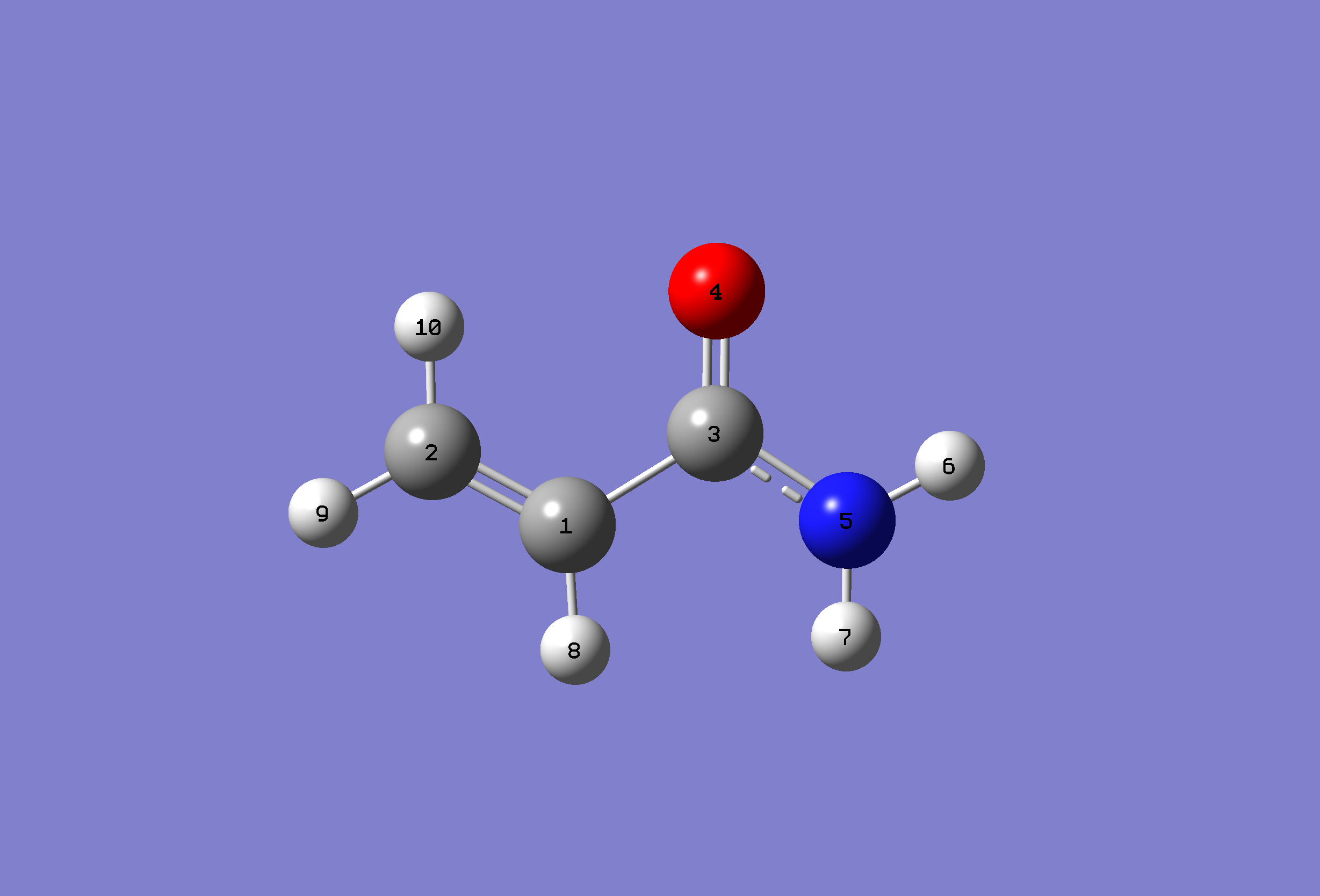
Acrylamide:



Standard orientation:

---------------------------------------------------------------------

Center Atomic Atomic Coordinates (Angstroms)

Number Number Type X Y Z

---------------------------------------------------------------------

1 6 0 -0.793203 -0.659633 -0.000778

2 6 0 -1.949141 -0.032481 0.000324

3 6 0 0.472196 0.125390 -0.000287

4 8 0 0.487264 1.344676 -0.000213

5 7 0 1.604322 -0.620663 0.000480

6 1 0 2.484046 -0.151283 0.000708

7 1 0 1.594865 -1.614995 0.000107

8 1 0 -0.726053 -1.729845 -0.000561

9 1 0 -2.884470 -0.556159 0.001095

10 1 0 -1.975867 1.039860 0.001437

---------------------------------------------------------------------

Bond Lengths:

