User's Guide for ABACUS 2.1.0

The ABACUS team

December 10, 2020

Contents

Chapter 1

Introduction

ABACUS (Atomic-orbital Based Ab-initio Computation at UStc) is an open-source computer code package, aiming at performing large-scale, first-principles electronic-structure calculations and molecular dynamics simulations. The ABACUS package was initially developed at the CAS Key Laboratory of Quantum Information, University of Science and Technology of China (USTC), and its recent development also reflects a significant contribution from the High Performance Computing Center at 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. These basis functions are generated using an optimization scheme developed by Chen, Guo, and He (CGH) [1] in 2010, and can be grouped into hierarchical basis sets of increasing size, via which the computational accuracy can be systematically improved. In addition to atomic basis sets, ABACUS also allows to use plane-wave basis set as an alternative option. This dual basis-set feature offers a convenient consistency and accuracy check within ABACUS.

The current released version of ABACUS is 2.1.0. ABACUS 2.1.0 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 using both conjugated gradient (CG) and Broyden–Fletcher–Goldfarb–Shanno (BFGS) [5] algorithms.
- Semi-empirical van der Waals energy correction using the Grimme DFT-D methods, including D2 [6], D3(0) [12] and D3(BJ) [13].
- NVT molecular dynamics simulation using the Nosé-Hoover thermostat [7, 8].
- Stress calculation and cell relaxation.
- Electric polarization calculation using Berry Phase theory.
- Interface to the Wannier90 package [9, 10].
- Real-time time dependent density functional theory (TDDFT)
- Electrostatic potential
- Mulliken charge

- Projected density of states (PDOS)

Compared to ABACUS 2.0.0, the last four items, namely, TDDFT, electrostatic potential, Mulliken charge, and PDOS are newly added features in ABACUS 2.1.0. Other functionalities including hybrid density functionals and spin-orbital coupling are under active development, and will be available in future release.

ABACUS can run both on desktop computers and high performance supercomputers under the unix/linux-based environment. The code is massively parallelized, scaling up to O(10^3) CPU cores.

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

> [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. X. He, Large-scale ab initio simulations based on systematically improvable atomic basis, Comput. Mater. Sci.,**112**, 503-517 (2016).

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, users 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 `2.1.0`.

2.2 Structure of the package

After obtaining the package, untar it and the entire package will be extracted into a directory called `ABACUS`.

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

Under the ABACUS directory, there are the following subdirectories,

- `bin/`
  where the executable program resides;
- `examples/`
  which contains test examples;
- `source/`
  which contains the source code and makefiles;
- `tools/`
  which currently contains the script for generating the numerical atomic orbitals;

The source directory further contains the following folders, where the source files of ABACUS are located

- `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, users should make sure that the following prerequisites are present,

1. C++ compiler, supporting C++11. For example, Intel C++ compiler (https://software.intel.com/en-us/c-compilers) or GCC(https://gcc.gnu.org/);
2. Fortran compiler.
3. MPI compiler. The recommended version are Intel MPI (https://software.intel.com/en-us/mpi-library) or MPICH (https://www.mpich.org/);
4. Boost C++ library (https://www.boost.org/);
5. The ScaLAPACK library. For example, Intel MKL (https://software.intel.com/en-us/mkl) or Netlib ScaLAPACK(http://www.netlib.org/scalapack/);
6. The FFTW library (http://www.fftw.org/). ABACUS now supports both FFTW2 and FFTW3.
7. The ELPA library (https://elpa.mpcdf.mpg.de/)

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.vars using any editor tools you like, e.g., `vi`

```
vi Makefile.vars
```

Specify the location of the compiler and libraries present in your own machine

```
CPLUSPLUS =
CPLUSPLUS_MPI =
FORTRAN =
LAPACK_DIR =
FFTW3_DIR =
```

```
BOOST_DIR =
ELPA_DIR =
```

If there are Inel C++ compiler, Intel MPI and Intel MKL, file Makefile.vars can be set as follows:

```
CPLUSPLUS = icpc
CPLUSPLUS_MPI = mpiicpc
FORTRAN = ifort
LAPACK_DIR = /opt/intel/.../mkl/lib/intel64/
FFTW3_DIR = /opt/fftw-3.3.8/
BOOST_DIR = /opt/boost/1.64.0/
ELPA_DIR = /opt/elpa/2016.05.004/
```

Then, 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.

If there are just GCC, MPICH and ScaLAPACK, set file Makefile.vars like:

```
CPLUSPLUS = g++
CPLUSPLUS_MPI = mpicxx
FORTRAN = gfortran
SCALAPACK_DIR = /opt/scalapack/
FFTW3_DIR = /opt/fftw-3.3.8/
BOOST_DIR = /opt/boost/1.64.0/
ELPA_DIR = /opt/elpa/2016.05.004/
```

Next, set parameter LIBS in Makefile.system like:

```
LIBS = \
-lgfortran -lm \
-openmp -lpthread \
${SCALAPACK_DIR}/lib/libscalapack.a \
/opt/lapack/lib/liblapack.a \
```

```
/opt/blas/lib/libblas.a \
/opt/blacs/lib/libblacs.a \
${FFTW_LIB} \
${ELPA_LIB}
```

Finally, type

```
make fp_mpi
```

In version 2.1.0, the default option of FFTW libraries is FFTW3. If one wants to use FFTW2, please set the parameters in file Makefile.system

```
HONG_FFTW = -D__FFTW2
FFTW_LIB = -L${FFTW_LIB_DIR} -lfftw -Wl,-rpath=${FFTW_LIB_DIR}
```

Chapter 3

Quickstart Quide

The following files are the central input files for ABACUS. Before executing the program, please make sure these files are prepared and stored in the working 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 either in 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 of each element's pseudopotential needs to be specified in the STRU file, if the the pseudopotential files are already present in the working directory. However, in case that the pseudopotential files are stored in some other directories, then a full path to access the pseudopotential files have 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 for each element's numerical orbital basis needs to be specified in the STRU file. However, in case that the numerical orbital files are stored in a location different from the working directory, then a full path to access the orbital files have to be specified in the STRU file.

