

# Теория Фильтрации и Прогнозирование данных

Липатов Данила МСМТ 243  
Лабораторная работа №3

## Пункт 1

Для построения CWT было принято использовать вейвлет Морле (morl) в библиотеке pywt.  
Результаты для scale от 1 до 91 представлены ниже:

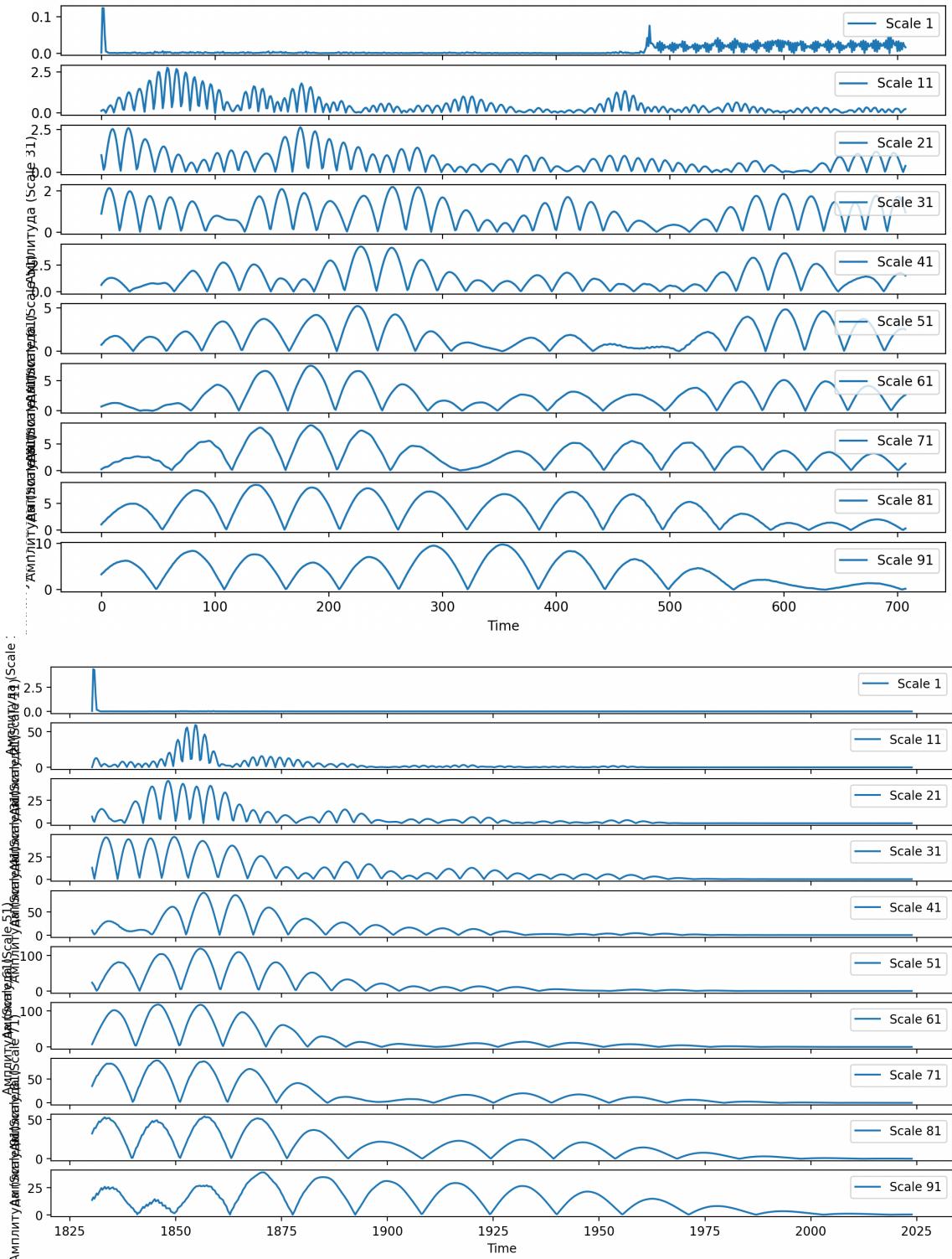


Рис. 1 Графики X , Y относительно времени в годах.

Y затухает после, примерно, 1975 года. График отно времени представлен ниже:

