

Лабораторная работа №9
Дифференцирование и интегрирование

Смирнов Никита

5 мая 2021 г.

Оглавление

1	Упражнение 9.1	4
2	Упражнение 9.2	5
3	Упражнение 9.3	8
4	Упражнение 9.4	12
5	Упражнение 9.5	14
6	Выводы	18

Список иллюстраций

2.1	Треугольный сигнал	5
2.2	Визуализация при использовании <code>diff</code>	6
2.3	Визуализация при использовании <code>differentiate</code>	7
3.1	Прямоугольный сигнал	8
3.2	Визуализация при использовании <code>cumsum</code>	9
3.3	Визуализация при использовании <code>integrate</code>	10
3.4	Сравнение волн	10
4.1	Создание сигнала	12
4.2	Двойное применение <code>integrate</code>	13
5.1	Создание сигнала	14
5.2	Применение <code>diff</code>	15
5.3	16
5.4	Визуализация двух фильтров(синий - <code>diff</code> , оранжевый - <code>deriv</code>)	16

Листинги

2.1	Создание треугольного сигнала	5
2.2	Визуализация при <code>diff</code>	5
2.3	Визуализация при <code>differentiate</code>	6
3.1	Создание сигнала	8
3.2	Визуализация при <code>cumsum</code>	8
3.3	Визуализация при <code>integrate</code>	9
3.4	Сравнение волн	10
3.5	Разница между реализациями	11
4.1	Создание сигнала	12
4.2	двойное применение <code>integrate</code>	12
5.1	Создание сигнала	14
5.2	Применение <code>diff</code>	14
5.3	Визуализация двух фильтров	16

Глава 1

Упражнение 9.1

В данном упражнении нам нужно открыть `chap09.ipynb`, прочитать пояснения и запустить примеры. Поэтому я просто изучил все примеры с комментариями.

Глава 2

Упражнение 9.2

Создаю треугольный сигнал `TriangleSignal`:

```
1 in_wave = thinkdsp.TriangleSignal(freq=30).make_wave(duration=0.1,  
    framerate=44100)  
2 in_wave.plot()  
3 thinkplot.config(xlabel='Time (s)')
```

Листинг 2.1: Создание треугольного сигнала

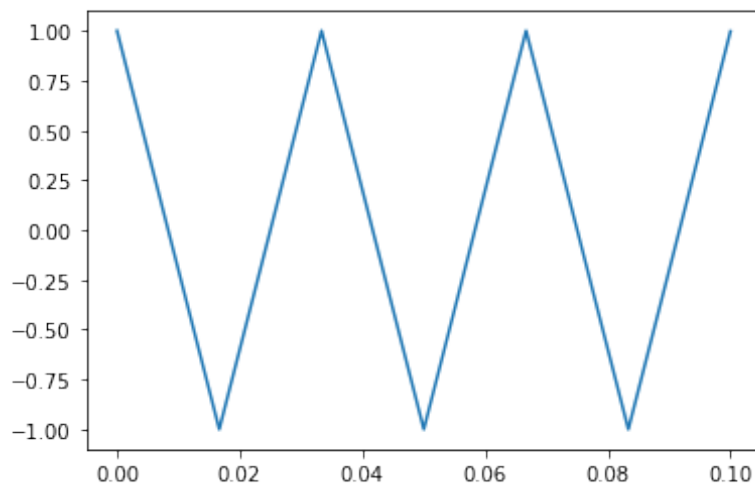


Рис. 2.1: Треугольный сигнал

Применяю `diff` от треугольной функции - прямоугольная функция. Можно сказать, что гармоники прямоугольной и треугольной гармоники совпадают по $1/f$ и $1/f^2$.

```

1 out_wave = in_wave.diff()
2 out_wave.plot()

