
SALMON document

Release v.1.2.1

SALMON developers

May 21, 2019

CONTENTS

1	Introduction	3
1.1	About SALMON	3
1.2	SALMON features	3
1.3	License	4
1.4	SALMON at Github	4
1.5	List of developers	4
1.6	Acknowledgements for SALMON developments	5
2	Install and Run	7
2.1	Prerequisites	7
2.2	Download	7
2.3	Build and Install	8
2.4	Files necessary to run SALMON	9
2.5	Run SALMON	10
2.6	Appendix	11
2.7	Troubleshooting of the Installation Process	12
3	Inputs	15
3.1	&units	15
3.2	&calculation	15
3.3	&control	15
3.4	&functional	16
3.5	&system	16
3.6	&pseudo	17
3.7	&atomic_coor	18
3.8	&atomic_red_coor	18
3.9	&rgrid	19
3.10	&kgrid	19
3.11	&scf	19
3.12	&hartree	20
3.13	&tgrid	20
3.14	&propagation	20
3.15	&emfield	21
3.16	&analysis	22
3.17	&multiscale	23
3.18	¶llel	23
4	Exercises	25
4.1	Getting started	25
4.2	C2H2 (isolated molecules)	26

4.3	Crystalline silicon (periodic solids)	41
4.4	Maxwell + TDDFT multiscale simulation	64
4.5	FDTD simulation	75
5	List of all input keywords	83
5.1	&calculation	84
5.2	&control	84
5.3	&units	85
5.4	¶llel	85
5.5	&system	86
5.6	&atomic_red_coor	87
5.7	&atomic_coor	87
5.8	&pseudo	87
5.9	&functional	87
5.10	&rgrid	88
5.11	&kgrid	88
5.12	&tgrid	88
5.13	&propagation	89
5.14	&scf	89
5.15	&emfield	90
5.16	&multiscale	91
5.17	&maxwell	92
5.18	&analysis	93
5.19	&hartree	94
5.20	&ewald	95
5.21	&opt (Trial)	95
5.22	&md (Trial)	95
5.23	&misc	96
5.24	&group_fundamental (Trial)	96
5.25	&group_parallel (Trial)	96
5.26	&group_hartree (Trial)	97
5.27	&group_file (Trial)	97
5.28	&group_others (Trial)	97
	Bibliography	99

May 21, 2019

SALMON is an open-source software based on first-principles time-dependent density functional theory to describe optical responses and electron dynamics in matters induced by light electromagnetic fields.

INTRODUCTION

1.1 About SALMON

SALMON is an open-source computer program for ab-initio quantum-mechanical calculations of electron dynamics at the nanoscale that takes place in various situations of light-matter interactions. It is based on time-dependent density functional theory, solving time-dependent Kohn-Sham equation in real time and real space with norm-conserving pseudopotentials.

SALMON was born by unifying two scientific programs: ARTED, developed by Univ. Tsukuba group, that describes electron dynamics in crystalline solids, and GCEED, developed by Institute for Molecular Science group, that describes electron dynamics in molecules and nanostructures. It can thus describe electron dynamics in both isolated and periodic systems. It can also describe coupled dynamics of electrons and light-wave electromagnetic fields.

To run the program, SALMON requires MPI Fortran/C compiler with LAPACK libraries. SALMON has been tested and optimized to run in a number of platforms, including Linux PC Cluster with x86-64 CPU, Fujitsu FX100 super-computer system, K-computer, and supercomputer system with Intel Xeon Phi (Knights Landing).

1.2 SALMON features

SALMON describes electron dynamics in both isolated (molecules and nanostructures) and periodic (crystalline solids) systems. SALMON first carries out ground-state calculations in the density functional theory to prepare initial configurations. SALMON then calculates electron dynamics induced by applied electric field. Employing a weak impulsive external field, SALMON can be used to calculate linear response properties such as a polarizability of molecules and a dielectric function of crystalline solids. Using pulsed electric fields, SALMON describes electron dynamics in matters induced by intense and ultrashort laser pulses.