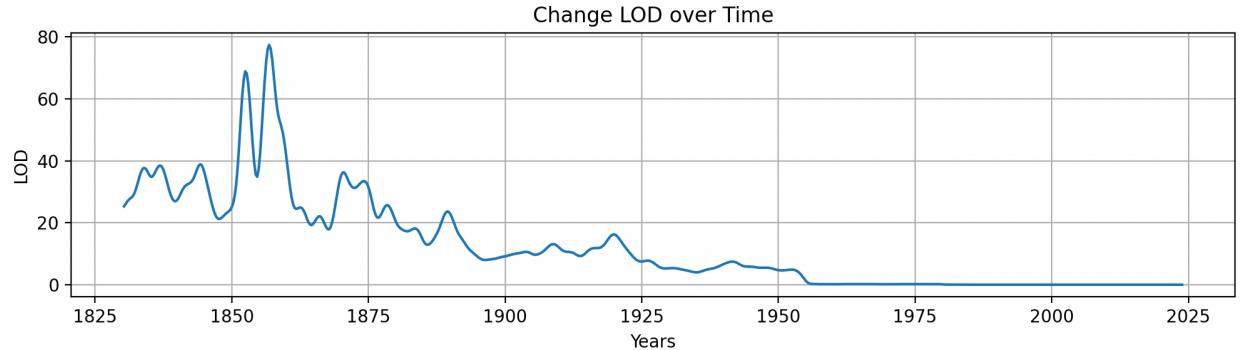
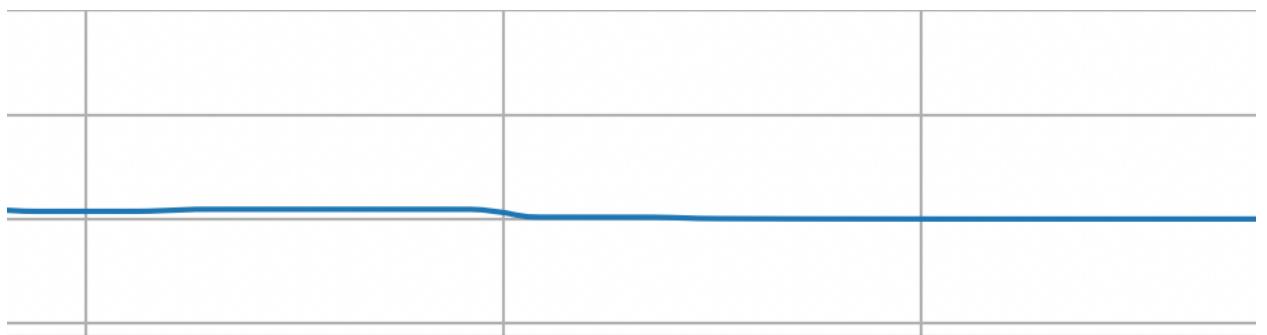
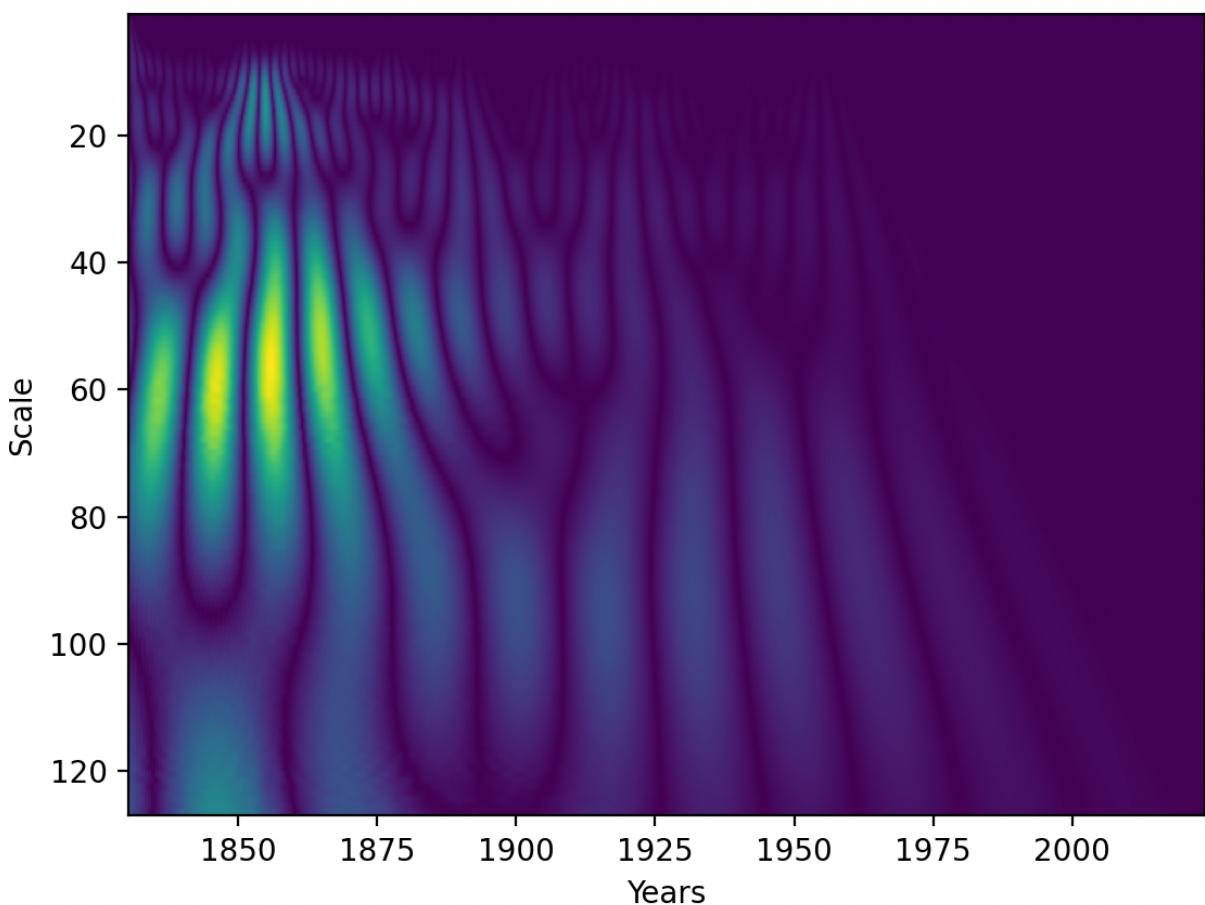


Рис. 1.2 График Y отн-но времени в годах.

Причем он продолжает колебаться после 1975 года, но не так сильно:



Вейвлет-скелограмма:



Самые большие колебания были до 1900 года, это можно связать с тем, что после 1900 года (в частности после 1959-ых годов) были изобретены технологии и подходы, которые позволили не только фиксировать мелкие колебания, но и более детально анализировать их, что сделало возможным выделение влияния сезонных, атмосферных, океанических и приливных эффектов на вращение Земли.

