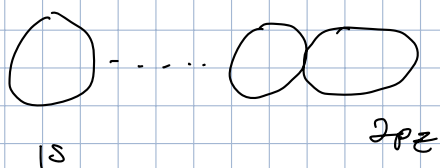
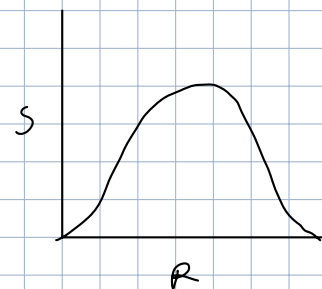


## Problem #2

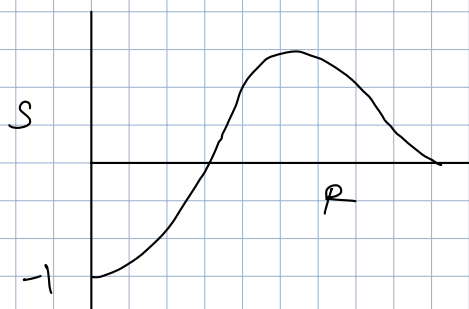


$$R \rightarrow 0 \quad S = 0$$

$$R \rightarrow \infty \quad S = 0$$



## Problem #3



$$R \rightarrow 0 \quad S = -1$$

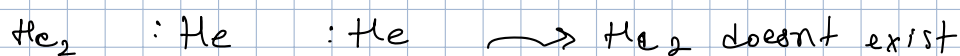
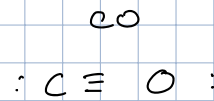
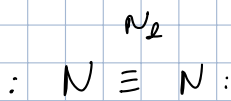
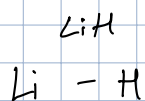
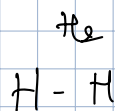
$$R \rightarrow \infty \quad S = 0$$



## Problem #4

Energies of the two atoms should match

## Problem #5



- He is stronger because they have similar energies
- N<sub>2</sub> is stronger because it has a triple bond instead of a single bond
- Difficult to say N<sub>2</sub> is stronger because they are similar in energy
- H<sub>2</sub> is stronger because they have similar energies

# 1 Chem 30324, Spring 2024, Homework 8

## 1.1 Computational chemistry.

1.1.1 Today properties of a molecule are more often than not calculated rather than inferred. Quantitative molecular quantum mechanical calculations require specialized numerical solvers like [Orca](#). Following are instructions for using Orca with the [Webmo](#) graphical interface.

1.1.2 Now, let's set up your calculation (you may do this with a partner or partners if you choose):

1. Log into the Webmo server <https://www.webmo.net/demoserver/cgi-bin/webmo/login.cgi> using "guest" as your username and password.
2. Select New Job-Creat New Job.
3. Use the available tools to sketch a molecule.
4. Use the right arrow at the bottom to proceed to the Computational Engines.
5. Select Orca
6. Select "Molecular Energy," "B3LYP" functional and the default def2-SVP basis set.
7. Select the right arrow to run the calculation.
8. From the job manager window choose the completed calculation to view the results.

The molecule you are to study depends on your last name. Choose according to the list: + A-G: **CO** + H-R: **BN** + S-Z: **BeO**

For your convenience, here are the total energies (in Hartree, 27.212 eV/Hartree) of the constituent atoms, calculated using the B3LYP DFT treatment of  $v_{ee}$  and the def2-SVP basis set:

Atom	Energy	Atom	Energy
B	-24.61703	N	-54.51279
Be	-14.64102	O	-74.98784
C	-37.79271	F	-99.60655

1.1.3 6. Construct a potential energy surface for your molecule. Using covalent radii, guess an approximate equilibrium bond length, and use the Webmo editor to draw the molecule with that length. Specify the "Molecular Energy" option to Gaussian and the "Routine" basis set for better accuracy. Calculate and plot out total molecular energy vs. bond distance in increments of 0.05 Å about your guessed minimum, including enough points to encompass the actual minimum. (You will find it convenient to subtract off the individual atom energies from the molecular total energy and to convert to more convenient units, like eV or kJ/mol.) By fitting the few points nearest the minimum, determine the equilibrium bond length. How does your result compare to literature?

Literature Bond length CO 1.128 Å, BN 1.325 Å, BeO 1.331 Å.

