

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.

```

1 from ase.data.molecules import molecule
2 from jasp import *
3
4 # Define molecules
5 CO = molecule('CO')
6 CO.set_cell([8, 8, 8], scale_atoms=False)
7 CO.center()
8
9 O2 = molecule('O2')
10 O2.set_cell([8, 8, 8], scale_atoms=False)
11 O2.center()
12
13 CO2 = molecule('CO2')
14 CO2.set_cell([8, 8, 8], scale_atoms=False)
15 CO2.center()
16
17 # Create calculators to get the energies
18 with jasp('molecules/hw03/CO-{}'.format(250),
19          xc='PBE',
20          encut=250,
21          ismear=1,
22          ibrion=2,
23          nsw=10,
24          atoms=CO) as calc:
25     try:
26         eCO = CO.get_potential_energy()
27         print 'CO Forces\n=====',
28         print CO.get_forces()
29     except (VaspSubmitted, VaspQueued):
30         eCO = None
31 with jasp('molecules/hw03/O2-{}'.format(250),
32          xc='PBE',
33          encut=250,
34          ismear=1,
35          ibrion=2,
36          nsw=10,
37          atoms=O2) as calc:
38     try:
39         eO2 = O2.get_potential_energy()
40         print 'O2 Forces\n=====',
41         print O2.get_forces()
42     except (VaspSubmitted, VaspQueued):
43         eO2 = None
44 with jasp('molecules/hw03/CO2-{}'.format(250),
45          xc='PBE',
46          encut=250,
47          ismear=1,
48          ibrion=2,
49          nsw=10,
50          atoms=CO2) as calc:
51     try:
52         eCO2 = CO2.get_potential_energy()
53         print 'CO2 Forces\n=====',
54         print CO2.get_forces()
55     except (VaspSubmitted, VaspQueued):
56         eCO2 = None
57
58 if None in (eCO, eO2, eCO2):
59     pass
60 else:
61     dE = eCO2 - eCO - 0.5*eO2

```

```
62     print 'Delta E = {0:1.3f} eV'.format(dE)
63     print 'Delta E = {0:1.3f} kcal/mol'.format(dE*23.06035)
64     print 'Delta E = {0:1.3f} kJ/mol'.format(dE*96.485)
```

Output:

CO Forces

=====

```
[[ 0.    0.    0.004]
 [ 0.    0.   -0.004]]
```

O2 Forces

=====

```
[[ 0.    0.    0.051]
 [ 0.    0.   -0.051]]
```

CO2 Forces

=====

```
[[ 0.    0.    0.   ]
 [ 0.    0.    0.008]
 [ 0.    0.   -0.008]]
```

