# Problem 1

## Subproblem 1.1

The code is in file hw03\_1\_1.py and shown below. The reaction energy is -322.61 kJ/mol.

```
from ase.data.molecules import molecule
  from jasp import *
  # Define molecules
5 \mid CO = molecule('CO')
  CO. set_cell([8, 8, 8], scale_atoms=False)
  CO.center()
  O2 = molecule('O2')
  O2.set\_cell([8, 8, 8], scale\_atoms=False)
11 O2. center ()
13
  CO2 = molecule('CO2')
14 CO2.set_cell([8, 8, 8], scale_atoms=False)
15
  CO2. center()
17
  # Create calculators to get the energies
   with jasp ('molecules/hw03/CO-\{0\}'. format (250),
18
       xc = PBE'
19
       encut = 250,
20
       ismear=1,
21
22
       ibrion=2,
23
       nsw=10,
       atoms=CO) as calc:
24
25
            eCO = CO.get\_potential\_energy()
26
            print 'CO Forces\n=
27
            print CO.get_forces()
28
       except (VaspSubmitted, VaspQueued):
29
            eCO = None
   with jasp ('molecules/hw03/O2-\{0\}'. format (250),
32
       xc = 'PBE'
       encut = 250,
33
       ismear=1,
34
35
       ibrion=2.
       nsw=10,
36
37
       atoms=O2) as calc:
38
39
            eO2 = O2.get_potential_energy()
            print 'O2 Forces\n=
40
            print O2.get_forces()
41
       except (VaspSubmitted, VaspQueued):
42
            eO2 = None
43
   with jasp ('molecules/hw03/CO2-\{0\}'. format (250),
44
       xc = PBE
45
46
       encut = 250,
47
       ismear=1,
       ibrion=2,
48
       nsw=10,
49
       atoms=CO2) as calc:
50
51
52
            eCO2 = CO2.get_potential_energy()
            print 'CO2 Forces\n=
53
54
            print CO2.get_forces()
       \begin{array}{ll} \textbf{except} & (VaspSubmitted\;,\;\; VaspQueued): \end{array}
55
56
            eCO2 = None
57
58
   if None in (eCO, eO2, eCO2):
59
       pass
60
       {\rm dE} \, = \, {\rm eCO2} \, - \, {\rm eCO} \, - \, \, 0.5\!*\!{\rm eO2}
```

October 12, 2012 1 of 8

#### Output:

```
CO Forces
=======
[[ 0.
           0.
                   0.004]
 Γ0.
           0.
                  -0.004]]
02 Forces
=======
[[ 0.
           0.
                   0.051]
                  -0.051]]
 [ 0.
           0.
CO2 Forces
=======
[[ 0.
           0.
                   0.
                         ]
 ΓО.
                   0.008]
           0.
 Γ0.
           0.
                  -0.008]]
Delta E = -3.344 \text{ eV}
Delta E = -77.106 \text{ kcal/mol}
Delta E = -322.613 \text{ kJ/mol}
```

### Subproblem 1.2

The code is in file hw03\_1\_2.py and shown below.

```
from ase.data.molecules import molecule
  from jasp import *
  # Define molecules
  CO = molecule ('CO')
  CO.set\_cell([8, 8, 8], scale\_atoms=False)
  CO. center()
  O2 = molecule('O2')
  O2.\,set\_cell\,([\,8\,\,,\  \, 8\,,\  \, 8\,]\,\,,\  \, scale\_atoms{=}False\,)
11
  O2.center()
CO2 = molecule('CO2')
  CO2.set\_cell([8, 8, 8], scale\_atoms=False)
14
  CO2.center()
15
  encuts = [250, 350, 450, 500]
17
  DCO,\ DO2,\ DCO2,\ Drxn\ =\ [\ ]\ ,\ \ [\ ]\ ,\ \ [\ ]
  for encut in encuts:
19
       with jasp ('molecules/hw03/CO-{0}'.format(encut),
20
            xc='PBE',
21
            encut=encut,
            ismear=1,
            \#sigma=0.01,
24
25
            ibrion=2,
26
            nsw=10,
            atoms=CO) as calc:
27
            try:
```

