1a. sinewave

In [86]:

```
### 1a. sinewave
import math
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.pyplot import figure
import wave
import A1b_zxc701_package

### subewave
def sinewave(t, f, d):
    phi = 2 * np.pi * d * f
    # A vector version of the function (if you are not using broadcasting)
    if(type(t) == list):
        t = np.array(t)
    return np.sin(2 * np.pi * f * t + phi)

### Show sinewave
def plot_sinewave(t, f, d):
    figure(figsize=(8, 6), dpi = 80)
    sine_vals = sinewave(t, f = 5, d = 1.0)
    plt.plot(t, sine_vals, label = 'd = 1.0')
    plt.title('$sinewave, f = 5HZ$')
    plt.ylabel("Amplitude")
    plt.xlabel("Time(s)")
    plt.legend(loc = 'upper left')
    plt.show()

### show delayed sinewave
def plot_delayed_sinewave(t, f, d):
    figure(figsize=(8, 6), dpi=80)
    sine_vals = sinewave(t, f, 0.0)
    sine_vals_delayed = sinewave(t, f, d)
    plt.title('$sinewave, f = 5HZ$')
    plt.plot(t, sine_vals, label = 'd = 0.0')
    plt.plot(t, sine_vals_delayed, linestyle = '--', label = 'd = 0.05')
    plt.ylabel("Amplitude")
    plt.xlabel("Time(s)")
    plt.legend(loc = 'upper left')
    plt.show()
```

Convert the formulas and unit analysis

Requirement: t for the time value, f for the frequency, and d for the delay in seconds, using f_s as the sampling frequency and i as the sample index (zero-based).

1:

Formula:

$f_s \times \mathrm{t} = i$

Unit Analysis:

$$\mathrm{Hz} \times \mathrm{seconds} = \frac{\mathrm{samples}}{\mathrm{second}} \times \mathrm{seconds} = \mathrm{samples}$$

2:

Formula:

$$\text{- } \mathrm{delay} \times f_s \times 2\pi = \mathrm{phase} = \phi$$

Unit Analysis:

$$\mathrm{seconds} \times \frac{\mathrm{cycles}}{\mathrm{second}} \times \frac{\mathrm{radians}}{\mathrm{second}} = \mathrm{radians}$$