### 1.1.4 CO molecule

```
[1]: "CO molecule"
import numpy as np
import matplotlib.pyplot as plt

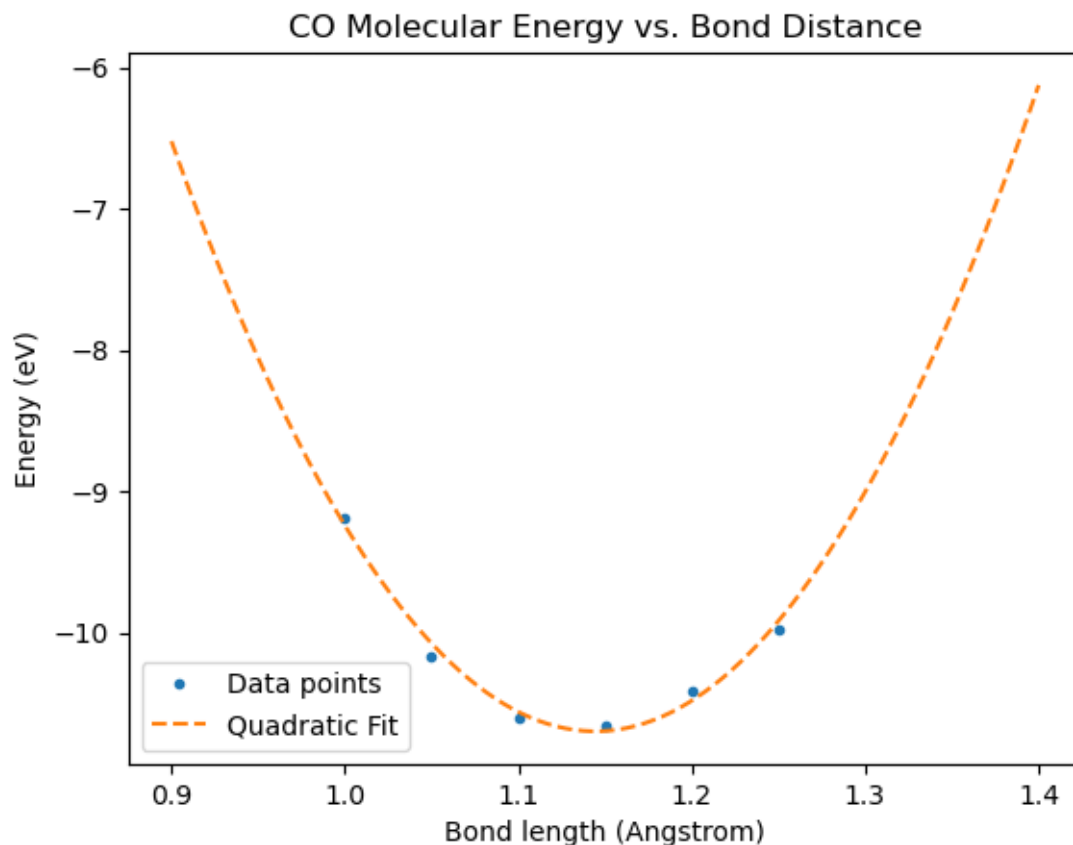
"quadratic fit"
E_C = -37.79271 # Ha
E_O = -74.98784 # Ha
R_CO = [1.00, 1.05, 1.10, 1.15, 1.2, 1.25] # distance, angstrom
E_total_CO = [-113.118, -113.154, -113.170, -113.172, -113.163, -113.147] # Ha, total
    ↳energy of the molecule
E_CO = [] # eV, subtract off the individual atom energies from the molecular
    ↳total energy and convert to eV
for i in E_total_CO:
    E_CO.append((i-E_C-E_O)*27.212)
fit = np.polyfit(R_CO, E_CO, 2) # quadratic fit
print(fit)

[ 69.97371429 -160.14650743  80.93179231]
```

```
[2]: "plot energy vs. bond length"
x = np.linspace(0.9, 1.4, 100)
z = 69.97371429*x**2 - 160.14650743*x + 80.93179231 # quadratic fit
E_min_CO = min(z) # minimum energy
print('The energy minimum is {:.5f} eV.'.format(E_min_CO))
plt.plot(R_CO, E_CO, '.', label='Data points')
plt.plot(x, z, '--', label='Quadratic Fit')
plt.xlabel('Bond length (Angstrom)')
plt.ylabel('Energy (eV)')
plt.title('CO Molecular Energy vs. Bond Distance')
plt.legend()
plt.show()

"find equilibrium bond length"
import sympy as sp
x = sp.symbols('x')
z = 69.97371429*x**2 - 160.14650743*x + 80.93179231 # quadratic fit
l = sp.solve(sp.diff(z,x),x)
print('The equilibrium bond length is {:.4f} angstroms.'.format(l[0])) #
    ↳equilibrium bond length
```