Ниже представлено изображение с большим скейлом

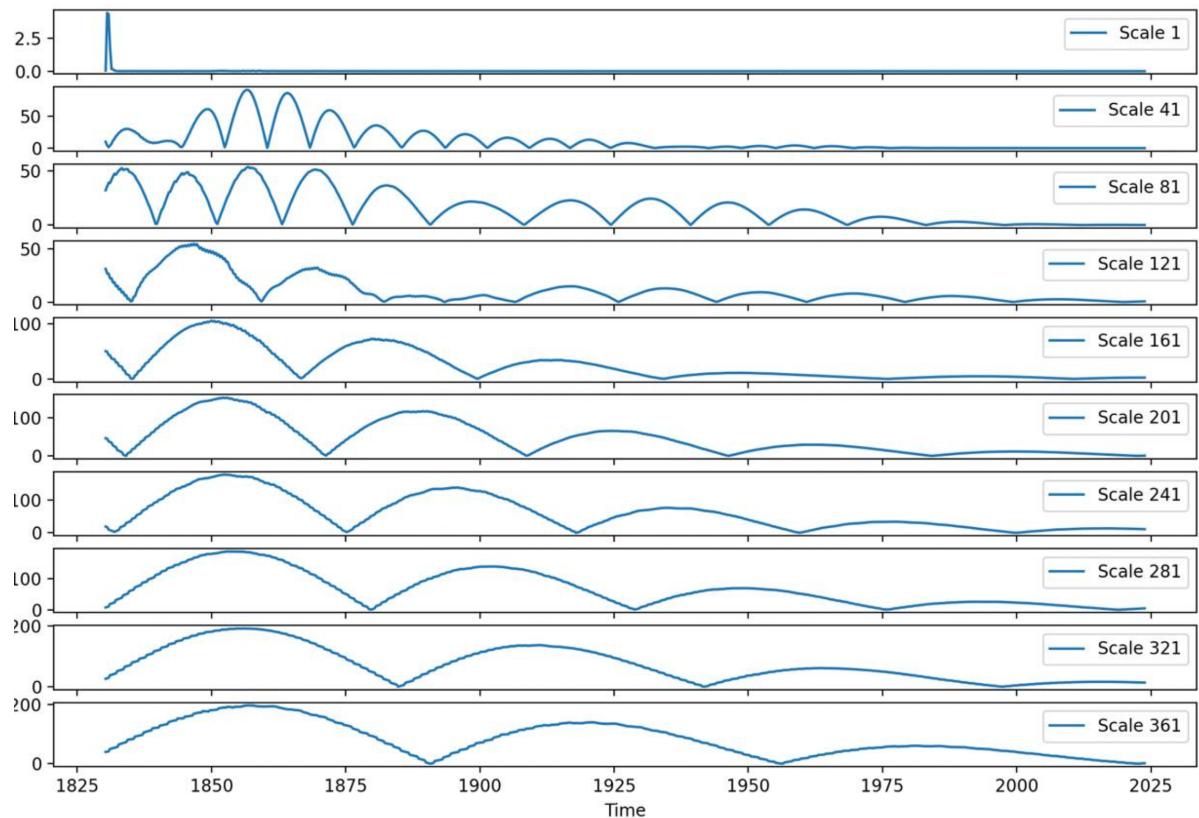


Рис. 1.3 Вейвлет преобразование LOD(Y)

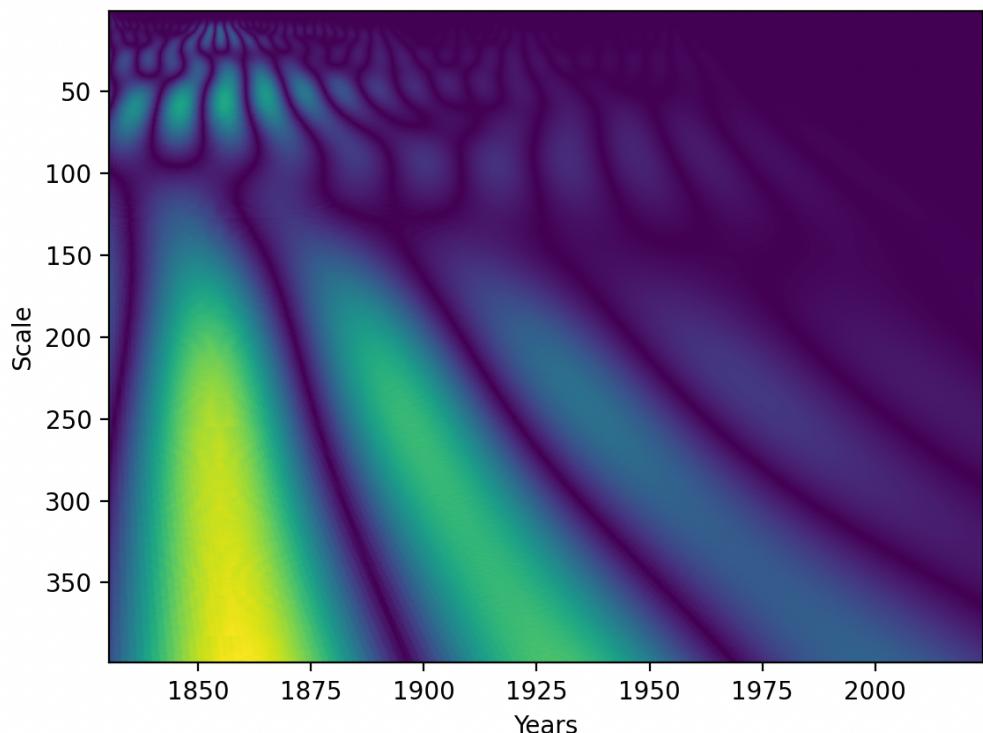


Рис. 1.4 Вейвлет преобразование скейлограмма LOD(Y)

