hwk3.org Due 10/12/2012

John Kitchin

2012-09-22 Sat

Contents

1 Reaction energy of CO oxidation

1.1 Compute the reaction energy for CO + 1/2 O_2 → CO_2

Use a cutoff energy of 250 eV. The molecules should all be relaxed to their lowest energy geometry (perform a geometry optimization). Demonstrate that all the forces on the molecule are less than 0.05 eV/Å.

```
from jasp import *
from ase.data.molecules import molecule

```

```
atoms = molecule(’CO’)
atoms.set_cell((6,6,6), scale_atoms=False)
atoms.center()

with jasp(’hwk3/co-250’,
          xc=’PBE’,
          encut=250,
          ibrion=2, nsw=20, ediffg=-0.05,
          atoms=atoms) as calc:
    try:
        co = atoms.get_potential_energy()
    except (VaspSubmitted, VaspQueued):
        co = None

atoms = molecule(’CO2’)
atoms.set_cell((6,6,6), scale_atoms=False)
atoms.center()

with jasp(’hwk3/co2-250’,
          xc=’PBE’,
          encut=250,
          ibrion=2, nsw=20,
          atoms=atoms) as calc:
    try:
        co2 = atoms.get_potential_energy()
    except (VaspSubmitted, VaspQueued):
        co2 = None

atoms = molecule(’O2’)
atoms.set_cell((6,6,6), scale_atoms=False)
atoms.center()
atoms.set_initial_magnetic_moments([1,1])

with jasp(’hwk3/o2-250’,
          xc=’PBE’,
          encut=250,
          ispin=2,
          ibrion=2, nsw=20,
          atoms=atoms) as calc:
    try:
        o2 = atoms.get_potential_energy()
    except (VaspSubmitted, VaspQueued):
        o2 = None

print co, o2, co2

if None not in (co, o2, co2):
    Erxn = co2 - co - 0.5*o2
    print ’The reaction energy is {0:1.3f} eV’.format(Erxn)
```

```
from jasp import *
with jasp(’hwk3/o2-250’) as calc:
    atoms = calc.get_atoms()
    print ’O2’
    print atoms.get_forces()
```

```

with jasp(’hwk3/co2-250’) as calc:
    atoms = calc.get_atoms()
    print ’CO2’
    print atoms.get_forces()


with jasp(’hwk3/co-250’) as calc:
    atoms = calc.get_atoms()
    print ’CO’
    print atoms.get_forces()
```

For a reason I am not clear on, VASP converged the CO calculation with a residual force of 0.44 eV/ang until I specified via EDIFFG to make the force less than 0.05 eV/ang. This could be because the change in total energy between two steps was less than the EDIFFG default setting which defaults to 1 meV. Lesson: you must specify the criteria you want converged! In this case, that means set EDIFFG=-0.05. The negative sign means it is a force criteria, not an energy criteria.

1.2 Convergence test

Repeat the previous problem at 350, 450, and 500 eV. Reoptimize the geometry at each ENCUT value. Compare (in a graph) the convergence of the total energy of each species with the convergence of the reaction energy. Which converges faster?

```
from jasp import *
from ase.data.molecules import molecule

for encut in [350, 450, 500]:
    atoms = molecule(’CO’)
    atoms.set_cell((6,6,6), scale_atoms=False)
    atoms.center()

    with jasp(’hwk3/co-{0}’.format(encut),
              xc=’PBE’,
              encut=encut,
              ibrion=2, nsw=20,
              atoms=atoms) as calc:
        try:
            co = atoms.get_potential_energy()
        except (VaspSubmitted, VaspQueued):
            co = None

    atoms = molecule(’CO2’)
    atoms.set_cell((6,6,6), scale_atoms=False)
    atoms.center()

    with jasp('hwk3/co2-{0}'.format(encut),
              xc='PBE',
              encut=encut,
              ibrion=2, nsw=20,
              atoms=atoms) as calc:
        try:
            co2 = atoms.get_potential_energy()
        except (VaspSubmitted, VaspQueued):
            co2 = None

    atoms = molecule('O2')
    atoms.set_cell((6,6,6), scale_atoms=False)
    atoms.center()
    atoms.set_initial_magnetic_moments([1,1])

    with jasp('hwk3/o2-{0}'.format(encut),
              xc='PBE',
              encut=encut,
              ispin=2,
              ibrion=2, nsw=20,
              atoms=atoms) as calc:
        try:
            o2 = atoms.get_potential_energy()
        except (VaspSubmitted, VaspQueued):
            o2 = None

    if None not in (co, o2, co2):
        Erxn = co2 - co - 0.5*o2

        print 'At {1} eV the reaction energy is {0:1.3f} eV'.format(Erxn, encut)
```

```
from jasp import *
import numpy as np
import matplotlib.pyplot as plt

encuts = [250, 350, 450, 500]

co, co2, o2 = [], [], []

for encut in encuts:
    with jasp('hwk3/co-{0}'.format(encut)) as calc:
        atoms = calc.get_atoms()
        co.append(atoms.get_potential_energy())

    with jasp('hwk3/co2-{0}'.format(encut)) as calc:
        atoms = calc.get_atoms()
        co2.append(atoms.get_potential_energy())

    with jasp('hwk3/o2-{0}'.format(encut)) as calc:
        atoms = calc.get_atoms()
        o2.append(atoms.get_potential_energy())

co = np.array(co)
co2 = np.array(co2)
o2 = np.array(o2)
```

```

plt.plot(encuts, co - co.min())
plt.plot(encuts, co2 - co2.min())
plt.plot(encuts, o2 - o2.min())
plt.xlabel(’ENCUT (eV)’)
plt.ylabel(’$\Delta E$ (eV)’)
plt.legend([’CO’, ’CO2’, ’O2’], loc=’best’)
plt.savefig(’hwk3/molecular-convergence.png’)

plt.figure()
plt.plot(encuts, co2 - co - 0.5*o2)
plt.xlabel(’ENCUT (eV)’)
plt.ylabel(’$H_{rxn}$ (eV)’)
plt.savefig(’hwk3/rxn-encut-convergence.png’)

plt.show()
```

