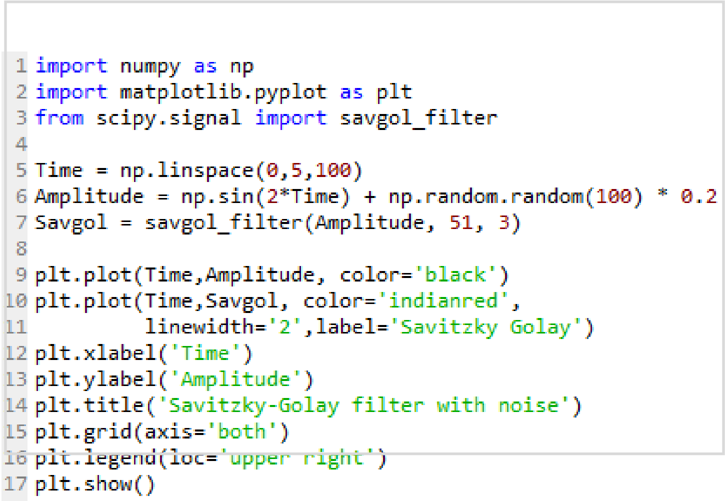
- Savitzky–Golay filter

It’s smoothed the data by taken the least square to retreat small window of the data into polynomial then take this polynomial to estimate the center of the window then the window is shifted one point forward. This process is repeated until each point perfectly set relative to its neighbors.

savgol_filter function is used after its import from scipy.signal library with parameter (1,2,3)

1. **Signal after applying noise.**
2. **Window size and its most be positive odd integer (coefficient number).**
3. **Polynomial order and it’s most be less than window length.**

Example :



```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import savgol_filter

Time = np.linspace(0,5,100)
Amplitude = np.sin(2*Time) + np.random.random(100) * 0.2
Savgol = savgol_filter(Amplitude, 51, 3)

plt.plot(Time,Amplitude, color='black')
plt.plot(Time,Savgol, color='indianred',
         linewidth='2',label='Savitzky Golay')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.title('Savitzky-Golay filter with noise')
plt.grid(axis='both')
plt.legend(loc='upper right')
plt.show()
```

Figure 1 Code for Savitzky–Golay filter

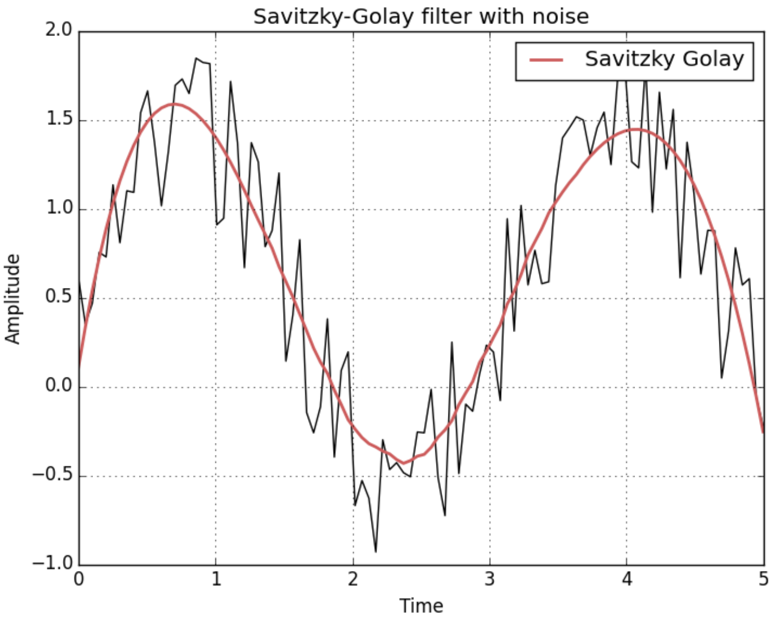


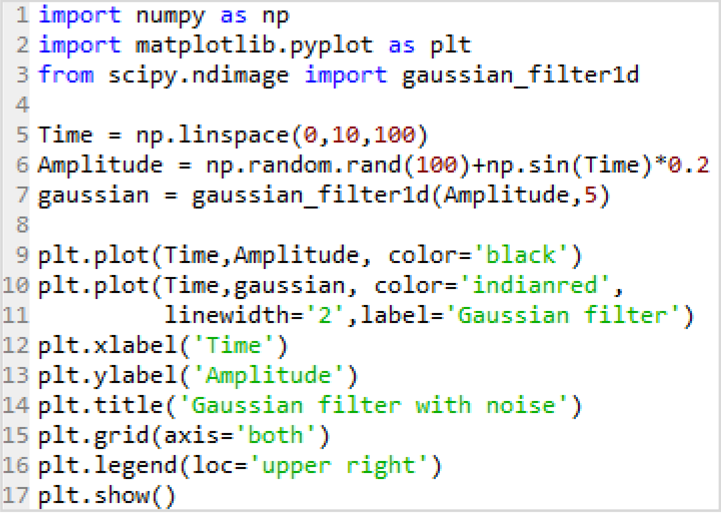
Figure 2 Savitzky–Golay filter output

- ## Gaussian filter :

Make little change to the data as they come closer to 0 and if standard deviation (which control the width)is closer to 0 its give result muach like Moving Average

gaussian_filter1d function is used after its import from scipy.ndimage library with parameter (1,2)