## Пункт 2

Для пункта 2 рассматривался сигнал из лабораторной работы №1 и, соответственно, добавлялся шум

```
ar_params = [0.125, -0.55, -0.15, 0.35] # коэффициенты AR(4)
```

```
noise = np.random.normal(0, 1.1, len(X))
```

```
ar_noise = np.zeros(len(X))
for t in range(4, len(X)):
    ar_noise[t] = 0 + ar_params[0] * ar_noise[t - 1] + ar_params[1] * ar_noise[t - 2] +
    ar_params[2] * ar_noise[t - 3] + ar_params[3] * ar_noise[t - 4] + noise[t]

# Сигнал с добавленным шумом
noisy_signal = X + ar_noise
```

Полученные графики на Рис. 2:

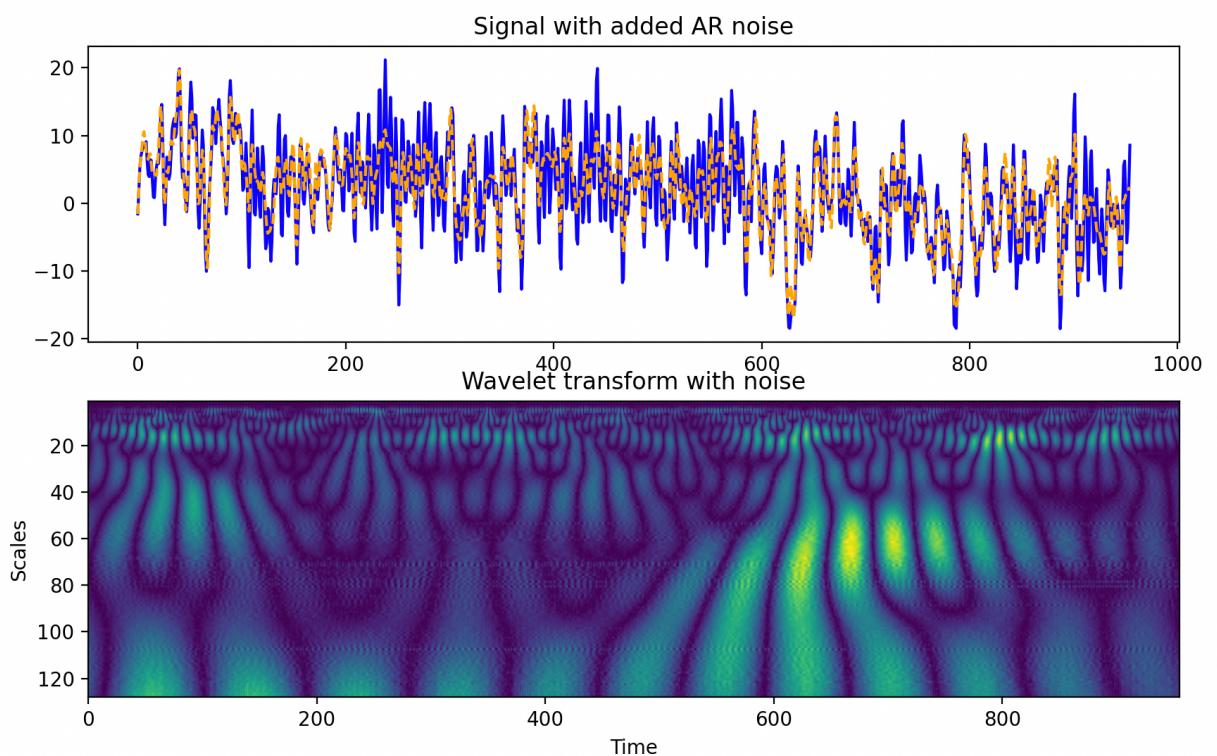


Рис. 2 Вейвлет для сигнала из Лаб.1

Сигнал с шумом наглядно отличается от исходного сигнала, а его Вейвлет преобразование отражает колебания

Масштаб контролирует "размер" вейвлета: высокий масштаб соответствует низкой частоте, а низкий масштаб — высокой частоте.

Окрашивание указывает на амплитуду вейвлет-коэффициентов на определённой частоте (или масштабе) в определённый момент времени. Чем ярче цвет, тем выше амплитуда вейвлет-коэффициента для этой частоты и времени, что говорит о сильном присутствии этой частотной компоненты в данном временном участке сигнала.

То есть, примерно, на 610 точке массива данных амплитуда довольно высокая, обращаясь к графику выше заметно резкое падение до значения -20 именно в этом диапазоне.

### Пункт 3

В Python можно так же визуализировать графики в пространстве, например, рассмотрим вейвлет Морле для того же сигнала с шумом:

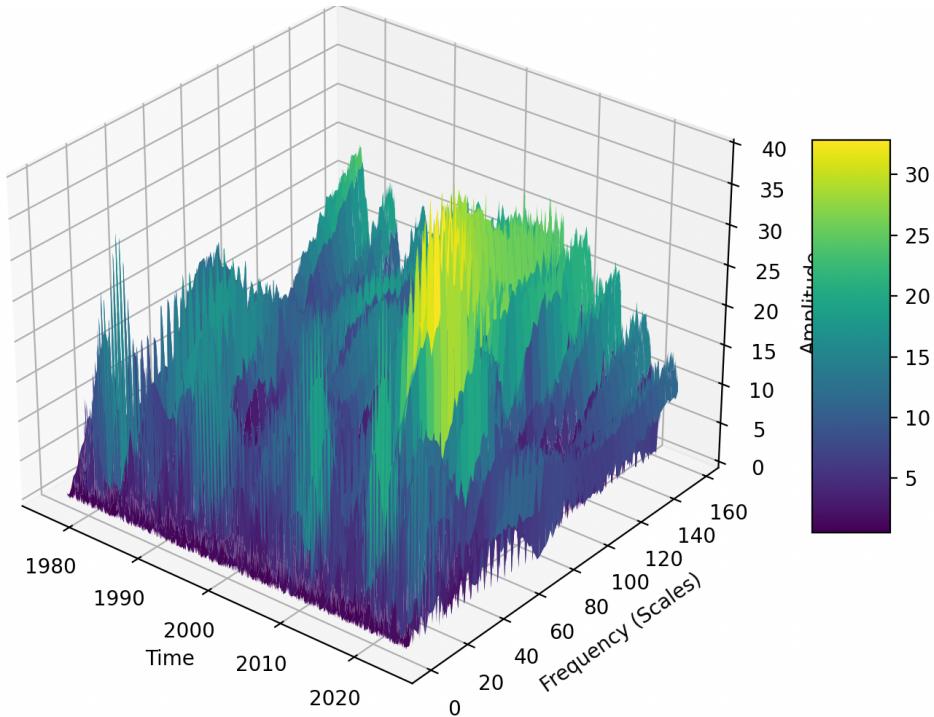


Рис. 3 3D вейвлет для X и Y

Если обратиться к предыдущей лабораторной работе, то как раз наибольшие колебания происходят после 2010-ых годов, что в целом совпадает с представленным сигналом с наличием шума в данных.

### Пункт 4

Для построения сигнала с импульсом необходимо индекс вы массиве сигнала увеличить, например:

```
impulse_position = [100, 200, 300, 400, 542, 232, 111]
impulse_magnitude = 100
impulsive_signal = noisy_signal.copy()
for ind in impulse_position:
    impulsive_signal[ind] += impulse_magnitude
```

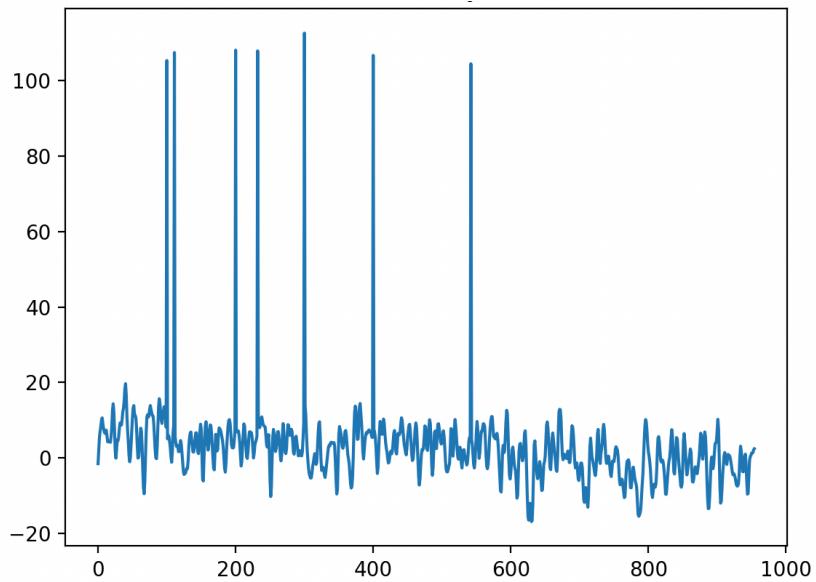


Рис. 4 Сигнал X с шумом и импульсом  
Wavelet transform of a impulse signal

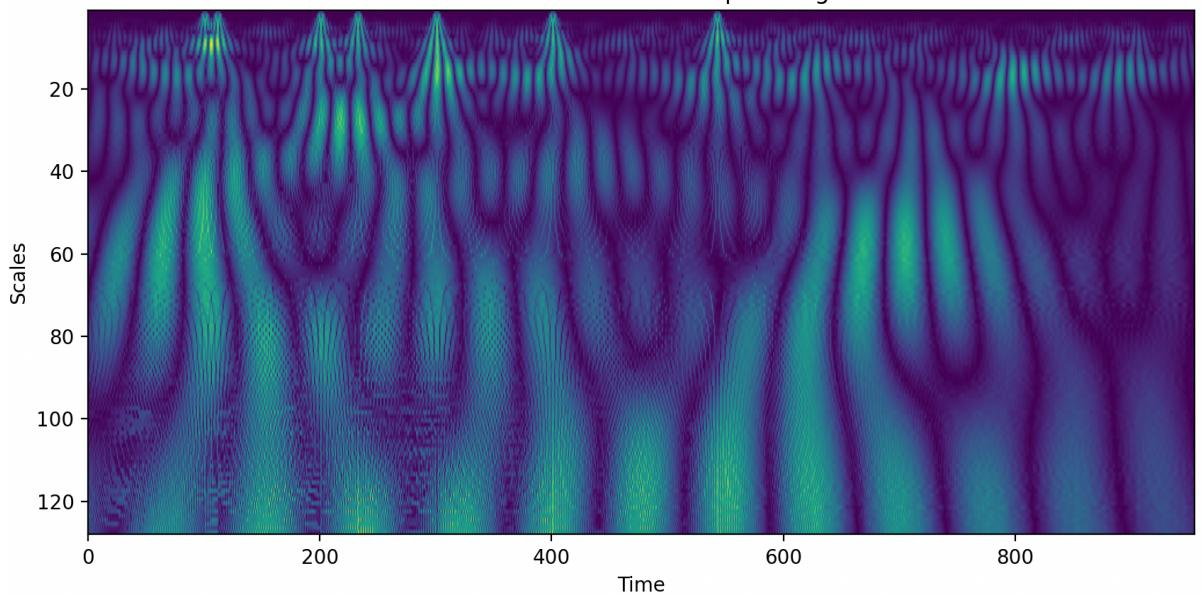


Рис. 5 Вейвлет-преобразование сигнала X с шумом и импульсом

Чтобы заметить импульс, достаточно заметить более яркие пики сверху вейвлет-скелограммы, именно в тех индексах, куда был добавлен импульс заметны более яркие пики.

