(v1.1)

User's Guide for ABACUS

The ABACUS team

November 18, 2015

Contents

Chapter 1

Introduction

ABACUS (Atomic-orbtial Based Ab-initio Computation at UStc) is an open-source computer code package aiming at large-scale electronic-structure simulations from first principles, developed at the Key Laboratory of Quantum Information, Unviersity of Science and Technology of China (USTC). ABACUS uses norm-conserving pseudopotentials to describe the interactions between nuclear ions and valence electrons. As its name indicates, ABACUS primarily employs numerically tabulated atom-centered orbitals as its basis functions to expand electronic wave functions. These basis functions are generated by using an optimization scheme developed by Chen, Guo, and He (CGH) [1] early on, and form a series of hierarchical basis sets, through which the computational accuracy can be systematically improved by increasing the basis set size. Besides atomic basis sets, ABACUS also allows to use plane-wave basis set as an alternative option. This dual basis-set feature allows for a convenient consistency and accuracy check within ABACUS.

Currently ABACUS provides the following features and functionalities.

- Ground-state total energy calculations using Kohn-Sham (KS) density functional theory (DFT) [2, 3] with local-density or generalized gradient approximations (LDA/GGAs).
- Brillouin zone sampling using the Monkhorst-Pack special **k**-points [4].
- Geometry relaxation with both Conjugated Gradient (CG) and BroydenFletcherGoldfarb-Shanno (BFGS) [5] methods.
- Semi-empirical van der Waals energy correction using the Grimme DFT-D2 [6] scheme.
- NVT molecular dynamics simulation using the Nosé-Hoover thermostat [7].

Other functionalities including hybrid density functionals and time-dependent DFT are under active development, and will be available for the next release.

ABACUS can run both on desktop computers and high performance supercomputers, for the moment under the unix/linux-based environment. It can scale up to O(10^3) CPU cores.

For any use of ABACUS, please cite the following paper,

> [1] M. Chen, G-C Guo, and L. He, Systematically improvable optimized atomic sets for ab initio calculations, J. Phys.: Condens. Matter **22**, 445501 (2010).
>
> [2] P. F. Li, X. H. Liu, M. H. Chen, P. Z. Lin, X. G. Ren, L. Lin, and L. He, Large-scale ab initio simulations based on systematically improvable atomic basis, Comput. Mater. Sci., in print; DOI:10.1016/j.commatsci.2015.07.004

Chapter 2

Obtaining and Installing ABACUS

2.1 How to obtain ABACUS

ABACUS can be downloaded from its official website:

http://abacus.ustc.edu.cn/

From the DOWNLOAD webpage, the user can obtain the ABACUS code package in a gzipped tar file, "`ABACUS_v$num.tar.gz`", where `$num` is the version number. The current release of ABACUS is version `1.0`.

2.2 Structure of the program

After obtaining the package, untar it and the entire package will extract itself into a directory `ABACUS`.

```
tar -zxvf ABACUS_v$num.tar.gz
```

Under the ABACUS directory, there exist the following subdirectories,

- `bin/`
  where the executable program resides;
- `data/`
  which contains a number of chemical elements' numerical orbitals, the UPF-type LDA and/or PBE pseudopotentials, as well as various structures;
- `examples/`
  which contains test examples;
- `source/`
  which contains the source code and various makefiles for different hardware platforms;
- `tools/`
  which currently contains the script for generating the numerical atomic orbitals;

The source directory further contains the following subdirectories, where the source code of ABACUS resides:

- `source/src_pw/`
- `source/src_lcao/`
- `source/src_ions/`
- `source/src_global/`
- `source/src_parallel/`
- `source/src_external/src_pdiag/`

2.3 Installation

2.3.1 Prerequisites

In order to compile ABACUS from the source code, the user needs to have the following prerequisites

1. a minimal Unix environment: basically a command shell (e.g., `bash`) and the utilities `make`;
2. C compilers, MPI libraries and a parallel compiler (e.g OpenMpi http://www.open-mpi.org/);
3. FFTW libraries (http://www.fftw.org/) for Fast Fourier Transforms;
4. CLAPACK libraries (http://www.netlib.org/lapack/) for linear algebra (or Intel mkl library);
5. Boost C++ libraries (http://www.boost.org/)

2.3.2 Building the program

To compile the ABACUS program, go to the source directory

```
cd source/
```

Then open and edit the file Makefile.system using any editor tool you like, e.g., `vi`

```
vi Makefile.system
```

Change the compiler and libraries to the proper path in your own machine

```
CPLUSPLUS_MPI =
FFTW_DIR =
LIBS =
BOOST =
```

Then go to the file Makefile, and set the parameters in the following way

```
HONG = ${HONG_MPI}
HONG_MPI = -D__FP -D__FFTW2 -D__MPI
```

if you need a parallel computing. Finally, to build the program, simply type

```
make fp_mpi
```

where ‘fp_mpi’ is one of the make options, and means a parallel version.
If the compilation finishes without error messages (except perhaps for some warnings), an executable program `ABACUS.mpi.$num` will be created in directory `bin/`, where `$num` denotes the version number.

Chapter 3

Quick Start Quide

The following files are the central input files of ABACUS. Before executing the program, please make sure these files are in the work directory.

1. The INPUT file.

   The file named INPUT contains the setting parameters used in the calculation, which informs the program "what to do and how to do it". Most parameters are supplied with default values, but some important parameters must be explicitly set by the user. For a complete list of the input parameters, please consult the Appendix 5.1 'List of keywords'.

   Attention: Users cannot change the filename "INPUT" to other names.

2. The structure file, whose default name is STRU.

   This can however be changed to a different name by explicitly specifying the name in the INPUT file. As its name indicates, the STRU file contains the structural information about the system, e.g., lattice constant, lattice vectors, and positions of the atoms within a unit cell. The positions can be given in either direct or Cartesian coordinates. Moreover, the name (and location of the pseudopotential and numerical orbital files, see below) need to be specified in the STRU file.

3. The k-point file, whose default name is KPT.

   The KPT file contains the information of the **k**-grid setting for the Brillouin zone sampling.

4. The pseudopotential files.

   Norm-conserving pseudopotentials are used in ABACUS, in the UPF file format. The filename (and sometimes location) of each element's pseudopotential needs to be specified in the STRU file.

5. The numerical orbital file.

   When doing calculations with atomic orbital basis, it's necessary to prepare a numerical orbital file for each element in the system. Generally, the numerical orbital file should be prepared by the user, which will be described later. The filename (and sometimes location) of each element's numerical orbital needs to be specified in the STRU file.

3.1 Input files

3.1.1 The INPUT file

Some of the most important parameters that need to be set in the INPUT file are given below

```
INPUT_PARAMETERS
#Parameters (General)
ntype           1
nbands          4

#Parameters (Accuracy)
ecutwfc         50
```

Parameters list starts with key word `INPUT_PARAMETERS`. Any content before `INPUT_PARAMETERS` will be ignored. Each parameter value is provided by specifying the name of the input variable and then putting the value after the name, separated by one or more blank characters(space or tab). The following characters($\leq$ 150) in the same line will be neglected. Depending on the input variable, the value may be an integer, a real number or a string. The parameters can be given in any order, but only one parameter should be given per line. Furthermore, if a given parameter name appeared more than once in the input file, only the last value will be taken. Last, if a name is not recognized by the program, the program will stop with an error message.