Energy in Hartrees: -244.45026749 Hartrees

Energy in Joules: -1.070919 J/atom

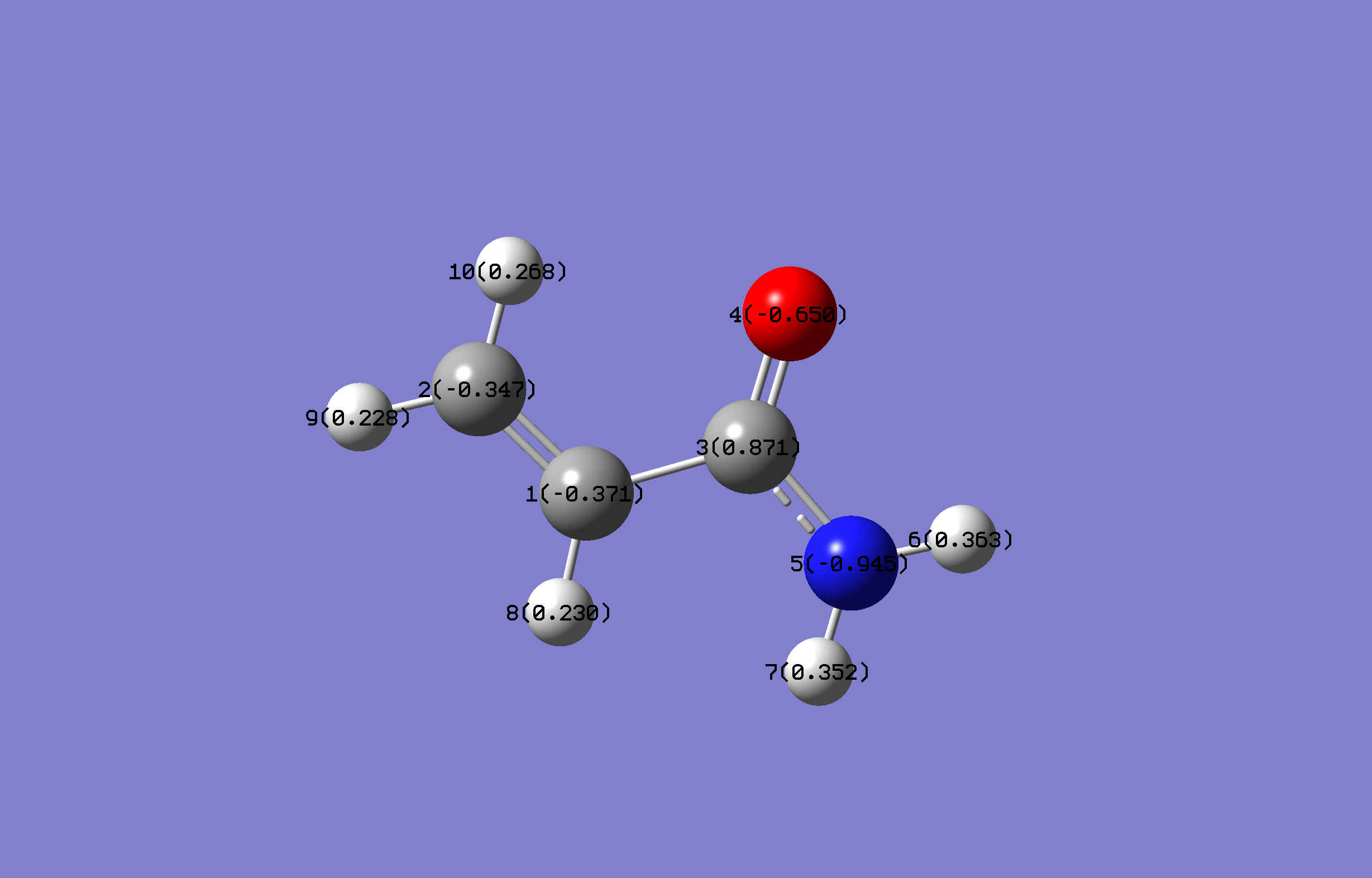
Energy in kJ/mol: -641789.6

Energy in MJ/mol: -641.7

Dipole moment: 3.9298 Debye

1.24752936e-29 C\*m

Dippr: **1.32000E-29 C\*m**

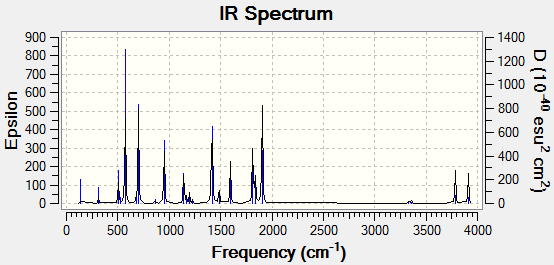


The Nitrogen is strongly impacting the electrons around it, and could be bonded to four surrounding molecules stably. Carbon 3 is deprived of electrons by the two electronegative atoms surrounding it, which induces electronegativity in Carbon 1.

#2

CO Stretch: 1905 Symmetric H-N-H: 3874

Antisymmetric H-N-H: 3911



Ideal Gas heat capacities:

C\_p: 18.884 Cal/mol-K -> 79.011 J/mol K

C\_v: 16.897 Cal/mol-K -> 70.697 J/mol K

Dippr:

C\_p: 80.742 J /mol K

C\_v: 72.427 J/mol K

Ideal Gas Entropy:

S: 70.935 Cal/mol-K -> 296.79 J/mol K

Dippr:

301.1 J/mol K

Moments of Inertia:

166.49080, 418.94595, 585.43663

Radius of Gyration:

2.9204x10-10 m

Dippr:

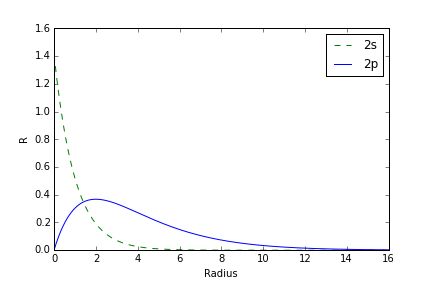
2.95x10-10 m

