In [1]:
```
# Tejas Acharya
# EE-541
# Homework 02
# Problem 01
# 06-06-2023
```

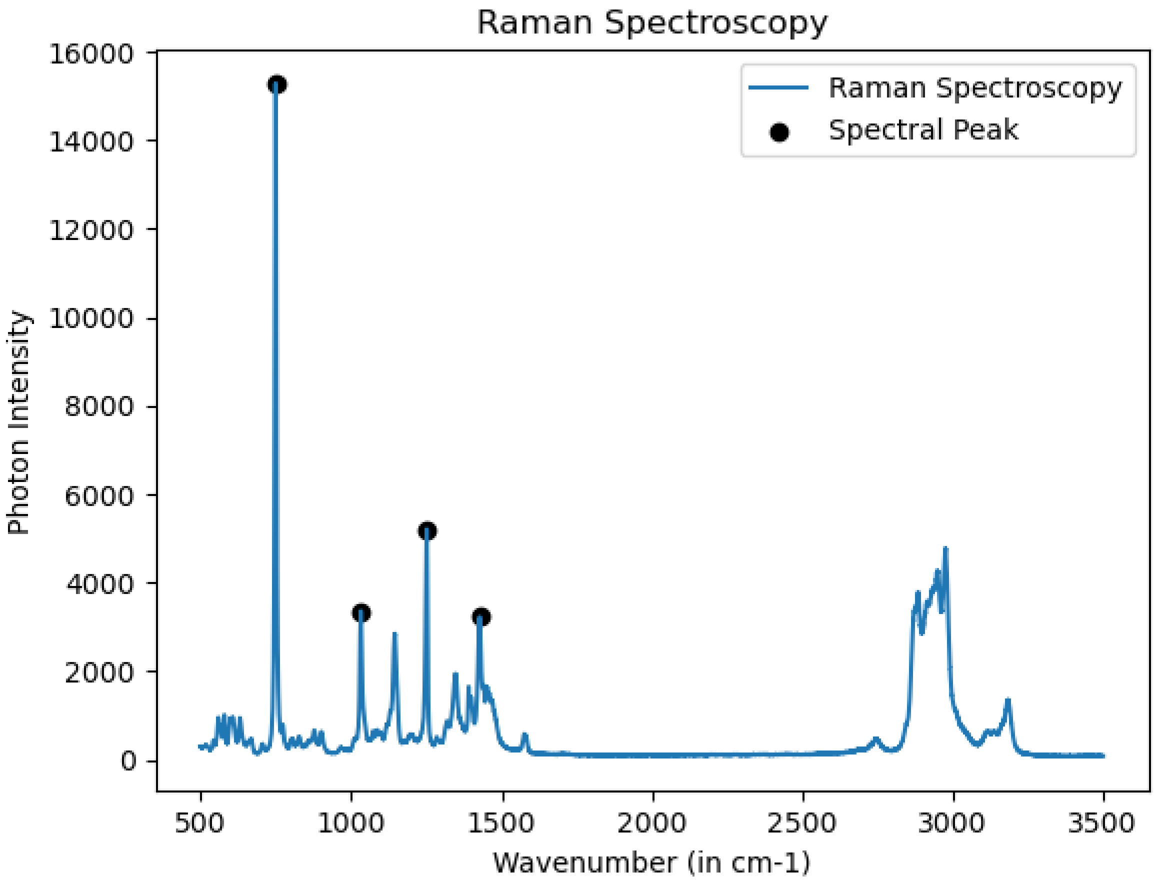
In [2]:
```
#Importing Libraries
import sys
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import find_peaks
from scipy.interpolate import CubicSpline
```

In [3]:
```
#Constants
RAMAN_DATASET_FILE = './raman.txt'
NUM_REGIONS_OF_INTEREST = 4
```

In [4]:
```
#Read the Data
raman_data = np.loadtxt(RAMAN_DATASET_FILE)
wavenumber = raman_data[:, 0]
intensity = raman_data[:, 1]
```

In [5]:
```
#Find Spectral Peaks
peak_idx = find_peaks(intensity, height=3000, distance = 300)[0][:NUM_REGIO
```