The energy minimum is -10.69845 eV.



The equilibrium bond length is 1.1443 angstroms.

### 1.1.5 BN molecule

```
[19]: "BN molecule"
import numpy as np
import matplotlib.pyplot as plt

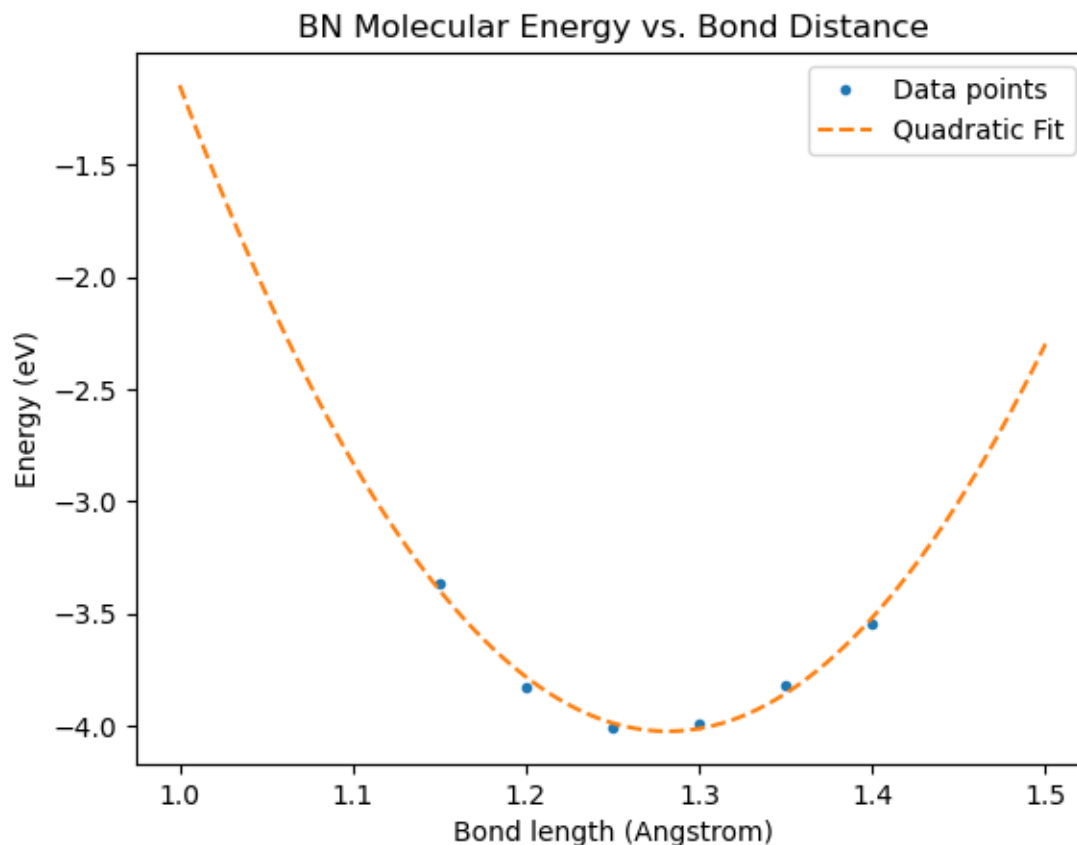
"quadratic fit"
E_B = -24.61703 # Ha
E_N = -54.51279 # Ha
R_BN = [1.15, 1.2, 1.25, 1.3, 1.35, 1.4] # distance, angstrom
E_total_BN = [-79.2535, -79.2704, -79.2771, -79.2764, -79.2702, -79.2601] # Ha, total
    ↳energy of the molecule
E_BN = [] # eV, subtract off the individual atom energies from the molecular
    ↳total energy and convert to eV
for i in E_total_BN:
    E_BN.append((i-E_B-E_N)*27.212)
fit = np.polyfit(R_BN, E_BN, 2) # quadratic fit
print(fit)
```

```
[ 36.26970857 -92.9806828  55.56616539]
```

```
[20]: "plot energy vs. bond length"
x = np.linspace(1.0, 1.5, 100)
z = 36.26970857*x**2 - 92.9806828*x + 55.56616539 # quadratic fit
E_min_BN = min(z) # minimum energy
print('The energy minimum is {:.5f} eV.'.format(E_min_BN))
plt.plot(R_BN, E_BN, '.', label='Data points')
plt.plot(x, z, '--', label='Quadratic Fit')
plt.xlabel('Bond length (Angstrom)')
plt.ylabel('Energy (eV)')
plt.title('BN Molecular Energy vs. Bond Distance')
plt.legend()
plt.show()

"find equilibrium bond length"
import sympy as sp
x = sp.symbols('x')
z = 36.26970857*x**2 - 92.9806828*x + 55.56616539 # quadratic fit
l = sp.solve(sp.diff(z,x),x)
print('The equilibrium bond length is {:.4f} angstroms.'.format(l[0])) #  
→equilibrium bond length
```

