ASE: A programmable environment for calculations with many electronic structure codes

Ask Hjorth Larsen

Center for Atomic-scale Materials Design, Technical University of Denmark

16th ETSF workshop
September 28 2011

Atomic Simulation Environment

ASE is a set of tools for atomic-scale simulation tasks.

What can ASE do?

- Set up and view structures
- Run calculations such as structure optimizations and molecular dynamics
- Read and write structures in many file formats
- ASE is open source / free software (LGPLv2.1)

What is so special about ASE?

- ASE unifies many electronic structure codes within a single environment
- Calculations are written as Python scipts

Motivation

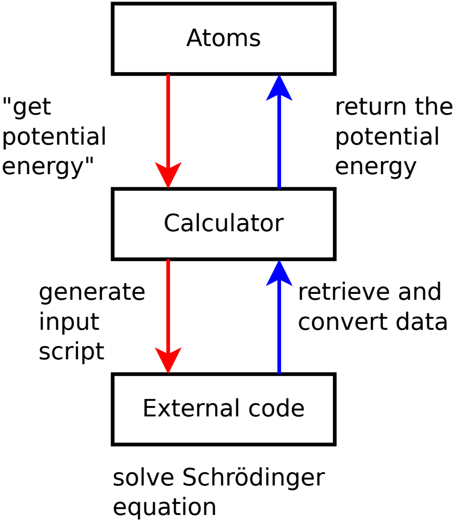
- Electronic structure codes usually take different input parameters and have incompatible file formats
- Each code requires learning a new input syntax
- And yet, much functionality is common to all codes: atomic geometries, structure optimization algorithms, ...

- ASE identifies common functionality and separates it from the specific parameters of different codes
- Electronic structure codes are abstracted as calculators. Calculators implement a specific programming interface

A bit of history

- ASE was developed as an interface to the FORTRAN-based ultrasoft pseudopotential code Dacapo
- S. R. Bahn and K. W. Jacobsen, *An object-oriented scripting interface to a legacy electronic structure code*, Comput. Sci. Eng. 4, 56 (2002), ISSN 1521-9615.
- ASE is a part of the CAMPOS (CAMP Open Source) project `https://wiki.fysik.dtu.dk/`
- Other CAMPOS codes have been written with ASE interfaces: ASAP and GPAW
- ASE development is driven in part by our own requirements, so ASE/GPAW is a strongly correlated system

Atoms and calculators


Atoms
"get potential energy"
return the potential energy
Calculator
generate input script
retrieve and convert data
External code
solve Schrödinger equation

- Atoms objects represent geometry: positions, atomic numbers, unit cell, momenta, . . .
- Calculators can be based on anything that calculates energies, forces (DFT, classical potentials, ...)

Example: The ASE/GPAW “hello world”

ASE calculators so far

abinit
Asap
CASTEP
DFTB+
EMT
exciting
aims
www.flapw.de
Fleur
GPAW!
Hb
Hotbit
JACAPO
LAMMPS
siesta
TURBOMOLE
VASP

Supported calculators

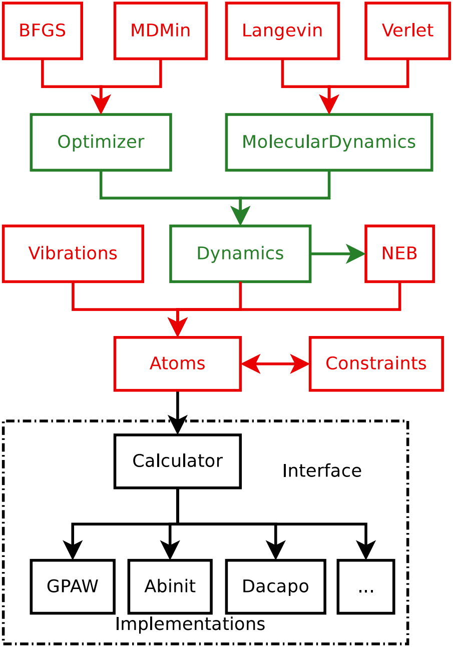
GPAW	Grid-based real-space PAW code	DFT, HF
Asap	Highly efficient EMT code (C++)	EMT
Dacapo	planewave USPP code	DFT
emt	Effective Medium Theory calculator	EMT
abinit	A planewave pseudopotential code	DFT
siesta	LCAO pseudopotential code	DFT
dftb	DftbPlus DFT based tight binding	DFT
turbomole	Fast atomic orbital code	DFT, HF
castep	Planewave pseodopotential code	DFT, HF
vasp	Planewave PAW code	DFT
FHI-aims	NAO full-potential code	DFT, HF
exciting	Full Potential LAPW code	DFT, LAPW
fleur	Full Potential LAPW code	DFT, LAPW
lammps	Classical molecular dynamics code	