```

Листинг 2.2: Визуализация при `diff`

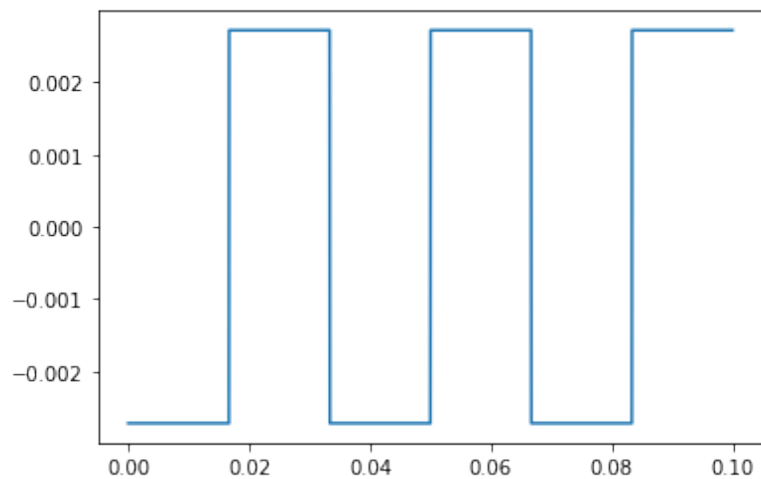


Рис. 2.2: Визуализация при использовании `diff`

Когда мы берём спектральную производную, мы получаем "звон" вокруг разрывов.

```

1 out_wave2 = in_wave.make_spectrum().differentiate().make_wave()
2 out_wave2.plot()

```

Листинг 2.3: Визуализация при `differentiate`

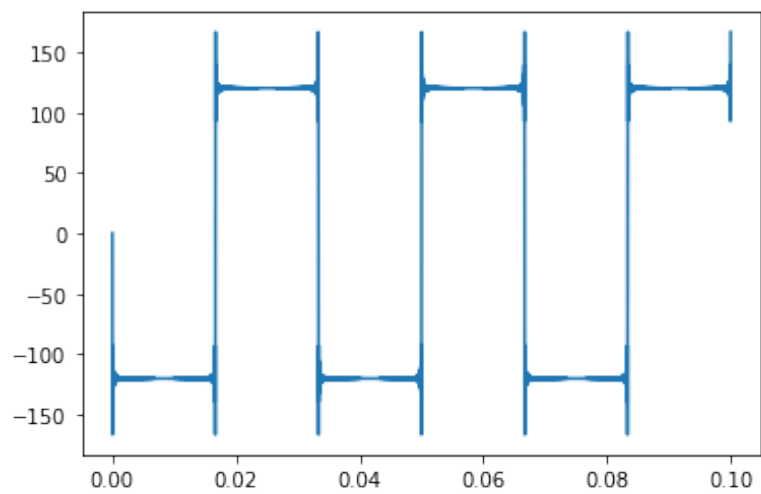


Рис. 2.3: Визуализация при использовании `differentiate`

Различия между `diff` и `differentiate` заключается в том, что производная треугольной волны не определена в точках треугольника.

Глава 3

Упражнение 9.3

Для начала я создал прямоугольный сигнал.

```
1 in_wave = thinkdsp.SquareSignal(freq=30).make_wave(duration=0.1,  
    framerate=44100)  
2 in_wave.plot()
```