The energy minimum is -4.02490 eV.



The equilibrium bond length is 1.2818 angstroms.

### 1.1.6 BeO molecule

```
[16]: "BeO molecule"
import numpy as np
import matplotlib.pyplot as plt

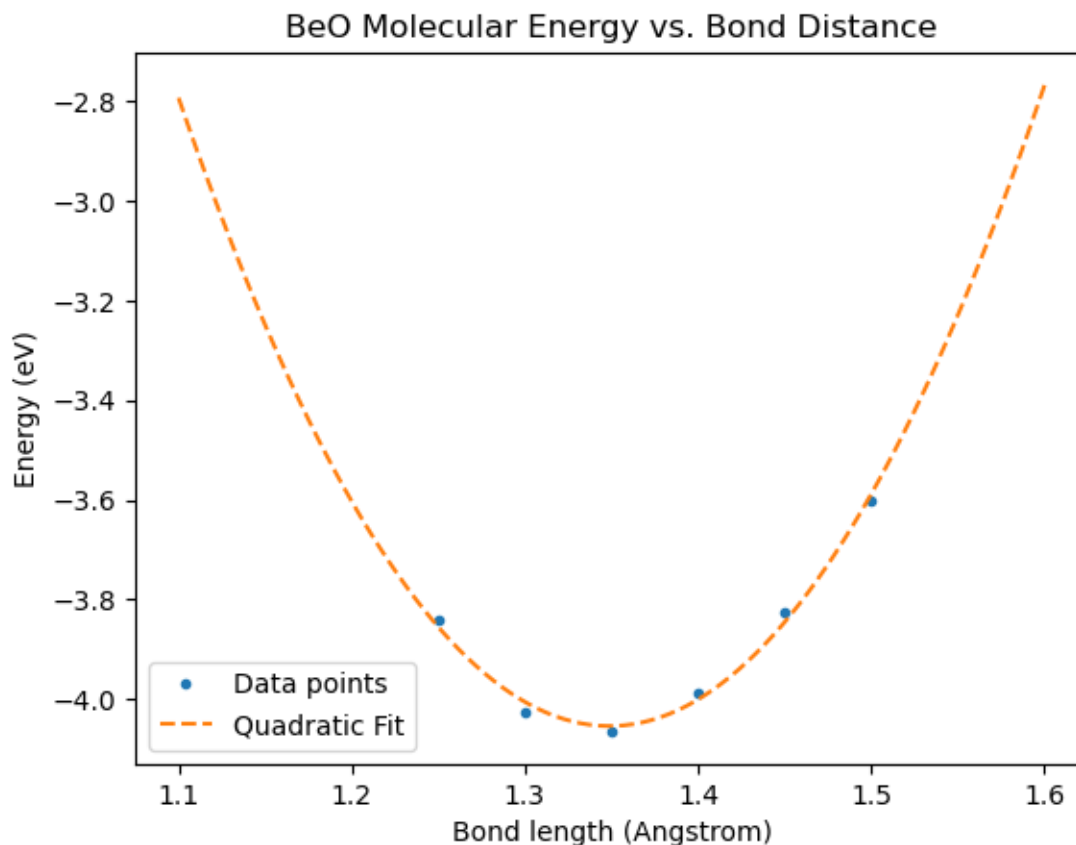
"quadratic fit"
E_Be = -14.64102 # Ha
E_O = -74.98784 # Ha
R_BeO = [1.25, 1.3, 1.35, 1.4, 1.45, 1.5] # distance, angstrom
E_total_BeO = [-89.770, -89.7768, -89.7782, -89.7754, -89.7695, -89.7612] # Ha, total
    ↳ energy of the molecule
E_BeO = [] # eV, subtract off the individual atom energies from the molecular
    ↳ total energy and convert to eV
for i in E_total_BeO:
    E_BeO.append((i-E_Be-E_O)*27.212)
fit = np.polyfit(R_BeO, E_BeO, 2) # quadratic fit
print(fit)
```

```
[ 20.35068857 -54.8961282  32.96722138]
```

```
[6]: "plot energy vs. bond length"
x = np.linspace(1.1, 1.6, 100)
z = 20.35068857*x**2 - 54.8961282*x + 32.96722138 # quadratic fit
E_min_BeO = min(z) # minimum energy
print('The energy minimum is {:.5f} eV.'.format(E_min_BeO))
plt.plot(R_BeO, E_BeO, '.', label='Data points')
plt.plot(x, z, '--', label='Quadratic Fit')
plt.xlabel('Bond length (Angstrom)')
plt.ylabel('Energy (eV)')
plt.title('BeO Molecular Energy vs. Bond Distance')
plt.legend()
plt.show()

"find equilibrium bond length"
import sympy as sp
x = sp.symbols('x')
z = 20.35068857*x**2 - 54.8961282*x + 32.96722138 # quadratic fit
l = sp.solve(sp.diff(z,x),x)
print('The equilibrium bond length is {:.4f} angstroms.'.format(l[0])) #
    ↳equilibrium bond length
```