3.1 Input files

3.1.1 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         60
```

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. Note: if a parameter 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: Rydberg)

For more information about the input parameters, see Appendix 5.1

3.1.2 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 28.00 Si_ONCV_PBE-1.0.upf  // label; mass; pseudo_file

NUMERICAL_ORBITAL
Si_gga_8au_60Ry_2s2p1d.orb  //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.'Si_ONCV_PBE-1.0.upf' is the pseudopotential file. When the path is not specified, the file is assumed to be located in work dierectory. Otherwise, please explicitly specify the location of the pseudopotential files.

2. `NUMERICAL_ORBITAL` Numerical atomic orbitals are only needed for LCAO calculations. Thus this section will be neglected in calcultions with plane wave basis. 'Si_gga_8au_60Ry_2s2p1d.orb' 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 given 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 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 Pseudopotential file

As mentioned above, 'Si_ONCV_PBE-1.0.upf' is a pseudopotential file, which contains the norm-conserving pseudopotential of Si element. The file is in UPF file format and begins like

```
<UPF version="2.0.1">
  <PP_INFO>

 This pseudopotential file has been produced using the code
 ONCVPSP  (Optimized Norm-Conservinng Vanderbilt PSeudopotential)
 scalar-relativistic version 2.1.1, 03/26/2014 by D. R. Hamann
 The code is available through a link at URL www.mat-simresearch.com.
 Documentation with the package provides a full discription of the
 input data below.


 While it is not required under the terms of the GNU GPL, it is
 suggested that you cite D. R. Hamann, Phys. Rev. B 88, 085117 (2013)
 in any publication using these pseudopotentials.
...
```

In ABACUS there are pseudopotentials of most elements commonly used. Users can download these pseudopotentials from website:

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

3.1.5 Numerical orbital file

'Si_gga_8au_60Ry_2s2p1d.orb' is a numerical orbital file of the Si element. The name indicates that this particular atomic basis set is generated using the pseudopotential with GGA functional, with an energy cutoff of 60 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)              60
Radius Cutoff(a.u.)            8
Lmax                           2
Number of Sorbital-->          2
Number of Porbital-->          2
Number of Dorbital-->          1
---------------------------------------------------------------------------
SUMMARY  END
...
```

ABACUS provides atomic basis sets of different accuracy levels for most elements commonly used. Users can download these basis sets from website:

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

Moreover, 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_ONCV_PBE-1.0.upf also to this directory.

```
cp $ABACUS_DIR/examples/01-Si2_diamond-scf/Si_ONCV_PBE-1.0.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'                 *
*                                                       *
*          Website: http://abacus.ustc.edu.cn/          *
*                                                       *
*********************************************************
Tue May 21 15:56:25 2019
MAKE THE DIR         : OUT.ABACUS/
DONE(0.0248992  SEC) : SETUP UNITCELL
DONE(0.0268722  SEC) : INIT K-POINTS
---------------------------------------------------------
This calculation is self-consistent
---------------------------------------------------------
SPIN    KPOINTS         PROCESSORS
1       64              1
---------------------------------------------------------
Use plane wave basis
---------------------------------------------------------
ELEMENT NATOM       XC
Si      2           PBE
---------------------------------------------------------
Initial plane wave basis and FFT box
---------------------------------------------------------
DONE(0.172609   SEC) : INIT PLANEWAVE
UNIFORM GRID DIM     : 36 * 36 * 36
UNIFORM GRID DIM(BIG): 36 * 36 * 36
MEMORY FOR PSI (MB)  : 8.28125
DONE(0.191006   SEC) : LOCAL POTENTIAL
DONE(0.229162   SEC) : NON-LOCAL POTENTIAL
START POTENTIAL      : atomic
DONE(0.314842   SEC) : INIT POTENTIAL
DONE(0.87122    SEC) : INIT BASIS
-------------------------------------------
SELF-CONSISTENT :
-------------------------------------------
ITER   ETOT(eV)       EDIFF(eV)      DRHO2      CG_ITER    TIME(S)
CG1    -2.172664e+02  0.000000e+00   6.055e-02  7.156e+00  3.510e+00
CG2    -2.142577e+02  3.008676e+00   3.512e-03  3.102e+00  1.940e+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 an 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               60
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/examples/01-Si2_diamond-scf/STRU ./
cp $ABACUS_DIR/examples/01-Si2_diamond-scf/Si_ONCV_PBE-1.0.upf ./
```

```
cp $ABACUS_DIR/examples/01-Si2_diamond-scf/Si_gga_8au_60Ry_2s2p1d.orb ./
```

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           genelpa
```

- `ks_solver` The method to solve the Kohn-Sham equation. For plane wave basis, default value is `cg`, the conjugate gradient method. For lcao basis, default value is genelpa, and another option is hpseps, which means using a High Performance Symmetric Eigenproblem Solvers package (HPSESP) [11].

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,

```
*********************************************************
*                                                       *
*                  WELCOME TO ABACUS                    *
*                                                       *
*            'Atomic-orbital Based Ab-initio            *
*                  Computation at UStc'                 *
*                                                       *
*          Website: http://abacus.ustc.edu.cn/          *
*                                                       *
*********************************************************
Tue May 21 16:24:09 2019
MAKE THE DIR         : OUT.ABACUS/
DONE(0.0244381  SEC) : SETUP UNITCELL
DONE(0.0262482  SEC) : INIT K-POINTS
Set NonLocal Pseudopotential Projectors
---------------------------------------------------------
This calculation is self-consistent
---------------------------------------------------------
SPIN    KPOINTS         PROCESSORS  NBASE       VNA
1       64              1           26          No
---------------------------------------------------------
Use Systematically Improvable Atomic bases
---------------------------------------------------------
ELEMENT ORBITALS        NBASE       NATOM       XC
Si      2s2p1d-8au      13          2           PBE
---------------------------------------------------------
```