In the above example, the meanings of the parameters are

- `ntype` how many types of elements in the unit cell
- `nbands` the number of bands to be calculated
- `ecutwfc` the plane-wave energy cutoff for the wave function expansion (UNIT: Rydeberg)

For more information about the input parameters, see Appendix 5.1

3.1.2 The STRU file

The STRU file contains the information about the name(s) and/or location(s) of the pseudopotential and numerical orbital files, as well as the structural information about the system. Take the diamond as an example, there are only two atoms in the primitive cell, and its STRU file looks as follows:

```
ATOMIC_SPECIES
Si 1.000 Si.pz-vbc.UPF  // label; mass; pseudo_file

NUMERICAL_ORBITAL
Si_lda_8.0au_50Ry_2s2p1d  //numerical_orbital_file

LATTICE_CONSTANT
10.2  // lattice scaling factor (Bohr)

LATTICE_VECTORS
0.5 0.5 0.0 // latvec1
0.5 0.0 0.5 // latvec2
0.0 0.5 0.5 // latvec3
```

```
ATOMIC_POSITIONS
Direct //Cartesian or Direct coordinate.
Si  // Element type
0.0 // magnetism
2   // number of atoms
0.00 0.00 0.00 0 0 0
0.25 0.25 0.25 1 1 1
```

The STRU file contains several sections, and each section must start with a keyword like `ATOMIC_SPECIES`, `NUMERICAL_ORBITAL`, or `LATTICE_CONSTANT`, etc. to signify what type of information that comes below.

In this example, only the names of the pseudopotential and numerical orbital files are given, but not the location. This simply means these files are located in the work directory.

1. `ATOMIC_SPECIES`

   This section provides information about the type of chemical elements contained the unit cell. Each line defines one type of element. The user should specify the name, the mass, and the pseudopotential file used for each element. The mass of the elment is only used in molecular dynamics simulations. For electronic-structure calculations, the actual mass value isn't important, and it is set to 1.000 here. 'Si.pz-vbc.UPF' is the pseudopotential file. Here, no explicit path is specified, and this simply means that the file is located in work directory. If this is not the case, please explicitly specify the precise location of the pseudopotential file.

2. `NUMERICAL_ORBITAL` Numerical atomic orbitals are only needed for LCAO calculations. Thus this section will be neglected in PW calcultions. 'Si_50Ry_8.0au_dzp' is name of the numerical orbital file. Again here the path is not specified, which means that this file is located in the work directory.

3. `LATTICE_CONSTANT`

   The lattice constant of the system in unit of Bohr.

4. `LATTICE_VECTORS`

   The lattice vectors of the unit cell. It is a 3×3 matrix written in 3 lines. Please note that the lattice vectors here are scaled by the lattice constant.

5. `ATOMIC_POSITIONS`

   This section specifies the positions and other information of individual atoms. The first line signifies whether atom positions are given in `Cartesian` or `Direct` coordinates.

   The following three lines tells the elemental type (Si), the initial magnetic moment (0.0), and the number of atoms for this particular element (2) repsectively.

   The last two lines in this example are the coordinates of atomic positions. There are six numbers in each line: the first three specifies the atomic positions and the last three control how the atom move in geometry relaxation calculations. The numbers "`0 0 0`" following the coordinates of the first atom means this atom are not allowed to move in all three directions, and the numbers "`1 1 1`" following the coordinates of the second atom means this atom can move in all three directions.

3.1.3 The KPT file

The k-point file contains a few lines as follows,

```
K_POINTS
0
Gamma
4 4 4 0.0 0.0 0.0
```

The first line is a keyword, and it can be set as `K_POINTS`, or `KPOINTS` or just `K`.

The second line is an integer, and its value determines how to get k-points. In this example, '0' means using Monkhorst-Pack (MP) method to generate k-points automatically.

The third line tells the input type of k-points, 'Gamma' or 'MP', different Monkhorst Pack (MP) method. Monkhorst-Pack(MP) is a method which uses the uniform k-points sampling in Brillouin-zone, while 'Gamma' means the Γ-centered Monkhorst-Pack method.

The first three numbers of the last line are integers, which give the MP **k** grid dimensions, and the rest three are real numbers, which give the offset of the **k** grid. In this example, the numbers "`0 0 0`" means that there is no offset, and this is the a standard 4×4×4 **k** grid.

3.1.4 The pseudopotential file

As mentioned above, 'Si.pz-vbc.UPF' is a pseudopotential file, which contains the norm-conserving pseudopotential of Si element. The file is in UPF file format and begins like

```
<PP_INFO>
Generated using ld1 code
Author: P. Giannozzi    Generation date: 1990
Info: H LDA 1s1 VonBarth-Car local
    0        The Pseudo was generated with a Non-Relativistic Calculation
  0.00000000000E+00    Local Potential cutoff radius
nl pn  l   occ               Rcut            Rcut US             E pseu
1S  0  0  1.00      0.00000000000      0.00000000000      0.00000000000
</PP_INFO>
...
```

In ABACUS there are pseudopotentials of most elements commonly used in the directory $ABACUS_DIR/data/elements/ . Please specify the filename and location of pseudopotentials in the STRU file.

3.1.5 The numerical orbital file

'Si_lda_8.0au_50Ry_2s2p1d' is a numerical orbital file of the Si element. The name indicates that this particular atomic basis set is generated using the LDA pseudopotential, with an energy cutoff of 50 Rydeberg and a radius cutoff of 8.0 Bohr. Furthermore, this is a DZP (double ζ orbitals + one polarized orbital) basis, containing two s functions, two p functions, and one d function. The orbital file begins with basic information like

```
---------------------------------------------------------------------------
Element                    Si
Energy Cutoff(Ry)           50
Radius Cutoff(a.u.)         8
Lmax                        2
Number of Sorbital-->       2
Number of Porbital-->       2
Number of Dorbital-->       1
---------------------------------------------------------------------------
SUMMARY  END
...
```

ABACUS provide atomic basis sets of different accuracy levels(low, medium and high) in directory $ABACUS_DIR/data/elements/ . Users can generate basis themselves, and the procedure to do so will be introduced in detail in Sec. 4.1.1. The filename and location should be set correctly in the STRU file.

3.2 Running the program

Suppose now you are in your work directory, first prepare INPUT, STRU and KPT files mentioned above here, And then copy the pseudopotential file Si.pz-vbc.UPF also to this directory.

```
cp $ABACUS_DIR/data/elements/14_Si/1_LDA/Si.pz-vbc.UPF ./
```

Where $ABACUS_DIR denotes the path to the ABACUS package. Run the program by typing

```
$ABACUS_DIR/bin/ABACUS.mpi.$num
```

The following typical output information will be printed to the screen:

```
 *********************************************************
*                                                       *
*                  WELCOME TO ABACUS                    *
*                                                       *
*            'Atomic-orbital Based Ab-initio            *
*                  Computation at UStc'                 *
*                                                       *
*********************************************************
Wed Sep  9 21:44:58 2015
MAKE THE DIR         : OUT.ABACUS/
```

```
DONE(0.013026   SEC) : SETUP UNITCELL
DONE(0.0143099  SEC) : INIT K-POINTS
---------------------------------------------------------
This calculation is self-consistent
---------------------------------------------------------
SPIN    KPOINTS         PROCESSOR
1       64              1
---------------------------------------------------------
Use plane wave basis
---------------------------------------------------------
ELEMENT NATOM       XC
Si      2           PZ-LDA
---------------------------------------------------------
Initial plane wave basis and FFT box
---------------------------------------------------------
DONE(5.13402    SEC) : INIT PLANEWAVE
UNIFORM GRID DIM     : 36 * 36 * 36
UNIFORM GRID DIM(BIG): 36 * 36 * 36
MEMORY FOR PSI (MB)  : 6.26562
DONE(5.14667    SEC) : LOCAL POTENTIAL
DONE(5.16344    SEC) : NON-LOCAL POTENTIAL
START POTENTIAL      : atomic
DONE(5.17689    SEC) : INIT POTENTIAL
DONE(6.7622     SEC) : INIT BASIS
-------------------------------------------
SELF CONSISTENT :
-------------------------------------------
ITER   ETOT(eV)       EDIFF(eV)      DRHO2      CG_ITER    TIME(S)
CG1    -2.192279e+02  0.000000e+00   6.928e-02  3.000e+00  3.420e+00
CG2    -2.154958e+02  3.732172e+00   5.745e-03  2.871e+00  3.370e+00
...
```

This is a self-consistent ground-state calculation with the plane wave basis. The first few lines show some basic information of the calculation. And the last few lines describes the electronic steps iterated to self-consistency. The ‘ITER’ column is electron iteration steps, and ’CG’ means that the conjugate gradient method is used here to solve Kohn-Sham equation. ‘ETOT’ is the total energy of the system in every iteration step in unit of eV. ‘EDIFF’ is the total energy difference between an iteration step and the last step in unit of eV. ‘DRHO2’ is the error between input and output charge density every iteration step. ‘CG_ITER’ is average iteration number in ccgdiag. ‘TIME’ is time used every iteration step in unit of seconds.