Checking the functions using different data

In [87]:
```
### Question 1
A1b_zxc701_package.timetoindex(0.25, 1000)
```

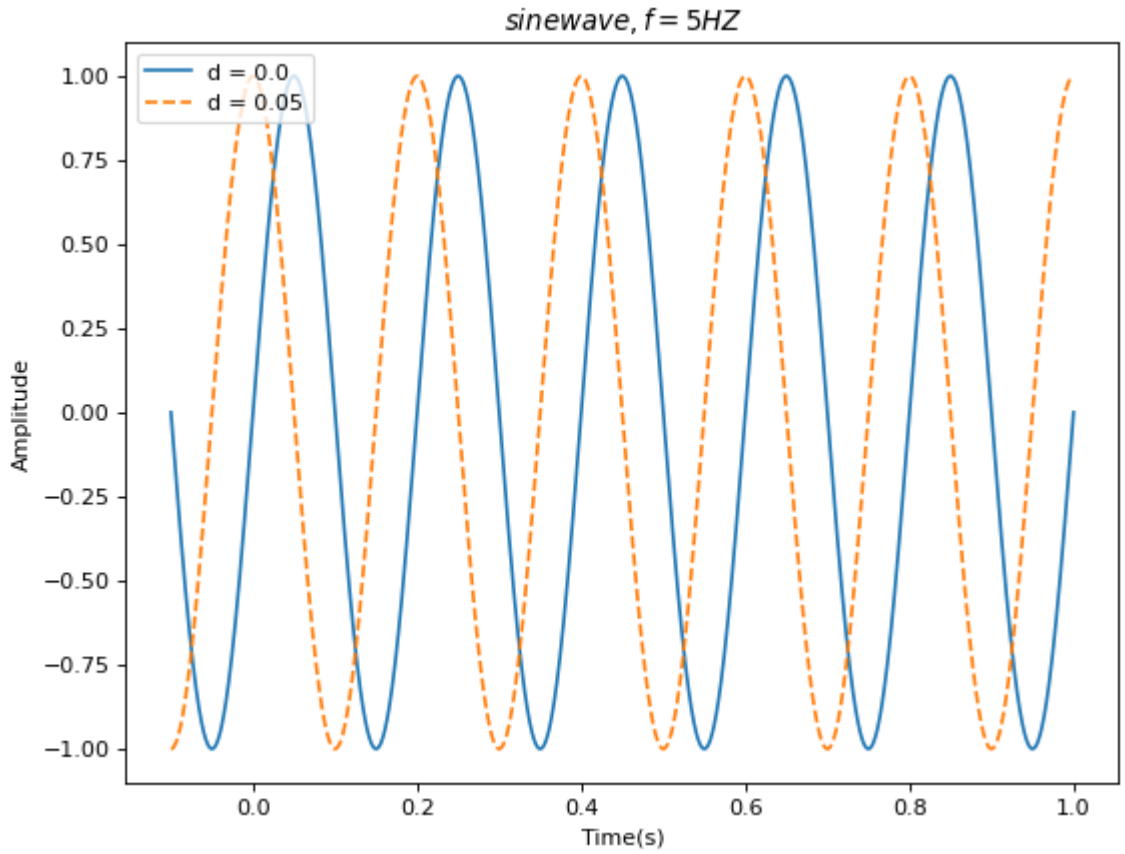
Out[87]:
```
250.0
```

In [88]:
```
### Question 2
sinewave(0, 5, 0.05)
```

Out[88]:
```
1.0
```

In [131…

```
### Question 11
t_1000Hz = np.linspace(-0.1, 1.0, num = 1000 * 2 + 1) # 1000Hz
plot_delayed_sinewave(t_1000Hz, f = 5, d = 0.05)
```



1b. gabor

Checking the code functions different data

In [90]:

```
### Question 3
t = 0
f = 100
sigma = 0.01
A1b_zxc701_package.gabore(t, sigma, f)
```

Out[90]:
```
1.0
```

In [91]:
```
### Question 4
t = -3
f = 0.0625
sigma = 8
A1b_zxc701_package.gaboro(t, sigma, f)
```

Out[91]:
```
0.8611504148937256
```

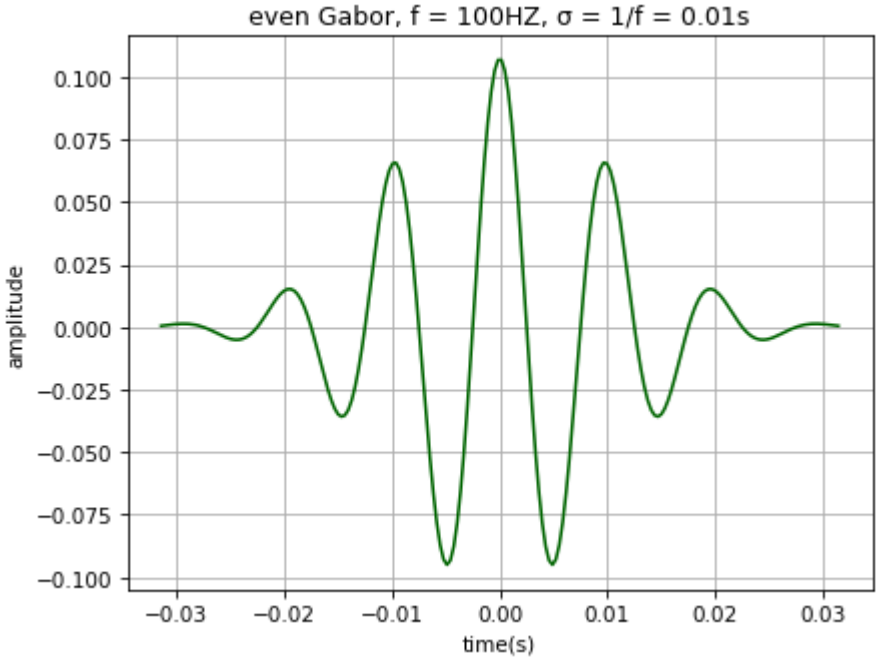
In [92]:
```
### Question 5
f = 100
sigma = 0.01
fs = 10000
A1b_zxc701_package.gabore_norm(fs, sigma, f)
```

