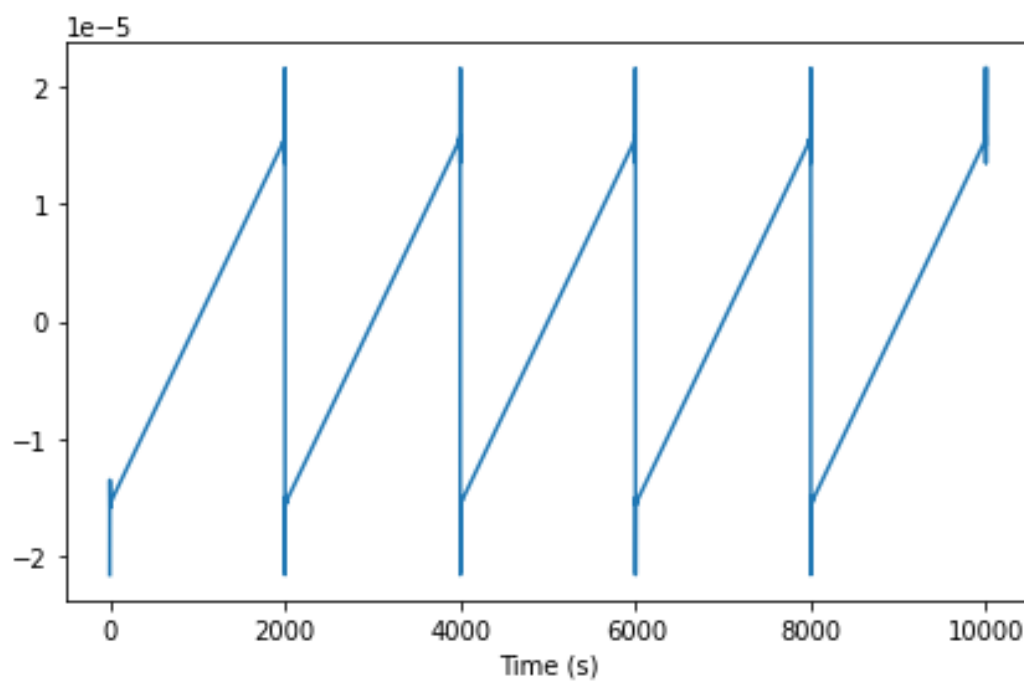
В этом задании требуется исследовать влияние двойного дифференцирования на кубический сигнал. Создадим и выведем его:



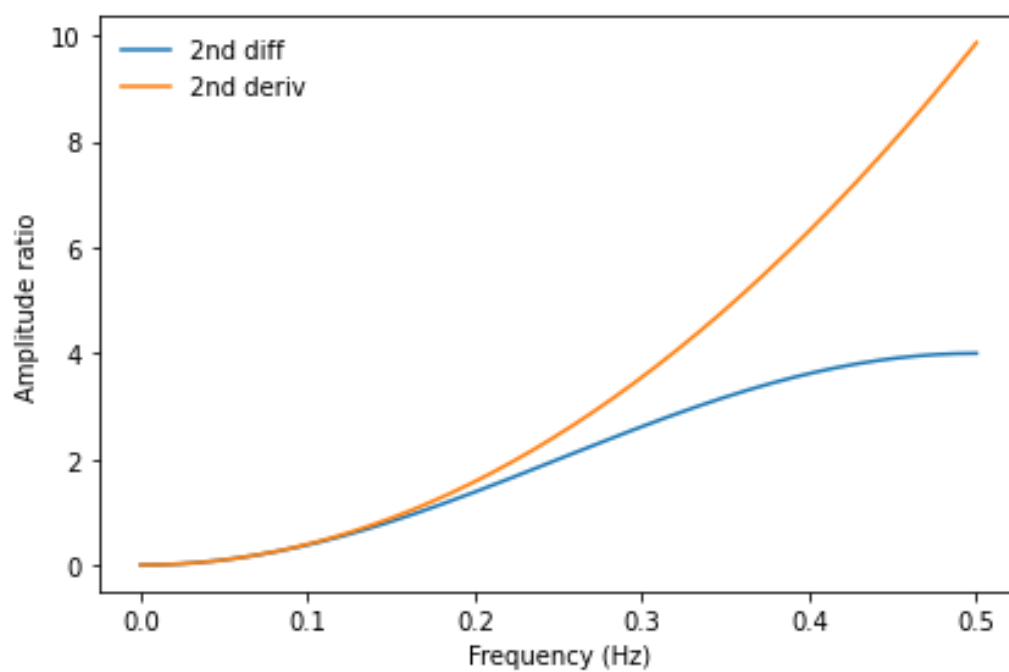
Двукратно применим `diff` и получим (ожидаемо) пилообразный сигнал:



Подобен тому, что было проделано в предыдущих упражнениях, вычислим спектр и дважды применим `differentiate`. Результат аналогичен, в точках разрыва функция зашумлена:



Наконец, можно сравнить фильтры для второй разницы и второй производной:



Как легко заметить по графику, оба фильтра усиливают высокие частоты, причем вторая производная делает это намного более явно с увеличением частоты, чем вторая разность.