In [6]:
```
#Figure
plt.figure()
plt.plot(wavenumber, intensity)
plt.scatter(wavenumber[peak_idx], intensity[peak_idx], c='#000000')

plt.xlabel('Wavenumber (in cm-1)')
plt.ylabel('Photon Intensity')
plt.title('Raman Spectroscopy')

plt.legend(['Raman Spectroscopy', 'Spectral Peak'])
plt.show()
```



In [7]:
```
good_widths = np.empty((NUM_REGIONS_OF_INTEREST))
```

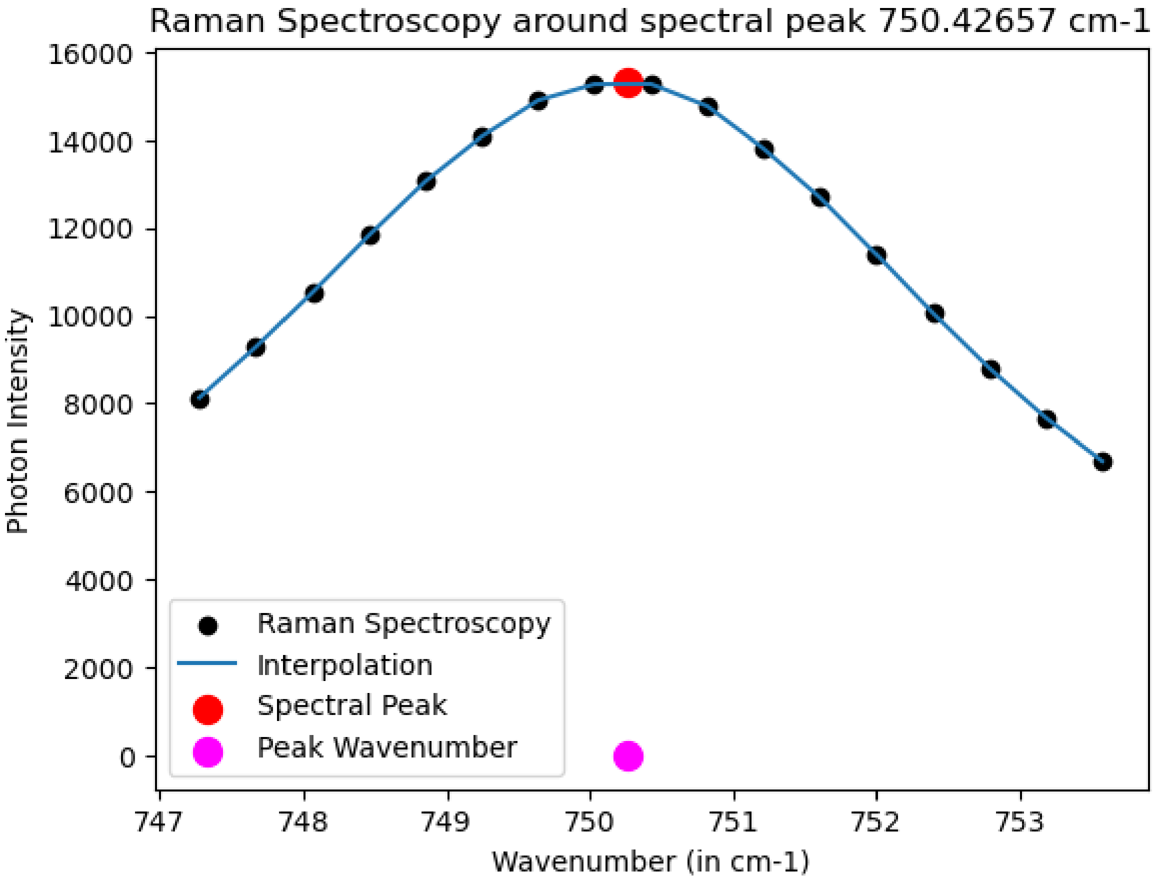
In [8]:
```
def get_good_widths(peak_idx):
    widths = np.arange(50, 0, -1)
    threshold = intensity[peak_idx] / 2
    for width in widths:
        if intensity[peak_idx] - intensity[peak_idx - width] < threshold :
            return width
```