```
Initial plane wave basis and FFT box
---------------------------------------------------------
DONE(1.67491    SEC) : INIT PLANEWAVE
UNIFORM GRID DIM     : 36 * 36 * 36
UNIFORM GRID DIM(BIG): 18 * 18 * 18
DONE(1.67712    SEC) : INIT CHARGE
DONE(1.67824    SEC) : INIT POTENTIAL
START POTENTIAL      : atomic
---------------------------------------------------------
SELF-CONSISTENT :
---------------------------------------------------------
ITER   ETOT(eV)       EDIFF(eV)      DRHO2      TIME(s)
GE1    -2.139898e+02  0.000000e+00   1.643e-01  1.361e+01
GE2    -2.140153e+02  -2.550386e-02  3.718e-02  1.295e+01
...
```

The string `GE`n in the first column means that the genelpa 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 expected to be 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 is 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 basis are generated with a tool called SIAB (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 `tools/` 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.fp_mpi.x
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   (L<=2)
Level        2    # num of levels to generate orbitals(<=5)
#(num)    (the max ang)   (num of S)  (num of P)  (num of D)
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 calculations:
   - `targets` the element type of which orbitals are to be generated. Its value has 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 (L<=2)
level   2    # num of levels to generate orbitals(<=5)
(num)  (the max ang) (num of S)  (num of P)  (num of D)
level1  1              1           1
level2  2              1           1         1
```

   Because of the property of symmetry, taking dimer or trimer as reference systems can not generate the f orbitals very well. And test results show that f orbital has little effect on improving the results. 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. The results will be written into several output files under the directory $element.id element/$Rcut/, for example 07_N/6/.

Some output files listed 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 softwares 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. To run a calculation, the pseudopotential needs to be set in `STRU` file. For example

```
ATOMIC_SPECIES
Si 28.00 Si_ONCV_PBE-1.0.upf
...
```

The user can download the pseudopotential files 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 the so-called 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’ , ‘genelpa’ 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 either be 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 GGA pseudopotential file Si_ONCV_PBE-1.0.upf

```
...
    <PP_HEADER
       generated="Generated using ONCVPSP code by D. R. Hamann"
       author="Martin Schlipf and Francois Gygi"
       date="150105"
       comment=""
       element="Si"
       pseudo_type="NC"
       relativistic="scalar"
       is_ultrasoft="F"
       is_paw="F"
       is_coulomb="F"
       has_so="F"
       has_wfc="F"
       has_gipaw="F"
       core_correction="F"
       functional="PBE"
       z_valence="    4.00"
       total_psenergy="  -3.74274958433E+00"
       rho_cutoff="   6.01000000000E+00"
...
```

The keyword pseudo_type="NC" indicates that the norm-conserving pseudopotential is used here. The keyword functional="PBE" shows that it uses the Perdew-Burke-Ernzerhof GGA 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

The user can set the XC functional type in INPUT file with the parameter,
`dft_functional:`

- 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 three Grimme DFT-D methods, including D2 [6], D3(0) [12] and D3(BJ) [13], to describe van der Waals interactions. Among them, the D3 method has been implemented in ABACUS based on the dftd3 program written by Stefan Grimme, Stephan Ehrlich and Helge Krieg (`www.chemie.uni-bonn.de/pctc/mulliken-center/software/dft-d3`).

The user can set the vdW-corrected methods in INPUT file with the parameter,
`vdw_method:`

- none: the vdW corrections are not included
- d2: DFT-D2 method
- d3_0: DFT-D3(0) method
- d3_bj: DFT-D3(BJ) method

For example, if you want to run a DFT-D2 calculation, you should add the following keyword to the INPUT file.

```
vdw_method                  d2
```

By this way, ABACUS will calculate total energies and forces with DFT-D2 correction. 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                 60
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 diagonalizes 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 `DOS1_smearing.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             60
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 `KPT` 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 the atomic forces for a given structure without ion relaxation, set '`calculation`' to '`scf`', set input parameter '`force`' to `1`.

```
calculation         scf
force               1
```

To relax the atom position without change cell shape, one needs to add a few more parameters in the `INPUT` file. Here is an example for the Si dimer. In this case, the forces are calculated by default.

```
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
```

- `calculation` relax, relax atom positions with fixed lattice vectors.
- `nstep` the maximal 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 Stress calculation and cell relaxation

To calculate the stress for a given structure, set input parameter '`stress`' to `1`. The stress can be calculated in all types of '`calculation`'.

```
stress                  1
```

To optimize the cell shape, one needs to setup a few parameters in the `INPUT` file. First, the parameter '`calculation`' should be set to '`cell-relax`'. The ion relaxation is automatically done in the cell shape optimization. To optimize the cell shape under external pressures, setup '`press1`, `press2`, `press3`' which are the external pressures along the a, b, c axes in KBar. The default values is zero. You may optimize the cell shape with some of axes are fixed. To do so, use '`fixed_axes`' as described below. The default value is `None`. The parameter 'stress_thr' is the threshold for stress.

Here is an example for the Si crystal.

```
suffix                  Si
calculation             cell-relax
nstep                   100
force_thr_ev            0.01
move_method             cg
out_stru                1
trust_radius_ini        0.5
press1                  0
press2                  0
press3                  0
fixed_axes              None
stress_thr              1
```

- `press1,2,3` the external pressures along three axes in KBar, the compressive stress is taken to be positive.
- `fixed_axes` which axes are fixed when do cell relaxation. Possible choices are:
  - `None` : default; all can relax
  - `volume` : relaxation with fixed volume
  - `a` : fix a axis during relaxation
  - `b` : fix b axis during relaxation
  - `c` : fix c axis during relaxation
  - `ab` : fix both a and b axes during relaxation
  - `ac` : fix both a and c axes during relaxation
  - `bc` : fix both b and c axes during relaxation
  - `abc` : fix all three axes during relaxation
- `stress_thr` the threshold for stress, below which the cell relaxation is considered to be converged. The unit is KBar. The default threshold is 10 KBar.

4.9 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 added.

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

#Parameters (Accuracy)
ecutwfc             60
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               1
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 means 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
- `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 is steady.