The energy minimum is -4.05342 eV.



The equilibrium bond length is 1.3488 angstroms.

- 1.1.7 7. Use the quadratic fit from Question 1 to determine the harmonic vibrational frequency of your molecule, in  $\text{cm}^{-1}$ . Recall that the force constant is the second derivative of the energy at the minimum, and that the frequency (in wavenumbers) is related to the force constant according to

$$\tilde{\nu} = \frac{1}{2\pi c} \sqrt{\frac{k}{\mu}}$$

#### 1.1.8 CO

```
[7]: "calculate harmonic vibrational frequency"
J = 1.6022e-19 # 1 eV = 1.6022e-19 J
A = 1e-10 # 1 angstrom = 1e-10 m
c = 2.99792e8 # m/s
m_C = 12.0107
m_O = 15.9994
mu_CO = m_C*m_O/(m_C+m_O)*1.6605e-27 # kg, reduced mass
k_CO = 2*69.97371429*J/A**2 # J/m**2
nu_CO = 1/(2*np.pi*c)*np.sqrt(k_CO/mu_CO)/100 # cm^-1, wavenumber
```



```
print('The harmonic vibrational frequency is {:.2f} cm-1'.format(nu_CO))
```

The harmonic vibrational frequency is 2355.28 cm<sup>-1</sup>.

### 1.1.9 BN

```
[21]: "calculate harmonic vibrational frequency"
J = 1.6022e-19 # 1 eV = 1.6022e-19 J
A = 1e-10 # 1 angstrom = 1e-10 m
c = 2.99792e8 # m/s
m_B = 10.811
m_N = 14.0067
mu_BN = m_B*m_N/(m_B+m_N)*1.6605e-27 # kg, reduced mass
k_BN = 2*36.26970857*J/A**2 # J/m**2
nu_BN = 1/(2*np.pi*c)*np.sqrt(k_BN/mu_BN)/100 # cm-1, wavenumber
print('The harmonic vibrational frequency is {:.2f} cm-1'.format(nu_BN))
```

The harmonic vibrational frequency is 1798.07 cm<sup>-1</sup>.