#3

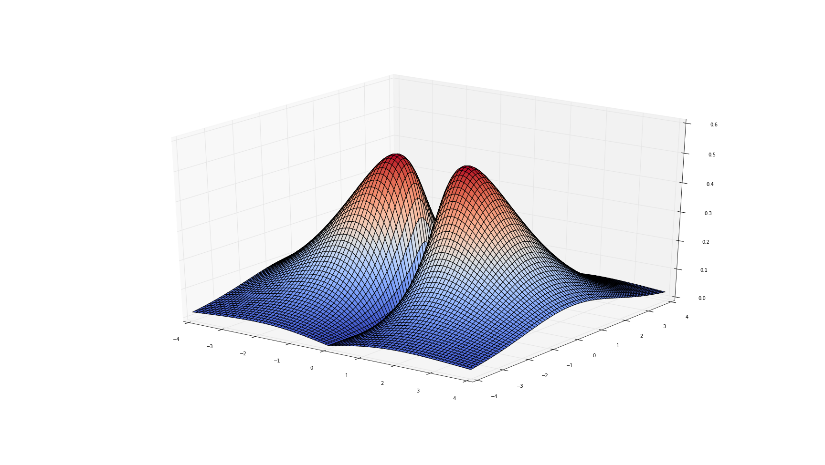
|  |  |  |
| --- | --- | --- |
|  | Cis | Trans |
|  |  |  |
| Dipole Moment | 3.0684 debye | 0 debye |
| Melting Point | 193.15 K | 223.35 K |
| Boiling Point | 333.65 K | 320.85 K |
| Heat Fusion @ Triple Point | 7.205x106 J/kmol | 1.198x107 J/kmol |
| Heat of Vaporization @ NBP | 2.9156x107 J/kmol | 2.8134x107 J/kmol |
| Critical Temperature | 544 K | 516.5 K |
| Liq. Density | 13.155 kmol/m3 | 12.881 kmol/m3 |

It is interesting that the boiling points and the melting points both seem strongly dependent on the dipole moment, as well as the heat of fusion. The density and the heat of vaporization were not affected as thoroughly. I was surprised that there was an effect on the critical temperature, but that is understandable. The reason there is a dipole moment on the cis, and not on the trans is because the electronegative Chlorine atoms pull equally (and synchronously) on the cis, while they pull equally (and antisynchronously) on the trans Carbon electrons.