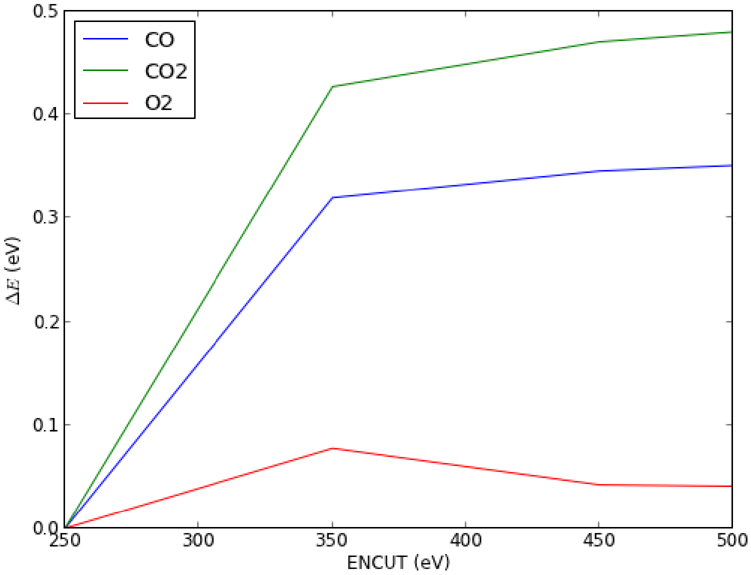

Figure 1: Convergence with ENCUT of molecules. The difference is with respect to the molecule energy at 250 eV.

It is probably safe to say that the reaction energy is within 0.02 eV of 3.11 eV. The energy has clearly not completely converged, even at 500 eV. Note the reaction energy change due to convergence is much smaller than the changes in the CO and CO2 molecular energies.

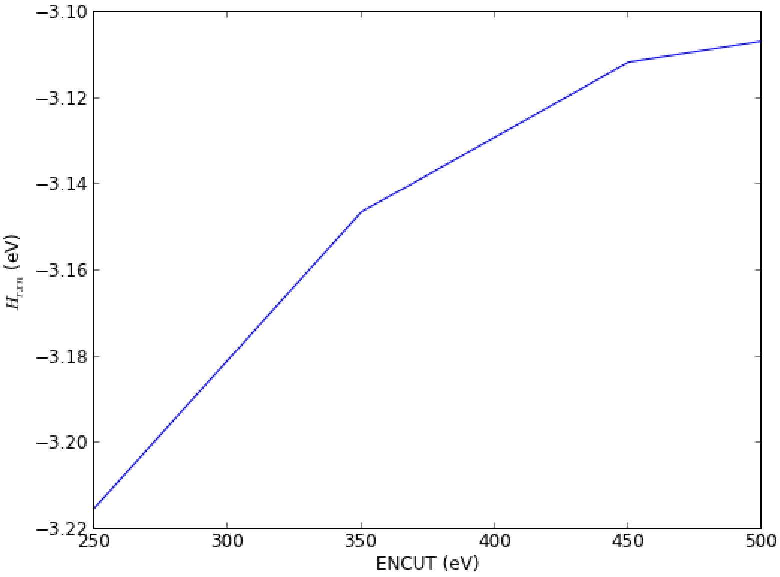

Figure 2: Convergence of the CO oxidation reaction.

2 Zero-point energy corrections

2.1 Compute vibrational modes for CO, CO_2 and O_2

Compute the vibrational modes of each molecule in the CO oxidation reaction. Do this at 350 eV cutoff energy only. Prepare a table of the vibrational modes for molecule.