Листинг 3.1: Создание сигнала

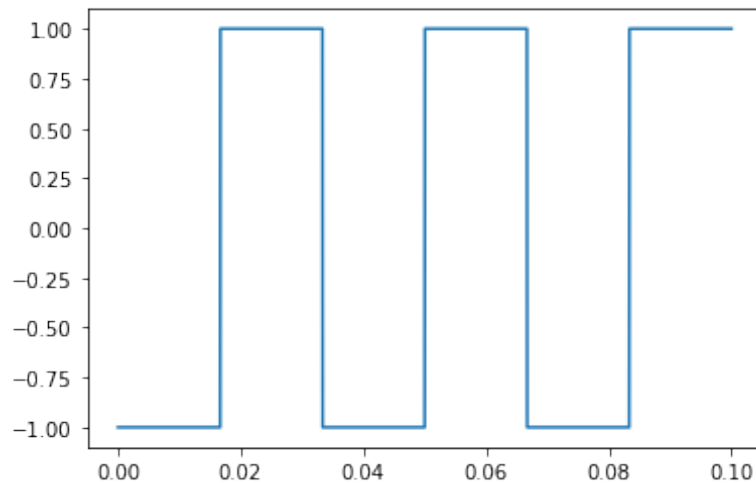


Рис. 3.1: Прямоугольный сигнал

Мы получим треугольный сигнал из прямоугольного, что довольно логично после выполнения предыдущего упражнения.

```
1 out_wave = in_wave.cumsum()  
2 out_wave.plot()
```

```
3 thinkplot.config(xlabel='Time (s)')
```

Листинг 3.2: Визуализация при cumsum

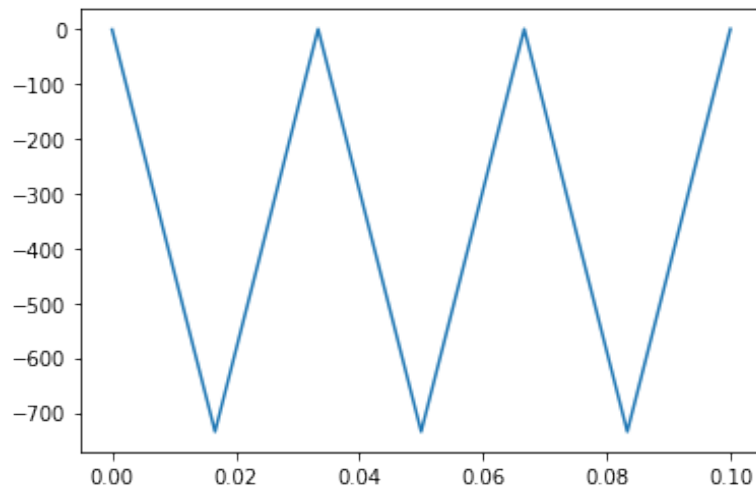


Рис. 3.2: Визуализация при использовании cumsum

Создаем спектр сигнала и используем функцию `integrate` и получаем новый сигнал на его основе.

```
1 spectrum = in_wave.make_spectrum().integrate()
2 spectrum.hs[0] = 0
3 out_wave2 = spectrum.make_wave()
4 out_wave2.plot()
```

Листинг 3.3: Визуализация при integrate

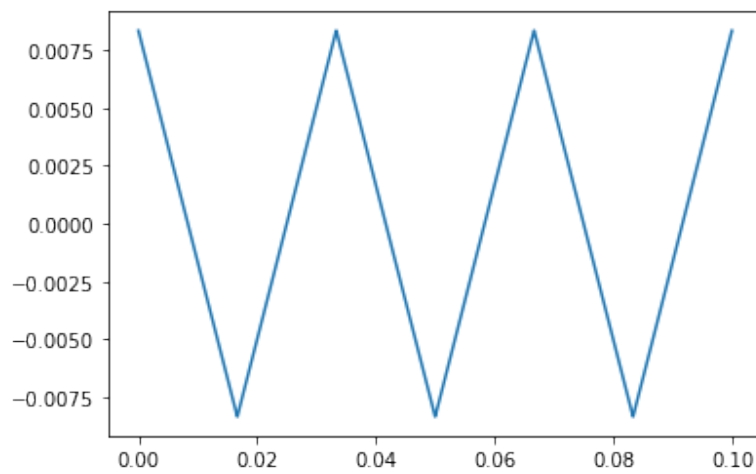


Рис. 3.3: Визуализация при использовании `integrate`

Если уравновесить и нормализовать две волны, они будут визуально похожи.

```

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