3.3 Output files

When the calculation finishes, the program will create an output directory (default: OUT.ABACUS/), into which the following output files will be generated:

1. INPUT: contains all input parameters, user’s input and default.
2. istate.info: information of energy eigenvalues.

3. running_scf.log: contains the running details.
4. STRU_READIN_ADJUST.cif: structure file in the cif formatter.
5. warning.log: errors and warning messages.
6. Si/ : element information
    - Si.NONLOCAL: non-local pseudopotential projectors.
    - Si-P.ORBITAL: pseudo atomic orbitals, p orbital
    - Si-S.ORBITAL: pseudo atomic orbitals, s orbital
    - v_loc_g.dat: vlocal in G space

3.4 Specific Examples

ABACUS allows to do various different kinds of calculations. After briefly introducing the input and output files of ABACUS, we described below with more details how to run ABACUS calculations using specific examples.

As alluded to above, ABACUS calculations can be done both with plane-wave basis and numerical atomic basis. These two types of calculations will be illustrated separately in the following.

3.4.1 Basic calculations with the PW basis set

We still take the silicon crystal in the diamond structure as the example. The basic task of the calculation is to obtain the ground-state charge density and total energy of the system. More parameters are explicitly set in this example to illustrate how to set up the input files.

First create a work directory, `mkdir -p test/Si_diamond_pw`

Then change to the directory and run a test calculation here.

```
cd test/Si_diamond_pw
```

Add a few more parameters to the `INPUT` file, in addition to those already shown in 3.1.1.

```
basis_type          pw
suffix              Si2_diamond
symmetry            1
niter               50
dr2                 1.0e-9
out_charge          1
```

The meanings of the above parameters are:

- `basis_type` The type of basis. The default value is `pw`, meaning that the plane wave set is used.
  Attention: This is a very important parameter in ABACUS. For more information, please see Appendix 5.1.

- `suffix`

  Suffix of output directory. In this example the name of the output directory will be `OUT.Si2_diamond`. The default value is ABACUS.

- `symmetry`

  Use symmetry(=`1`) or not(=`0`) in the calculation. The default value is 0.

- `niter`

  The maximal iteration number for electronic-structure calculations.

- `dr2`

  Tolerance of the difference of charge density, below which the self-consistent calculation is considered to be converged.

- `out_charge`

  Print out the charge density(=`1`) or not(=`0`).

The STRU and KPT files are the same as those shown in Sec. 3.1.2 and Sec. 3.1.3 respectively.

3.4.2 Basic calculations with the LCAO basis set

In this section we will describe how to do LCAO calculations using ABACUS. Again the crystal Si in the diamond structure will be taken as an example.

For convenience, first also create a subdirectory in directory ABACUS/

```
mkdir -p test/Si_diamond_lcao
```

Then change to this directory, and copy the STRU file, the pseudopotential file, and in addition the numerical atomic orbital file here

```
cd test/Si_diamond_lcao
cp $ABACUS_DIR/data/structures/0_collection/Si2_diamond.stru STRU
cp $ABACUS_DIR/data/elements/14_Si/1_LDA/Si.pz-vbc.UPF ./
cp $ABACUS_DIR/data/elements/14_Si/1_LDA/Si_lda_8.0au_50Ry_2s2p1d ./
```

The `INPUT` file is similar to that in section 3.4.1, and only the values of following two parameters are different:

```
basis_type          lcao
ks_solver           hpseps
```

- `ks_solver` The method to solve Kohn-Sham equation. Default value is `cg`, thus the conjugate gradient method. Here `hpseps` means using a High Performance Symmetric Eigenproblem Solvers package (HPSESP) [8].

The KPT file is the same as that in section 3.1.3.

```
$ABACUS_DIR/bin/ABACUS.mpi.$num
```

The information printed on the screen is different from that obtained using the plane-wave basis,

```
...
 ITER   ETOT(eV)        EDIFF(eV)      DRHO2     TIME(s)
 HP1    -2.151753e+02  0.000000e+00   1.685e-01  1.204e+01
 HP2    -2.152145e+02  -3.919739e-02  3.471e-02  1.169e+01
 HP3    -2.152153e+02  -8.138606e-04  3.748e-03  1.164e+01
 HP4    -2.152153e+02  -9.146924e-06  4.226e-05  1.221e+01
 ...
```

The string `HP`n in the first column means that the HPSEPS eigenvalue solver is used, and this is the n-th self-consistent KS iteration. In contrast, the output information from the PW calculation has the string `CG`n in its first column, indicating the Conjugate Gradients (CG) method is used to solve the Kohn-Sham equation.

In many cases, only one Γ point (i.e., k=0) is needed in the calculations. In these cases, one can set in the `INPUT` file:

- `gamma_only` If set to 1, only Γ point is used in the calculation. The `gamma_only` algorithm is much faster than the standard algorithm.

Chapter 4

Main Features and Functionalities

As a comprehensive first-principles computational package, ABACUS has many keywords. Most of them have default values and the users don't have to set explicitly by themselves. A complete list of keywords are given in the Appendix. Here we summarize the most relevant keywords that the users should know about, and explain how to set up the input files according to the features and functionalities of the code.

4.1 Basis sets

In ABACUS, there are 3 types of basis set choices:

1. PW
   Plane wave basis.
2. LCAO
   Localized atomic-orbital basis; these systematically improvable atomic orbitals are generated with a tool called SIAO (see Sec. 4.1.1).
3. LCAO_in_PW
   Expand the atomic basis in terms of plane waves, and use plane-waves techniques to construct the Hamiltonian matrix, but solve the eigenvalue problem within the LCAO basis set.

In the `INPUT` file, the keyword `basis_type` controls what basis type to use.

1. PW: `basis_type = pw`
2. LCAO: `basis_type = lcao`
3. LCAO_in_PW: `basis_type = lcao_in_pw`

The default value of `basis_type` is `pw`. When choosing `lcao` or `lcao_in_pw`, the user should prepare atomic orbitals first.

4.1.1 Orbital generation

In ABACUS, the atomic orbital bases are generated using a scheme developed in Ref. [1]. We provide a script named "generate_orbital.sh" under the directory $ABACUS/tools/ SIAB/2_Generate to generate the atomic orbitals bases. In order to run this script, an `ORBITAL_INPUT` file is required.

