Python interface for MyMD

Rodrigo, Leopold

Design Problems

- MyMD code to be completely reorganized
- No fixed API
- No compiling code to test against
- => Decision: ‘API’ = input file format

Getting started

- 4 * 1/2 day = not a lot
- usable interface = lot of work
- => What's already there?

Atomic Simulation Environment ¶

The Atomic Simulation Environment (ASE) is the common part of the simulation tools developed at CAMd. ASE provides Python modules for manipulating atoms, analyzing simulations, visualization etc.

> **Note:** The old ASE-2 webpage has moved to http://wiki.fysik.dtu.dk/ase2.

Supported `calculators`:

Calculator Class

```
"""This package defines an ASE interface to MyMD

MyMD is an educational project, written during the 2013
Workshop on Computer Programming and Advanced Tools for Scientific Research Work
at ICTP Trieste, Italy.
"""

import os
import sys
import subprocess

import numpy as np

from ase.atoms import Atoms
from ase.calculators.calculator import FileIOCalculator, Parameters, ReadError

import potentials
import io

class FileIOMyMD(FileIOCalculator):
```

Example

```
""" Example 03

This example shows how to use the built-in atom viewer
of ASE as well as data visualization with matplotlib.
"""
from ase.lattice.cubic import SimpleCubic
from ase.md.velocitydistribution import MaxwellBoltzmannDistribution
import ase.units as units
import ase.calculators.mymd as mymd

# Set up a cube of 125 Argon atoms in a cube of (15 Angstroms)**3.
myatoms = SimpleCubic('Ar', latticeconstant=3.0, size=(5,5,5))
scalef = 8.31446714569e-7       # adjust velocity units
MaxwellBoltzmannDistribution(myatoms, temp=50*scalef, force_temp=True)
```

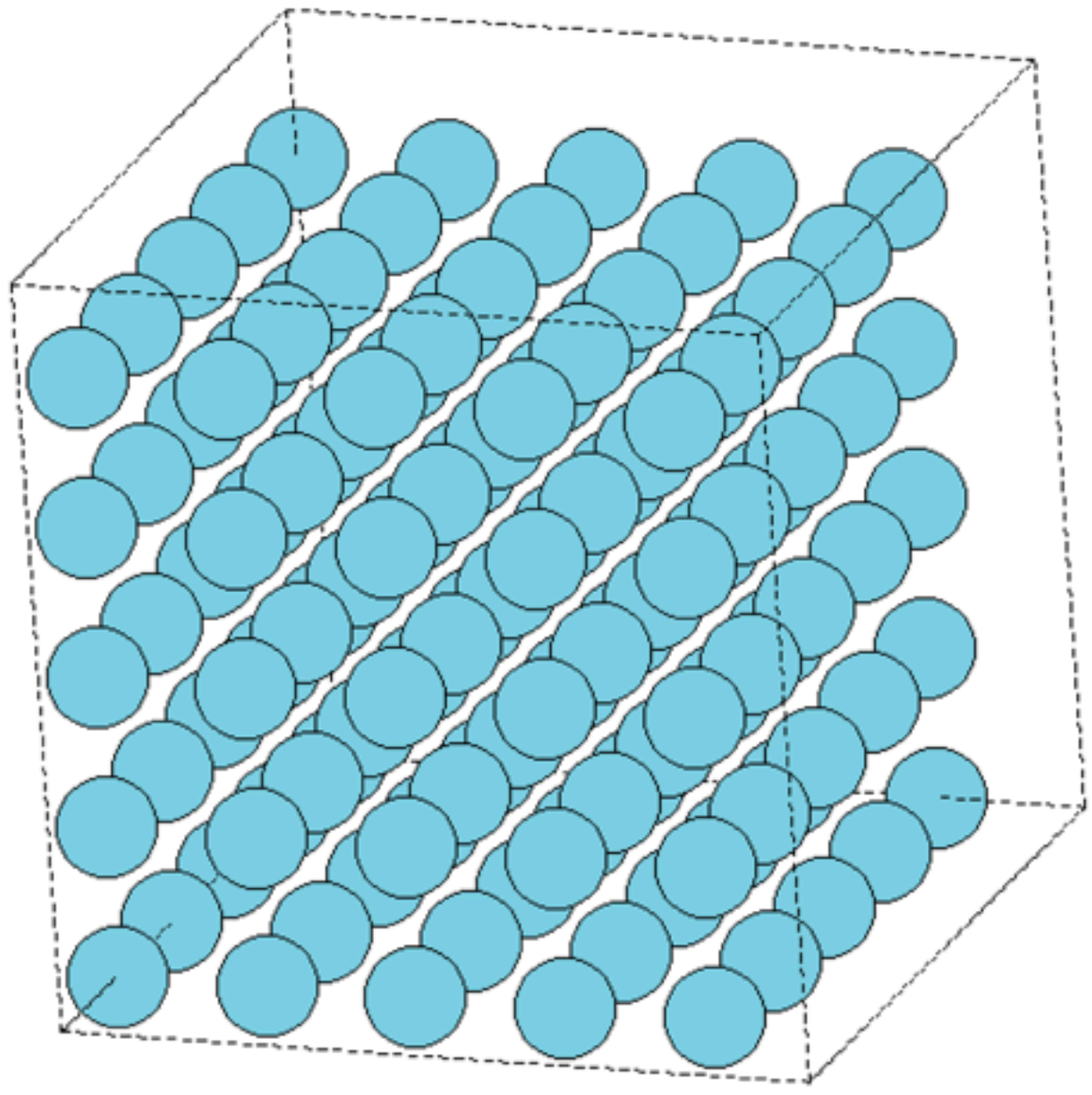
Example

```
from ase.visualize import view
view(myatoms)


# Set up MyMD calculator and run
calc  = mymd.FileIOMyMD(label='mymd',
                        nsteps=10000,
                        dt=2.0,
                        nprint=100)
calc.run_md(myatoms)

# View final state of the atoms
view(calc.state)

# View trajectory of the atoms
trajectory = calc.frames.collect('atoms
view(trajectory)
```



Example

```
# Visualize data using matplotlib
import matplotlib.pyplot as plt

n    = calc.frames.collect('index')
ekin = calc.frames.collect('ekin')
epot = calc.frames.collect('epot')
etot = calc.frames.collect('etot')

plt.plot(n,epot, label='Potential Energy
plt.plot(n,ekin, label='Kinetic Energy [k
plt.plot(n,etot, label='Total Energy [kca
plt.legend()

plt.show()
```