#4

a) 

b) This is consistent with what I know about 2p orbitals, as there needs to be a node at zero, and a dumbbell shape for a given probability. The peaks represent the most probable x,z position for the electrons to be observed, and the trough represents space taken by the 2s orbital.



#5) Code used to generate plot:

from scipy.interpolate import interp1d

import matplotlib.pyplot as plt

from matplotlib import lines

from scipy.optimize import curve\_fit

%matplotlib qt

Energy = np.array([-276.83222734,-276.83356166,-276.83222734,-276.82958388,-276.82838551,-276.82979468,\

-276.83185709,-276.8314893 ,-276.82754547,-276.82220242,-276.81966900,\

-276.82220242,-276.82754547,-276.8314893 ,-276.83185709,-276.82979468,\

-276.82838551,-276.82958388,-276.83222734,-276.83356166,-276.83222734])

Angle = np.array([-200,-180,-160,-140,-120,-100,-80,-60,-40,-20,0,20,40,60,\

80,100,120,140,160,180,200])

function = interp1d(Angle,Energy,kind='cubic')

def lineFit(x,a,b,c,d):

ans = a+b\*np.cos(x\*np.pi/180.)+c\*np.cos(x\*np.pi/180.)\*\*2+d\*np.cos(x\*np.pi/180.)\*\*3

return ans

fit\_params,fit\_covar = curve\_fit(lineFit,Angle,Energy)

a,b,c,d=fit\_params

xAngle = np.linspace(-200,200,100)

Epred = function(xAngle)

plt.figure()

plt.plot(Angle,Energy,linestyle='None',marker='.',color='blue',label='Calculated Points')

#plt.plot(xAngle,Epred,linestyle='-')

plt.plot(xAngle,lineFit(xAngle,a,b,c,d),linestyle='-',color='green',label='Fitted')

y\_formatter = matplotlib.ticker.ScalarFormatter(useOffset=False)

ax = plt.gca()

ax.yaxis.set\_major\_formatter(y\_formatter)

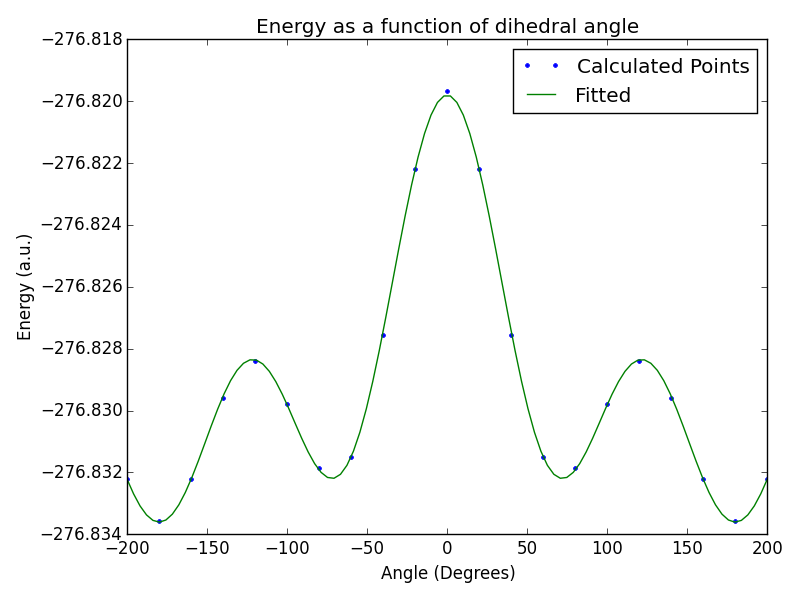
plt.xlabel('Angle (Degrees)')

plt.ylabel('Energy (a.u.)')

plt.title('Energy as a function of dihedral angle')

plt.legend(loc='best')

plt.tight\_layout()



|  |  |
| --- | --- |
| Energy (AU) | Dihedral angle |
| -276.83356166 | 180 |
| -276.83222734 | 160 |
| -276.82958388 | 140 |
| -276.82838551 | 120 |
| -276.82979468 | 100 |
| -276.83185709 | 80 |
| -276.8314893 | 60 |
| -276.82754547 | 40 |
| -276.82220242 | 20 |
| -276.81966900 | 0 |

#6)

a) def gs(x,alpha=1):

C = (2\*alpha/np.pi)\*\*0.75

ans = C\*np.exp(-alpha\*x\*\*2)

return ans

def gx(x,alpha=1):