4.10 Macroscopic polarization calculation

From version 2.0.0, ABACUS is capable of calculating macroscopic polarization of insulators by using the Berry phase method, known as the “modern theory of polarization” [14]. To calculate the polarization, you need first to do a self-consistent calculation as described in 4.6.1 to get the converged charge density. Then, do a non-self-consistent calculation with `berry_phase` setting to 1. You need also to specify the direction of the polarization you want to calculate. As an example, we give below the INPUT file (nscf) and KPT file (nscf) to calculate the electric polarization along c axis for $PbTiO_3$:

```
INPUT_PARAMETERS
suffix              PbTiO3
ntype               3
nbands              25          // number of bands
calculation         nscf        // non-self-consistent calculation
ecutwfc             50          // Ry
symmetry            0           // turn off symmetry
dr2                 1.0e-8
smearing            gaussian    // gaussian smearing
sigma               0.002       // Ry
basis_type          lcao        // atomic basis
ks_solver           genelpa
start_charge        file        // read charge from files

#Parameters(berryphase)
berry_phase         1           // calculate Berry phase
gdir                3           // calculate polarization along c axis
```

Note: You need to turn off the symmetry when do Berry phase calculations. Currently, ABACUS support Berry phase calculation with nspin=1 and nspin=2. The Berry phase can be calculated in both `pw` and `lcao` bases.

- `berry_phase` : 1, calculate berry phase; 0, no calculate berry phase.
- `gdir` : 1, 2, 3, the lattice vector direction of the polarization you want to calculate.

The KPT file need to be modified according to `gdir` in the INPUT file. Generally, you need denser k points along this direction. For example, in the following KPT file, 4 k-points are taken along the a and b axes, and 8 k-points are taken along the c-axis. You should check the convergence of the k points when calculating the polarization.

```
K_POINTS
0
Gamma
4 4 8 0 0 0
```

The results of the berry phase calculation are written in the "running_nscf.log" in the OUT folder. You may these results by searching for keywords "POLARIZATION CALCULATION". The results are shown as follows:

```
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
|                                                                    |
| POLARIZATION CALCULATION:                                          |
|                  Modern Theory of Polarization                     |
| calculate the Macroscopic polarization of a crystalline insulator  |
| by using Berry Phase method.                                       |
|                                                                    |
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

VALUES OF POLARIZATION

 The Ionic Phase:   -0.10600
Electronic Phase:    0.92355

The calculated polarization direction is in R3 direction

P=7.3956703 (mod 18.0922373)( 0.0000000, 0.0000000, 7.3956703)(e/Omega).bohr

P=0.0155501 (mod  0.0380407)( 0.0000000, 0.0000000, 0.0155501)e/bohr^2

P=0.8890277 (mod  2.1748536)( 0.0000000, 0.0000000, 0.8890277)C/m^2
```

The electric polarization **P** is multivalued, which modulo a quantum $e\mathbf{R}/V_{cell}$. Note: the values in parentheses are the components of the **P** along the c axis in the x, y, z Cartesian coordinates when set gdir = 3 in INPUT file.

4.11 ABACUS-wannier90 interface

Wannier90 (`http://www.wannier.org`) is a useful package to calculate the maximally-localized Wannier functions (MLWFs) [15] and related quantities. ABACUS provides an interface to Wannier90 package. The users are assumed to be familiar with the use of Wannier90. To construct the MLWFs using the wave functions of ABACUS generally requires four steps. Here we use the diamond as an example. First, you need prepare a Wannier90 input file, e.g., "diamond.win", which is the master input file for Wannier90. It contains the specification of the system and any parameters for the run. You should run "wannier90 -pp diamond.win" to generate "diamond.nnkp" file which will be read in by ABACUS later.

```
num_wann        =  4
num_iter        = 20
wvfn_formatted  = .true.

begin atoms_frac
C   -0.12500  -0.1250    -0.125000
C    0.12500   0.1250     0.125000
end atoms_frac

begin projections
f=0.0,0.0,0.0:s
f=0.0,0.0,0.5:s
f=0.0,0.5,0.0:s
f=0.5,0.0,0.0:s
end projections

begin unit_cell_cart
-1.613990   0.000000   1.613990
 0.000000   1.613990   1.613990
-1.613990   1.613990   0.000000
end unit_cell_cart

mp_grid : 4 4 4

begin kpoints
0.0000  0.0000     0.0000
0.0000  0.2500     0.0000
...
end kpoints
```

Second, do a self-consistent calculation as described in Sec. 4.6.1 and get the converged charge density. Third, do a non-self-consistent calculation. You should setup the ABACUS INPUT parameters and KPT file according to the "diamond.win" file. The ABACUS INPUT file and KPT file for the nscf calculation for the fifth example (diamond) in the Wannier90 package are given below.

```
INPUT_PARAMETERS

ntype                    1
ecutwfc                  40
nbands                   4              ! number of bands
calculation              nscf           ! non-self-consistent calculation
```

```
niter                    40
ethr                     1.0e-8
dr2                      1.0e-8
start_charge             file          ! read charge from files

#Parameters(wannier90)
towannier90              1             ! use the ABACUS-wannier90 interface
nnkpfile                 diamond.nnkp  ! specify .nnkp file
```

```


K_POINTS
64
Direct
0.0000  0.0000   0.0000   0.0156250
0.0000  0.2500   0.0000   0.0156250
0.0000  0.5000   0.0000   0.0156250
0.0000  0.7500   0.0000   0.0156250
0.2500  0.0000   0.0000   0.0156250
0.2500  0.2500   0.0000   0.0156250
0.2500  0.5000   0.0000   0.0156250
0.2500  0.7500   0.0000   0.0156250
0.5000  0.0000   0.0000   0.0156250
0.5000  0.2500   0.0000   0.0156250
0.5000  0.5000   0.0000   0.0156250
0.5000  0.7500   0.0000   0.0156250
...
```

Here, we have two parameters:

- `towannier90` : 1, generate files for wannier90 code; 0, do not generate.
- `nnkpfile` : the name of the .nnkp file

If you use nspin=2, there is one additional parameter in the INPUT:

- `wannier_spin` : up, calculate the Wannier functions for the spin up components ; down, calculate the Wannier functions spin down components.