An example of this `ORBITAL_INPUT` file can be found in $ABACUS/tools/ SIAB/2_Generate.

```
#1.exe_dir
#----------------------------------------------------------------------------
EXE_pw           bin/ABACUS.mpi.$num
EXE_orbital      bin/SIA_s.exe
#----------------------------------------------------------------------------
#( In this part, the direction of the two used exe is provided )

#2.electronic calculatation
#----------------------------------------------------------------------------
targets     07_N          # element
ref_bands   5                  # reference bands
nbands      8             # num of bands for calculate ( > reference bands)
Ecut        50                # cutoff energy (in Ry)
Rcut        6                 # cutoff radius (in a.u.)
Pseudo_dir  ./
Pseudo      N.LDA.UPF
sigma       0.01        # energy range for gauss smearing (in Ry)
#----------------------------------------------------------------------------
#( In this part , some parameters of calculating are given )
#3.structure information
#----------------------------------------------------------------------------
Dis  1.0    1.1    1.5    2.0    3.0
#----------------------------------------------------------------------------
#( In this part , it gives us the bond length of the reference system( in
#angstrom) )

#4.orbital calculatation
#----------------------------------------------------------------------------
maxL         2    # the max angular momentum
Level        2    # num of levels to generate orbitals(<=5)
#(num)  (the max ang) (num of S) (num of P) (num of D) (num of F) (num of G)
level1        1            1          1
level2        2            1          1          1
#----------------------------------------------------------------------------
#( In this part, some information of orbital is given )

#5.Metropolis parameters (in most cases do not need to change)
#----------------------------------------------------------------------------
Start_tem_S  1.0e-4 # start temperature for optimize Spillage(default 1.0e-4)
Start_tem_K  1.0e-2 # start temperature for optimize Kinetic (default 1.0e-2)
Step_S       20      # num of steps for optimize Spillage (default 20)
Step_K       15     # num of steps for optimize Kinetic  (default 15)
Delta_kappa  0.01    # delta kappa (default 0.01)
#----------------------------------------------------------------------------
#(In this part, some parameters of Metropolis is given. In most cases, they
```

```
#do not need to be changed , only when you run into a situation , that the
#Kinnetic energy is larger than the  maximum value allowed , you can enlarge
#the start temperature appropritely, or you can enlarge the delta_kappa, e.g.
#start_tem_k 1.0e-2 to 5.0e-1, delta_kappa 0.01 to 0.02. more steps can make
#the orbitals better , too)
```

The `ORBITAL_INPUT` file contains 5 parts :

1. `exe_dir`
   The paths of two executable files:
   - `EXE_pw` executable file of ABACUS
   - `EXE_orbital` executable file of orbital generation

   We can get the exe file of orbital generation as below:

   ```
   cd ABACUS/tools/SIAB/1_Source/
   make s
   ```

2. `electronic calculation`
   Parameters for electronic calculation:
   - `targets` the element type of which orbitals are to be generated. Its value has the form of 'element.id_element', for example 07_N.
   - `ref_bands` the number of reference bands for orbital generation. We usually take the number of occupied bands of the system. For the N element, we take its "dimers" as the reference systems, so the number of ref_bands should be 5 (valence electrons of this element)*2(number of the atoms of the system)/2 (1 band contain 2 electrons) =5. While for Na element, we take trimer as reference systems, and the number of the ref_bands should be 1*3/2=1.5 for 1.5 is not a integer, here we use 2 for its ref_bands. Most elements use dimer as reference systems, except for Li, Na, K, Ca, which use trimer instead.
   - `nbands` the number of bands to be calculated in electronic calculation. Here, we use the gaussian smearing for the electronic structure calculation, so the value of this parameter can not be smaller than the value of ref_bands
   - `Ecut` the Energy cutoff (in Ry)
   - `Rcut` the radius cutoff of atomic orbital (in a.u)
   - `Pseudo_dir` the path to the directory where the pseudopotential file is.
   - `Pseudo` the file name of pseudopotential
   - `sigma` The gaussian smearing (in Ry) for scf calculations. The default vaule is 0.01. In case that the scf iterations don't converage (which could happen, e.g., for transition metal dimers), the user may increase this parameter, say, to 0.05.

3. `structure information`
   This part gives the bond lengths of the reference systems (dimer or trimer). Generally, the bond lengths are chosen to distribute on both sides of the equilibrium value. For example, for N dimer we use (in Å):

- `Dis` 1.0 1.1 1.5 2.0 3.0

It means we take 5 reference systems (dimer), and the bond lengths are 1.0 1.1 1.5 2.0 3.0 angstrom, respectively. Every element has reference systems with different bond lengths, which could be found in file $ABACUS/tools/SIAB/2_Generate/DIS.txt.

4. `orbital generation`
   The main parameters for orbital generation

   - `maxL` the max angular momentum for orbitals to be generated
   - `level` number of levels to generate orbitals.
     In the main part of this section, `level1`, `level2`... provide the information of each layer, the max angular momentum and the number of s, p, d orbitals.

   For example, if we want to use 2 steps to generate DZP basis for N, we can set this part like this

```
maxL    2    # the max angular momentum
level   2    # num of levels to generate orbitals(<=5)
(num)(the max ang)(num of S)(num of P)(num of D)(num of F)(num of G)
level1     1           1         1
level2     2           1         1         1
```

   It means we generate one s orbital and one p orbital in first step (level1), and generate one s, p, d orbital in the second step (level2).

5. `Metropolis parameters`
   The main parameters for Metropolis optimization.

   - `Start_tem_S` start temperature for spillage optimization
   - `Start_tem_K` start temperature for kinetic energy optimization
   - `Step_S` number of steps for spillage optimization
   - `Step_K` number of steps for kinetic energy optimization
   - `Delta_kappa` the accept rise of spillage when optimizing the kinetic energy

After preparing the `ORBITAL_INPUT` file, one just needs to run the script and wait for the results.

```
./generate_orbital.sh
```

The results will be written into several output files under the directory $element.id_element/$Rcut/, for example 07_N/6/.

Some output files here are useful.

- ORBITAL_RESULTS.txt this file shows some important information of the orbital, we can see the spillage of the orbital to judge whether the orbital is good enough for us to use.

- running_1.txt the details of generating orbitals

- ORBITAL_PLOTU.dat you can open it by using any drawing software to visualize the shape of the orbital
- ORBITAL_7U.dat the general type is ORBITAL_(element.id)U.dat . This is the orbital file we will use in the calculation. And you can rename it as anything you want. We usually use the covention "element_xc_rcut_ecut_$XsYpZd$", e.g., N_lda_6.0au_50Ry_$2s2p1d$, which tells the key parameters for the basis set construction.

For some elements, you can download the reference `ORBITAL_INPUT` files and pseudopotentials from our website:

http://abacus.ustc.edu.cn/pseudo.html.

A file README is also given and you can decide the parameters with it as a reference.

In most cases, you just need to modify the parameters in Section 1, 2. Section 4 may be partially modified if you need higher precision orbitals. The users are not encouraged to change the settings in sections 5, unless you are very familiar with the code generating algorithms.

4.2 Pseudopotentials

Currently ABACUS uses norm-conserving pseudopotentials in the (old) format of UPF, which is the standard pseudopotential format for Quantum ESPRESSO. ABACUS supports both LDA and PBE pseudopotentials. Take the element Si as an example:

- Its LDA pseudopotential is in file Si.pz-vbc.UPF, which is in the directory ABACUS/data/elements/14_Si/1_LDA;
- Its PBE pseudopotential is in file Si.PBE.UPF, which is in $ABACUS/data/elements/14_Si/2_PBE

To run a calculation, the pseudopotential needs to be set in `STRU` file. For example

```
ATOMIC_SPECIES
Si 1.000 Si.pz-vbc.UPF
...
```

The user can download the pseudopotential files together with a sample input file ORBITAL_INPUT from

http://abacus.ustc.edu.cn/pseudo.html

For more information of different types of pseudopotentials, please visit

http://www.quantum-espresso.org/pseudopotentials/

4.3 Boundary conditions and k-points

ABACUS uses periodic boundary conditions for both crystals and finite systems. For isolated systems, such as atoms, molecules, clusters, etc., one uses so called the supercell model. Lattice vectors of the supercell are set in the STRU file.

For the input k-point (KPT) file, the file should either contain the k-point coordinates and weights or the mesh size for creating the k-point gird. Both options are allowed in ABACUS.

1. To generate k-mesh automatically
   To generate k-mesh automatically, it requires the input subdivisions of the Brillouin zone in each direction and the origin for the k-mesh. ABACUS uses the Monkhorst-Pack method to generate k-mesh, and the following is the input k-point (KPT) file used in ABACUS.

```
K_POINTS    //keyword for start
0           //total number of k-point, '0' means generate automatically
Gamma       //which kind of Monkhorst-Pack method, 'Gamma' or 'MP'
2 2 2 0 0 0 //first three number: subdivisions along recpri. vectors
              last three number: shift of the mesh
```

   The third line controls generating k-mesh with origin at the Γ point('Gamma') or using the original Monkhorst-Pack scheme('MP').

