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.

```python
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()

CO2 = molecule('CO2')
CO2.set_cell([8, 8, 8], scale_atoms=False)
CO2.center()

# Create calculators to get the energies
with jasp('molecules/hw03/CO-{0}'.format(250),
    xc='PBE',
    encut=250,
    ismear=1,
    ibrion=2,
    nsw=10,
    atoms=CO) as calc:
    try:
        eCO = CO.get_potential_energy()
        print 'CO Forces\n=========='
        print CO.get_forces()
    except (VaspSubmitted, VaspQueued):
        eCO = None
with jasp('molecules/hw03/O2-{0}'.format(250),
    xc='PBE',
    encut=250,
    ismear=1,
    ibrion=2,
    nsw=10,
    atoms=O2) as calc:
    try:
        eO2 = O2.get_potential_energy()
        print 'O2 Forces\n=========='
        print O2.get_forces()
    except (VaspSubmitted, VaspQueued):
        eO2 = None
with jasp('molecules/hw03/CO2-{0}'.format(250),
    xc='PBE',
    encut=250,
    ismear=1,
    ibrion=2,
    nsw=10,
    atoms=CO2) as calc:
    try:
        eCO2 = CO2.get_potential_energy()
        print 'CO2 Forces\n=========='
        print CO2.get_forces()
    except (VaspSubmitted, VaspQueued):
        eCO2 = None

if None in (eCO, eO2, eCO2):
    pass
else:
    dE = eCO2 - eCO - 0.5*eO2
    print 'Delta E = {0:1.3f} eV'.format(dE)
    print 'Delta E = {0:1.3f} kcal/mol'.format(dE*23.06035)
    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.

```
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()

CO2 = molecule('CO2')
CO2.set_cell([8, 8, 8], scale_atoms=False)
CO2.center()

encuts = [250, 350, 450, 500]
DCO, DO2, DCO2, Drxn = [], [], [], []
for encut in encuts:
    with jasp('molecules/hw03/CO-{0}'.format(encut),
        xc='PBE',
        encut=encut,
        ismear=1,
        #sigma=0.01,
        ibrion=2,
        nsw=10,
        atoms=CO) as calc:
        try:
            eCO = CO.get_potential_energy()
            DCO.append(eCO)
        except (VaspSubmitted, VaspQueued):
            eCO = None
    with jasp('molecules/hw03/O2-{0}'.format(encut),
        xc='PBE',
        encut=encut,
        ismear=1,
        #sigma=0.01,
        ibrion=2,
        nsw=10,
        atoms=O2) as calc:
        try:
            eO2 = O2.get_potential_energy()
            DO2.append(eO2)
        except (VaspSubmitted, VaspQueued):
            eO2 = None
    with jasp('molecules/hw03/CO2-{0}'.format(encut),
        xc='PBE',
        encut=encut,
        ismear=1,
        #sigma=0.01,
        ibrion=2,
        nsw=10,
        atoms=CO2) as calc:
        try:
            eCO2 = CO2.get_potential_energy()
            DCO2.append(eCO2)
        except (VaspSubmitted, VaspQueued):
            eCO2 = None

    if None not in (eCO, eO2, eCO2):
        dE = eCO2 - eCO - 0.5*eO2
        Drxn.append(dE)

import matplotlib.pyplot as plt
plt.figure()
plt.subplot(221)
plt.plot(encuts, DCO, '-bo')
plt.xlabel('ENCUT (eV)')
plt.ylabel('CO Energy (eV)')
plt.subplot(222)
plt.plot(encuts, DO2, '-ro')
plt.xlabel('ENCUT (eV)')
plt.ylabel('O$_2$ Energy (eV)')
plt.subplot(223)
plt.plot(encuts, DCO2, '-go')
plt.xlabel('ENCUT (eV)')
plt.ylabel('CO$_2$ Energy (eV)')
plt.subplot(224)
plt.plot(encuts, Drxn, '-ko')
plt.xlabel('ENCUT (eV)')
plt.ylabel('CO Oxidation Energy (eV)')
plt.tight_layout()
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.