### 1.1.10 BeO

```
[9]: "calculate harmonic vibrational frequency"
J = 1.6022e-19 # 1 eV = 1.6022e-19 J
A = 1e-10 # 1 angstrom = 1e-10 m
c = 2.99792e8 # m/s
m_Be = 9.01218
m_O = 15.9994
mu_BeO = m_Be*m_O/(m_Be+m_O)*1.6605e-27 # kg, reduced mass
k_BeO = 2*20.35068857*J/A**2 # J/m**2
nu_BeO = 1/(2*np.pi*c)*np.sqrt(k_BeO/mu_BeO)/100 # cm-1, wavenumber
print('The harmonic vibrational frequency is {:.2f} cm-1'.format(nu_BeO))
```

The harmonic vibrational frequency is 1385.63 cm<sup>-1</sup>.

### 1.1.11 8. Use your results to determine the zero-point-corrected bond energy of your molecule. How does this model compare with the experimental value?

Experimental value: CO 1072 kJ/mol, BN 385 kJ/mol, BeO 445 kJ/mol.

### 1.1.12 CO

```
[10]: "determine the zero-point-corrected bond energy"
h = 6.62607e-34 # J*s
NA = 6.02214e23
EO_CO = 0.5*h*nu_CO*100*c # J, zero point energy for harmonic oscillator
E_Bond_CO = (E_min_CO*J + EO_CO)*NA/1000 # kJ/mol, zero-point-corrected bond
↪energy
print('The zero-point-corrected bond energy is {:.4f} kJ/mol.'.
↪format(-E_Bond_CO))
```

The zero-point-corrected bond energy is 1018.1704 kJ/mol.

### 1.1.13 BN

```
[22]: "determine the zero-point-corrected bond energy"
h = 6.62607e-34 # J*s
NA = 6.02214e23
EO_BN = 0.5*h*nu_BN*100*c # J, zero point energy for harmonic oscillator
E_Bond_BN = (E_min_BN*J + EO_BN)*NA/1000 # kJ/mol, zero-point-corrected bond
↪energy
print('The zero-point-corrected bond energy is {:.4f} kJ/mol.'.
↪format(-E_Bond_BN))
```

The zero-point-corrected bond energy is 377.5943 kJ/mol.

### 1.1.14 BeO

```
[12]: "determine the zero-point-corrected bond energy"
h = 6.62607e-34 # J*s
NA = 6.02214e23
EO_BeO = 0.5*h*nu_BeO*100*c # J, zero point energy for harmonic oscillator
E_Bond_BeO = (E_min_BeO*J + EO_BeO)*NA/1000 # kJ/mol, zero-point-corrected bond
↪energy
print('The zero-point-corrected bond energy is {:.4f} kJ/mol.'.
↪format(-E_Bond_BeO))
```

The zero-point-corrected bond energy is 382.8133 kJ/mol.

## 1.2 Computational chemistry, part deux

**1.2.1** Diatomics are a little mundane. These same methods can be used to compute the properties of much more complicated things. As example, the OQMD database <http://oqmd.org/> contains results for many solids. We don't have time to get this complicated in class, but at least you can compute properties of some molecules.

**1.2.2** 9. Working with some of your classmates, compute the equilibrium structures of C<sub>2</sub>H<sub>6</sub>, C<sub>2</sub>H<sub>4</sub>, and C<sub>2</sub>H<sub>2</sub>. Compare their equilibrium C-C bond lengths. Do they vary in the way you expect?

1. Log into the Webmo server <https://www.webmo.net/demoserver/cgi-bin/webmo/login.cgi> using "guest" as your username and password.
2. Select New Job-Creat New Job.
3. Use the available tools to sketch a molecule. Make sure the bond distances and angles are in a plausible range.
4. Use the right arrow at the bottom to proceed to the Computational Engines.
5. Select Orca
6. Select "Geometry optimization," "B3LYP" functional and the default def2-SVP basis set.
7. Select the right arrow to run the calculation.
8. From the job manager window choose the completed calculation to view the results.