Как было сказано ранее, масштаб(скейл) - определяет частотный диапазон анализа. Более высокие масштабы соответствуют более низким частотам (большим периодам), а более низкие масштабы – высоким частотам (коротким периодам). Выбор правильного диапазона масштабов позволяет анализировать как кратковременные, так и долгосрочные колебания в данных.

То есть, когда используются малые значения скейла, вейвлет сжимается, что позволяет ему лучше наблюдать короткие и быстрые изменения сигнала, т.е. высокочастотные колебания. В то же время, большие скейлы реагируют на медленные, долговременные изменения в сигнале, отображая низкие частоты.

Ниже представлен пример того же сигнала с шумом, но скейл был взят меньше от 1 до 40, по итогу мы видим больше ярких участков

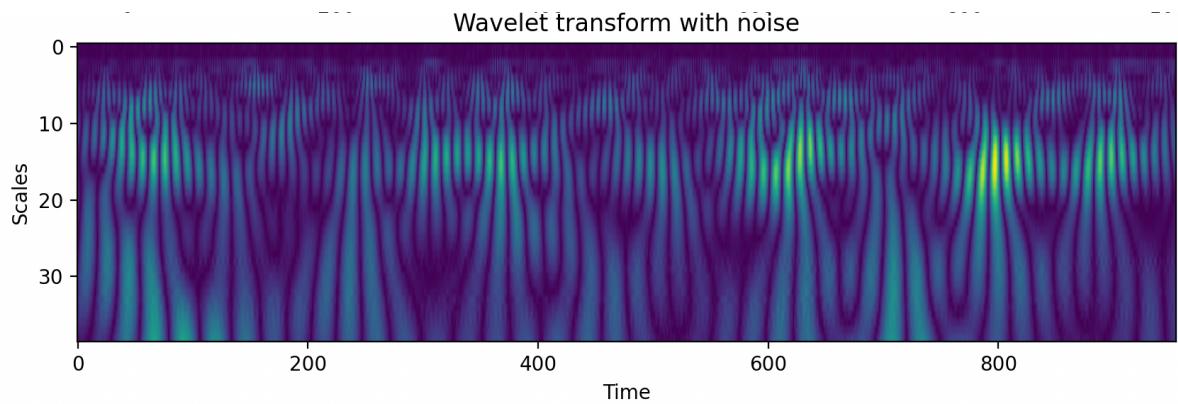


Рис. 6 Вейвлет-преобразование сигнала X с шумом и импульсом, с обозначением треугольника влияния на малых скейлах

Вейвлет `amor` напрямую не реализован в Python. Судя по доступным документациям, данный тип реализован именно в Matlab.

Однако, судя по документации, в Python есть вейвлет `cmor`, который является аналогом `amor`. Приведем результаты вейвлет преобразования