Out[92]:
```
9.413962637767147
```

In [93]:
```
### Question 6
t = 0.003
f = 100
sigma = 0.01
A1b_zxc701_package.gaboro(t, sigma, f)
```

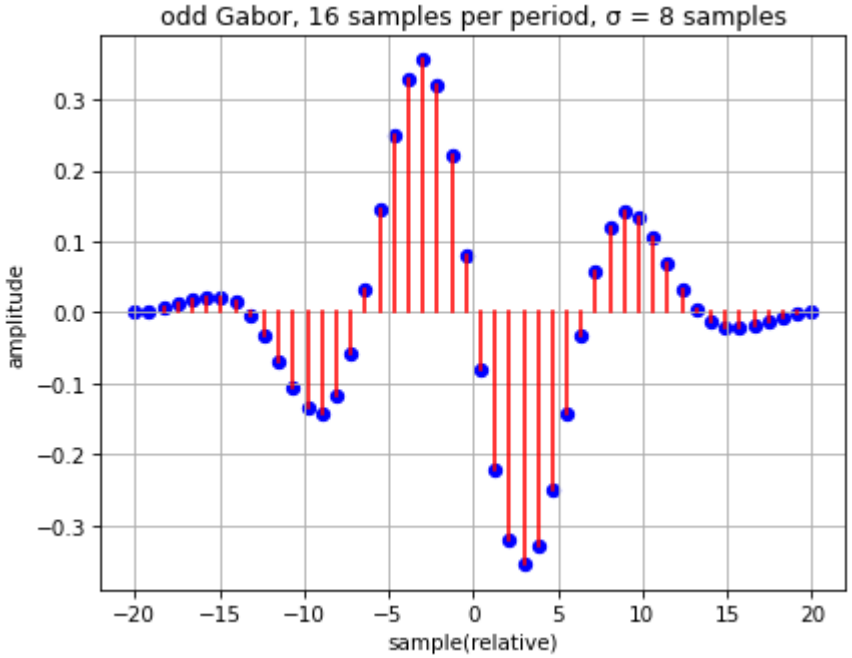
Out[93]:
```
-0.9092076346591274
```

In [125…
```
### Quesition 12
t = np.linspace(-np.pi/100, np.pi/100, 200, endpoint = True)
y = np.vectorize(A1b_zxc701_package.gabore_a)(t, 100, 0.01, 100)
x = np.linspace(-np.pi/100, np.pi/100, len(y))
plt.figure(dpi=75)
plt.plot(x, y/7, color = 'darkgreen')
plt.title('even Gabor, f = 100HZ, σ = 1/f = 0.01s')
plt.xlabel('time(s)')
plt.ylabel('amplitude')
plt.grid()
plt.show()
```



