The universal interface for testing atomistic potentials

Bohumir Jelinek

Center for Advanced Vehicular Systems

Mississippi State University

Testing of atomistic potentials

Task at hand

- validate *Modified Embedded Atom Method* (MEAM) interatomic potentials for Al, Si, Mg, Cu, Fe, and their alloys

Method

1. calculate basic structural properties of single crystals, formation energies of defects, and structural and elastic properties of simple compounds using MEAM
2. compare with other interatomic potentials and *ab-initio* methods

Issues

Need to learn formats of input parameter files, atomic configuration files, and output files of

- classical MD code implementing MEAM (LAMMPS)
- *ab-initio* code (VASP)

Need to create atomic configurations for

- single crystal structures, basic crystallic compounds, point defects (vacancies, interstitials, substitutions), planar defects (varying surfaces, stacking faults), and strained structures

What would help

A tool applicable to quickly evaluate basic properties from classical interatomic potentials and *ab-initio* methods.

Ideally, a single universal tool would be able to

- create basic atomic configurations and manipulate them
- serve these atomistic configurations as inputs to a variety of methods/simulation codes and obtain energies

Anything like that available?

Atomistic Simulation Environment (ASE)[1]

- universal Python interface to many DFT codes (calculators), with visualization, simple GUI, documentation, and tutorials
- creates molecules, crystal structures, surfaces, nanotubes, analyzes symmetry and spacegroups
- provides support for Equation of state, structure optimization, dissociation, diffusion, constrains, NEB, vibration analysis, phonon calculations, infrared intensities, MD in NVE, NVT, and NPT ensembles, STM, and electron transport
- recent support for LAMMPS by Jörg Meyer (TU München)

[1] S. R. Bahn and K. W. Jacobsen, An object-oriented scripting interface to a legacy electronic structure code, Comput. Sci. Eng., Vol. 4, 56-66, 2002, https://wiki.fysik.dtu.dk/ase/

Calculators[1] working with ASE

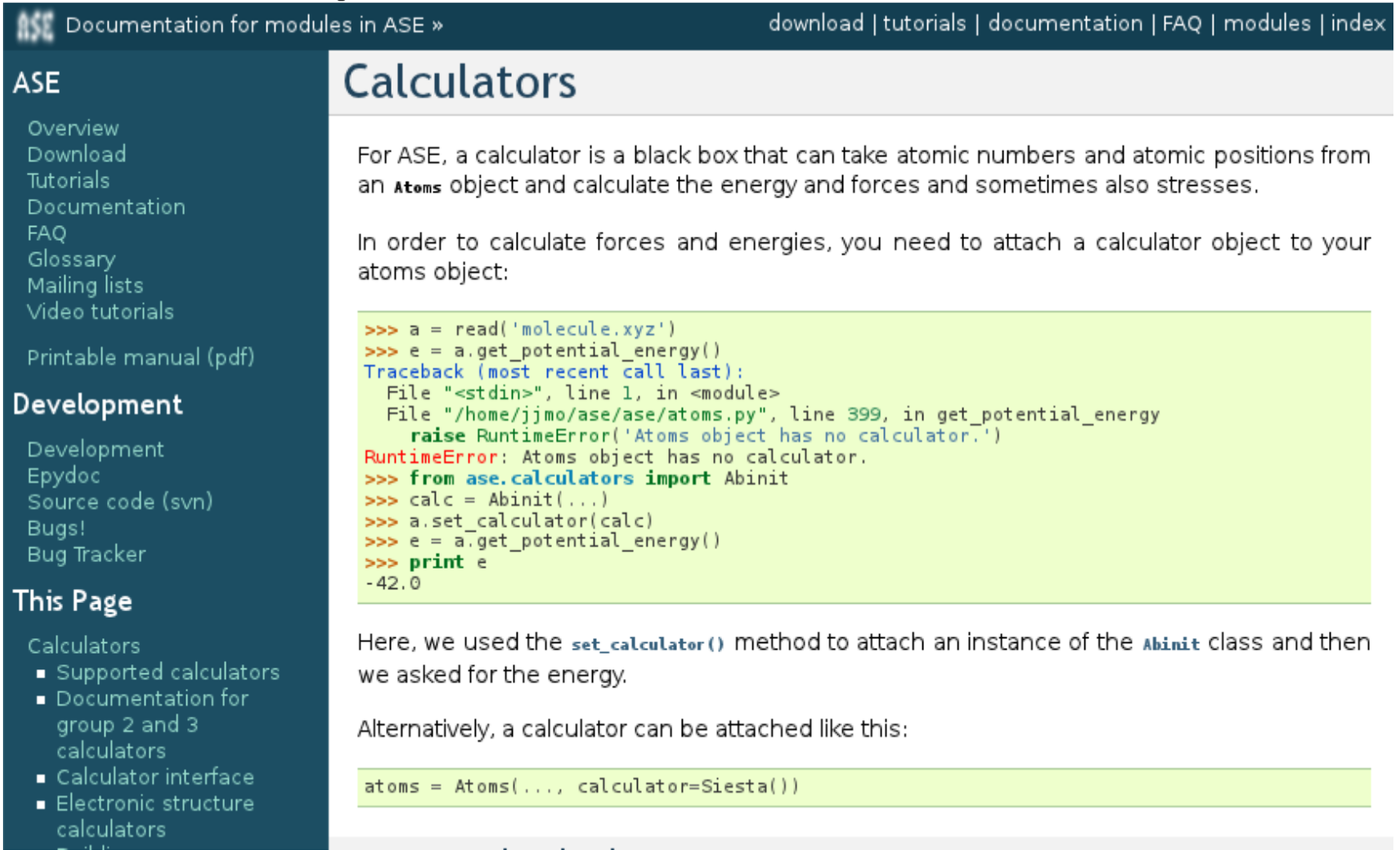
Code	Description	Type
vasp	Planewave PAW code	DFT
abinit	A planewave pseudopotential code	DFT
siesta	LCAO pseudopotential code	DFT
exciting	Full Potential LAPW code	DFT, LAPW
jacapo	ASE interface to Dacapo, planewave ultra-soft pseudopotentials	DFT
dftb	DftbPlus DFT based tight binding	DFT
turbomole	Fast atom orbital code Turbomole	DFT, HF
FHI-aims	Numeric Atomic Orbital, full potential code	DFT, HF
fleur	Full Potential LAPW code	DFT, LAPW
emt	Effective Medium Theory calculator	EMT
Asap	Highly efficient EMT code (written in C++)	EMT
GPAW	Grid-based real-space PAW code	DFT, HF
Dacapo	Old interface to Dacapo. Requires Numeric python and ASE2	DFT
lammps	Classical molecular dynamics code	CMD