In [9]:
```
for i in range(len(peak_idx)):
    good_widths[i] = get_good_widths(peak_idx[i])
```

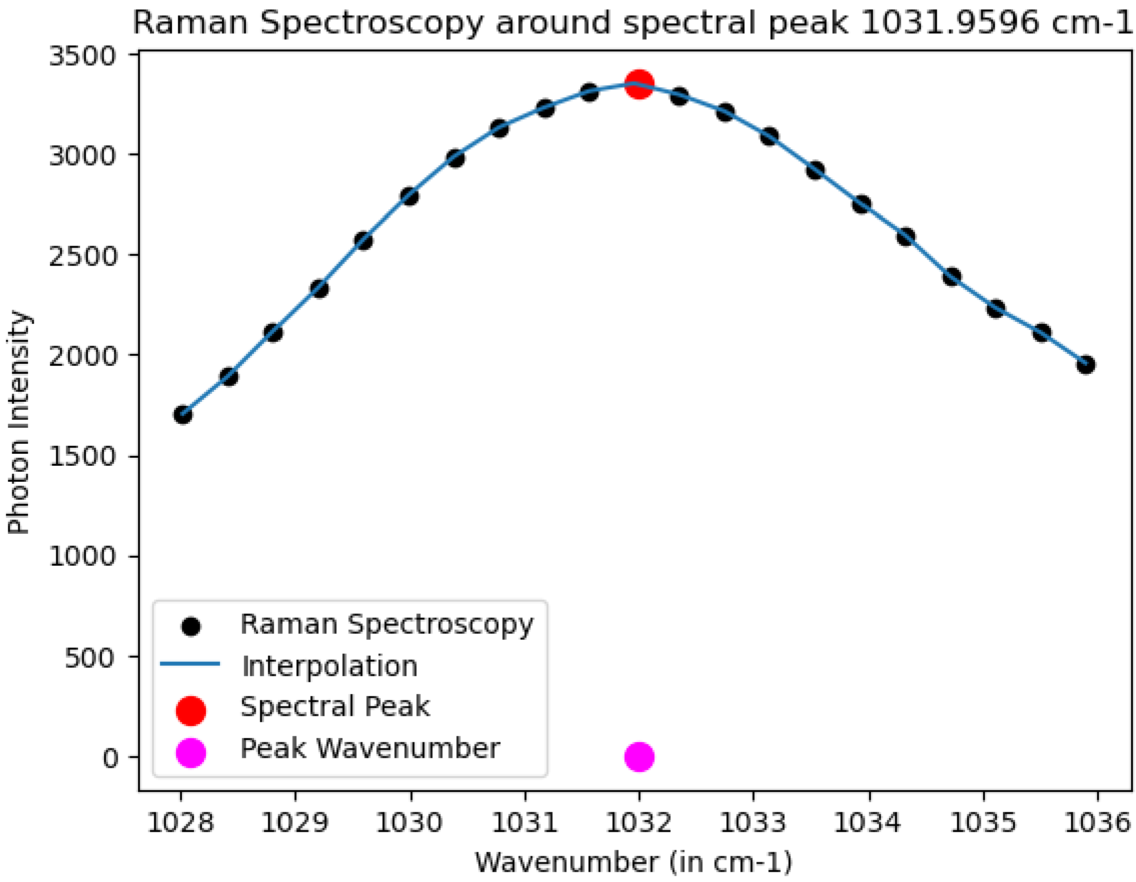
In [10]:
```
print('#' * 50)
for j in range(NUM_REGIONS_OF_INTEREST):
    neighborhood = np.arange(peak_idx[j] - good_widths[j], peak_idx[j] + go
    intensity_interpolate = CubicSpline(wavenumber[neighborhood], intensity
    wavenumber_list = np.linspace(wavenumber[neighborhood[0]], wavenumber[n
    derivative_list = np.array(list(map(lambda x: (intensity_interpolate(x 
    zero_crossing = list(filter(lambda x: x < 0, derivative_list))[0]
    peak_wavenumber = wavenumber_list[derivative_list == zero_crossing][0]

    plt.figure()
    plt.scatter(wavenumber[neighborhood], intensity[neighborhood], c='#0000
    plt.plot(wavenumber[neighborhood], intensity_interpolate(wavenumber[nei
    plt.scatter(peak_wavenumber, intensity_interpolate(peak_wavenumber), c=
    plt.scatter(peak_wavenumber, 0, c='#FF00FF', s=100)

    print('Peak Wavenumber: ')
    sys.stdout.write(f'{peak_wavenumber}')

    plt.xlabel('Wavenumber (in cm-1)')
    plt.ylabel('Photon Intensity')
    plt.title(f'Raman Spectroscopy around spectral peak {wavenumber[peak_id
    plt.legend(['Raman Spectroscopy', 'Interpolation', 'Spectral Peak', 'Pe
```

```
    plt.show()
    print('#' * 50)
```