Scripting with Python in ASE

- Access to a complete language provides very high flexibility
- Python is interpreted, so no compilation required
- Well suited for scripting due to high-level features and simple syntax
- Cross-platform (although some supercomputers can be finicky)

Scientific programming in Python

- numpy: module for array calculations
- scipy: module for high-level mathematics with numpy (optimization, integration, root finding)
- matplotlib: module for plotting



Overview of ASE classes

- Only the most common objects shown
- Red: user interface
  Green: classes for internal use

Demonstration: structure optimization

Structure generation

Molecules

- ASE includes G1, G2 and G3 datasets

```
from ase.data.molecules import molecule
system = molecule(’CH3CH2OH’)
```

Crystal structures

- Bulk
- Surfaces
- Clusters

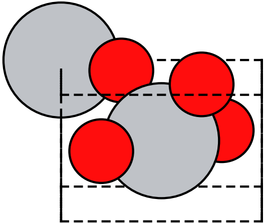
IO

- ASE supports a couple dozen file formats

Demonstration: Gold 111 surface

Bulk rutile

```
from ase.lattice.spacegroup import crystal
a = 4.6
c = 2.95
rutile = crystal(['Ti', 'O'],
                 basis=[(0, 0, 0),
                        (0.3, 0.3, 0.0)],
                 spacegroup=136,
                 cellpar=[a, a, c, 90, 90, 90])
```



Demonstration: Pt cluster with GUI

Future plans or work in progress

Electronic structure interface

- Get density of states
- Export wavefunctions, densities (basis set conversions)
- More uniform input parameters

Automation

- Bulk submit script generation?
- “Run this calculator on this set of structures”

And more

- Interface to deal with electronic excitations
- Debian packages for Debian Open Science
- “Scratch your own itch”

List of committers

Andrew Peterson
David Landis
Lars Grabow
Jakob Blomquist
Jens J Mortensen
John Kitchin
Jesper Kleis
Michael Walter
Christian Glinsvad
Mikkel Strange
Anthony Goodrow
Troels K Jacobsen

Ask Hjorth Larsen
Marcin Dulak
Heine Anton Hansen
Jon B. Maronsson
Jesper Friis
Jussi Enkovaara
Karsten W Jacobsen
Poul Georg Moses
Jakob Schiotz
Tao Jiang
Carsten Rostgaard

Jonas Bjork
George Tritsaris
Felix Hanke
Janne Blomqvist
Jingzhe Chen
Kristen Kaasbjerg
Markus Kaukonen
Marco Vanin
Mattias Slabanja
Thomas Olsen
Janosch M Rauba

Conclusion

- Pluggable calculators
- Python scripting
- Object oriented interface
- Structure generation and I/O
- Open source / free software

Links and info

- CAMPOS: `https://wiki.fysik.dtu.dk/`
- ASE: `https://wiki.fysik.dtu.dk/ase/`
- Mailing list: `ase-users@listserv.fysik.dtu.dk`
- IRC: `#gpaw` on `irc.freenode.net` (some of the same people)
- Python: http://www.python.org/