In [124…

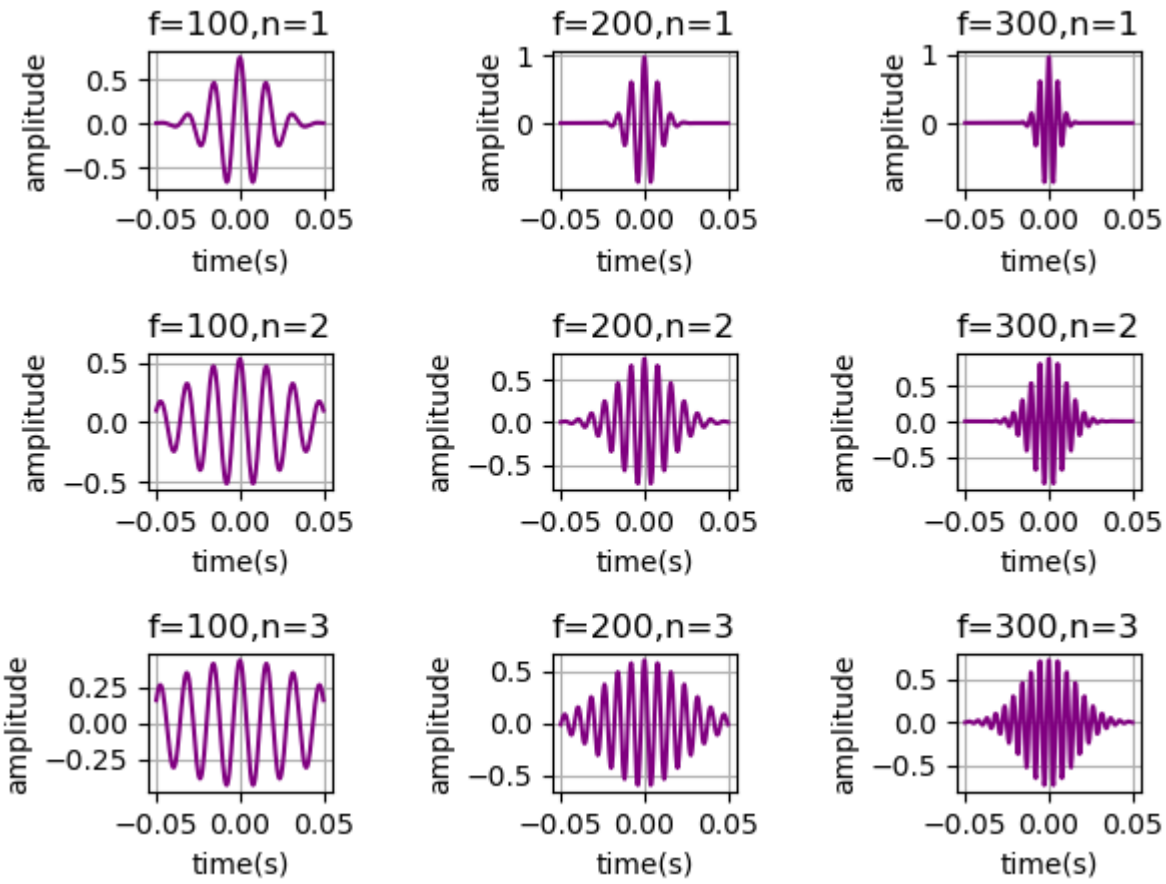
```
### Question 13
t = np.linspace(-24, 24, 48, endpoint = True)
y = np.vectorize(A1b_zxc701_package.gaboro_a)(t, f = 1/16, sigma = 8, fs = 16)
x = np.linspace(-20, 20, len(y))
plt.figure(dpi=75)
plt.scatter(x, y * 4, marker='o', color='blue')
for i in range(1, len(y)):
    plt.plot([x[i], x[i]], [0,y[i]*4], linestyle = '-', color = 'red')
plt.grid()
plt.title('odd Gabor, 16 samples per period, σ = 8 samples')
plt.xlabel('sample(relative)')
plt.ylabel('amplitude')
plt.show()
```



In [96]:

```
### Question 14
import matplotlib.pyplot as plt
plt.subplots_adjust(left=None,bottom=None,right=None,top=None,wspace=1.2,hspace=1.2)
t=np.linspace(-np.pi/100,np.pi/100,200,endpoint=True)

for i in range(1,10):
    plt.subplot(3,3,i)
    plt.grid()
    plt.xlabel('time(s)')
    plt.ylabel('amplitude')

for i in range(1,4):
    plt.subplot(3,3,i)
    hez=i*100
    y=np.vectorize(A1b_zxc701_package.gabore_a)(t,f=hez,sigma=1/hez,fs=100)
    x=np.linspace(-0.05,0.05,len(y))
    plt.plot(x,y,color = 'purple')
    plt.title('f=%d,n=1'%hez)

for i in range(4,7):
    plt.subplot(3,3,i)
    hez=(i-3)*100
    y=np.vectorize(A1b_zxc701_package.gabore_a)(t,f=hez,sigma=2/hez,fs=100)
```

```
    x=np.linspace(-0.05,0.05,len(y))
    plt.plot(x,y,color = 'purple')
    plt.title('f=%d,n=2'%hez)

for i in range(7,10):
    plt.subplot(3,3,i)
    hez=(i-6)*100
    y=np.vectorize(A1b_zxc701_package.gabore_a)(t,f=hez,sigma=3/hez,fs=100)
    x=np.linspace(-0.05,0.05,len(y))
    plt.plot(x,y,color = 'purple')
    plt.title('f=%d,n=3'%hez)
```