After the nscf calculation, ABACUS will generate “diamond.amn”, “diamond.mmn”, “diamond.eig”, “UNK*” files in the OUT.* folder. Finally, you should run “wannier90 diamond.win”, to get the MLWFs.

Notes:

- The ABACUS-wannier90 interface can be used in both `pw` and `lcao` basis.
- You must set “wvfn_formatted = .true.” in “diamond.win”, otherwise Wannier90 code cannot read files generated by ABACUS because these files are not binary files.

4.12 Real-time time dependent density functional theory

Real-time time dependent density functional theory (rt-TDDFT) approaches, directly provide time domain evolution of electronic wave functions together with ionic movements, presenting a versatile way for the real time tracking of ultrafast dynamics and phenomena either in perturbative or non-perturbative regime.

To run Real-time TDDFT (Time-Dependent DFT) 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 TDDFT simulation is given below, which contains additional keywords that need to be added.

```
INPUT_PARAMETERS
#Parameters (General)
pseudo_dir ./
ntype         1
nbands 12
calculation md

#Parameter (Accuracy)
ecutwfc 50
niter 30

smearing gaussian
sigma 0.02
basis_type lcao
out_charge 1
gamma_only 0
nstep 2000
dr2             1.0e-6

md_mdtype 0
md_dt 0.01
md_tfirst           30
md_mdoutpath       mdtest
md_rstmd            0
md_fixtemperature   20000
md_nresn            3
md_nyosh            3
md_qmass            0
md_tlast            30
md_dumpmdfred       1
md_domsd            0
md_domsdatom        0
md_outputstressperiod    0

tddft 1
val_elec_01 5
vext 1
vext_dire 2
```

Note: The TDDFT simulation is based on molecular dynamics in ABACUS. Accomplished with these MD parameters, TDDFT needs some new key words. The TDDFT simulation can be calculated only in both `lcao` bases.

- `tddft` : 1, calculate TDDFT; 0, no calculate TDDFT.
- `vext` : 1, a laser material interaction is included; 0, no extern laser field.
- `vext_dire` : 1, 2, 3, the direction of the extern laser field added x, y, z.

The results of the TDDFT calculation are dependent on what you want. Dynamics, High-order harmonic generation and occupation could be achieved when you do TDDFT calculation. Users can also do some post-processing according to the basic TDDFT calculation.

4.13 Electrostatic potential

From version 2.1.0, ABACUS has the function of outputing electrostatic potential, which consists of Hartree potential and the local pseudopotential. To use this function, set '`out_potential`' to '2' in the `INPUT` file. Here is an example for the Si-111 surface, and the `INPUT` file is

```
INPUT_PARAMETERS
#Parameters (1.General)
calculation     scf
ntype           1
nbands          100
gamma_only      0

#Parameters (2.Iteration)
ecutwfc         50
dr2             1e-8
niter           200

#Parameters (3.Basis)
basis_type      lcao
ks_solver       genelpa

#Parameters (4.Smearing)
smearing        gaussian
sigma           0.01

#Parameters (5.Mixing)
mixing_type     pulay
mixing_beta     0.4

out_potential   2
```

the `STRU` file is

```
ATOMIC_SPECIES
Si 1.000 Si_ONCV_PBE-1.0.upf

NUMERICAL_ORBITAL
Si_gga_8au_60Ry_2s2p1d.orb

LATTICE_CONSTANT
1.8897162

LATTICE_VECTORS
7.6800298691  0.0000000000 0.0000000000
-3.8400149345 6.6511009684 0.0000000000
0.0000000000  0.0000000000 65.6767997742

ATOMIC_POSITIONS
Cartesian
Si
0.0
40
3.840018749  2.217031479 2.351520061 0 0 0
3.840014935  0.000000000 3.135360003 0 0 0
3.840018749  2.217031479 5.486879826 0 0 0
3.840014935  0.000000000 6.270720005 0 0 0
3.840018749  2.217031479 8.622240067 0 0 0
3.840014935  0.000000000 9.406080246 0 0 0
3.840018749  2.217031479 11.757599831 0 0 0
3.840014935  0.000000000 12.541440010 0 0 0
3.840018749  2.217031479 14.892959595 0 0 0
3.840014935  0.000000000 0.000000000 0 0 0
1.920011044  5.542582035 2.351520061 0 0 0
1.920007467  3.325550556 3.135360003 0 0 0
1.920011044  5.542582035 5.486879826 0 0 0
1.920007467  3.325550556 6.270720005 0 0 0
1.920011044  5.542582035 8.622240067 0 0 0
1.920007467  3.325550556 9.406080246 0 0 0
1.920011044  5.542582035 11.757599831 0 0 0
1.920007467  3.325550556 12.541440010 0 0 0
1.920011044  5.542582035 14.892959595 0 0 0
1.920007467  3.325550556 0.000000000 0 0 0
0.000003815  2.217031479 2.351520061 0 0 0
0.000000000  0.000000000 3.135360003 0 0 0
0.000003815  2.217031479 5.486879826 0 0 0
0.000000000  0.000000000 6.270720005 0 0 0
0.000003815  2.217031479 8.622240067 0 0 0
0.000000000  0.000000000 9.406080246 0 0 0
0.000003815  2.217031479 11.757599831 0 0 0
0.000000000  0.000000000 12.541440010 0 0 0
0.000003815  2.217031479 14.892959595 0 0 0
0.000000000  0.000000000  0.000000000 0 0 0
```

```
-1.920003772 5.542582035  2.351520061 0 0 0
-1.920007467 3.325550556  3.135360003 0 0 0
-1.920003772 5.542582035  5.486879826 0 0 0
-1.920007467 3.325550556  6.270720005 0 0 0
-1.920003772 5.542582035  8.622240067 0 0 0
-1.920007467 3.325550556  9.406080246 0 0 0
-1.920003772 5.542582035  11.757599831 0 0 0
-1.920007467 3.325550556  12.541440010 0 0 0
-1.920003772 5.542582035  14.892959595 0 0 0
-1.920007467 3.325550556   0.000000000 0 0 0
```

the `KPT` file is

```
K_POINTS
0
Gamma
4 4 2 0 0 0
```

Run the program, and you will see the following two files in the output directory,

- ElecStaticPot: contains electrostatic potential (unit: Rydberg) in realspace. This file can be visually viewed by the software of VESTA.
- ElecStaticPot_AVE: contains electrostatic potential (unit: Rydberg) along the z-axis (here z-axis is the default direction of vacuum layer) in realspace.