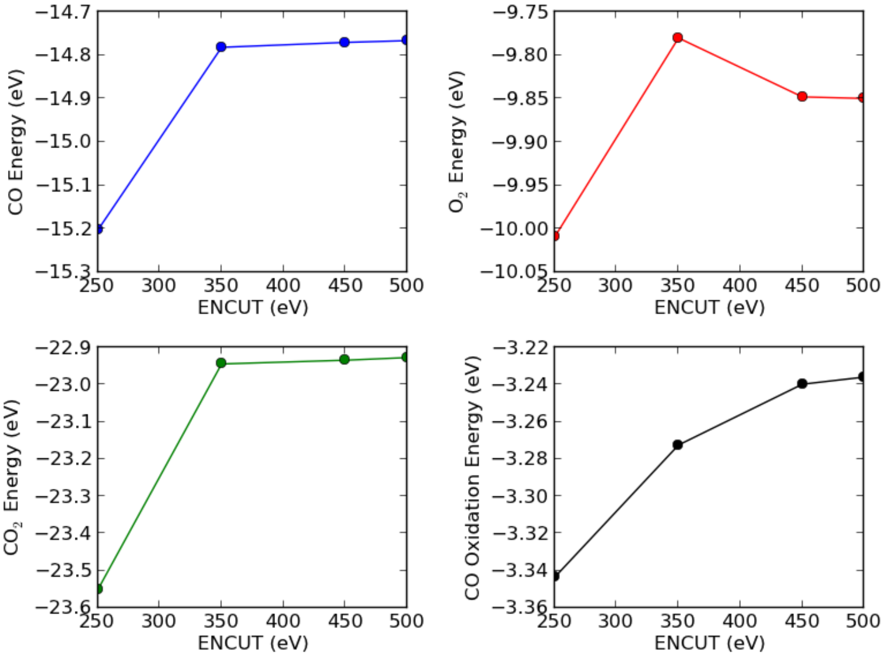

Figure 1: Convergence of energies for Subproblem 1.2.

Problem 2

Subproblem 2.1

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

```
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()

CO2 = molecule('CO2')
CO2.set_cell([8, 8, 8], scale_atoms=False)
CO2.center()

with jasp('molecules/hw03/CO_vib-{0}'.format(350),
    encut=400,
    ismear=0,# Gaussian smearing
    ibrion=6,# finite differences with symmetry
    nfree=2, # central differences (default)
    potim=0.015, # default as well
    ediff=1e-8,
    nsw=1,
    atoms=CO) as calc:
    try:
        CO.get_forces()
        energies, modes = calc.get_vibrational_modes()
        print 'CO Energies\n========'
        for i, e in enumerate(energies):
            print '{0:02d}: {1} eV'.format(i, e)
    except (VaspSubmitted, VaspQueued):
        pass

with jasp('molecules/hw03/O2_vib-{0}'.format(350),
    encut=400,
    ismear=0,# Gaussian smearing
    ibrion=6,# finite differences with symmetry
    nfree=2, # central differences (default)
    potim=0.015, # default as well
    ediff=1e-8,
    nsw=1,
    atoms=O2) as calc:
    try:
        O2.get_forces()
        energies, modes = calc.get_vibrational_modes()
        print 'O2 Energies\n========'
        for i, e in enumerate(energies):
            print '{0:02d}: {1} eV'.format(i, e)
    except (VaspSubmitted, VaspQueued):
        pass

with jasp('molecules/hw03/CO2_vib-{0}'.format(350),
    encut=400,
    ismear=0,# Gaussian smearing
    ibrion=6,# finite differences with symmetry
    nfree=2, # central differences (default)
    potim=0.015, # default as well
    ediff=1e-8,
    nsw=1,
    atoms=CO2) as calc:
    try:
        CO2.get_forces()
        energies, modes = calc.get_vibrational_modes()
        print 'CO2 Energies\n========'
        for i, e in enumerate(energies):
            print '{0:02d}: {1} eV'.format(i, e)
    except (VaspSubmitted, VaspQueued):
        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.