2. To set up the **k**-points explicitly
   If the user wants to set up the **k**-points explicitly, the input k-point file should contain the k-point coordinates and weights. An example is given as follows:

```
K_POINTS    //keyword for start
8           //total number of k-point
Direct      //'Direct' or 'Cartesian' coordinate
0.0 0.0 0.0 0.125   //coordinates and weights
0.5 0.0 0.0 0.125
0.0 0.5 0.0 0.125
0.5 0.5 0.0 0.125
0.0 0.0 0.5 0.125
0.5 0.0 0.5 0.125
0.0 0.5 0.5 0.125
0.5 0.5 0.5 0.125
```

3. Band structure calculations

   ABACUS uses specified high-symmetry directions of the Brillouin zone for band structure calculations. The third line of k-point file should start with 'Line' or 'Line_Cartesian' for line mode 'Line' means the positions below are in Direct coordinates, while 'Line_Cartesian' means in Cartesian coordinates

```
K_POINTS              # keyword for start
6                     # number of high symmetry lines
Line                  # line-mode
0.5   0.0   0.5  20   # X
0.0   0.0   0.0  20   # G
0.5   0.5   0.5  20   # L
0.5   0.25  0.75 20   # W
0.375 0.375 0.75 20   # K
0.0   0.0   0.0  1    # G
```

The fourth line and the following are special **k**-point coordinates and number of k-points between this special k-point and the next.

4.4 Kohn-Sham solver

For different types of basis set choice, different methods are used to solve the Kohn-Sham equation. For PW basis, there are CG and Blocked Davidson methods for solving the eigenvalue probelm. For LCAO basis/LCAO_in_PW basis, one uses direct diagnolization method. In the `INPUT` file, the parameter 'ks_solver' controls what method to use for solveing the Kohn-Sham equation for each basis.

- PW: `ks_solver` = 'cg' or 'dav'
- LCAO: `ks_solver` = 'hpseps' or 'lapack'
- LCAO_in_PW: `ks_solver` = 'lapack'

If you set `ks_solver`='hpseps' for `basis_type`='pw', the program will be stopped with an error message:

`hpseps can not be used with plane wave basis.`

Then the user has to correct the input file and restart the calculation.

4.5 Exchange-correlation functionals

In our package, the XC functional can be either set explicitly using the `dft_functional` keyword as explained below, or set implicitly according to the XC functional information read from pseudopotential file. The user should ensure that the XC functional set in the INPUT file and the pseudopotential file are consistent.

To be specific, we briefly explain the format of the pseudopotential file and the key information it contains. There are a few lines in Si's LDA pseudopotential file Si.pz-vbc.UPF

```
<PP_HEADER>
   0                    Version Number
   Si                   Element
   NC                   Norm - Conserving pseudopotential
```

```
   F                    Nonlinear Core Correction
 SLA  PZ   NOGX NOGC   PZ   Exchange-Correlation functional
   4.00000000000       Z valence
   0.00000000000       Total energy
...
</PP_HEADER>
```

The fourth line indicates that the norm-conserving pseudopotential is used here. The sixth line shows that it uses the Perdew-Zunger exchange-correlation functional in the pseudopotential file. In ABACUS, currently the Perdew-Zunger LDA and the Perdew-Burke-Ernzerhof GGA XC functional are available.

4.5.1 LDA and GGAs

- none: the functional is specified implicity by the input pseudopotential file
- lda: Perdew-Zunger local density approximation
- pbe: Perdew-Burke-Ernzerhof general gradient approximation

If the functional specified by the user is not consistent with the pseudopotential file, the program will stop with an error message.

4.5.2 DFT + dispersion calculations

LDA and GGAs cannot describe van der Waals (vdW) interactions in a physically correct way. In order to describe materials where vdW interactions are important, one needs to go beyond LDA and GGAs. To this end, one simple and popular approach is to add a Lennard-Jones type term in terms of atom-based pairwise C_6/R^6 summations to the existing GGA functionals. There are different ways to do this. Currently ABACUS provides the Grimme DFT-D2 method [6] to describe van der Waals interactions.

To run a DFT-D2 calculation, add the following keyword to the INPUT file

```
vdwD2                1
```

When set to 1, ABACUS calculates total energies and forces with DFT-D2 correction, otherwise the corrections are not included. For more parameters please see Appendix 5.1.

4.6 Electronic structure calculations

4.6.1 Ground-state energy and charge-density calculation

The ground-state energy calculation is the primary functionality and the default option of ABACUS calculations.

In the example given in section 3.4.2, the program has calculated the charge density and the ground-state total energy. The charge density is contained in the file `SPIN1_CHG` in directory OUT.Si2_diamond. The ground-state energy information can be extracted by typing

```
grep FINAL_ETOT_IS OUT.Si2_diamond/running_scf.log
```

which give the converged final total energy.

4.6.2 DOS, wave functions

The main task of this section is to calculate the density of states (DOS) of the system. At first, do a ground-state energy calculation as in 4.6.1 and get the converged charge density, which is contained in the file `SPIN1_CHG`. Next, do a non-self-consistent calculation. In this section the potential should be constructed from the ground-state charge density from the proceeding calculation. Now the `INPUT` file is like

```
INPUT_PARAMETERS
#Parameters (General)
suffix              Si2_diamond
ntype               1
nbands              8
calculation         nscf

#Parameters (Accuracy)
ecutwfc             50
symmetry             1
niter               50
dr2                 1.0e-9
ethr                1.0e-7

#Parameters (File)
start_charge        file
out_dos             1
dos_sigma           0.07

#Parameters (Smearing)
smearing             gaussian
sigma               0.02
```

New parameters in the `INPUT` file are:

- `calculation` choose which kind of calculation: scf calculation, nscf calculation, structure relaxation or Molecular Dynamics. Now we need to do one step of nscf calculation.
  Attention: This is a main variable of ABACUS, and for its more information please see Appendix 5.1.

- `ethr` threshold for the CG method which digonalizes the Hamiltonian to get eigenvalues and eigen wave functions. If one wants to do nscf calculation, `ethr` needs to be changed to a smaller account, typically smaller than 1.0e-3. Note that this parameter only apply to plane-wave calculations that employ the CG method to diagonalize the Hamiltonian. For LCAO calculations, this parameter will be neglected !

- `start_charge` the type of starting density. When doing scf calculation, this variable can be set "atomic". When doing nscf calculation, the charge density already exists(eg. in `SPIN1_CHG`), and the variable should be set as "file". It means the density will be read from the existing file `SPIN1_CHG`. For more information please see Appendix 5.1.
- `out_dos` output density of state(DOS).
- `dos_sigma` the gaussian smearing parameter(DOS).

To have an accurate DOS, one needs to have a denser **k**-point mesh. For example, the `KPT` file can be set as:

```
K_POINTS
0
Gamma
8 8 8 0 0 0
```

Run the program, and you will see a file named `DOS1_smearing.dat` in the output directory. Next, plot file `DAT` with drawing software, and you'll get the DOS.

4.6.3 Band structure

The task of this section explains how to calculate the energy band structure. The `STRU` file is the same with those in 4.6.2. The `INPUT` can be changed to:

```
INPUT_PARAMETERS
#Parameters (General)
suffix              Si2_diamond
ntype               1
nbands              8
calculation         nscf

#Parameters (Accuracy)
ecutwfc             50
niter               50
dr2                 1.0e-9
ethr                1.0e-7

#Parameters (File)
start_charge        file
out_band             1
```

```
#Parameters (Smearing)
smearing             gaussian
sigma               0.02
```

New parameters in file `INPUT`:

- `out_band` output the band structure.