4.14 Mulliken charge

From version 2.1.0, ABACUS has the function of Mulliken population analysis. To use this function, set '`mulliken`' to '`1`' in the `INPUT` file. After calculation, there will be an output file named `mulliken.txt` in the output directory. In the file, there are contents like

```
 CALCULATE THE MULLIkEN ANALYSIS FOR EACH ATOM
160 (Total charge all spin 1)
160 (Total charge of the system)
  Decomposed Mulliken populations
0                Si          Up spin          Down spin      Sum        Diff
              multiple
s                0          0.69860591      0.69860591      1.3972118    0
  sum over m                0.69860591      0.69860591      1.3972118    0
s                1         -0.0050780298   -0.0050780298   -0.01015606   0
  sum over m               -0.0050780298   -0.0050780298   -0.01015606   0
  sum over m+mul            0.69352788      0.69352788      1.3870558    0
px               0          0.41584881      0.41584881      0.83169763   0
```

```
py                0          0.40471071      0.40471071      0.80942142   0
pz                0          0.40471061      0.40471061      0.80942123   0
  sum over m                 1.2252701       1.2252701       2.4505403    0
px                1         -0.0033434318   -0.0033434318   -0.0066868636 0
py                1         -0.0026261277   -0.0026261277   -0.0052522554 0
pz                1         -0.0026261162   -0.0026261162   -0.0052522323 0
  sum over m                -0.0085956757   -0.0085956757   -0.017191351  0
  sum over m+mul             1.2166745       1.2166745       2.4333489    0
d3z^2-r^2         0          0.017451998     0.017451998     0.034903996  0
dx^2-y^2          0          0.022845186     0.022845186     0.045690372  0
dxy               0          0.022845275     0.022845275     0.04569055   0
dxz               0          0.019370371     0.019370371     0.038740743  0
dyz               0          0.019370408     0.019370408     0.038740816  0
  sum over m                 0.10188324      0.10188324      0.20376648   0
  sum over m+mul             0.10188324      0.10188324      0.20376648   0
Total Charge on atom  Si     4.0241712

1                 Si          Up spin          Down spin      Sum        Diff
               multiple
s                 0          0.71418721      0.71418721      1.4283744    0
  sum over m                 0.71418721      0.71418721      1.4283744    0
s                 1         -0.010437756    -0.010437756    -0.020875513  0
  sum over m                -0.010437756    -0.010437756    -0.020875513  0
  sum over m+mul             0.70374946      0.70374946      1.4074989    0
...
```

The file gives Mulliken charge in turn according to the order of atoms in the system. For example, the following block is for the first atom in system,

```
0                 Si          Up spin          Down spin      Sum        Diff
...
Total Charge on atom  Si     4.0241712
```

And the next block is for the second atom in system, and so on.

```
1                 Si          Up spin          Down spin      Sum        Diff
...
```

For each atom, the file gives detailed Mulliken population analysis at different levels,

- magnetic quantum number level: such as lines beigin with ‘s,px,py,pz,...’
- azimuthal quantum number level: such as lines begin with ‘sum over m’.

- principal quantum number level: such as lines begin with 'sum over m+mul'. Here 'mul' equals 'multiple' in the file, which means how many radial atomic orbitals there are for a given orbital angular momentum.
- atomic level: such as lines begin with 'Total Charge on atom'.

4.15 Projected density of states (PDOS)

From version 2.1, ABACUS has the function of PDOS calculation. To use this function, set '`out_dos`' to '`1`' in the `INPUT` file. After calculation, there will be three kinds of output files in the output directory.

- TDOS: Total density of states.In the file, there are contents like

```
Energy(eV)                    Dos Spin Up                    Dos Spin Down
-4.62606                        8.05985                        8.05985
-4.61606                        7.89703                        7.89703
-4.60606                        7.42804                        7.42804
-4.59606                        6.70747                        6.70747
-4.58606                        5.81456                        5.81456
-4.57606                        4.83892                        4.83892
-4.56606                        3.86593                        3.86593
-4.55606                        2.96505                        2.96505
-4.54606                        2.18316                        2.18316
```

- PDOS: Projected density of states.In the file, there are contents like

```
<pdos>
<nspin>2</nspin>
<norbitals>26</norbitals>
<energy_values units="eV">
-4.62606
-4.61606
-4.60606
 .
 .
 .
</energy_values>
<orbital
index="                                      1"
atom_index="                                      1"
species="Si"
l="                                      0"
m="                                      0"
z="                                      1"
>
<data>
5.90206                       5.90206
```

```
5.78283                        5.78283
5.4394                         5.4394
.
.
.
</data>
</orbital>
</pdos>
```

- Written in XML format.The first part is the energy window.And the following is the projected density of states corresponding to every atomic orbital in the basis set.
- Orbital: Contains the order of the orbitals.In the file, there are contents like

```
io    spec    l    m    z   sym
1      Si    0    0    1    s
2      Si    0    0    2    s
3      Si    1    0    1    px
4      Si    1    1    1    py
5      Si    1    2    1    pz
6      Si    1    0    2    px
7      Si    1    1    2    py
8      Si    1    2    2    pz
9      Si    2    0    1    d3z^2-r^2
10     Si    2    1    1    dx^2-y^2
11     Si    2    2    1    dxy
12     Si    2    3    1    dxz
13     Si    2    4    1    dyz
14      C    0    0    1    s
15      C    0    0    2    s
16      C    1    0    1    px
17      C    1    1    1    py
18      C    1    2    1    pz
19      C    1    0    2    px
20      C    1    1    2    py
21      C    1    2    2    pz
22      C    2    0    1    d3z^2-r^2
23      C    2    1    1    dx^2-y^2
24      C    2    2    1    dxy
25      C    2    3    1    dxz
26      C    2    4    1    dyz


io =Orbital index in supercell
spec =Atomic species label
l =Angular mumentum quantum number
m =Magnetic quantum number
z =Zeta index of orbital
sym =Symmetry name of real orbital
```