```

Листинг 3.4: Сравнение волн

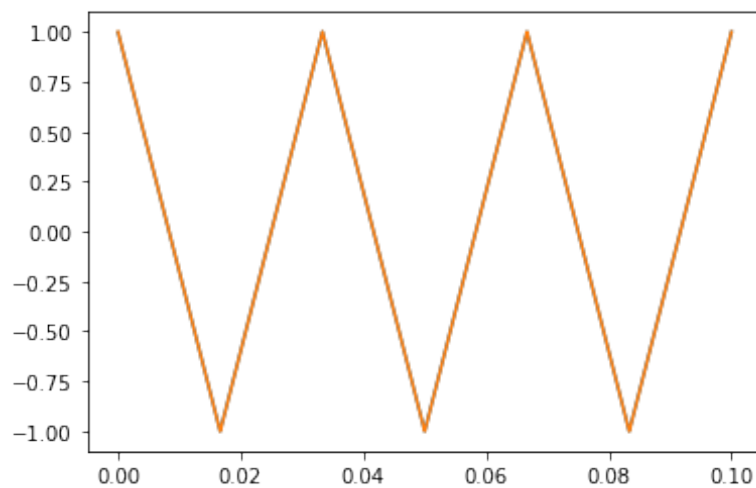


Рис. 3.4: Сравнение волн

```
1 max(abs(out_wave.ys - out_wave2.ys))
```

Листинг 3.5: Разница между реализациями

Считаем разницу между реализациями и получаем 0.0027210884353741083.

Глава 4

Упражнение 9.4

Создаю пилообразный сигнал.

```
1 in_wave = thinkdsp.SawtoothSignal(freq=30).make_wave(duration=0.1,  
    framerate=44100)  
2 in_wave.plot()
```

Листинг 4.1: Создание сигнала

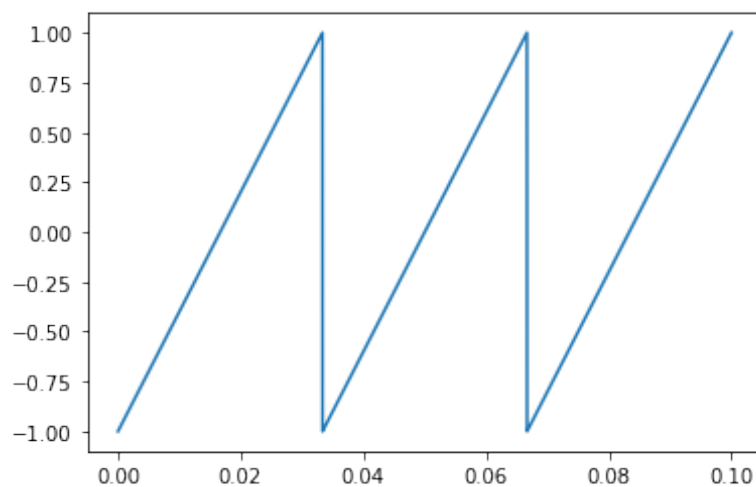


Рис. 4.1: Создание сигнала

Вычисляем спектр и применяем к нему функцию `integrate` два раза.

```
1 out_wave = in_wave.cumsum()  
2 out_wave.unbias()  
3 out_wave.plot()  
4 out_wave = in_wave.cumsum()
```

```
5 out_wave.unbias()  
6 out_wave.plot()
```