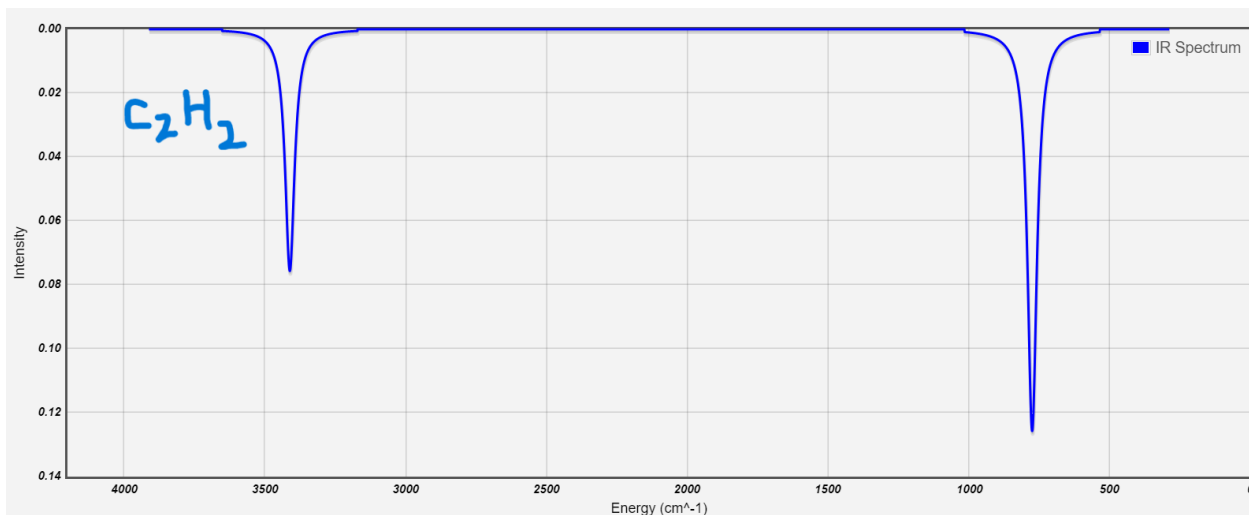
Values I got for the C-C bond lengths are listed below, they vary in length as expected with the triple bond being the shortest and the single bond the longest.

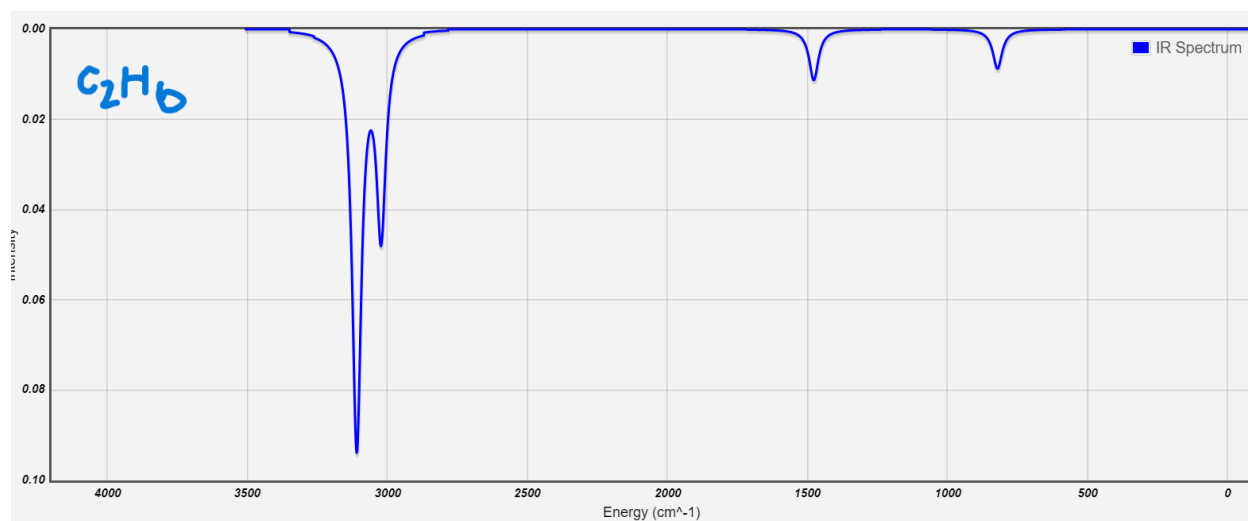
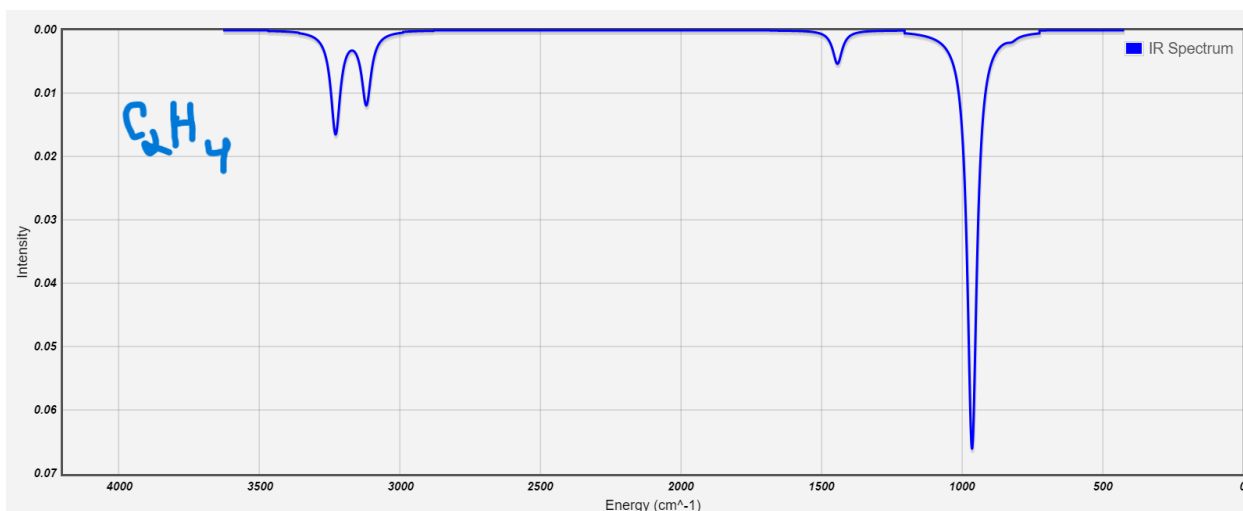
Molecule	C-C Bond Length ( $\text{\AA}$ )
$\text{C}_2\text{H}_6$	1.542
$\text{C}_2\text{H}_4$	1.333
$\text{C}_2\text{H}_2$	1.209