Delta E = -3.344 eV

Delta E = -77.106 kcal/mol

Delta E = -322.613 kJ/mol

Subproblem 1.2

The code is in file hw03_1_2.py and shown below.

```
1 from ase.data.molecules import molecule
2 from jasp import *
3
4 # Define molecules
5 CO = molecule('CO')
6 CO.set_cell([8, 8, 8], scale_atoms=False)
7 CO.center()
8
9 O2 = molecule('O2')
10 O2.set_cell([8, 8, 8], scale_atoms=False)
11 O2.center()
12
13 CO2 = molecule('CO2')
14 CO2.set_cell([8, 8, 8], scale_atoms=False)
15 CO2.center()
16
17 encuts = [250, 350, 450, 500]
18 DCO, DO2, DCO2, Drxn = [], [], [], []
19 for encut in encuts:
20     with jasp('molecules/hw03/CO-{0}'.format(encut),
21             xc='PBE',
22             encut=encut,
23             ismear=1,
24             #sigma=0.01,
25             ibrion=2,
26             nsw=10,
27             atoms=CO) as calc:
28         try:
```

```
29         eCO = CO.get_potential_energy()
30         DCO.append(eCO)
31     except (VaspSubmitted, VaspQueued):
32         eCO = None
33     with jasp('molecules/hw03/O2-{0}'.format(encut),
34             xc='PBE',
35             encut=encut,
36             ismear=1,
37             #sigma=0.01,
38             ibrion=2,
39             nsw=10,
40             atoms=O2) as calc:
41         try:
42             eO2 = O2.get_potential_energy()
43             DO2.append(eO2)
44         except (VaspSubmitted, VaspQueued):
45             eO2 = None
46     with jasp('molecules/hw03/CO2-{0}'.format(encut),
47             xc='PBE',
48             encut=encut,
49             ismear=1,
50             #sigma=0.01,
51             ibrion=2,
52             nsw=10,
53             atoms=CO2) as calc:
54         try:
55             eCO2 = CO2.get_potential_energy()
56             DCO2.append(eCO2)
57         except (VaspSubmitted, VaspQueued):
58             eCO2 = None
59
60     if None not in (eCO, eO2, eCO2):
61         dE = eCO2 - eCO - 0.5*eO2
62         Drxn.append(dE)
63
64 import matplotlib.pyplot as plt
65 plt.figure()
66 plt.subplot(221)
67 plt.plot(encuts, DCO, '-bo')
68 plt.xlabel('ENCUT (eV)')
69 plt.ylabel('CO Energy (eV)')
70 plt.subplot(222)
71 plt.plot(encuts, DO2, '-ro')
72 plt.xlabel('ENCUT (eV)')
73 plt.ylabel('O$ _2$ Energy (eV)')
74 plt.subplot(223)
75 plt.plot(encuts, DCO2, '-go')
76 plt.xlabel('ENCUT (eV)')
77 plt.ylabel('CO$ _2$ Energy (eV)')
78 plt.subplot(224)
79 plt.plot(encuts, Drxn, '-ko')
80 plt.xlabel('ENCUT (eV)')
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.

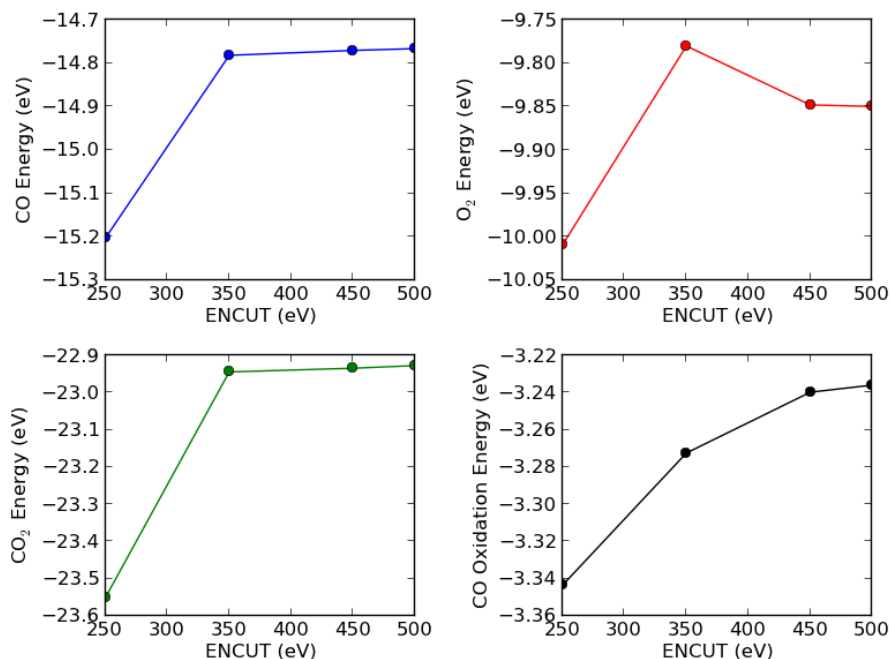


Figure 1: Convergence of energies for Subproblem 1.2.

Problem 2

Subproblem 2.1

The code is in file `hw03_2_1.py` and shown below.