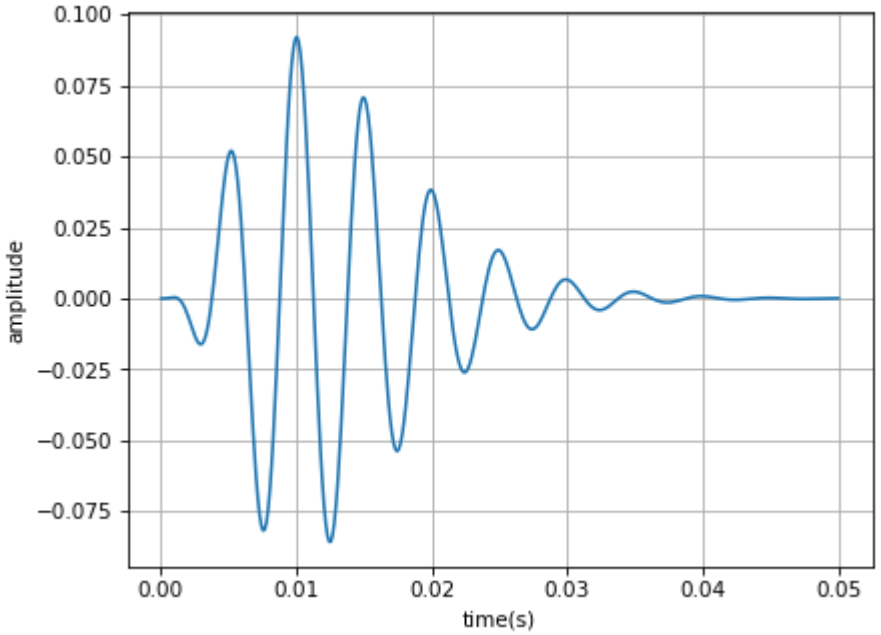
1c. gammatone

Checking the functions using different data

In [97]:
```
### Question 7
t = 0.01
f = 100
fs = 10000
A1b_zxc701_package.gammatone(t,fs,f)
```

Out[97]:
```
0.10873332003195081
```

In [123…
```
### Question 15
t = np.linspace(0, 0.05, 400, endpoint = True)
y=np.vectorize(A1b_zxc701_package.gammatone)(t, f = 200, fs = 100)
x=np.linspace(0, 0.05, len(y))
plt.figure(dpi=75)
plt.grid()
plt.xlabel('time(s)')
plt.ylabel('amplitude')
plt.plot(x, y/10)
plt.show()
```