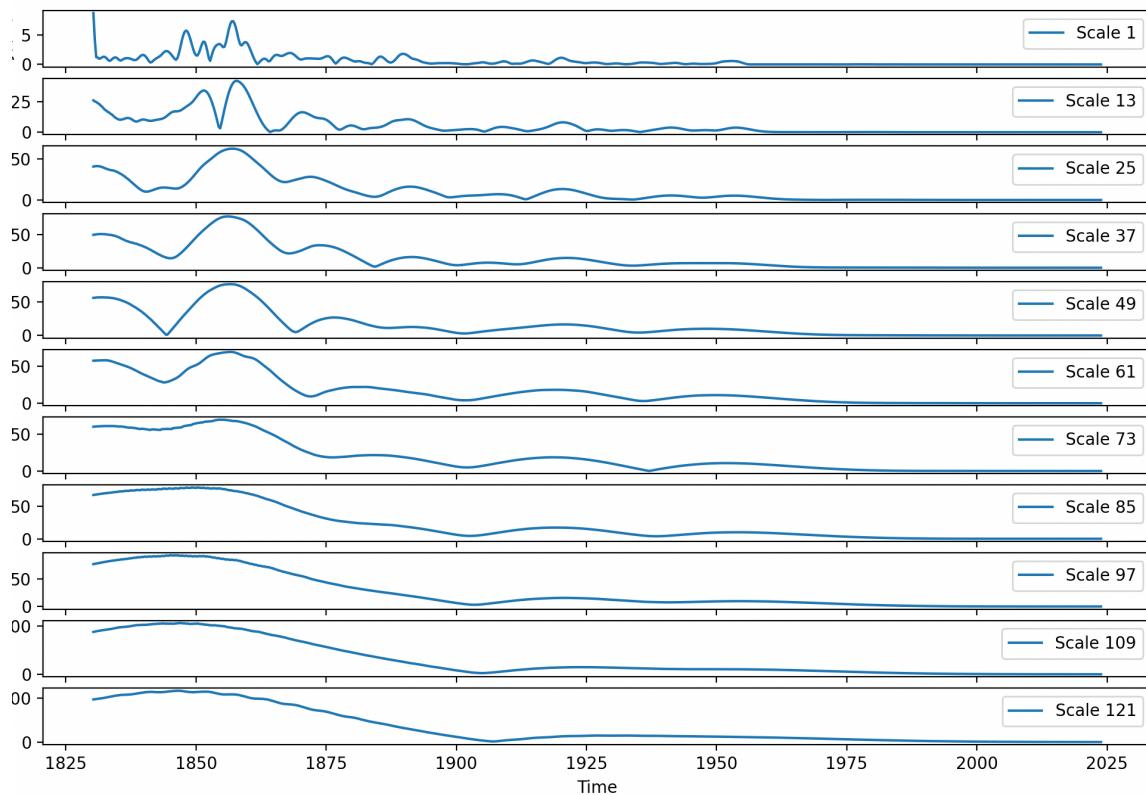


Рис. 7 Вейвлет-преобразование сигнала LOD(Y) `cmor`

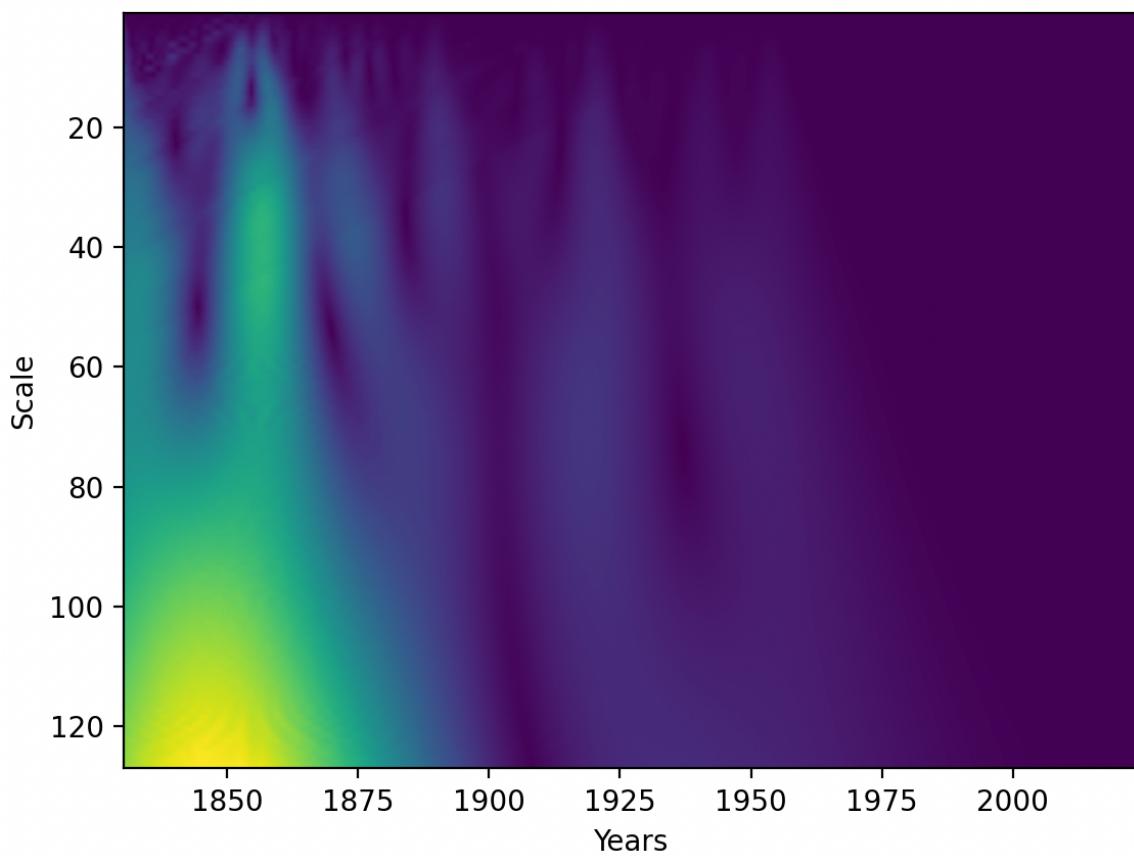


Рис. 8 Вейвлет-преобразование сигнала LOD(Y) смор

В отличии от morl, смор более плавный и более расплывчатый, не видно каких то резких всплесков, как на предыдущих рисунках, как на обычный графиках, так и на скейлограмме