```python
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)])
    except (IOError):
        COfreq = None

with jasp('molecules/hw03/O2_vib-{0}'.format(350)) as calc:
    try:
        eO2 = calc.get_potential_energy(calc.get_atoms())
        O2freq = calc.get_vibrational_frequencies()
        O2ZPE = np.sum([0.5*h*f*c for f in O2freq if isinstance(f, float)])
    except (IOError):
        O2freq = None

with jasp('molecules/hw03/CO2_vib-{0}'.format(350)) as calc:
    try:
        eCO2 = calc.get_potential_energy(calc.get_atoms())
        CO2freq = calc.get_vibrational_frequencies()
        CO2ZPE = np.sum([0.5*h*f*c for f in CO2freq if isinstance(f, float)])
    except (IOError):
        CO2freq = None

```

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

with jasp('molecules/hw03/CO2-{0}'.format(350)) as calc:
    try:
        atoms = calc.get_atoms()
        x, y, z, cd = calc.get_charge_density()
        mlab.figure(bgcolor=(1, 1, 1))

        # plot the atoms as spheres
        for atom in atoms:
            mlab.points3d(atom.x,
                          atom.y,
                          atom.z,
                          scale_factor=vdw_radii[atom.number]/5.,
                          resolution=20,
                          # a tuple is required for the color
                          color=tuple(cpk_colors[atom.number]),
                          scale_mode='none')

        # draw the unit cell - there are 8 corners, and 12 connections
        a1, a2, a3 = atoms.get_cell()
        origin = [0, 0, 0]
        cell_matrix = [[origin,  a1],
                       [origin,  a2],
                       [origin,  a3],
                       [a1,      a1 + a2],
                       [a1,      a1 + a3],
                       [a2,      a2 + a1],
                       [a2,      a2 + a3],
                       [a3,      a1 + a3],
                       [a3,      a2 + a3],
                       [a1 + a2, a1 + a2 + a3],
                       [a2 + a3, a1 + a2 + a3],
                       [a1 + a3, a1 + a3 + a2]]

        for p1, p2 in cell_matrix:
            mlab.plot3d([p1[0], p2[0]], # x-positions
                        [p1[1], p2[1]], # y-positions
                        [p1[2], p2[2]], # z-positions
                        tube_radius=0.02)


        # Now plot the charge density
        mlab.contour3d(x, y, z, cd, transparent=True)

        # this view was empirically found by iteration
        mlab.view(azimuth=-90, elevation=90, distance='auto')

        mlab.savefig('images/co2-centered-cd.png')
        mlab.show()
    except (VaspSubmitted, VaspQueued, IOError):
        pass
```

The electron density of CO_2 is shown in Figure 2.

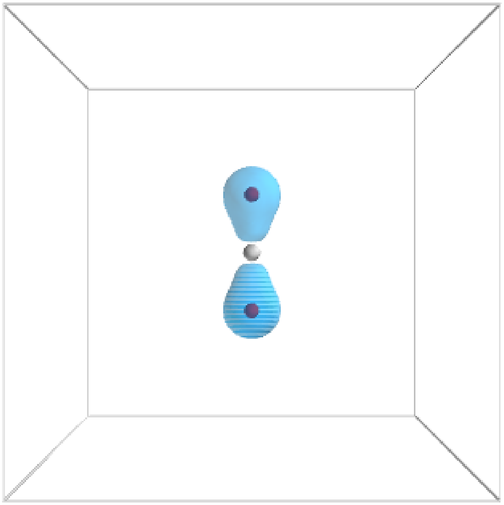

Figure 2: Charge density for CO_2 for Problem 3.