Листинг 4.2: двойное применение `integrate`

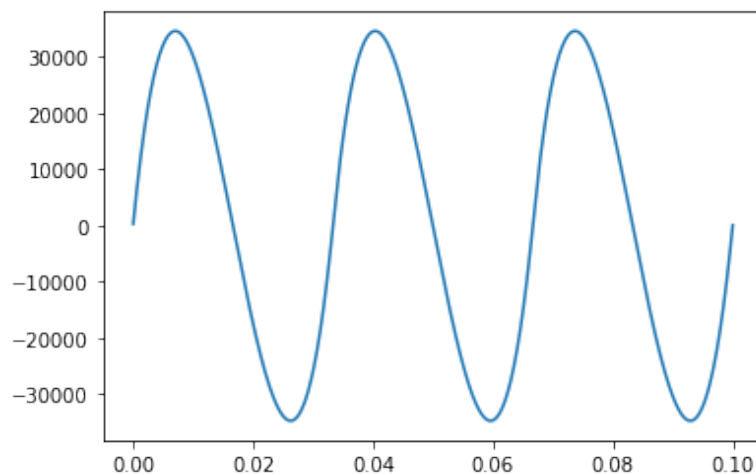


Рис. 4.2: Двойное применение `integrate`

Двойное интегрирование дает кубическую кривую. На этом этапе результат всё больше и больше напоминает синусоиду. Причина в том, что интеграция действует как фильтр нижних частот.

Глава 5

Упражнение 9.5

Создадим кубически сигнал.

```
1 in_wave =  
    thinkdsp.CubicSignal(freq=0.0005).make_wave(duration=10000,  
    framerate=1)  
2 in_wave.plot()
```

Листинг 5.1: Создание сигнала

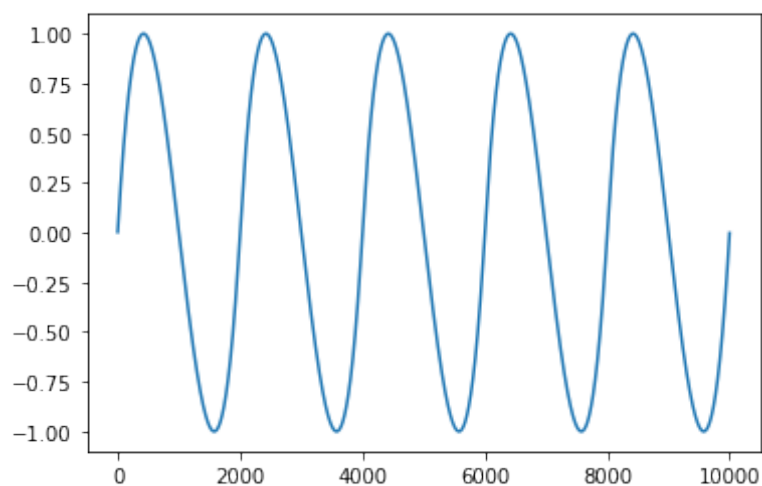


Рис. 5.1: Создание сигнала

Дважды применим функцию `diff` и получи пилообразный сигнал, что довольно логично после выполнения предыдущего упражнения.

```
1 out_wave = in_wave.diff().diff()
```

```
2 out_wave.plot()
```

Листинг 5.2: Применение diff

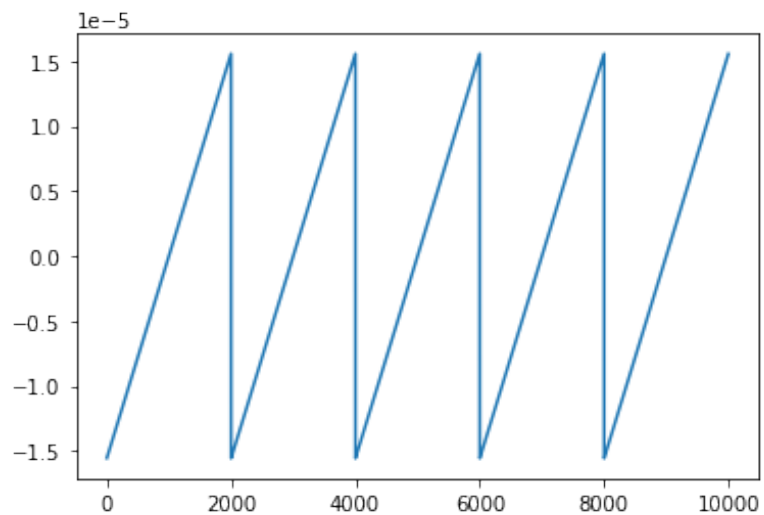


Рис. 5.2: Применение diff

Когда мы дифференцируем дважды, получаем пилообразную форму с некоторым звоном. Проблема в том, что производная параболического сигнала в точках не определена.

```
1 spectrum = in_wave.make_spectrum().differentiate().differentiate()
2 out_wave2 = spectrum.make_wave()
3 out_wave2.plot()
```