### 1.2.3 10. Compute the corresponding vibrational spectra. Could you distinguish these molecules by their spectra?

1. Log into the Webmo server <https://www.webmo.net/demoserver/cgi-bin/webmo/login.cgi> using “guest” as your username and password.
2. Select the job with the optimized geometry and open it.
3. Use the right arrow at the bottom to proceed to the Computational Engines.
4. Select Orca
5. Select “Vibrational frequency,” “B3LYP” functional and the default def2-SVP basis set.
6. Select the right arrow to run the calculation.
7. From the job manager window choose the completed calculation to view the results.

Attached are screenshots of the calculated vibrational spectra of the three molecules. The three spectra have unique features due to the differences in C-C bonding and the number of H bonds, and thus could be distinguished by their spectra.





1.2.4 11. Compute the structure and energy of  $H_2$ . Use it to compare the energies to hydrogenate acetylene to ethylene and ethylene to ethane. Which is easier to hydrogenate? Can you see why selective hydrogenation of acetylene to ethylene is difficult to do?

```
[15]: E_H2 = -1.1672# Hartree
      E_ethane = -79.702# Hartree
      E_ethylene = -78.4735# Hartree
      E_acetylene = -77.2215# Hartree

      E_ace_to_ethyl = (E_ethylene - (E_acetylene + E_H2))*27.212# eV
      print('Energy to hydrogenate acetylene to ethylene is', E_ace_to_ethyl, 'eV')

      E_ethyl_to_eth = (E_ethane - (E_ethylene + E_H2))*27.212# eV
      print('Energy to hydrogenate ethylene to ethane is', E_ethyl_to_eth, 'eV')
```

Energy to hydrogenate acetylene to ethylene is -2.3075776000000356 eV

Energy to hydrogenate ethylene to ethane is  $-1.6680956000000762$  eV

Calculations suggest that it is easier to hydrogenate acetylene, since more energy is released in the reaction. However, since both reactions release energy it is difficult to selectively hydrogenate acetylene to ethylene instead of the reaction continuing to ethane.