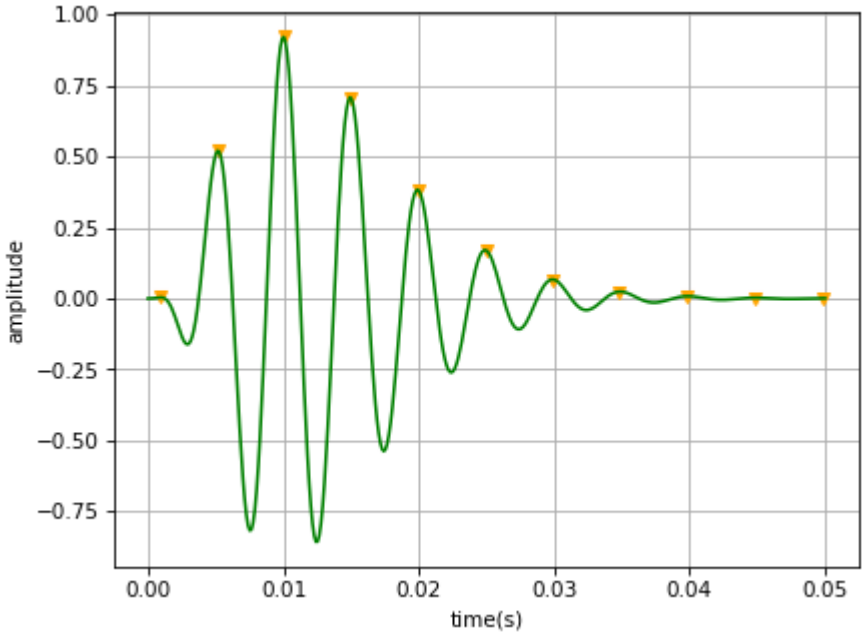
2a. localmaxima

Checking the functions using different data

In [99]:
```
### Question 8
i,arry=A1b_zxc701_package.localmaxima([1,3,2,-2,2,4,8,6])
i
```

Out[99]:
```
[1, 6]
```

In [122…
```
### Question 16
t=np.linspace(0, 0.05, 400, endpoint = True)
y=np.vectorize(A1b_zxc701_package.gammatone)(t, f=200, fs=100)
indices,arry = A1b_zxc701_package.localmaxima(y)
plt.figure(dpi=75)
for i in range(0, len(indices)):
    plt.plot(t[indices[i]], arry[i], color='orange', marker='v')
x = np.linspace(0, 0.05, len(y))
plt.plot(x, y, color = 'green')
plt.grid()
plt.xlabel('time(s)')
plt.ylabel('amplitude')
plt.show()
```



2b. crossings

Checking the functions using different data

In [101…

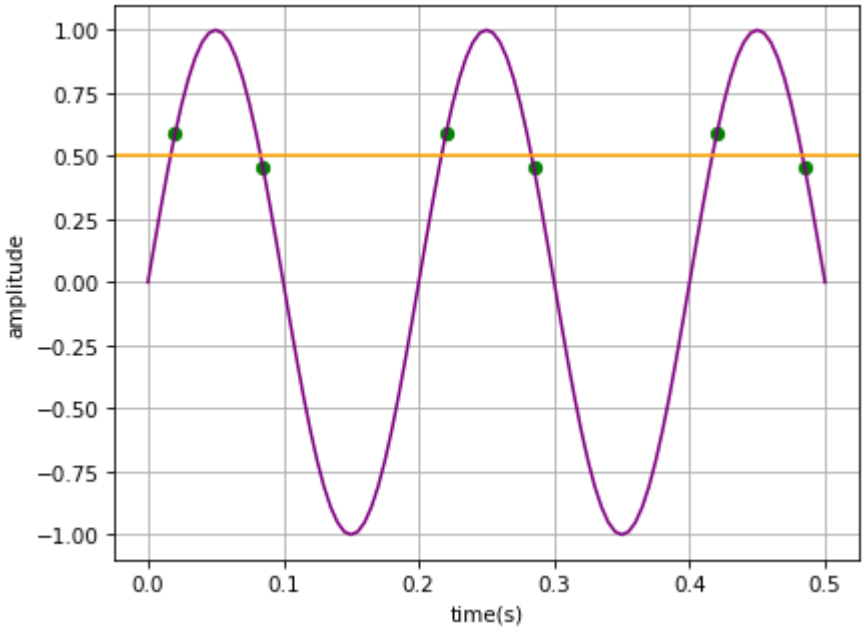
```
### Question 9
A1b_zxc701_package.crossings([-1,0,1,2,0,-2], 1, "both")
```

Out[101]:

```
[2, 4]
```

In [121…

```
### Question 17
t1 = 0
t2 = 0.5
fs = 200
f = 5
d = 0
y = A1b_zxc701_package.pltsinwave(t1, t2, fs, f, d)
x = np.linspace(t1, t2, len(y))
above = A1b_zxc701_package.crossings(y, 0.5, "negpos")
below = A1b_zxc701_package.crossings(y, 0.5, "posneg")
plt.figure(dpi=75)
for i in above:
    plt.scatter(x[i], y[i], color = 'g')
for i in below:
    plt.scatter(x[i], y[i], color = 'g')
plt.plot(x, y, color = 'purple')
plt.grid()
plt.xlabel('time(s)')
plt.ylabel('amplitude')
plt.axhline(y = 0.5, color = 'orange', linestyle = '-')
plt.show()
```