- There also is a fortran script called pdos.f so that we can use this script to extract the projected density of states from the PDOS file.For example,if we want to extract the p orbital density of states of C, We can type the following command:

```
gfortran pdos.f
./a.out
Input file name (PDOS):
PDOS
Output file name :
Op
Extract data for atom index (enter atom NUMBER, or 0 to select all),
or for all atoms of given species (enter its chemical LABEL):
2
Extract data for l= ... (-1 for all l ):
1
Extract data for m= ... (9 for all m ):
9
```

- then there will be a file called Op in the output.

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 quit. *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 (nspin=1)*: `max(1.2*occupied_bands, occupied_bands+10)` *Default (nspin=2)*: `max(1.2*nelec, nelec+20)`
`nelec`	*Type*: Real *Description*: Input number of electrons. *Default*: `0.0`
`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, cell information, atom positions, 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 INPUT 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. • $cell - relax$: do cell relaxation calculation. • *nscf*: do the non self-consistent electronic structure calculations. 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_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 geometry relaxation 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. For plane-wave basis, • `cg`: *cg* method. • `david`: *david* is the Davidson algorithm. For atomic orbitals basis, • `genelpa`: This method should be used if you choose localized orbitals. • `hpseps`: old method, still used. • `lapack`: *lapack* can be used for localized orbitals, but is only used for single processor. *Default*: `cg (pw) or genelpa (lcao).`
`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 *ks_solver = cg* or *ks_solver = david*. It indicates the maximal iteration number for `cg/david` method. *Default*: `40`
`diago_david_ndim`	*Type*: Integer *Description*: Only useful when you use *ks_solver = 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-06`
`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_potential`	*Type*: Integer *Description*: If set to 1, ABACUS will output the local potential on real space grid. The name of the file is $SPIN1\_POT$ and $SPIN2\_POT$ (if $nspin = 2$). If set to 2, ABACUS will output the electrostatic potential on real space grid. The name of the file is ElecStaticPot and $ElecStaticPot\_AVE$ (along the z-axis). *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*: Used in pw basis, write wave functions into file WAVEFUNC.dat (set to 1) *Default*: `0`
`mulliken`	*Type*: Integer *Description*: If set to 1, ABACUS will output the Mulliken population analysis result. The name of the output file is mulliken.txt *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`
`force`	*Type*: Integer *Description*: If set to 1, calculate the force at the end of the electronic iteration. 0 means the force calculation is turned off. *Default*: `0`
`nstep`	*Type*: Integer *Description*: The maximal 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`

`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*: `0.5`
`stress`	*Type*: Integer *Description*: If set to 1, calculate the stress at the end of the electronic iteration. 0 means the stress calculation is turned off. *Default*: `0`
`stress_thr`	*Type*: Real *Description*: The threshold of the stress convergence, it indicates the largest stress among all the directions, the unit is $KBar$, *Default*: `10`
`press1, 2, 3`	*Type*: Real *Description*: the external pressures along three axes,the compressive stress is taken to be positive, the unit is $KBar$, *Default*: `0`

`fixed_axes`	*Type*: String *Description*:which axes are fixed when do cell relaxation. Possible choices are: • `None` : default; all can relax • `volume` : relaxation with fixed volume • `a` : fix a axis during relaxation • `b` : fix b axis during relaxation • `c` : fix c axis during relaxation • `ab` : fix both a and b axes during relaxation • `ac` : fix both a and c axes during relaxation • `bc` : fix both b and c axes during relaxation • `abc` : fix all three axes during relaxation *Default*: `None`
`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_type`	*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 vdW-corrected related parameters.

Name	**Description**

`vdw_method`	*Type*: String *Description*: If set to `d2` ,`d3_0` or `d3_bj`, ABACUS will calculate corresponding vdW correction, which is DFT-D2, DFT-D3(0) or DFT-D3(BJ) method. And this correction includes energy and forces. ”none” means that no vdW-corrected method has been used. *Default*: `none`
`vdw_s6`	*Type*: Real *Description*: This scale factor is to optimize the interaction energy deviations. For DFT-D2, it is found to be 0.75 (PBE), 1.2 (BLYP), 1.05 (B-P86), 1.0 (TPSS), and 1.05 (B3LYP). For DFT-D3, recommended values of this parameter with different DFT functionals can be found on the webpage `www.chemie.uni-bonn.de/pctc/mulliken-center/software/dft-d3`. The default values of this parameter in ABACUS is available for PBE. This variable will be set to `default`, if no vdW-corrected method has been used. *Default*: `default`
`vdw_s8`	*Type*: Real *Description*: This scale factor is only the parameter of DFT-D3 approachs including D3(0) and D3(BJ). Recommended values of this parameter with different DFT functionals can be found on the webpage `www.chemie.uni-bonn.de/pctc/mulliken-center/software/dft-d3`. The default values of this parameter in ABACUS is available for PBE. This variable will be set to `default`, if no vdW-corrected method has been used. *Default*: `default`
`vdw_a1`	*Type*: Real *Description*: This damping function parameter is only the parameter of DFT-D3 approachs including D3(0) and D3(BJ). Recommended values of this parameter with different DFT functionals can be found on the webpage `www.chemie.uni-bonn.de/pctc/mulliken-center/software/dft-d3`. The default values of this parameter in ABACUS is available for PBE. This variable will be set to `default`, if no vdW-corrected method has been used. *Default*: `default`
`vdw_a2`	*Type*: Real *Description*: This damping function parameter is only the parameter of DFT-D3(BJ) approach. Recommended values of this parameter with different DFT functionals can be found on the webpage `www.chemie.uni-bonn.de/pctc/mulliken-center/software/dft-d3`. The default values of this parameter in ABACUS is available for PBE. This variable will be set to `default`, if no vdW-corrected method has been used. *Default*: `default`
`vdw_d`	*Type*: Real *Description*: The variable is to control the dumping speed of dumping function of DFT-D2. *Default*: `20`