C = (128\*alpha\*\*5/np.pi\*\*3)\*\*0.25

ans = C\*x\*np.exp(-alpha\*x\*\*2)

return ans

def gxx(x,alpha=1):

C = (2048\*alpha\*\*7/(9\*np.pi\*\*3))\*\*0.25

return C\*x\*\*2\*np.exp(-alpha\*x\*\*2)

x = np.linspace(-5,5,150)

plt.subplot(1,3,1)

plt.plot(x,gs(x),label='$\\alpha=1$')

plt.plot(x,gs(x,1.5),label='$\\alpha=1.5$')

plt.title('$g\_s$')

plt.subplot(1,3,2)

plt.plot(x,gx(x),label='$\\alpha=1$')

plt.plot(x,gx(x,1.5),label='$\\alpha=1.5$')

plt.legend(loc='lower right')

plt.title('$g\_x$')

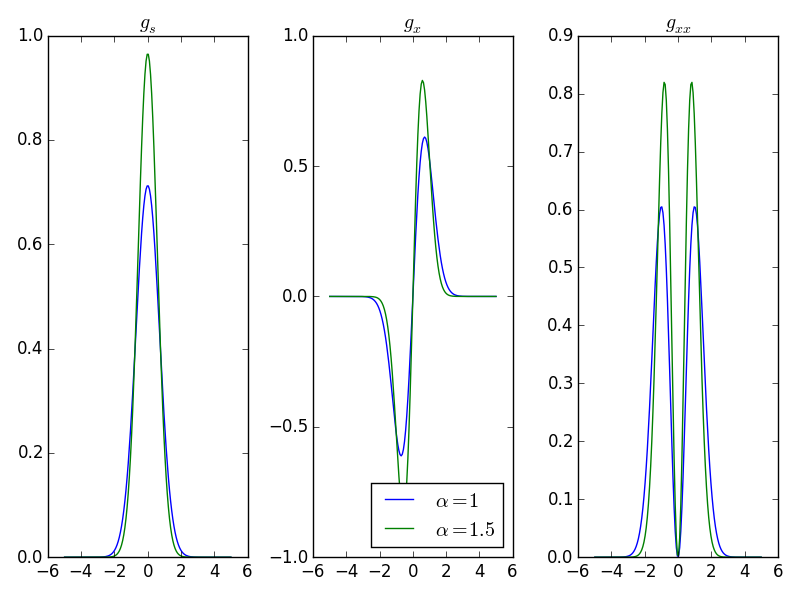
plt.subplot(1,3,3)

plt.plot(x,gxx(x),label='$\\alpha=1$')

plt.plot(x,gxx(x,1.5),label='$\\alpha=1.5$')

plt.title('$g\_{xx}$')

plt.tight\_layout()

 The orbital exponent changes the position of the peaks, and the value of alpha blurs the peak width.

b) from matplotlib import cm

def gxy(x,y):

alpha = 1

r = np.sqrt(x\*\*2 + y\*\*2+1)

C = (2048\*alpha\*\*7/(9\*np.pi\*\*3))\*\*0.25

return C\*x\*y\*np.exp(-alpha\*r\*\*2)

x = np.arange(-3,3,.01)

y = x.copy()

X,Y = np.meshgrid(x,y)

Z = gxy(X,Y)

import matplotlib.pyplot as plt

from mpl\_toolkits.mplot3d import Axes3D

fig = plt.figure()

ax = fig.add\_subplot(111, projection='3d')

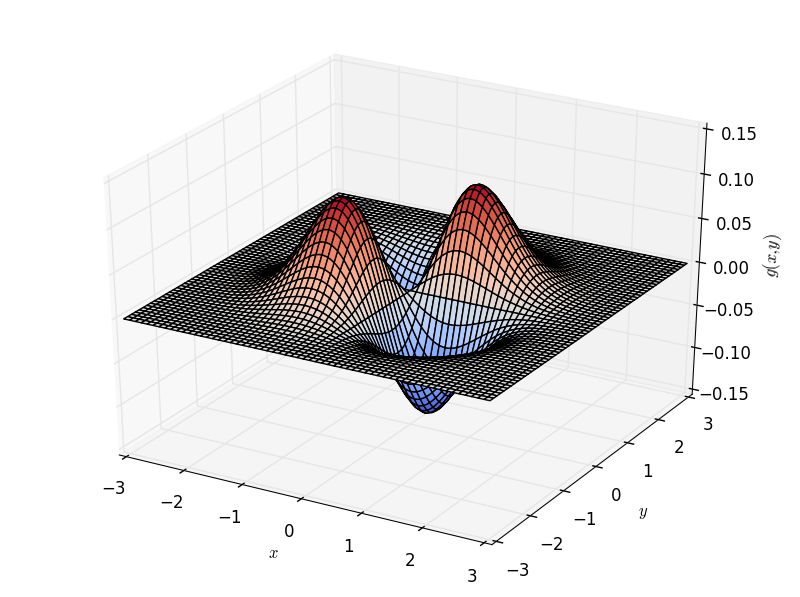
ax.plot\_surface(X,Y,Z,cmap=cm.coolwarm)