October 12, 2012 2 of 8

```
eCO = CO.get potential energy()
                DCO. append (eCO)
30
31
           except (VaspSubmitted, VaspQueued):
                eCO = None
32
33
       with jasp ('molecules/hw03/O2-{0}'.format(encut),
           xc='PBE',
34
           encut=encut,
           ismear=1,
36
37
           \#sigma=0.01
           ibrion=2,
38
39
           nsw=10,
           atoms=O2) as calc:
40
41
                eO2 = O2.get potential energy()
42
                DO2. append (e\overline{O}2)
43
44
           except (VaspSubmitted, VaspQueued):
                eO2 = None
45
46
       with jasp ('molecules/hw03/CO2-{0}'.format(encut),
           xc='PBE',
47
48
           encut=encut.
49
           ismear=1,
           \#sigma = 0.01,
50
51
           ibrion=2,
           nsw=10,
           atoms=CO2) as calc:
53
54
                eCO2 = CO2.get potential energy()
                DCO2. append (eCO2)
56
           except (VaspSubmitted, VaspQueued):
57
58
                eCO2 = None
       if None not in (eCO, eO2, eCO2):
60
           {\rm dE} \, = \, {\rm eCO2} \, - \, {\rm eCO} \, - \, \, 0.5\!*\!{\rm eO2}
61
62
           Drxn.append(dE)
  import matplotlib.pyplot as plt
64
  plt.figure()
  plt.subplot(221)
67 plt.plot(encuts, DCO, '-bo')
68 plt.xlabel('ENCUT (eV)')
69
  plt.ylabel('CO Energy (eV)')
  plt.subplot(222)
  plt.plot(encuts, DO2, '-ro')
72 plt.xlabel('ENCUT (eV)')
73 plt.ylabel('O$_2$ Energy (eV)')
  plt.subplot(223)
74
  plt.plot(encuts, DCO2, '-go')
76 plt.xlabel('ENCUT (eV)')
77 plt.ylabel('CO$ 2$ Energy (eV)')
  plt.subplot(224)
  plt.plot(encuts, Drxn, '-ko')
plt.xlabel('ENCUT (eV)')
79
81 plt.ylabel ('CO Oxidation Energy (eV)')
82 plt.tight layout()
83 plt.savefig('images/problem-1-2-convergence.png')
```

Figure 1 shows the convergence results for individual molecules and reaction energy. The energies converge faster for the individual molecules as they reach a plateau for lower cut-off energy than the reaction energy.

October 12, 2012 3 of 8

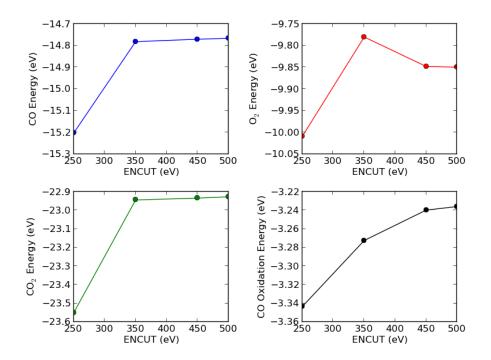


Figure 1: Convergence of energies for Subproblem 1.2.

# Problem 2

# Subproblem 2.1

The code is in file hw03\_2\_1.py and shown below.

```
from ase.data.molecules import molecule
  from jasp import *
  # Define molecules
  CO = molecule('CO')
  CO.set_cell([8, 8, 8], scale_atoms=False)
  CO. center()
  O2 = molecule('O2')
  O2.set_cell([8, 8, 8], scale_atoms=False)
  O2.center()
11
12
13
  CO2 = molecule('CO2')
  CO2. set_cell([8, 8, 8], scale_atoms=False)
14
  CO2.center()
15
  with jasp ('molecules/hw03/CO vib-\{0\}'. format (350),
18
19
      ismear=0,# Gaussian smearing
       ibrion=6,# finite differences with symmetry
20
21
       nfree = 2, # central differences (default)
      potim = 0.015, # default as well
23
       ediff=1e-8,
      nsw=1,
24
25
      atoms=CO) as calc:
26
27
          CO.get_forces()
           energies, modes = calc.get vibrational modes()
```