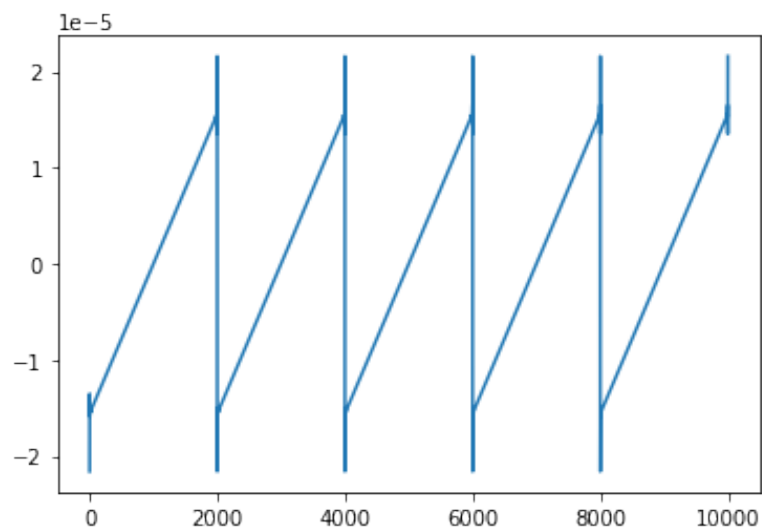



Рис. 5.3:

В конце работы я вывел фильтры для второй разницы и второй производной фильтры и сравнил их:

Рассмотрим два фильтра в одном масштабе:

```

1 diff_filter.plot(label='2nd diff')
2 deriv_filter.plot(label='2nd deriv')

```

Листинг 5.3: Визуализация двух фильтров

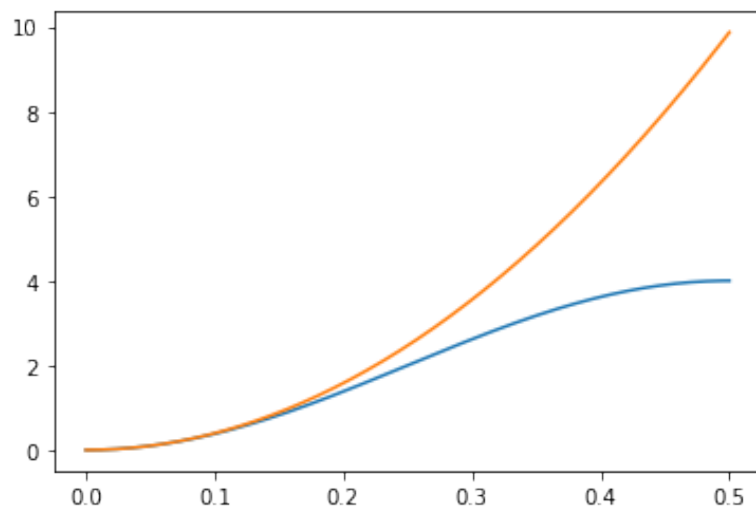


Рис. 5.4: Визуализация двух фильтров(синий - diff, оранжевый - deriv)

Теперь мы можем видеть, что оба являются фильтрами верхних частот, которые усиливают компоненты самых высоких частот. Вторым `deriv` параболический, поэтому он сильнее всего усиливает самые высокие частоты. Вторым `diff` - хорошее приближение второй производной только на самых низких частотах, затем он существенно отклоняется.

Глава 6

Выводы

Во время выполнения лабораторной работы получены навыки работы с взаимосвязью между окнами во временной области и фильтрами в частотной области. Также изучалось влияние окна конечных разностей, которое приближает дифференцирование, и операции накопления суммы, которая приближает интегрирование.