Here the the relevant k-point file is no longer `KPT`, but rather `KLINES`. It looks like,

```
K_POINTS              # keyword for start
6                     # number of high symmetry lines
Line                  # line-mode
0.5   0.0   0.5  20   # X
0.0   0.0   0.0  20   # G
0.5   0.5   0.5  20   # L
0.5   0.25  0.75 20   # W
0.375 0.375 0.75 20   # K
0.0   0.0   0.0  1    # G
```

- `6` number of k points, here means 6 k points:
  (0.5, 0.0, 0.5) (0.0, 0.0, 0.0) (0.5, 0.5, 0.5) (0.5, 0.25, 0.75) (0.375, 0.375, 0.75) (0.0, 0.0, 0.0)
- `20/1` number of k points along the segment line, which is constructed by two adjacent k points.

Run the program, and you will see a file named `BANDS_1.dat` in the output directory. Plot it to get energy band structure.

4.6.4 Magnetic properties

For spin polarized calculations, the users should turn out the spin option and give an initial magnetism. The input parameter that controls spin options is

```
nspin               2
```

`nspin`=1, 2
`nspin`=1, the default value, meaning spin-unpolarized calculation.
`nspin`=2, collinear spin polarized calculation.

Initial magnetic moments are set in the STRU file, in the third line of ‘ATOMIC_POSITIONS’ part. For example,

```
...

ATOMIC_POSITIONS
Direct //Cartesian or Direct coordinate.
Si  // Element type
0.0 //initial magnetism
2   // number of atoms
...
```

For each element in the system, users should give their initial magnetism when `nspin=2`.

4.7 Force calculation and structure relaxation

To calculate forces that the atoms experience, one needs to add a few more parameters in the `INPUT` file. Below is an example for the Si dimer.

```
suffix              Si2_dimer
calculation         relax
gamma_only          1
nstep               100
force_thr_ev        0.01
move_method         cg
out_stru            1
trust_radius_ini    0.5
```

- `nstep` the number of ionic iteration steps.
- `force_thr_ev` the threshold for the force, below which the geometry relaxation is considered to be converged. The unit is eV/Angstrom.
- `move_method` the algorithm used for geometry optimization. Possible choices are:
  - `cg` : Conjugate Gradient (CG) algorithm
  - `bfgs` : Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm
  - `sd` : Steepest descend algorithm

  The CG method is recommended.
- `out_stru` output the structure of each step or not.
- `trust_radius_ini` the initial radius of the relaxation. We advise you not to change this parameter, unless you are sure that the initial structure is close to the final structure.

4.8 Molecular dynamics

To run molecular dynamics (MD) simulations with ABACUS, the KPT file and the STRU file should be set in the same way as in Sec. 4.7. A typical INPUT file for MD simulation is given below, which contains additional keywords that need to be added.

```
INPUT_PARAMETERS
#Parameters (General)
suffix              Sn_md_test
ntype               1
nbands              200
calculation         md

#Parameters (Accuracy)
ecutwfc             50
symmetry         0
niter               200
dr2                 1.0e-6

#Parameters (Smearing)
smearing         gaussian
sigma               0.02

#Parameters (LCAO)
basis_type          lcao
mixing_beta         0.1

#Parameters (Technique)
gamma_only          1
charge_extrap       second-order

#Parameters (Ions)
force_thr_ev        0.01
move_method         cg

#Parameters (md)
md_mdtype          1      //choose ensemble
md_dt              1      //time step
md_tfirst          700    //the first target temperature
md_rstmd           0      //whether restart md
md_qmass           1      //mass of themostat
md_dumpmdfred      10     //The period to dump MD
                            information for monitoring and restarting MD
```

These MD parameters mean that ABACUS will use NVT ensemble with Nosé-hoover themostat; the time step is 1fs, and target temperature is 700K; start renew without restart file, set the mass of themostat as 1g/mol, and calculate the MSD and diffusion coefficent from first step.

Note: Please turn off symmetry when do MD simulation.

- `md_mdtype` : 0, NVE; 1, NVT; 2, velocity scaling
- `md_dt` : time step in md simulation (fs)
- `md_tfirst` : target temperature in md simulation(K), you should set parameter md_tlast and md_fixtemperature when you want to change temperature during md simulation.
- `md_rstmd` : 0, no need of restart ; 1, restart with restart file, you must repalce STRU file with STRU_MD before you run the restart task.
- `md_qmass` : mass of thermostat, set by experience, if you don't know how to set, set it to 0 will have a number autosetted by ABACUS
- `md_dumpmdfred` : frequency for output consequence of md simulation

Run the program, and see results in the output directory. The following files are about MD:

- `md_pos_$num.cif` optimized structure in direct coordinate
- `md_pos_$num.xyz` optimized structure in cartesian coordinate
- `MD_RESTART` output the information of md for restart
- `STRU_MD` if you want to restart md, you must replace the STRU with this file.

see MD information in file running_md.log or in file MD_OUT

```
NVT Temperature      :     469.72 (K)       //temperature at this second
NVT Kinetic energy   :   0.285593 (Hartree) //kinetic energy of system
NVT Potential energy :   -35.1068 (Hartree) //potential energy of system
NVT Conservation     :   -34.8212 (Hartree) //total energy of system

Thermostat Position : -8.49027e-05          //themostat's information
Thermostat vel : -1.04976e-05
scaling factor of vel : 1.00006
```

Check these information to confirm whether temperature and conservation are steady.

Chapter 5

Appendix: List of Keywords

5.1 List of input variables

1 System	**Description**
`suffix`	*Type*: String *Description*: In each run, ABACUS will generate a subdirectory in the working directory. This subdirectory contains all the information of the run. The subdirectory name has the format: OUT.suffix, where the 'suffix' is the name you can pick up for your convenience. *Default*: `ABACUS.`
`ntype`	*Type*: Integer *Description*: Number of different atom species in this calculations. This value must be set. If the number you set is smaller than the atom species in the STRU file, ABACUS only read the 'wrong number' of atom information. If the number is larger than the atom species in the STRU file, ABACUS may stop and quite. *Default*: `No default value.`
`nbands`	*Type*: Integer *Description*: Number of bands to calculate. It is recommended you setup this value, especially when you use smearing techniques, more bands should be included. Default value is the number of fully occupied states plus 10 more bands *Default*: `the number of fully occupied states.`
`atom_file`	*Type*: String *Description*: This parameter specifies the name of structure file which contains various information about atom species, including pseudopotential files, local orbitals files, nonlocal projectors files, cell information, atom positions, local orbitals that will be used and whether atoms should be allowed to move. *Default*: `STRU.`

`kpoint_file`	*Type*: String *Description*: This parameter specifies the name of k-points file. Note that if you use atomic orbitals as basis, and you only use gamma point, you don't need to have k-point file in your directory, ABACUS will automatically generate 'KPT' file. Otherwise, if you use more than one k-point, please do remember the algorithm in ABACUS is different for gamma only and various k-point dependent simulations. So first you should turn off the k-point algorithm by set *gamma_only* = 0 in INCAR and then you should setup your own k-points file. *Default*: `KPT`.
`pseudo_dir`	*Type*: String *Description*: This parameter specifies pseudopotential directory. *Default*: `./`
`nbands_istate`	*Type*: Integer *Description*: Only used when *calculation* = *ienvelope* or *calculation* = *istate*, this variable indicates how many bands around Fermi level you would like to calculate. *ienvelope* means to calculate the envelope functions of wave functions $\Psi_i = \sum_\mu C_{i\mu}\Phi_\mu$, where Ψ_i is the ith wave function with the band index i. Φ_{mu} is the localized atomic orbital set. *istate* means to calculate the density of each wave function $\|\Psi_i\|^2$. Specifically, suppose we have highest occupied bands at 100th wave functions. And if you set this variable to 5, it will print five wave functions from 96th to 105th. But before all this can be carried out, the wave functions coefficients $C_{i\mu}$ should be first calculated and written into a file by setting the flag *out_lowf* = 1. *Default*: `5`.
`nspin`	*Type*: Integer *Description*: Number of spin components of wave functions. There are only two choices now: 1 or 2, meaning non spin or collinear spin. *Default*: `1`.
`calculation`	*Type*: String *Description*: Specify the type of calculation. • *scf*: do self-consistent electronic structure calculation • *relax*: do structure relaxation calculation, one can ues 'nstep' to decide how many ionic relaxations you want. • *nscf*: do the non self-consistent electronic structure calculations only. For this option, you need a charge density file. • *istate*: Please see the explanation for variable *nbands_istate*. • *ienvelope*: Please see the explanation for variable *nbands_istate*. • *md*: molecular dynamics *Default*: `scf`.