2c. envelope

Checking the functions using different data

In [103…
```
### Question 10
y = [5, 5, 2, 3, 4, 3, -6, -9, 0, -3, 9,-7]
ylower,yupper,blockindices = A1b_zxc701_package.envelope(y, 3)
```

In [104…
```
yupper
```

Out[104]:
```
[5, 4, 9]
```

In [105…
```
ylower
```

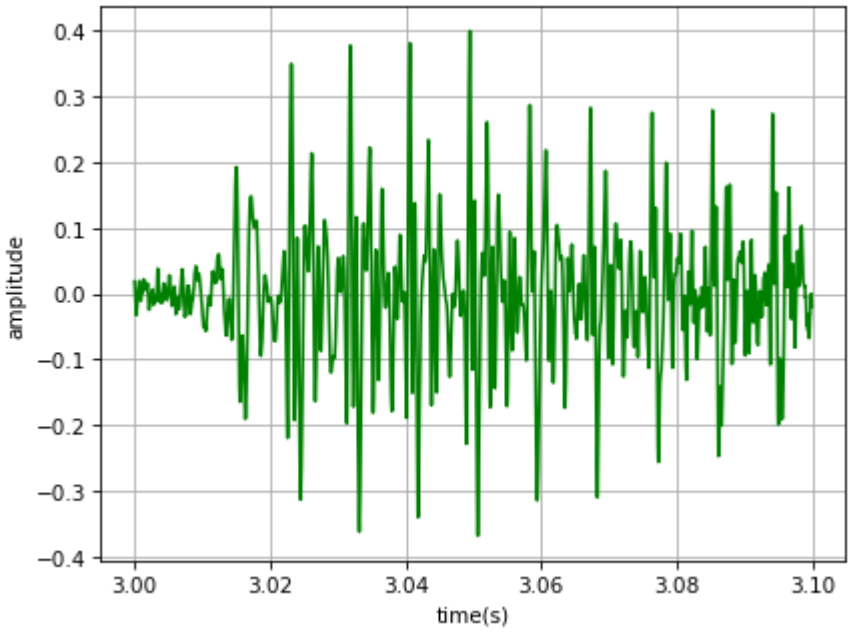
Out[105]:
```
[2, -9, -7]
```

In [106…
```
blockindices
```

Out[106]:
```
[0, 4, 8]
```

In [117…

```
### Question 18
fs,data=A1b_zxc701_package.wavfile.read('speech.wav')
ylower,yupper,blockindices=A1b_zxc701_package.envelope(data,500)
length=len(data)/fs
t=np.arange(0,length,1/fs)
xx=A1b_zxc701_package.deepcopy(data)
xx=xx-np.mean(xx)
x=xx/np.max(np.abs(xx))
plt.figure(dpi=75)
plt.plot(t[3*fs:int(3.1*fs)], x[3*fs:int(3.1*fs)]*2/5, color = 'green')
plt.grid()
plt.xlabel('time(s)')
plt.ylabel('amplitude')
plt.show()
```



In [127…

```
### Question 18
fs,data=A1b_zxc701_package.wavfile.read('speech.wav')
ylower,yupper,blockindices=A1b_zxc701_package.envelope(data,500)
length=len(data)/fs
t=np.arange(0,length,1/fs)
xx=A1b_zxc701_package.deepcopy(data)
xx=xx-np.mean(xx)
x=xx/np.max(np.abs(xx))
plt.figure(dpi=75)
```

```
plt.plot(t[0*fs:int(6*fs)], x[0*fs:int(6*fs)]*0.4, color = 'gray')
plt.grid()
plt.xlabel('time(s)')
plt.ylabel('amplitude')
plt.show()
```