```

1 from ase.data.molecules import molecule
2 from jasp import *
3
4 # Define molecules
5 CO = molecule('CO')
6 CO.set_cell([8, 8, 8], scale_atoms=False)
7 CO.center()
8
9 O2 = molecule('O2')
10 O2.set_cell([8, 8, 8], scale_atoms=False)
11 O2.center()
12
13 CO2 = molecule('CO2')
14 CO2.set_cell([8, 8, 8], scale_atoms=False)
15 CO2.center()
16
17 with jasp('molecules/hw03/CO_vib-{}'.format(350),
18          encut=400,
19          ismear=0, # Gaussian smearing
20          ibrion=6, # finite differences with symmetry
21          nfree=2, # central differences (default)
22          potim=0.015, # default as well
23          ediff=1e-8,
24          nsw=1,
25          atoms=CO) as calc:
26     try:
27         CO.get_forces()
28         energies, modes = calc.get_vibrational_modes()

```

```
29     print 'CO Energies\n=====',
30     for i, e in enumerate(energies):
31         print '{0:02d}: {1} eV'.format(i, e)
32 except (VaspSubmitted, VaspQueued):
33     pass
34
35 with jasp('molecules/hw03/O2_vib-{0}'.format(350),
36          encut=400,
37          ismear=0, # Gaussian smearing
38          ibrion=6, # finite differences with symmetry
39          nfree=2, # central differences (default)
40          potim=0.015, # default as well
41          ediff=1e-8,
42          nsw=1,
43          atoms=O2) as calc:
44     try:
45         O2.get_forces()
46         energies, modes = calc.get_vibrational_modes()
47         print 'O2 Energies\n=====',
48         for i, e in enumerate(energies):
49             print '{0:02d}: {1} eV'.format(i, e)
50 except (VaspSubmitted, VaspQueued):
51     pass
52
53 with jasp('molecules/hw03/CO2_vib-{0}'.format(350),
54          encut=400,
55          ismear=0, # Gaussian smearing
56          ibrion=6, # finite differences with symmetry
57          nfree=2, # central differences (default)
58          potim=0.015, # default as well
59          ediff=1e-8,
60          nsw=1,
61          atoms=CO2) as calc:
62     try:
63         CO2.get_forces()
64         energies, modes = calc.get_vibrational_modes()
65         print 'CO2 Energies\n=====',
66         for i, e in enumerate(energies):
67             print '{0:02d}: {1} eV'.format(i, e)
68 except (VaspSubmitted, VaspQueued):
69     pass
```

Output:

CO Energies

=====

00: 0.257043282 eV
01: 0.020844782 eV
02: 0.020844782 eV
03: (1.18e-05+0j) eV
04: (0.000982597+0j) eV
05: (0.000982597+0j) eV

O2 Energies

=====

00: 0.186040739 eV
01: 0.021139118 eV
02: 0.021139118 eV
03: 2e-09 eV

```
04: 0.0 eV
05: (1e-09+0j) eV
CO2 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
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.

```
1 import numpy as np
2 from jasp import *
3 c = 3e10 # speed of light cm/s
4 h = 4.135667516e-15 # eV/s
5
6 # Get frequencies and calculate ZPE
7 with jasp('molecules/hw03/CO_vib-{0}'.format(350)) as calc:
8     try:
9         eCO = calc.get_potential_energy(calc.get_atoms())
10        COfreq = calc.get_vibrational_frequencies()
11        COZPE = np.sum([0.5*h*f*c for f in COfreq if isinstance(f, float)])
12    except IOError:
13        COfreq = None
14
15 with jasp('molecules/hw03/O2_vib-{0}'.format(350)) as calc:
16     try:
17         eO2 = calc.get_potential_energy(calc.get_atoms())
18         O2freq = calc.get_vibrational_frequencies()
19         O2ZPE = np.sum([0.5*h*f*c for f in O2freq if isinstance(f, float)])
20     except IOError:
21         O2freq = None
22
23 with jasp('molecules/hw03/CO2_vib-{0}'.format(350)) as calc:
24     try:
25         eCO2 = calc.get_potential_energy(calc.get_atoms())
26         CO2freq = calc.get_vibrational_frequencies()
27         CO2ZPE = np.sum([0.5*h*f*c for f in CO2freq if isinstance(f, float)])
28     except IOError:
29         CO2freq = None
30
```

```

31 if None not in (COfreq, O2freq, CO2freq):
32     dE = eCO2 - eCO - 0.5*eO2
33     dZPE = CO2ZPE - COZPE - 0.5*O2ZPE
34     print 'Delta E = {0:1.3f} eV'.format(dE)
35     print 'Delta E = {0:1.3f} kcal/mol'.format(dE*23.06035)
36     print 'Delta E = {0:1.3f} kJ/mol'.format(dE*96.485)
37     print 'Delta ZPE = {0:1.3f} eV'.format(dZPE)
38     print 'Delta ZPE = {0:1.3f} kcal/mol'.format(dZPE*23.06035)
39     print 'Delta ZPE = {0:1.3f} kJ/mol'.format(dZPE*96.485)
40     print 'Delta Total E = {0:1.3f} eV'.format(dE + dZPE)
41     print 'Delta Total E = {0:1.3f} kcal/mol'.format((dE + dZPE)*23.06035)
42     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.