1- **Signal after applying noise.**
2- **Standard deviation (sigma)**



```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.ndimage import gaussian_filter1d

Time = np.linspace(0,10,100)
Amplitude = np.random.rand(100)+np.sin(Time)*0.2
gaussian = gaussian_filter1d(Amplitude,5)

plt.plot(Time,Amplitude, color='black')
plt.plot(Time,gaussian, color='indianred',
         linewidth='2',label='Gaussian filter')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.title('Gaussian filter with noise')
plt.grid(axis='both')
plt.legend(loc='upper right')
plt.show()
```

Figure 3 Code for Gaussian filter

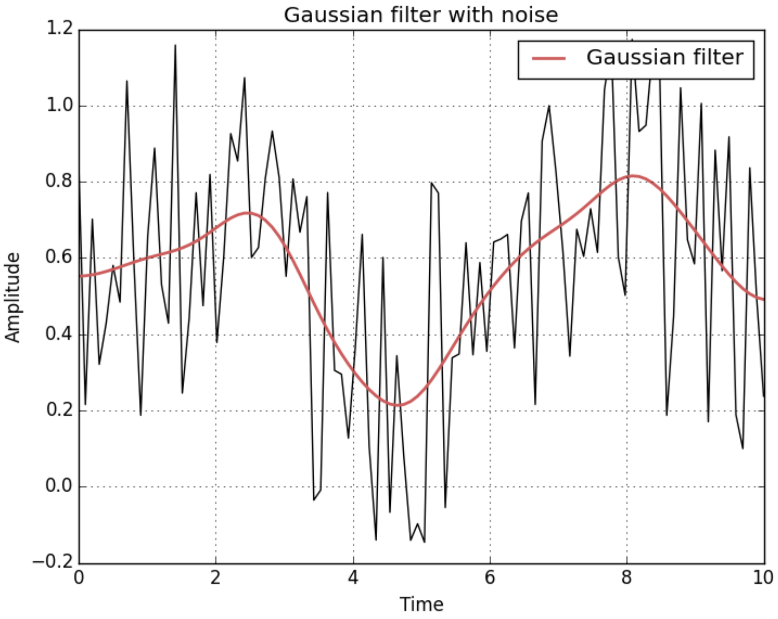


Figure 4 Gaussian filter output

- ## Median filter :

Because Savitzky-Golay and Moving Average have approximately incorrect edges Median filter is used since its maintain the edges and that why it's the best choice in digital image processing.

medfilt function is used after its import from scipy.signal library with parameter (1,2)

1. **Signal after applying noise.**
2. **Window (kernel) size and its most be positive odd integer (default is 3).**

Example from (mathworks.com) :

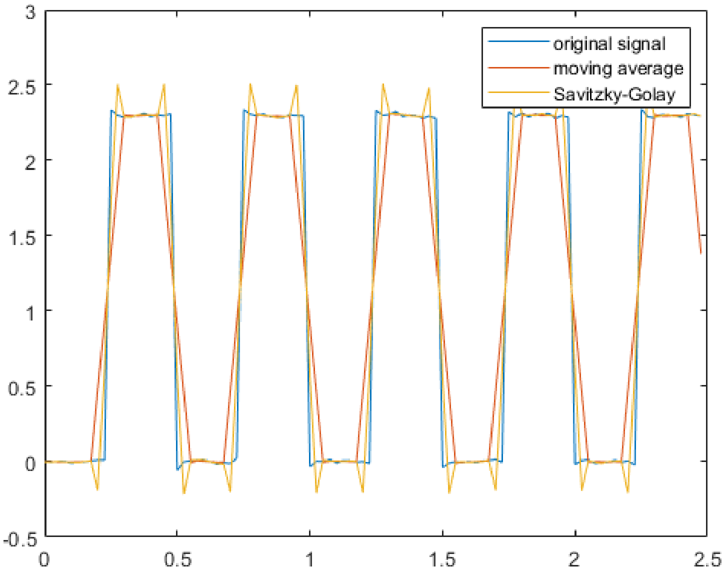


Figure 5 Comparing between filters

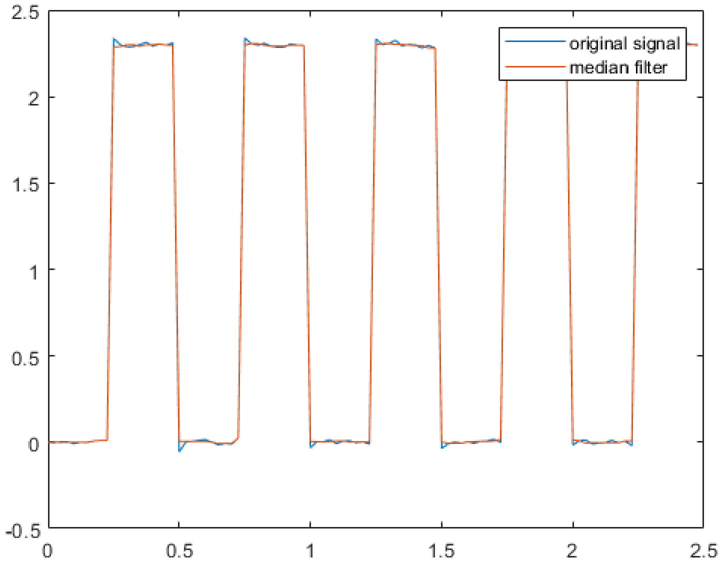


Figure 6 Median filter