October 12, 2012 4 of 8

```
print 'CO Energies\n==
             for i, e in enumerate(energies):
30
        \begin{array}{c} print \ \ '\{0:02d\}\colon \ \{1\} \ \ eV \ \ '. \ format(i\ ,\ e) \\ except \ \ (VaspSubmitted\ ,\ VaspQueued): \end{array}
31
32
33
   with jasp ('molecules/hw03/O2_vib-\{0\}'. format (350),
        encut = 400,
36
37
        ismear=0,# Gaussian smearing
        ibrion=6,# finite differences with symmetry
38
        nfree = 2, # central differences (default)
39
        \operatorname{potim} = 0.015, # default as well
40
        ediff=1e-8,
41
        nsw=1,
42
        atoms=O2) as calc:
43
44
        try:
             O2.get_forces()
45
46
             energies , modes = calc.get_vibrational_modes()
             print 'O2 Energies\n=
47
             for i, e in enumerate(energies):
48
        print '{0:02d}: {1} eV'.format(i, e)
except (VaspSubmitted, VaspQueued):
49
50
51
   with jasp ('molecules/hw03/CO2 vib-{0}'.format(350),
53
        encut = 400,
54
55
        ismear=0,# Gaussian smearing
        ibrion = 6,# finite differences with symmetry
56
        nfree=2, # central differences (default)
57
        potim = 0.015, # default as well
58
        ediff=1e-8,
        nsw=1,
60
        atoms=CO2) as calc:
61
62
        \mathbf{try}:
63
             CO2.get forces()
             energies , modes = calc.get_vibrational_modes()
64
             print 'CO2 Energies\n==='
for i, e in enumerate(energies):
65
66
                  print ~`\{0:02d\}:~\{1\}~eV~'.format(i~,~e)
67
        except (VaspSubmitted, VaspQueued):
68
             pass
```

#### Output:

#### CO Energies

October 12, 2012 5 of 8

```
04: 0.0 eV

05: (1e-09+0j) eV

C02 Energies

=======

00: 0.290887755 eV

01: 0.162520019 eV

02: 0.079241315 eV

03: 0.079241315 eV

04: 0.010559629 eV

05: 0.010559629 eV

06: (6.743e-05+0j) eV

07: (0.000333671+0j) eV

08: (0.000333671+0j) eV
```

The results are summarized in the table below.

Molecule	Vibrational Modes [eV]
CO	0.2570
$O_2$	0.1860
$\mathrm{CO}_2$	0.2909,  0.1625

### Subproblem 2.2

The code is in file  $hw03_2_2.py$  and shown below. The difference in energy when accounting for ZPE is  $10.63\,kJ/mol$ . Hence, the total difference in energy is  $-302.89\,kJ/mol$ .

```
import numpy as np
  from jasp import *
  c = 3e10 \# speed of light cm/s
  h = 4.135667516e - 15 \# eV/s
  # Get frequencies and calculate ZPE
   with jasp('molecules/hw03/CO_vib-{0}'.format(350)) as calc:
       try:
             eCO = calc.get_potential_energy(calc.get_atoms())
             COfreq = calc.get_vibrational_frequencies()
            COZPE = np.sum([0.5*h*f*c for f in COfreq if isinstance(f, float)])
12
        except (IOError):
             COfreq = None
13
14
   with jasp('molecules/hw03/O2\_vib-\{0\}'.format(350)) as calc:
16
             eO2 = calc.get_potential_energy(calc.get_atoms())
17
            \begin{array}{lll} O2freq = calc.get\_vibrational\_frequencies() \\ O2ZPE = np.sum([0.5*h*f*c for f in O2freq if isinstance(f, float)]) \end{array}
18
19
        except (IOError):
20
21
             O2freq = None
22
   with jasp('molecules/hw03/CO2\_vib-\{0\}'.format(350)) as calc:
23
24
             eCO2 = calc.get_potential_energy(calc.get_atoms())
25
            \begin{array}{lll} CO2 freq = calc.get\_vibrational\_frequencies() \\ CO2 ZPE = np.sum([0.5*h*f*c for f in CO2 freq if isinstance(f, float)]) \end{array}
26
27
28
        except (IOError):
             CO2freq = None
29
```