```

1 from jasp import *
2 from enthought.mayavi import mlab
3 from ase.data.vdw import vdw_radii
4 from ase.data.colors import cpk_colors
5 from ase.data.molecules import molecule
6
7 CO2 = molecule('CO2')
8 CO2.set_cell([8, 8, 8], scale_atoms=False)
9 CO2.center()
10
11 with jasp('molecules/hw03/CO2-{0}'.format(350)) as calc:
12     try:
13         atoms = calc.get_atoms()
14         x, y, z, cd = calc.get_charge_density()
15         mlab.figure(bgcolor=(1, 1, 1))
16
17         # plot the atoms as spheres
18         for atom in atoms:
19             mlab.points3d(atom.x,
20                           atom.y,
21                           atom.z,

```

```

22         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
28     # draw the unit cell - there are 8 corners, and 12 connections
29     a1, a2, a3 = atoms.get_cell()
30     origin = [0, 0, 0]
31     cell_matrix = [[origin, a1],
32                   [origin, a2],
33                   [origin, a3],
34                   [a1, a1 + a2],
35                   [a1, a1 + a3],
36                   [a2, a2 + a1],
37                   [a2, a2 + a3],
38                   [a3, a1 + a3],
39                   [a3, a2 + a3],
40                   [a1 + a2, a1 + a2 + a3],
41                   [a2 + a3, a1 + a2 + a3],
42                   [a1 + a3, a1 + a3 + a2]]
43
44     for p1, p2 in cell_matrix:
45         mlab.plot3d([p1[0], p2[0]], # x-positions
46                   [p1[1], p2[1]], # y-positions
47                   [p1[2], p2[2]], # z-positions
48                   tube_radius=0.02)
49
50
51     # Now plot the charge density
52     mlab.contour3d(x, y, z, cd, transparent=True)
53
54     # this view was empirically found by iteration
55     mlab.view(azimuth=-90, elevation=90, distance='auto')
56
57     mlab.savefig('images/co2-centered-cd.png')
58     mlab.show()
59 except (VaspSubmitted, VaspQueued, IOError):
60     pass

```

The electron density of CO₂ is shown in Figure 2.

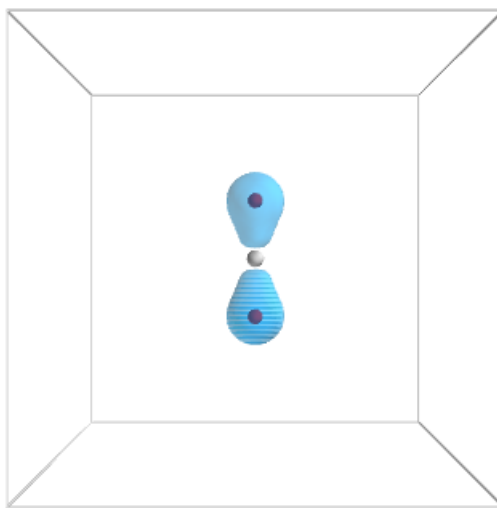


Figure 2: Charge density for CO₂ for Problem 3.