`vdw_abc`	*Type*: Integer *Description*: The variable is to control the calculation of three-body term of DFT-D3 approachs, including D3(0) and D3(BJ). If set to 1, ABACUS will calculate three-body term, otherwise, the three-body term is not included. *Default*: `0`
`vdw_C6_file`	*Type*: String *Description*: This variable which is useful only when set *vdw_method* to `d2` 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 86_Rn. 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`
`vdw_C6_unit`	*Type*: String *Description*: This variable which is useful only when set *vdw_method* to `d2` specifies unit of C_6 Parameters. Two kinds of unit is available: `Jnm6/mol`(means $J \cdot nm^6/mol$) and `eVA`(means $eV \cdot \mathring{A}$) *Default*: `Jnm6/mol`
`vdw_R0_file`	*Type*: String *Description*: This variable which is useful only when set *vdw_method* to `d2` 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 86_Rn. 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`
`vdw_R0_unit`	*Type*: String *Description*: This variable which is useful only when set *vdw_method* to `d2` specifies unit of R_0 Parameters. Two kinds of unit is available: `A`(means $\mathring{A}$) and `Bohr`. *Default*: `A`
`vdw_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 *vdw_period*. If set to `radius`, ABACUS will calculate a cubic periodic structure containing a sphere, whose radius is *vdw_radius* and centre is origin point. *Default*: `radius`
`vdw_radius`	*Type*: Real *Description*: If *vdw_model* is set to `radius`, this variable specifies the radius of the calculated sphere. For DFT-D2, the default value is 56.6918 ,while it is 95 for DFT-D3. This variable will be set to `default`, if no vdW-corrected method has been used. *Default*: `default`

`vdw_radius_unit`	*Type*: String *Description*: If *vdw_model* is set to `radius`, this variable specifies the unit of *vdw_radius*. Two kinds of unit is available: `A`(means *Å*) and `Bohr`. *Default*: `Bohr`
`vdw_cn_radius`	*Type*: Real *Description*: This cutoff is chosen for the calculation of the coordination number (CN) in DFT-D3 approachs, including D3(0) and D3(BJ). *Default*: `40.0`
`vdw_cn_radius_unit`	*Type*: String *Description*: This variable specifies the unit of *vdw_cn_radius*. Two kinds of unit is available: `A`(means *Å*) and `Bohr`. *Default*: `Bohr`
`vdw_period`	*Type*: Int Int Int *Description*: If *vdw_model* is set to `period`, these variables specify the number of x, y and z periodic. *Default*: `3 3 3`

This part of variables are used to control berry phase and wannier90 interfacae parameters.

Name	**Description**
`berry_phase`	*Type*: Integer *Description*: `1`, calculate berry phase; `0`, no calculate berry phase. *Default*: `0`
`gdir`	*Type*: Integer *Description*: - `1`: calculate the polarization in the direction of the lattice vector a_1 that is defined in STRU file. - `2`: calculate the polarization in the direction of the lattice vector a_2 that is defined in STRU file. - `3`: calculate the polarization in the direction of the lattice vector a_3 that is defined in STRU file. *Default*: `3`
`towannier90`	*Type*: Integer *Description*: `1`, generate files for wannier90 code; `0`, no generate. *Default*: `0`
`nnkpfile`	*Type*: String *Description*: the file name when you run "`wannier90 -pp ...`". *Default*: `seedname.nnkp`
`wannier_spin`	*Type*: String *Description*: If *nspin* is set to `2`, - `up`: calculate spin up for wannier function. - `down`: calculate spin down for wannier function. *Default*: `up`

`tddft`	*Type*: Integer *Description*: • `1`: calculate the real time time dependent density functional theory (TDDFT). • `0`: do not calculate TDDFT. *Default*: `0`
`vext`	*Type*: Integer *Description*: • `1`: add a laser material interaction (extern laser field). • `0`: no extern laser field. *Default*: `0`
`vext_dire`	*Type*: Integer *Description*: • `1`: the direction of external light field is along x axis. • `2`: the direction of external light field is along y axis. • `3`: the direction of external light field is along z axis. *Default*: `1`

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] W. G. Hoover, "Canonical dynamics: Equilibrium phase-space distributions," *Phys. Rev. A*, vol. 31, p. 1695, 1985.

[9] `www.wannier.org`.

[10] A. A. Mostofi, J. R. Yates, G. Pizzi, Y. S. Lee, I. Souza, D. Vanderbilt, and N. Marzari, "An updated version of wannier90: A tool for obtaining maximally-localised wannier functions," *Comput. Phys. Commun.*, vol. 185, p. 2309, 2014.

[11] 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.

[12] S. Grimme, J. Antony, S. Ehrlich, and H. Krieg, "A consistent and accurate ab initio parametrization of density functional dispersion correction (dft-d) for the 94 elements h-pu," *J. Chem. Phys.*, vol. 132, p. 154104, 2010.

[13] S. Grimme, S. Ehrlich, and L. Goerigk, "Effect of the damping function in dispersion corrected density functional theory," *Journal of computational chemistry*, vol. 32, no. 7, pp. 1456–1465, 2011.

[14] N. A. Spaldin, "A beginner's guide to the modern theory of polarization," *Journal of Solid State Chemistry*, vol. 195, no. 195, pp. 2–10, 2012.

[15] A. A. Mostofi, J. R. Yates, G. Pizzi, Y. S. Lee, I. Souza, D. Vanderbilt, and N. Marzari, "An updated version of wannier90 : A tool for obtaining maximally-localised wannier functions," *Computer Physics Communications*, vol. 185, no. 8, pp. 2309–2310, 2014.