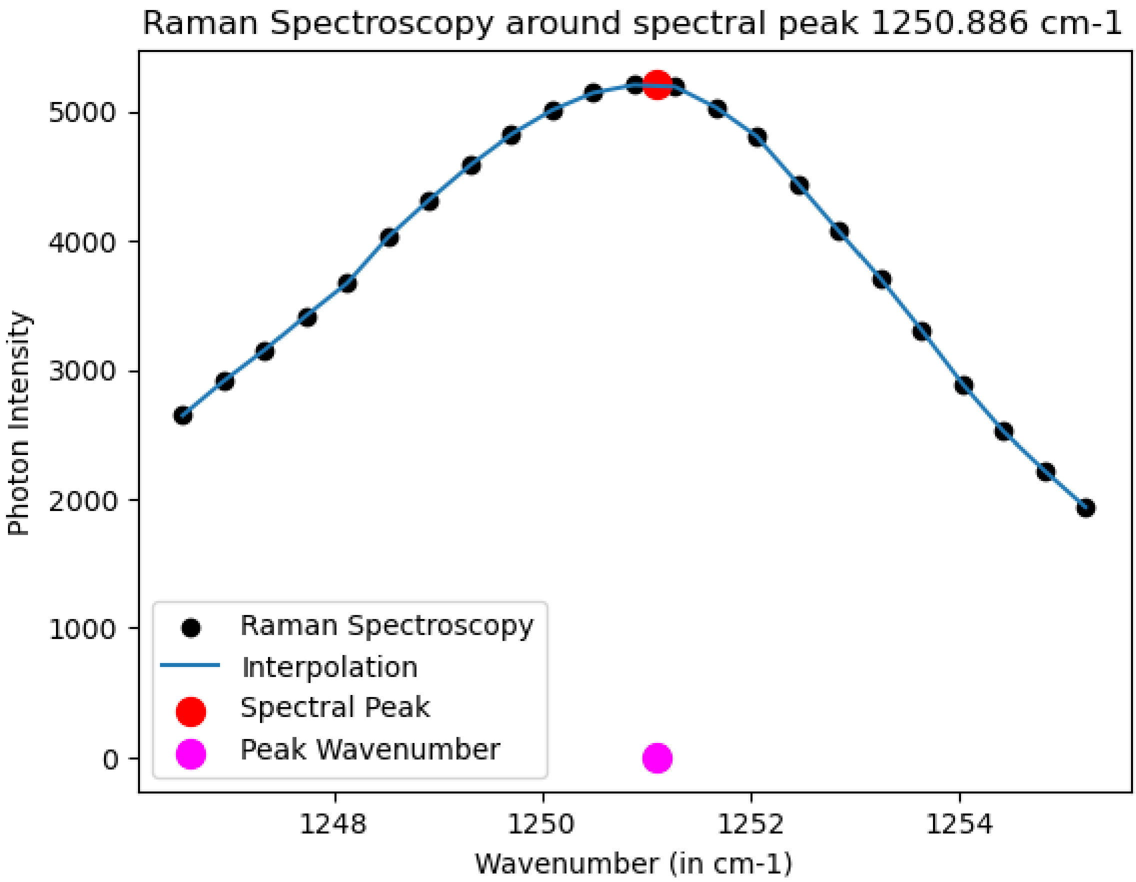
```
##################################################
Peak Wavenumber:
750.2674828282828
```



