Welcome to
CHE 384T: Computational Methods in Materials Science

Hands-on DFT with Abinit

Programming Day 5

The University of Texas at Austin
McKetta Department of Chemical Engineering
Cockrell School of Engineering

Programming Day Agenda

Hands-on session on DFT with Abinit:

- Hydrogen and oxygen molecule
- Convergence
- Dissociation of a water molecule

Electronic structure/DFT codes:

Planewave, pseudopotential codes:

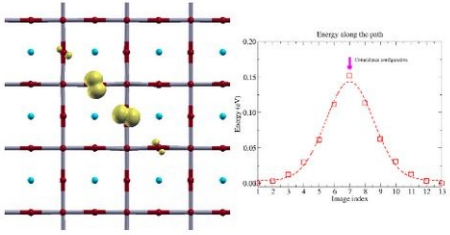
- VASP
- Quantum ESPRESSO
- CASTEP
- Abinit

Localized basis sets:

- Gaussian
- SIESTA
- GAMESS
- CP2K (mixed)

Abinit

Pseudos & PAW data | Input variables | Tutorials | Forum | Download | Compile | GitHub | GUI | Abipy



New : Abinit 10.2.3 available (production)

What is ABINIT ?

ABINIT is a software suite to calculate the optical, mechanical, vibrational, and other observable properties of materials. Starting from the quantum equations of **density functional theory**, you can build up to advanced applications with **perturbation theories** based on DFT, and **many-body Green's functions** (GW and DMFT). +

What is Abinit?

- Presentation
- What you can do with ABINIT
- Suggested acknowledgments
- Sponsors - Partners - Links
- Int'l Advisory Committee
- Legal information
- Past events - many with slides

Get Abinit

- Abinit packages
- Browse latest released sources
- Atomic data/Pseudopotentials
- Installation notes
- Abinit fallbacks
- Run Abinit in a Virtual Machine
- Other resources

Learn Abinit

- Guide for new users
- Tutorials
- Input variables
- Features
- Miscellaneous
- Benchmarks
- For developers

Get in Touch

- Contact information
- Mailing list
- Forum information
- Support the Abinit group
- Wiki (old)
- Developer's corner

Sponsors : cea | cecam

Partners : NOMAD

Socials :

Setting up Abinit

Download:
https://forge.abinit.org/abinit-10.2.3.tar.gz
(be sure to download with configure file)

Installation:
https://docs.abinit.org/installation/

Pre-compiled versions

MacOS
```
brew tap abinit/tap
brew install abinit
```

Ubuntu
```
sudo apt install abinit
```

Setting up Abinit

Set up your paths (where to find things)

```
#!/usr/bin/env sh

export ABI_HOME="replace_with_abinit_top_level_path"
echo "ABI_HOME set to:\n\t" $ABI_HOME

# Do not change these lines
export PATH=$ABI_HOME/src/98_main/:$PATH      # path to executable
export ABI_TESTS=$ABI_HOME/tests/             # path to tests dir
export ABI_PSPDIR=$ABI_TESTS/Psps_for_tests/ # path to pseudos dir

echo "PATH set to:\n\t" $PATH
echo "ABI_TESTS set to:\n\t" $ABI_TESTS
echo "ABI_PSPDIR set to:\n\t" $ABI_PSPDIR
```

Setting up Abinit

Other helpful items

- text editor
  - emacs (https://www.gnu.org/software/emacs/refcards/pdf/refcard.pdf)
  - vim (https://vim.rtorr.com/)
  - Nano (https://www.nano-editor.org/dist/latest/cheatsheet.html)

- Abinit documentation of input parameters
  - https://docs.abinit.org/variables/

- Visualize your structure
  - VESTA (https://jp-minerals.org/vesta/en/download.html)
  - XCrySDen (http://www.xcrysden.org/)

Building Abinit from src

See INSTALL* files

```
./configure && make && make install
```

```
→ abinit-10.2.3 ./configure && make && make install | tee log_install

 ================================================================================
 === Basic initialization                                                     ===
 ================================================================================

checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking target system type... x86_64-unknown-linux-gnu
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether UID '1000' is supported by ustar format... yes
checking whether GID '1000' is supported by ustar format... yes
checking how to create a ustar tar archive... gnutar
checking whether make sets $(MAKE)... (cached) yes
checking whether ln -s works... yes
checking for a sed that does not truncate output... /usr/bin/sed
checking for gawk... (cached) gawk
checking for grep that handles long lines and -e... /usr/bin/grep
checking for sh... /usr/bin/sh
checking for mv... /usr/bin/mv
checking for perl... /usr/bin/perl
checking for rm... /usr/bin/rm
checking for patch... patch
checking for tar... tar
checking for egrep... /usr/bin/grep -E
```