October 12, 2012 6 of 8

```
if None not in (COfreq, O2freq, CO2freq):
         \mathrm{dE} = \mathrm{eCO2} - \mathrm{eCO} - 0.5 * \mathrm{eO2}
33
         dZPE \,=\, CO2ZPE \,-\, COZPE \,-\, 0.5*O2ZPE
         print 'Delta E = \{0:1.3 f\} eV'. format (dE)
34
         print 'Delta E = \{0:1.3f\} kcal/mol'. format (dE*23.06035)
35
         print 'Delta E = \{0:1.3 f\} kJ/mol'.format(dE*96.485)
36
         print 'Delta ZPE = {0:1.3 f} eV'.format(dZPE)
print 'Delta ZPE = {0:1.3 f} kcal/mol'.format(dZPE*23.06035)
print 'Delta ZPE = {0:1.3 f} kJ/mol'.format(dZPE*96.485)
print 'Delta Total E = {0:1.3 f} eV'.format(dE + dZPE)
38
39
40
          print 'Delta Total E = \{0:1.3f\} kcal/mol'.format((dE + dZPE) *23.06035)
41
         print 'Delta Total E = \{0:1.3f\} kJ/mol'. format ((dE + dZPE) *96.485)
```

#### Output:

```
Delta E = -3.249 eV

Delta E = -74.933 kcal/mol

Delta E = -313.520 kJ/mol

Delta ZPE = 0.110 eV

Delta ZPE = 2.540 kcal/mol

Delta ZPE = 10.626 kJ/mol

Delta Total E = -3.139 eV

Delta Total E = -72.393 kcal/mol

Delta Total E = -302.894 kJ/mol
```

## Subproblem 2.3

The value for the reaction energy of CO combustion found in the literature is: -283.00 kJ/mol. Source: http://www.science.uwaterloo.ca/~cchieh/cact/c120/hess.html.

# 1 Problem 3

The code is in file  $hw03_3.py$  and shown below.

```
from jasp import *
  from enthought.mayavi import mlab
  from ase.data.vdw import vdw radii
  from ase.data.colors import cpk colors
  from ase.data.molecules import molecule
  CO2 = molecule('CO2')
  CO2. set cell([8, 8, 8], scale atoms=False)
  CO2.center()
10
  with jasp ('molecules/hw03/CO2-\{0\}'. format(350)) as calc:
12
          atoms = calc.get\_atoms()
          x, y, z, cd = calc.get_charge_density()
14
          mlab.figure(bgcolor=(1, 1, 1))
15
17
          # plot the atoms as spheres
          for atom in atoms:
18
19
               mlab.points3d(atom.x,
20
                              atom.y,
                              atom.z,
```

October 12, 2012 7 of 8

```
scale_factor=vdw_radii[atom.number]/5.,
23
                                 resolution = 20,
24
                                 # a tuple is required for the color
25
                                 color=tuple(cpk_colors[atom.number]),
26
                                 scale_mode='none')
27
            # draw the unit cell - there are 8 corners, and 12 connections
28
            a1, a2, a3 = atoms.get_cell()
29
30
            origin = [0, 0, 0]
            cell_matrix = [[origin,
31
32
                              [origin,
                                         a3] ,
33
                               origin,
                                         a1 + a2],
                              [a1,
34
35
                              [a1,
                                         a1 + a3,
                              a2,
                                         a2 + a1,
36
37
                              [a2,
                                         a2 + a3,
                              ĺa3,
                                         a1 + a3],
38
39
                              [a3,
                                         a2 + a3,
                              [a1 + a2, a1 + a2 + a3],
40
41
                              [a2 + a3, a1 + a2 + a3],
                              [a1 + a3, a1 + a3 + a2]
42
43
44
            for p1, p2 in cell_matrix:
                mlab.plot3d([p1[0], p2[0]], #x-positions
45
                               \left[\,\mathrm{p1}\left[\,1\right]\,,\ \mathrm{p2}\left[\,1\,\right]\,\right]\,,\ \#\ \mathrm{y-positions}
46
                               [p1[2], p2[2]], \# z-positions
47
                               tube radius = 0.02)
48
49
50
            # Now plot the charge density
51
            mlab.contour3d(x, y, z, cd, transparent=True)
53
            # this view was empirically found by iteration
54
55
            mlab.view(azimuth=-90, elevation=90, distance='auto')
56
            mlab.savefig('images/co2-centered-cd.png')
57
58
            mlab.show()
       except (VaspSubmitted, VaspQueued, IOError):
59
60
            pass
```

The electron density of  $CO_2$  is shown in Figure 2.

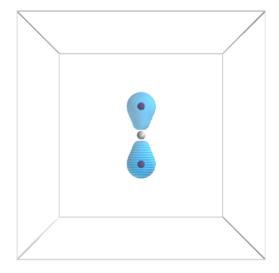


Figure 2: Charge density for CO<sub>2</sub> for Problem 3.

October 12, 2012 8 of 8