[1] https://wiki.fysik.dtu.dk/ase/ase/calculators/calculators.html

Transformation of general simulation cell into LAMMPS specific coordinate system

- important issue in making LAMMPS work with ASE
- LAMMPS supports non-orthogonal (triclinic) simulation boxes, but triclinic box vectors cannot be arbitrarily oriented
- transformation is needed from ASE general coordinate system to LAMMPS specific coordinate system and back
- implemented in LAMMPS calculator by Jörg Meyer

Examples from ASE documentation



ASE Documentation for modules in ASE » download | tutorials | documentation | FAQ | modules | index

ASE
- Overview
- Download
- Tutorials
- Documentation
- FAQ
- Glossary
- Mailing lists
- Video tutorials

Printable manual (pdf)

Development
- Development
- Epydoc
- Source code (svn)
- Bugs!
- Bug Tracker

This Page
- Calculators
  - Supported calculators
  - Documentation for group 2 and 3 calculators
  - Calculator interface
  - Electronic structure calculators

Calculators

For ASE, a calculator is a black box that can take atomic numbers and atomic positions from an `Atoms` object and calculate the energy and forces and sometimes also stresses.

In order to calculate forces and energies, you need to attach a calculator object to your atoms object:

```
>>> a = read('molecule.xyz')
>>> e = a.get_potential_energy()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/home/jjmo/ase/ase/atoms.py", line 399, in get_potential_energy
    raise RuntimeError('Atoms object has no calculator.')
RuntimeError: Atoms object has no calculator.
>>> from ase.calculators import Abinit
>>> calc = Abinit(...)
>>> a.set_calculator(calc)
>>> e = a.get_potential_energy()
>>> print e
-42.0
```

Here, we used the `set_calculator()` method to attach an instance of the `Abinit` class and then we asked for the energy.

Alternatively, a calculator can be attached like this:

```
atoms = Atoms(..., calculator=Siesta())
```

Development
Epydoc
Source code (svn)
Bugs!
Bug Tracker

This Page

Introduction: Nitrogen on copper
- Atoms
- Calculators
- Structure relaxation
- Input-output
- Visualization
- Molecular dynamics

Previous topic

What is Python?

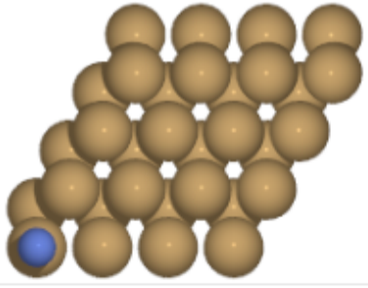
Next topic

Manipulating atoms

Show Source

Quick search

Go



Please have a look at the following script doc/tutorials/N2Cu.py:

```
from ase import Atoms
from ase.visualize import view
from ase.calculators.emt import EMT
from ase.constraints import FixAtoms
from ase.optimize import QuasiNewton
from ase.lattice.surface import fcc111,add_adsorbate

h = 1.85
d = 1.10

slab = fcc111('Cu', size=(4,4,2), vacuum=10.0)

slab.set_calculator(EMT())
e_slab = slab.get_potential_energy()

molecule = Atoms('2N', positions=[(0., 0., 0.), (0., 0., d)])
molecule.set_calculator(EMT())
e_N2 = molecule.get_potential_energy()

add_adsorbate(slab, molecule, h, 'ontop')
constraint = FixAtoms(mask=[a.symbol != 'N' for a in slab])
slab.set_constraint(constraint)
dyn = QuasiNewton(slab, trajectory='N2Cu.traj')
dyn.run(fmax=0.05)

print 'Adsorption energy:', e_slab + e_N2 - slab.get_potential_energy()

#view(slab)
```

ASE requirements

For basic ASE functionality:

- Python
- NumPy

Highly recommended (to create png and eps files, and for ASE GUI)

- matplotlib
- PyGTK

Conclusions

- ASE provides a universal interface to many electronic-structure codes and LAMMPS
- ASE interface for LAMMPS and VASP was utilized in testing Al-Si-Mg-Cu-Fe MEAM alloy potentials
- Following the LAMMPS example, ASE can provide support to other classical MD codes
- ASE simplifies and increases efficiency of atomistic simulation research

Copenhagen, Denmark

Christiansborg Palace, amusement park Tivoli, City Hall Square, ferry