PWDFT.jl: Density Functional Theory Calculations with Julia

Fadjar Fathurrahman

Engineering Physics Department
Research Center for Nanoscience and Nanotechnology
Institut Teknologi Bandung

24 September 2019

Software packages for DFT calculations

There are a lot of software packages for DFT calculations, for examples:

- Quantum ESPRESSO
- VASP
- ABINIT
- Gaussian series: G03, G09, G16
- NWchem

More extensive list: `https://en.wikipedia.org/wiki/List_of_quantum_chemistry_and_solid-state_physics_software`

Problems

- These packages are very helpful for doing various calculations based on DFT. These packages are suitable for black-box-type calculations where we are only concerned about the results.
- However they are generally rather difficult to extend.
  - New development on the DFT functionals: users generally need to wait for the next release of the package to use them (if these functionals are to be implemented at all).
  - Custom calculations or post-processing steps: Users need to know in some detail about how the data they are interested in is represented or implemented in the package's source code.

Write another DFT package?

Personal motivation:

learning how DFT or Kohn-Sham equations are solved

frustation when trying to extend available package

Programming languages for DFT

Programming languages used:

- Fortran and/or C/C++: ABINIT, VASP, Quantum Espresso, ...
- Python: GPAW
- MATLAB: KSSOLV, RESCU

Static languages: Fortran, C/C++

Dynamic languages: Python and MATLAB

Julia programming language

A rather new programming language (2012)

Syntax is familar to MATLAB or Python users

support for multidimensional array and linear algebra

Loop is fast!

Introducing PWDFT.jl

- A program package for doing DFT calculations using Julia
- plane wave basis + pseudopotentials (norm-conserving)
- LDA and GGA functionals can be used (via LibXC)
- symmetry detection (via SPGLIB)
- algorithms for solving KS problem: SCF and Emin (PCG and DCM)

Aims

- Friendly-to-developers DFT package: enables quick implementation of various algorithms
- educational purpose: simple yet powerful enough to carry out practical DFT calculations for molecular and crystalline systems.

Example: hydrogen molecule in a box

```
using PWDFT
atoms = Atoms( xyz_file="H2.xyz",
               LatVecs=gen_lattice_sc(16.0) )
pspfiles = ["../pseudopotentials/pade_gth/H-q1.gth"]
ecutwfc = 15.0
Ham = Hamiltonian( atoms, pspfiles, ecutwfc )
KS_solve_SCF!( Ham )
```

Converged Kohn-Sham energy components

PWDFT.jl's result:

```
------------------------
Final Kohn-Sham energies:
------------------------

Kinetic    energy:       1.0100082964
Ps_loc     energy:      -2.7127874818
Ps_nloc    energy:       0.0000000000
Hartree    energy:       0.9015209348
XC         energy:      -0.6314251867
PspCore    energy:      -0.0000012675
-----------------------------------
Electronic energy:      -1.4326847047
NN         energy:       0.3131700043
-----------------------------------
Total      energy:      -1.1195147004
```

ABINIT's result:

```
Components of total free energy (in Hartree) :

   Kinetic energy  =  1.01004059294567E+00
   Hartree energy  =  9.01545039301481E-01
   XC energy       = -6.31436384237843E-01
   Ewald energy    =  3.13170052325859E-01
   PspCore energy  = -1.26742500464741E-06
   Loc. psp. energy= -2.71283243086241E+00
   NL   psp  energy=  0.00000000000000E+00
   >>>>>>>>> Etotal= -1.11951439795224E+00
```

SCF options

Various mixing methods

Various diagonalization schemes

ChebySCF

Direct energy minimization

For systems with band gaps:

Conjugate gradient