ax.set\_xlabel('$x$')

ax.set\_ylabel('$y$')

ax.set\_zlabel('$g(x,y)$')

plt.tight\_layout()



We can have both positive and negative values because the electron density (probability of finding an electron) is the square of this function.

c)

def gx(x,y,z,alpha=1):

r = np.sqrt(x\*\*2+y\*\*2+z\*\*2)

C = (128\*alpha\*\*5/np.pi\*\*3)\*\*0.25

ans = C\*x\*np.exp(-alpha\*r\*\*2)

return ans\*\*2

Z = gx(X,Y,0)

fig = plt.figure()

ax = fig.add\_subplot(131, projection='3d')

ax.plot\_surface(X,Y,Z,cmap=cm.coolwarm)

ax.set\_xlabel('x')

ax.set\_ylabel('y')

ax.set\_zlabel('g(x,y,z=0)')

ax = fig.add\_subplot(132, projection='3d')

Z = gx(X,0,Y)

ax.plot\_surface(X,Y,Z,cmap=cm.coolwarm)

ax.set\_xlabel('x')

ax.set\_ylabel('z')

ax.set\_zlabel('g(x,y=0,z)')

ax = fig.add\_subplot(133, projection='3d')

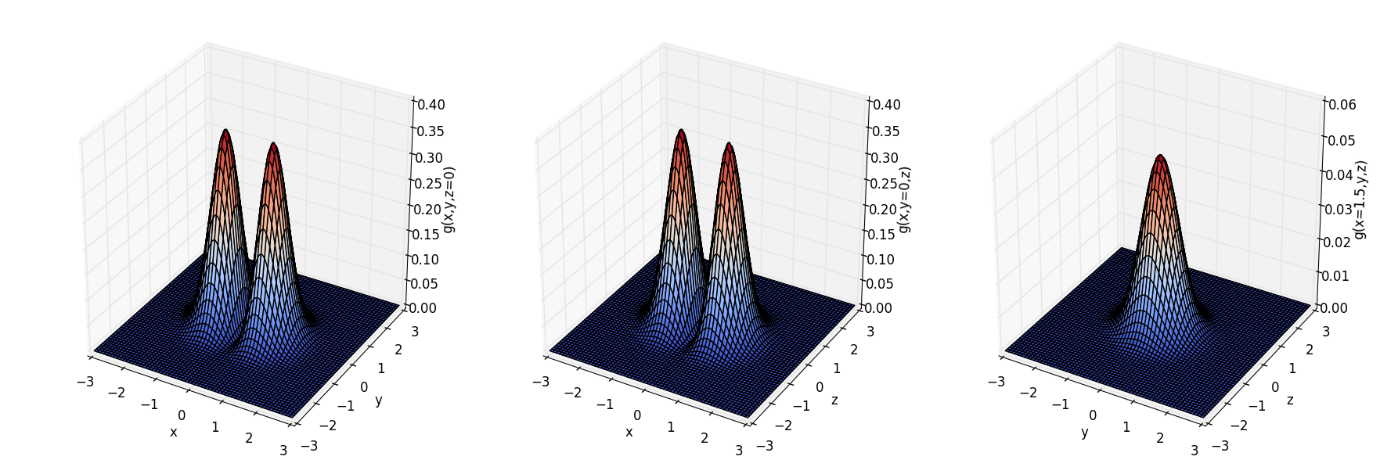
Z = gx(1.5,X,Y)

ax.plot\_surface(X,Y,Z,cmap=cm.coolwarm)

ax.set\_xlabel('y')

ax.set\_ylabel('z')

ax.set\_zlabel('g(x=1.5,y,z)')

plt.tight\_layout()  This does match what I would expect for a p orbital. There is zero probability of finding the electrons at x=0, and it has a dumbbell shape on any isosurface..