This part of variables are used to control the plane wave related parameters.

PW	**Description**

`ecutwfc`	*Type*: Real *Description*: Energy cutoff for plane wave functions, the unit is Rydberg. Note that even for localized orbitals basis, you still need to setup a energy cutoff for this system. Because our local pseudopotential parts and the related force are calculated from plane wave basis set, etc. Also, because our orbitals are generated by matching localized orbitals to a chosen set of wave functions from certain energy cutoff, so this set of localize orbitals are most accurate under this same plane wave energy cutoff. *Default*: `50.`
`ethr`	*Type*: Real *Description*: Only used when you use *diago_type* = *cg* or *diago_type* = *david*. It indicates the threshold for the first electronic iteration, from the second iteration the ethr will be updated automatically. *Default*: `0.01`
`start_wfc`	*Type*: String *Description*: Only useful for plane wave basis only now. It is the name of the starting wave functions. In the future we should also make this variable available for localized orbitals set. • *atomic*: • *file*: *Default*:`atomic.`
`start_charge`	*Type*: String *Description*: This variable is used for both plane wave set and localized orbitals set. It indicates the type of starting density. If set this to 'atomic', the density is starting from summation of atomic density of single atoms. If set this to 'file', the density will be read in from file. The file should be in the output directory. Besides, when you do 'nspin=1' calculation, you only need the density file *SPIN1_CHGCAR*. However, if you do 'nspin=2' calculation, you also need the density file *SPIN2_CHGCAR*. The density file should be output with these names if you set *out_charge* = 1 in INPUT file. • *atomic*: • *file*: *Default*:`atomic.`

This part of variables are used to control the electronic structure and force calculations.

Name	**Description**

`basis_type`	*Type*: String *Description*: This is very important parameters to choose basis set in ABACUS. • `pw`: Using Plane-Wave basis set only. • `lcao_in_pw`: Expand the localized atomic set in Plane-Wave basis. • `lcao`: Using localized atomic orbital sets. *Default*: `pw.`
`ks_solver`	*Type*: String *Description*: It's about choice of diagonalization methods for hamiltonian matrix expanded in a certain basis set. • `hpseps` This method should be used if you choose localized orbitals. • `cg` *cg* method is only used for plane wave basis calculations. • `david` *david* is the Davidson algorithm for plane wave basis calculations only. • `lapack` *lapack* can be used for localized orbitals, but is only used for single processor. *Default*: `cg.`
`smearing`	*Type*: String *Description*: It indicates which occupation and smearing method is used in the calculation. • `fixed`: use fixed occupations. • `gauss or gaussian`: use gaussian smearing method. • `mp`: use methfessel-paxton smearing method. *Default*: `fixed.`
`sigma`	*Type*: Real *Description*: energy range for smearing, the unit is Rydberg. *Default*: `0.01`
`mixing_type`	*Type*: String *Description*: Charge mixing methods. • `plain`: Just simple mixing. • `kerker`: Use kerker method, which is the mixing method in G space. • `pulay`: Standard Pulay method. • `pulay-kerker`: *Default*: `pulay.`

`mixing_beta`	*Type*: Real *Description*: mixing parameter: 0 means no new charge *Default*: `0.7`
`mixing_ndim`	*Type*: Integer *Description*: It indicates the mixing dimensions in Pulay, Pulay method use the density from previous *mixing_ndim* steps and do a charge mixing based on these density. *Default*: `8`
`mixing_gg0`	*Type*: Real *Description*: used in pulay-kerker mixing method *Default*: `1.5`
`gamma_only`	*Type*: Integer *Description*: It is an important parameter in localized orbitals set. It you set *gamma_only* = 1, ABACUS use gamma only, the algorithm is fast and you don't need to specify the k-points file. If you set *gamma_only* = 0, more than one k-point is used and the ABACUS is slower compared to gamma only algorithm. *Default*: `0`
`printe`	*Type*: Integer *Description*: Print out energy for each band for every *printe* steps *Default*: `100`
`niter`	*Type*: Integer *Description*:This variable indicates the maximal iteration number for electronic iterations. *Default*: `40`
`diago_cg_maxiter`	*Type*: Integer *Description*: Only useful when you use *diago_type* = *cg* or *diago_type* = *david*. It indicates the maximal iteration number for `cg/david` method. *Default*: `40`
`diago_david_ndim`	*Type*: Integer *Description*: Only useful when you use *diago_type* = *david*. It indicates the maximal dimension for `david` method. *Default*: `10`
`dr2`	*Type*: Real *Description*: An important parameter in ABACUS. It's the threshold for electronic iteration. It represents the charge density error between two sequential density from electronic iterations. Usually for local orbitals, usually 10^{-6} may be accurate enough. *Default*:`1e-09`
`charge_extrap`	*Type*: String *Description*: Methods to do extrapolation of density when ABACUS is doing geometry relaxations. • *atomic*: atomic extrapolation • *first-order*: first-order extrapolation • *second-order*: second-order extrapolation *Default*:`atomic`

`out_charge`	*Type*: Integer *Description*: If set to 1, ABACUS will output the charge density on real space grid. The name of the density file is $SPIN1\_CHGCAR$ and $SPIN2\_CHGCAR$ (if $nspin = 2$). Suppose each density on grid has coordinate (x, y, z). The circle order of the density on real space grid is: z is the outer loop, then y and finally x (x is moving fastest). *Default*:`0`
`out_dm`	*Type*: Integer *Description*: If set to 1, ABACUS will output the density matrix of localized orbitals, only useful for localized orbitals set. The name of the output file is $SPIN1\_DM$ and $SPIN2\_DM$ in the output directory. *Default*: `0`
`out_wf`	*Type*: Integer *Description*: Only used in localized orbitals set. When set this variable to 1, it outputs the coefficients of wave functions. The file name is $LOWF\_GAMMA\_S1.dat$ when using only GAMMA k-point for spin 1 component and $LOWF\_GAMMA\_S2.dat$ for spin 2 component. *Default*: `0`
`out_lowf`	*Type*: Integer *Description*: If set to 1, ABACUS will output the wave functions coefficients. *Default*: `0`
`out_alllog`	*Type*: Integer *Description*: When set to 1, ABACUS will generate a log file for each processor when parallel, it is very useful for debugging. *Default*: `0`
`nstep`	*Type*: Integer *Description*: The number of ionic iteration steps, the minimal value is 1. *Default*: `1`
`force_thr`	*Type*: Real *Description*: The threshold of the force convergence, it indicates the largest force among all the atoms, the unit is $Ry/Bohr$, *Default*: `0.000388935 Ry/Bohr = 0.01 eV/Angstrom`
`force_thr_ev`	*Type*: Real *Description*: The threshold of the force convergence, has the same function as $force\_thr$, just the unit is different, it is $eV/Angstrom$, you can choose either one as you like. The recommendation value for using atomic orbitals is $0.04eV/Angstrom$. *Default*: `0.01 eV/Angstrom`
`force_thr_ev2`	*Type*: Real *Description*: This may be a very useful parameter in localized orbitals set. Due to numerical issue, the force calculation using atomic orbitals is not accurate sometimes. When the force of an atom is smaller than $force\_thr\_ev2$, we set this force to be 0 and don't move the atom. This makes the geometry relaxation more stable. *Default*: `0`