Takes several minutes

```
make[5]: Entering directory '/home/wwwennie/wwwennie@gmail.com/UT-Austin/teaching/2024-Fall-CHE389T/Programming-Days/abin
t-tutorial/abinit-10.2.3/shared/common/src/mods'
There is no buildable file here
make[5]: Leaving directory '/home/wwwennie/wwwennie@gmail.com/UT-Austin/teaching/2024-Fall-CHE389T/Programming-Days/abini
-tutorial/abinit-10.2.3/shared/common/src/mods'
Making all in 02_clib
make[5]: Entering directory '/home/wwwennie/wwwennie@gmail.com/UT-Austin/teaching/2024-Fall-CHE389T/Programming-Days/abin
t-tutorial/abinit-10.2.3/shared/common/src/02_clib'
mpicc -DHAVE_CONFIG_H -I. -I../../../..  -I../../../../shared/common/src/incs -I../../../../shared/common/src/incs -I../.
/../../shared/common/src/mods -I../../../../shared/common/src/mods -I../../../../src/incs -I../../../../src/incs -I../../
./../src/mods -I../../../../src/mods -I../../../../shared/libpaw/incs -I../../../../shared/libpaw/incs -I../../../../shar
d/libpaw/mods -I../../../../shared/libpaw/mods   -I/home/wwwennie/anaconda3/include -I/home/wwwennie/anaconda3/include
-g -O2 -march=native       -MT calling_levmar.o -MD -MP -MF .deps/calling_levmar.Tpo -c -o calling_levmar.o calling_levma
.c
mv -f .deps/calling_levmar.Tpo .deps/calling_levmar.Po
mpicc -DHAVE_CONFIG_H -I. -I../../../..  -I../../../../shared/common/src/incs -I../../../../shared/common/src/incs -I../.
/../../shared/common/src/mods -I../../../../shared/common/src/mods -I../../../../src/incs -I../../../../src/incs -I../../
./../src/mods -I../../../../src/mods -I../../../../shared/libpaw/incs -I../../../../shared/libpaw/incs -I../../../../shar
d/libpaw/mods -I../../../../shared/libpaw/mods   -I/home/wwwennie/anaconda3/include -I/home/wwwennie/anaconda3/include
-g -O2 -march=native       -MT cclock.o -MD -MP -MF .deps/cclock.Tpo -c -o cclock.o cclock.c
mv -f .deps/cclock.Tpo .deps/cclock.Po
mpicc -DHAVE_CONFIG_H -I. -I../../../..  -I../../../../shared/common/src/incs -I../../../../shared/common/src/incs -I../.
/../../shared/common/src/mods -I../../../../shared/common/src/mods -I../../../../src/incs -I../../../../src/incs -I../../
./../src/mods -I../../../../src/mods -I../../../../shared/libpaw/incs -I../../../../shared/libpaw/incs -I../../../../shar
d/libpaw/mods -I../../../../shared/libpaw/mods   -I/home/wwwennie/anaconda3/include -I/home/wwwennie/anaconda3/include
-g -O2 -march=native       -MT etime.o -MD -MP -MF .deps/etime.Tpo -c -o etime.o etime.c
mv -f .deps/etime.Tpo .deps/etime.Po
mpicc -DHAVE_CONFIG_H -I. -I../../../..  -I../../../../shared/common/src/incs -I../../../../shared/common/src/incs -I../.
/../../shared/common/src/mods -I../../../../shared/common/src/mods -I../../../../src/incs -I../../../../src/incs -I../../
./../src/mods -I../../../../src/mods -I../../../../shared/libpaw/incs -I../../../../shared/libpaw/incs -I../../../../shar
d/libpaw/mods -I../../../../shared/libpaw/mods   -I/home/wwwennie/anaconda3/include -I/home/wwwennie/anaconda3/include
```

Getting familiar with Abinit

Run through

Tutorial 1: H_2 molecule without convergence studies
(https://docs.abinit.org/tutorial/base1/; ~2 hrs)

Tutorial 2: H_2 molecule with convergence studies
(https://docs.abinit.org/tutorial/base2/; ~1 hr)

Tutorial files are in the source code (typical for many codes):

```
export $ABI_HOME=replace_with_toplevel_abinit_path
export $ABI_TESTS=$ABI_HOME/tests
cd $ABI_TESTS/tutorial/
```

Tip: tutorials often have under-converged results for running expediency
Particularly for new modules/capabilities, codes have a “tutorialized” a published paper

Exercises with Abinit

Exercise 1 (atoms/molecules in a big box)

Use ABINIT to calculate and compare the equilibrium bond length for molecules, total energy, eigenvalues in the H atom, H_2 molecule, O atom and O_2 molecule.

You will need to do convergence tests for each calculation with respect to ecut and the size of the box, to avoid interaction between periodic images.

O_2 molecule requires a spin polarized calculation. Set flag : nsppol = 2
(For more information : https://docs.abinit.org/tutorial/spin/)

Compute the atomization (i.e., dissociation) energy to split H_2 and O_2 into H and O atoms, respectively from their total energy values. How do your computed values compare with measured values?

Exercises with Abinit

Exercise 2 (Water molecule, energy to split water)
Similar to Exercise 1, calculate the total energy, equilibrium bond-length and bond-angle of water molecule by using the manual method of finding the energy minimum with respect to two variables: bond-length and bond-angle.

As this exercise involves many calculations with manual placement of the H atoms, an accuracy of +/- 2% for the bond-angle is OK. Choose your angle step accordingly.

Compare the eigenvalues for all the atoms and molecules calculated and comment on the eigenvalues in an atom versus in a molecule. Sketch an energy diagram.

Compute the dissociation energy per water molecule into H_2 and O_2 molecule from the total energies calculated.