Wavelet transform of a impulse signal

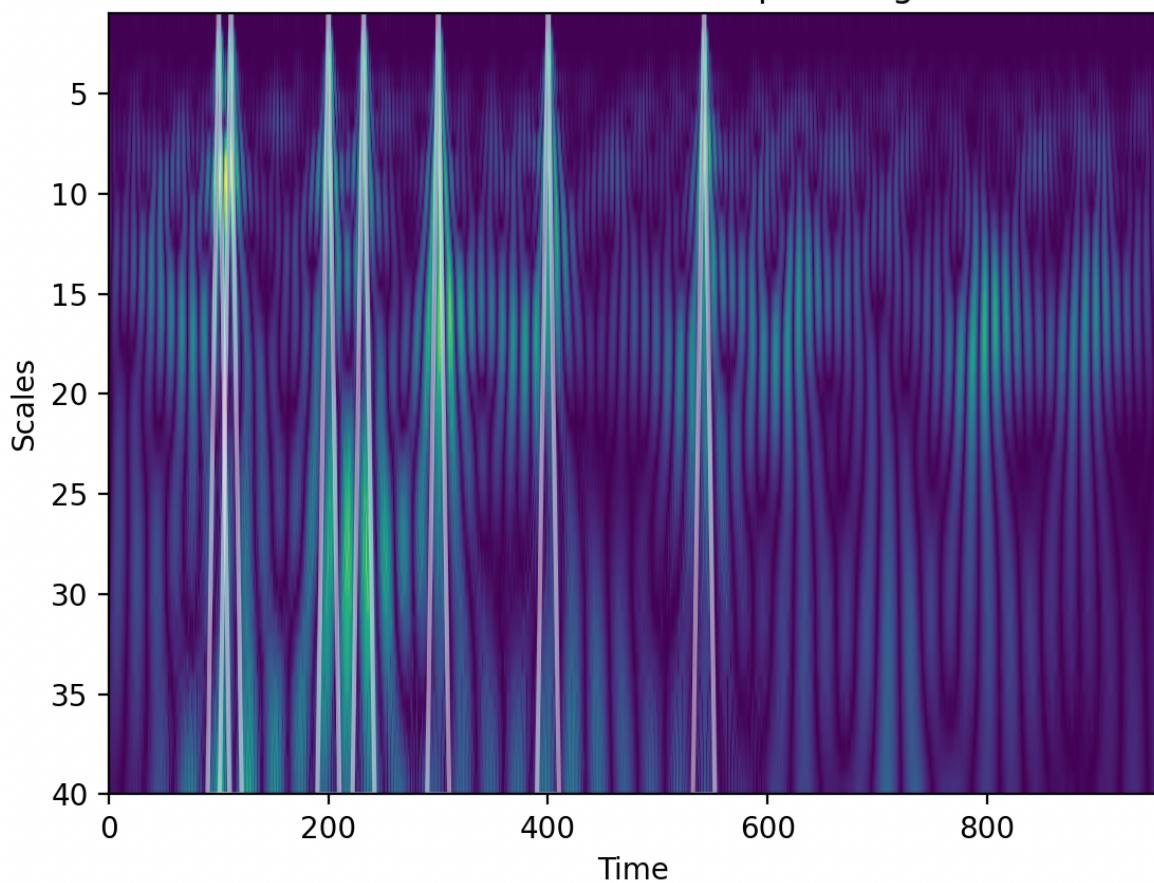


Рис. 7 Вейвлет-преобразование сигнала X с шумом и импульсом, с обозначением треугольника влияния на малых скейлах

Wavelet transform of a impulse signal

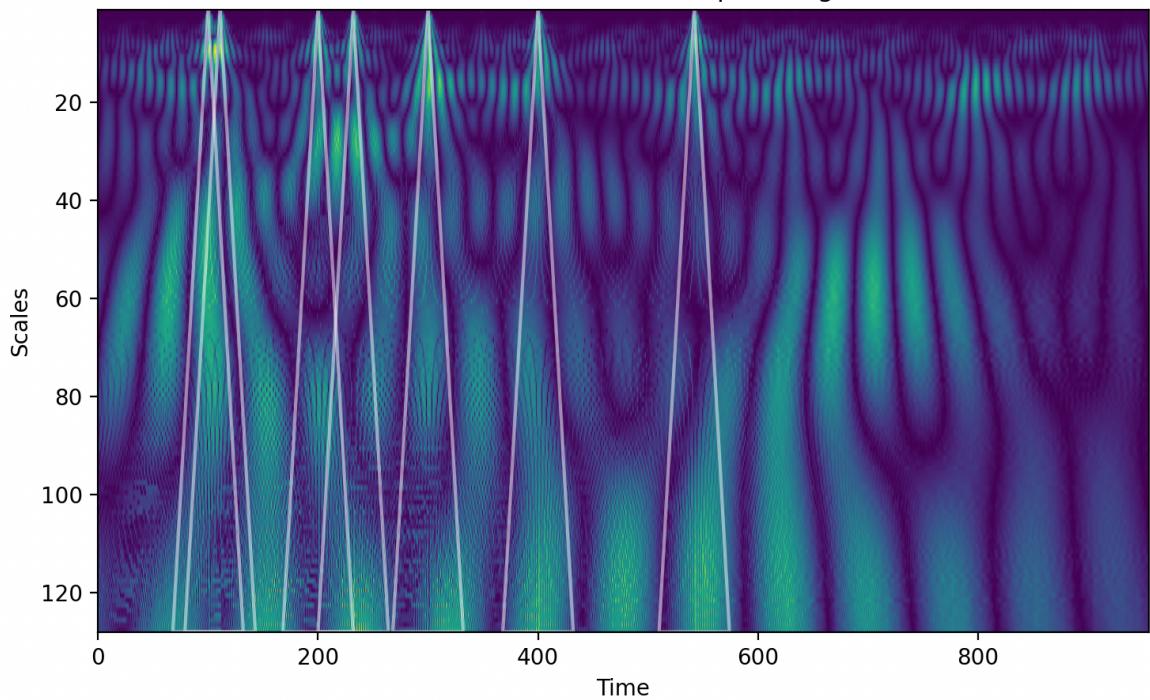


Рис. 8 Вейвлет-преобразование сигнала X с шумом и импульсом, с обозначением треугольника влияния на больши скейлах

В целом, что мы и хотели пронаблюдать треугольник влияния при больших скейлах, как бы, растягивается, захватывая большие отображения, нежели при малых скейлах.

Характерно для импульса, пики распознались однозначно.

#### Скрипт отображения

```
scales = np.arange(1, 128)
coeffs_imp, freqs_imp = pywt.cwt(impulsive_signal, scales=scales,
wavelet='morl')

plt.imshow(np.abs(coeffs_imp), extent=[0, len(X), 1, 128], cmap='viridis',
aspect='auto')
plt.title("Wavelet transform of a impulse signal")
# plt.show()
for pos in impulse_position:
    plt.plot([pos, pos - 64 / 2, pos + 64 / 2, pos],
             [128, 1, 1, 128], 'w-', alpha=0.5)
plt.show()
```