2.1.1 O2

```
from jasp import *

with jasp('hwk3/o2-350') as calc:
    calc.clone('hwk3/o2-350-vib')
    atoms = calc.get_atoms()

with jasp('hwk3/o2-350-vib',
          ibrion=5, nfree=2, nsw=1,
          atoms=atoms) as calc:
    try:
        calc.calculate()
        energies, modes = calc.get_vibrational_modes()
        print '#+caption: O2 vibrational energies'
        print ’| mode | energy (eV) |’
        print ’|-’
        for i, energy in enumerate(energies):
            print ’| {0} | {1} |’.format(i, energy)
    except (VaspSubmitted, VaspQueued):
        pass
```

2.1.2 CO2

```
from jasp import *

with jasp(’hwk3/co2-350’) as calc:
    calc.clone(’hwk3/co2-350-vib’)
    atoms = calc.get_atoms()

with jasp(’hwk3/co2-350-vib’,
          ibrion=5, nfree=2, nsw=1,
          atoms=atoms) as calc:
    try:
        calc.calculate()
        energies, modes = calc.get_vibrational_modes()
        print ’#+caption: CO2 vibrational energies’
        print ’| mode | energy (eV) |’
        print ’|-’
        for i, energy in enumerate(energies):
            print ’| {0} | {1} |’.format(i, energy)
    except (VaspSubmitted, VaspQueued):
        pass
```

2.1.3 CO

```
from jasp import *
with jasp(’hwk3/co-350’) as calc:
    calc.clone(’hwk3/co-350-vib’)
    atoms = calc.get_atoms()

with jasp(’hwk3/co-350-vib’,
          ibrion=5, nfree=2, nsw=1,
          atoms=atoms) as calc:
    try:
        calc.calculate()
        energies, modes = calc.get_vibrational_modes()
        print ’#+caption: CO vibrational energies’
        print ’| mode | energy (eV) |’
        print ’|-’
        for i, energy in enumerate(energies):
            print ’| {0} | {1} |’.format(i, energy)
    except (VaspSubmitted, VaspQueued):
        pass
```

2.2 Compute the CO oxidation reaction energy with zero-point energy corrections.

Compare the reaction energy with and without the zero-point energy correction.

We compute zero point energy as the sum of 0.5*h*nu_i

```
from jasp import *
import numpy as np
c = 3e10 # speed of light cm/s
h = 4.135667516e-15 # eV/s

with jasp(’hwk3/o2-350-vib’) as calc:
    freq = calc.get_vibrational_frequencies()

    o2_ZPE = 0.0
    for f in freq:
        if not isinstance(f,float):
            continue #skip complex numbers
        nu = f*c # convert to frequency
        o2_ZPE += 0.5*h*nu
    print o2_ZPE


with jasp(’hwk3/co-350-vib’) as calc:
    freq = calc.get_vibrational_frequencies()

    co_ZPE = 0.0
    for f in freq:
        if not isinstance(f,float):
            continue #skip complex numbers
        nu = f*c # convert to frequency
        co_ZPE += 0.5*h*nu
    print co_ZPE


with jasp(’hwk3/co2-350-vib’) as calc:
    freq = calc.get_vibrational_frequencies()

    co2_ZPE = 0.0
    for f in freq:
        if not isinstance(f,float):
            continue #skip complex numbers
        nu = f*c # convert to frequency
        co2_ZPE += 0.5*h*nu
    print co2_ZPE

rxn_ZPE = co2_ZPE - co_ZPE - 0.5*o2_ZPE

print ’The ZPE correction to the reaction is {0:1.3f} eV’.format(rxn_ZPE)

print ’E_rxn_zpe = {0:1.3f} eV’.format(-3.1 + rxn_ZPE)
```

2.3 Compare your computed energy to a value from the literature.

Provide a reference for your literature value.

Table 1: Standard enthalpies of formation (webbook.nist.gov)

	Standard enthalpy (kJ/mol)
CO	-110.53 ± 0.17
CO2	-393.51 ± 0.13
O2	0.0

The standard enthalpy of reaction is -282.98 kJ/mol which is about -2.95 eV.

Our calculated value is **remarkably** close to the experimental value after zero point energy correction! This is certainly due to fortuitous error cancellations. DFT is not usually this good.

3 Plot the electron density of the CO2 molecule.

Include the figure in your homework.

```python
from jasp import *
from enthought.mayavi import mlab
from ase.data import vdw_radii
from ase.data.colors import cpk_colors

with jasp('hwk3/co2-350') as calc:
    atoms = calc.get_atoms()
    x, y, z, cd = calc.get_charge_density()

mlab.figure(1, bgcolor=(1, 1, 1)) # make a white figure
mlab.contour3d(x, y, z, cd)
mlab.view(azimuth=-90, elevation=90, distance='auto')
mlab.savefig('hwk3/co2-cd.png')
```

It looks a lot like the CO charge density looking in the mirror!

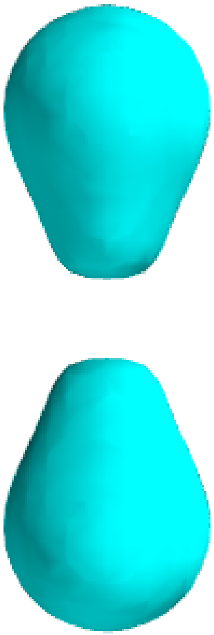

Figure 3: CO2 charge density.