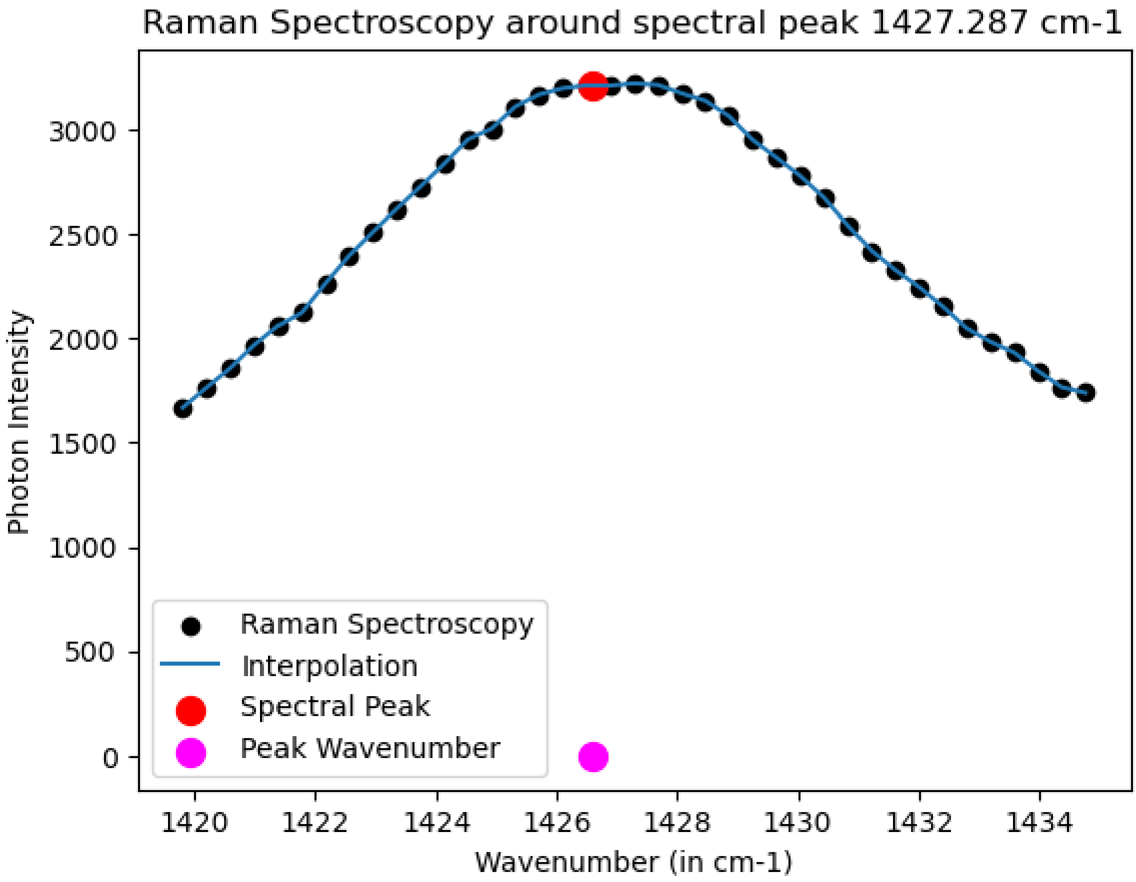

```
##################################################
Peak Wavenumber:
1031.9993727272727
```




```
##################################################
Peak Wavenumber:
1251.1047
```




```
##################################################
Peak Wavenumber:
1426.6069363636364
```


Raman Spectroscopy around spectral peak 1427.287 cm-1
Photon Intensity
Wavenumber (in cm-1)
Raman Spectroscopy
Interpolation
Spectral Peak
Peak Wavenumber
0
500
1000
1500
2000
2500
3000
1420
1422
1424
1426
1428
1430
1432
1434

##