

Code example 1: DFT optimization script

```
# call as
# $ python optimize.py filename.xyz output_prefix
import sys
import ase
from ase.optimize import BFGS
from ase.constraints import FixAtoms
from gpaw import GPAW, PW, FermiDirac
import gpaw.solvation as solv
from ase.calculators.vdwc correction import vdW Tkatchenko09pr1
from gpaw.analyse.hirshfeld import HirshfeldPartitioning
from gpaw.analyse.vdwradii import vdW radii

# Assigns VdW for metals, since not present at the moment
# this is needed for solvation module
# https://pubs.rsc.org/en/content/articlehtml/2013/dt/c3dt50599e
from ase.data.vdw import vdw_radii
vdw_radii.flags.writeable = True
vdw_radii[26] = 2.44 # Fe
vdw_radii[27] = 2.40 # Co
vdw_radii[28] = 2.40 # Ni

# DFT parameters
functional = 'PBE'
grid = 0.18
force_prec = 0.01
kgrid = (2,2,1)

# Input
fname = sys.argv[1]
write_name = sys.argv[2]

# Read in the structure and constrain several atoms
molecule = ase.io.read(fname)
molecule.set_constraint(FixAtoms(indices=(2,6,10,14,18,17,33,32,52,51, 3, 7, 11, 15,
19)))

# Calculator for optimization with solvation
calc = solv.SolvationGPAW(h = grid,
                        spinpol = True,
                        xc = functional,
                        kpts = kgrid,
                        maxiter = maxiter,
                        txt = write_name+'_solv_calc.out',
                        **solv.get_HWL4_water_kwargs())

molecule.set_calculator(calc)
dyn = BFGS(atoms=molecule, trajectory=write_name+'.traj',
restart=write_name+"_res.pckl")
dyn.run(fmax=force_prec)

# Writes the optimized structure to new file
ase.io.write(write_name+"_solv_opt.xyz",molecule)

# Apply TS09 dispersion correction at the end of the optimization for solvation
calc = solv.SolvationGPAW(h = grid,
                        spinpol = True,
                        xc = functional,
                        kpts = kgrid,
                        maxiter = maxiter,
                        txt = write_name+'_solv_calc_TS09.out',
                        **solv.get_HWL4_water_kwargs())

calc_TS = vdW Tkatchenko09pr1(HirshfeldPartitioning(calc),
vdW radii(molecule.get_chemical_symbols(), 'PBE'), calc);

molecule.set_calculator(calc_TS)
molecule.get_potential_energy()
```

Code example 2: Vibrational analysis script

```
# call as
# $ python vibration.py filename.xyz output_prefix Nof_adsorbate_atoms
import sys
import ase
from ase.constraints import FixAtoms
from ase.vibrations import Vibrations
from ase.thermochemistry import HarmonicThermo
import gpaw.solvation as solv

# OVERWRITE ASEs DEFAULT VALUE FOR SELECTED TM VdW
# Data from:
# https://pubs.rsc.org/en/content/articlehtml/2013/dt/c3dt50599e
from ase.data.vdw import vdw_radii
vdw_radii.flags.writeable = True
vdw_radii[26] = 2.44 # Fe
vdw_radii[27] = 2.40 # Co
vdw_radii[28] = 2.40 # Ni

# DFT parameters
functional = 'PBE'
grid = 0.18
force_prec = 0.01
kgrid = (2,2,1)
maxiter = 600

# File input
fname = sys.argv[1]
write_name = sys.argv[2]
No_vib_atoms = int(sys.argv[3])

molecule = ase.io.read(fname)
molecule.set_constraint(FixAtoms(indices=range(0,59)))

# Calculator with solvation
calc = solv.SolvationGPAW(h = grid,
                          spinpol = True,
                          xc = functional,
                          kpts = kgrid,
                          maxiter = maxiter,
                          txt = write_name+'_solv_thermo.out',
                          **solv.get_HWL4_water_kwargs())
molecule.set_calculator(calc)

# Run vibration calculations
vib = Vibrations(molecule, indices = range(59, 59+No_vib_atoms))
vib.run()
print(vib.summary())
for mode in range(3*No_vib_atoms):
    vib.write_mode(mode)
vib_energies = vib.get_energies()
thermo = HarmonicThermo(vib_energies = vib_energies)

# Print vibrational output
G = thermo.get_entropy(temperature=298.15)
G2= thermo.get_internal_energy(temperature=298.15)
```

Code example 3: Optimization and vibrational analysis script for H₂

```
# call as
# $ python small_opt_vib.py filename.xyz output_prefix
import sys
import ase
from ase.optimize import BFGS
from gpaw import GPAW, PW, FermiDirac, MethfesselPaxton
from ase.vibrations import Vibrations
from ase.thermochemistry import IdealGasThermo

molecule = ase.io.read(sys.argv[1])
write_name = sys.argv[2]

# DFT parameters
functional = 'PBE'
grid = 0.18
force_prec = 0.01
kgrid = (2,2,1)
occupations = {'name': 'fermi-dirac',
               'width': 0.1}
calc = GPAW(h = grid,
            xc = functional,
            kpts = kgrid,
            occupations = occupations,
            txt = '_term.out', symmetry='off')
molecule.set_calculator(calc)

# Optimize structure
dyn = BFGS(atoms=molecule, trajectory=write_name+'_nosolv.traj',
          restart=write_name+'_nosolv_res.pckl')
dyn.run(fmax = force_prec)
ase.io.write(write_name+'_nosolv_opt.xyz', molecule)

# Vibrational analysis
potentialenergy = molecule.get_potential_energy()
vib = Vibrations(molecule)
vib.run()
print(vib.summary())
vib_energies = vib.get_energies()

thermo = IdealGasThermo(vib_energies = vib_energies,
                       potentialenergy = potentialenergy,
                       atoms=molecule,
                       geometry='linear', # 'nonlinear' for H2O
                       symmetrynumber=2, # 2 for H2O
                       spin=0)
G = thermo.get_gibbs_energy(temperature=298.15,
                           pressure=101325.) # pressure=3500 for H2O
for mode in range(6):
    vib.write_mode(mode)
```