`bfgs_w1`	*Type*: Real *Description*: This variable controls the Wolfe condition for BFGS algorithm used in geometry relaxation. You can look into paper $Phys.Chem.Chem.Phys.$, $2000, 2, 2177$ for more information. *Default*: `0.01`
`bfgs_w2`	*Type*: Real *Description*: This variable controls the Wolfe condition for BFGS algorithm used in geometry relaxation. You can look into paper $Phys.Chem.Chem.Phys.$, $2000, 2, 2177$ for more information. *Default*: `0.5`
`trust_radius_max`	*Type*: Real *Description*: This variable is for geometry optimization. It indicates the maximal movement of all the atoms. The sum of the movements from all atoms can be increased during the optimization steps. However, it will not be larger than *trust_radius_max* Bohr. *Default*: `0.8`
`trust_radius_min`	*Type*: Real *Description*: This variable is for geometry optimization. It indicates the minimal movement of all the atoms. When the movement of all the atoms is smaller than *trust_radius_min* Bohr , and the force convergence is still not achieved, the calculation will break down. *Default*: `1e-5`
`trust_radius_ini`	*Type*: Real *Description*: This variable is for geometry optimization. It indicates the initial movement of all the atoms. The sum of the movements from all atoms is *trust_radius_ini* Bohr. *Default*: `0.2`
`move_method`	*Type*: String *Description*: The method to do geometry optimizations. If set to $bfgs$, using BFGS algorithm. If set to *cg*, using CG algorithm. If set to *sd*, using steepest-descent algorithm. *Default*: `cg`

This part of variables are used to control the molecular dynamics calculations.

MD	**Description**
`md_mdtype`	*Type*: Integer *Description*: control the ensemble to run md. • `0`: When set to 0, ABACUS will use NVE ensemble; • `1`: When set to 1, ABACUS will use NVT ensemble with Nose Hoover method; • `2`: When set to 2, ABACUS will use NVT ensemble with Velosity Scaling method; *Default*: `1`

`md_rstmd`	*Type*: Bool *Description*:to control whether restart md. • `0`:When set to 0, ABACUS will calculate md normolly. • `1`:When set to 1, ABACUS will calculate md from last step in your test before. *Default*: `0`
`md_dt`	*Type*: Double *Description*:This is the time step(fs) used in md simulation . *Default*: `No default`
`md_tfirst&md_tlast`	*Type*: Double *Description*:This is the temperature used in md simulation, md_tlast's default value is md_tfirst. If md_tlast is setted and be different from the md_tfirst, ABACUS will automatically generate a linear temperature gradient file named "ChangeTemp.dat", you can also set this file according to your needs instead. *Default*: `No default`
`md_qmass`	*Type*: Double *Description*:Inertia of extended system variable. Used only when md_type is 1 or 2, you should set a number which is larger than 0. If you want to autoset this by ABACUS,just set it to 0. *Default*: `0`
`md_nresn&md_nyosh`	*Type*: Integer *Description*:Used when md_type is 1 or 2, control the Nose-Hoover thermostat extended-system, you can only set them at 1,3,5. *Default*: `md_nresn=md_nyosh=3`
`md_dumpmdfred`	*Type*: Integer *Description*:This is the steps to control the frequence to output md information *Default*: `1`
`md_domsd`	*Type*: Integer *Description*:when set to 1, ABACUS will calculate mean square displacement and the diffusion of each element. *Default*: `1`
`md_fixtemperature`	*Type*: Integer *Description*: • `n`:when set to n($n > 1$), ABACUS will read the file "ChangeTemp.dat" and change system's temperature every n steps, • `0,1`:When set to 0 or 1, ABACUS won't change the temperature during running MD. *Default*: `1`
`md_msdstarttime`	*Type*: Integer *Description*:when set to n, ABACUS will calculate mean square displacement and the diffusion of each element from nth step. *Default*: `1`

This part of variables are used to control DFT-D2 related parameters.

Name	**Description**
`vdwD2`	*Type*: Integer *Description*: If set to 1, ABACUS will calculate DFT-D2 correction, including vdW energy and vdW forces. *Default*: `0`
`vdwD2_scaling`	*Type*: Real *Description*: The scale factors are to optimize the interaction energy deviations. They are found to be 0.75 (PBE), 1.2 (BLYP), 1.05 (B-P86), 1.0 (TPSS), and 1.05 (B3LYP). *Default*: `0.75`
`vdwD2_d`	*Type*: Real *Description*: The variable is to control the dumping speed of dumping function of DFT-D2. *Default*: `20`
`vdwD2_C6_file`	*Type*: String *Description*: This variable specifies the name of each elemetent's C_6 Parameters file. If you don't setup this, ABACUS will use the default C_6 Parameters stored in the programme already. We've stored elements from 1_H to 54_Xe. Otherwise, if you want to use some new C_6 Parameters, you should provide a file contained all the C_6 Parameters ordered by periodic table of elements, from 1_H to the last elements you want. *Default*: `default`
`vdwD2_C6_unit`	*Type*: String *Description*: This variable specifies unit of C_6 Parameters. Two kinds of unit is available: `Jnm6/mol`(means $J \cdot nm^6/mol$) and `eVA`(means $eV \cdot \AA^6$) *Default*: `Jnm6/mol`
`vdwD2_R0_file`	*Type*: String *Description*: This variable specifies the name of each elemetent's R_0 Parameters file. If you don't setup this, ABACUS will use the default R_0 Parameters stored in the programme already. We've stored elements from 1_H to 54_Xe. Otherwise, if you want to use some new R_0 Parameters, you should provide a file contained all the R_0 Parameters ordered by periodic table of elements, from 1_H to the last elements you want. *Default*: `default`
`vdwD2_R0_unit`	*Type*: String *Description*: This variable specifies unit of R_0 Parameters. Two kinds of unit is available: `A`(means $\AA$) and `Bohr`. *Default*: `A`
`vdwD2_model`	*Type*: String *Description*: To calculate the periodic structure, you can assign the number of lattice cells calculated. This variable specifies the kind of model to assign. If set to `period`, ABACUS will calculate a cubic periodic structure as assigned in *vdwD2_period*. If set to `radius`, ABACUS will calculate a cubic periodic structure containing a sphere, whose radius is *vdwD2_radius* and centre is origin point. *Default*: `radius`

`vdwD2_radius`	*Type*: Real *Description*: If *vdwD2_model* is set to `radius`, this variable specifies the radius of the calculated sphere. *Default*: `56.6918`
`vdwD2_radius_unit`	*Type*: String *Description*: If *vdwD2_model* is set to `radius`, this variable specifies the unit of *vdwD2_radius*. Two kinds of unit is available: `A`(means $\mathring{A}$) and `Bohr`. *Default*: `Bohr`
`vdwD2_period`	*Type*: Int Int Int *Description*: If *vdwD2_model* is set to `period`, these variables specify the number of x, y and z periodic. *Default*: `2 2 2`

Bibliography

[1] M. Chen, G.-C. Guo, and L. He, “Systematically improvable optimized atomic basis sets for ab initio calculations,” *J. Phys.: Condens. Matter*, vol. 22, p. 445501, 2010.

[2] P. Hohenberg and W. Kohn *Phys. Rev.*, vol. 136, p. B864, 1964.

[3] W. Kohn and L. J. Sham *Phys. Rev.*, vol. 140, p. A1133, 1965.

[4] H. J. Monkhorst and J. D. Pack, “Special points for brillouin zone integration,” *Phys. Rev. B*, vol. 13, p. 5188, 1976.

[5] W. K. Press, S. A. Teukolsky, W. T. Vetterlin, and B. T. Flannery, *Numerical Recipes, 3rd edition.* Cambridge: Cambridge University Press, 2007.

[6] S. Grimme, “Semiempirical gga-type density functional constructed with a long-range dispersion correction,” *J. Comput. Chem.*, vol. 27, p. 1787, 2006.

[7] S. Nosé, “A unified formulation of the constant temperature molecular-dynamics methods,” *J. Chem. Phys.*, vol. 81, p. 511, 1984.

[8] High Performance Symmetric Eigenproblem Solvers Package (HPSESP) is a library developed by Yonghua Zhao’s group in the internet center of Chinese Academy of Sciences.