- Ground state calculations
 - Kohn-Sham orbitals and energies
 - density of states
 - projected density of states
 - electron localization function
- Optical properties
 - Oscillator strength distribution (absorption spectrum)
 - dielectric function
- Light-induced electron dynamics
 - time evolution of Kohn-Sham orbitals

- density, current
- excitation energy
- number density of excited carriers
- Simultaneous description of electron dynamics and light pulse propagation
 - light pulse propagation as well as time evolution of Kohn-Sham orbitals
 - energy transfer from pulsed light to electrons

1.3 License

SALMON is available under Apache License version 2.0.

Copyright 2017 SALMON developers

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.4 SALMON at Github

SALMON is developed at [GitHub.com](https://github.com)

1.5 List of developers

(Alphabetic order)

- Isabella Floss (TU Wien, Austria)
- Yuta Hirokawa (University of Tsukuba, Japan)
- Kenji Iida (Institute for Molecular Science, Japan)
- Kazuya Ishimura (Institute for Molecular Science, Japan)
- Kyung-Min Lee (Max Planck Institute for the Structure and Dynamics of Matter, Germany)
- Katsuyuki Nobusada (Institute for Molecular Science, Japan)
- Masashi Noda (University of Tsukuba, Japan)
- Tomohito Otobe (National Institutes for Quantum and Radiological Science and Technology, Japan)
- Shunsuke Sato (Max Planck Institute for the Structure and Dynamics of Matter, Germany)
- Yasushi Shinohara (University of Tokyo, Japan)
- Takashi Takeuchi (University of Tsukuba, Japan)
- Xiao-Min Tong (University of Tsukuba, Japan)

- Mitsuharu Uemoto (University of Tsukuba, Japan)
- Kazuhiro Yabana (University of Tsukuba, Japan)
- Atsushi Yamada (University of Tsukuba, Japan)
- Shunsuke Yamada (University of Tsukuba, Japan)
- Maiku Yamaguchi (University of Tokyo, Japan)

1.6 Acknowledgements for SALMON developments

SALMON has been developed by the SALMON developers under supports by Center for Computational Sciences, University of Tsukuba, and Institute for Molecular Science. SALMON has been supported by Strategic Basic Research Programs, CREST, Japan Science and Technology Agency, under the Grand Number JPMJCR16N5, in the research area of Advanced core technology for creation and practical utilization of innovative properties and functions based upon optics and photonics. SALMON has been also supported by Ministry of Education, Culture, Sports and Technology of Japan as a social and scientific priority issue (Creation of new functional devices and high-performance materials to support next-generation industries: CDMSI) to be tackled by using post-K computer.

INSTALL AND RUN

2.1 Prerequisites

In this guide, it is assumed that readers have a basic knowledge of Unix and its command line operations. For the installation of SALMON, following packages are required.

- Fortran90/C compiler. SALMON assumes users have one of the following compilers:
 - GCC (Gnu Compiler Collection)
 - Intel Fortran/C Compiler
 - Fujitsu Compiler (at FX100 / K-Computer)
- One of the following library packages for linear algebra:
 - BLAS/LAPACK
 - Intel Math Kernel Library (MKL)
 - Fujitsu Scientific Subroutine Library 2 (SSL-II)
- Build tools:
 - CMake

If you use other compilers, you may need to change build scripts (CMake). See *Additional options in configure.py script*. If no numerical library is installed on your computer system, you may need to install BLAS/LAPACK by yourself. See *Troubleshooting of the Installation Process*.

For the installation of SALMON, we adopt the CMake tools as the first option. If there were any problems to use CMake tools in your environment, you may use the GNU make tools. See *Troubleshooting of the Installation Process*.

2.2 Download

The newest version of SALMON can be downloaded from [download page](#). To extract files from the downloaded file SALMON-<VERSION>.tar.gz, type the following command in the command-line:

```
$ tar -zxvf ./salmon-<VERSION>.tar.gz
```

After the extraction, the following directories will be created:

```
SALMON
|- src          Source codes
|- example     Samples
```

(continues on next page)

(continued from previous page)

```
| - cmakefiles    CMake related files
| - gnumakefiles  GNU Makefiles for building
```

2.3 Build and Install

To compile SALMON to create executable the binary files, we adopt to use CMake tools as the first option. In case you fail to build SALMON using CMake in your environment, we may use Gnu Make. See *Build using GNU Makefile*.

2.3.1 Checking CMake availability

First, examine whether CMake is usable in your environment or not. Type the following in Unix command-line:

```
$ cmake --version
```

If CMake is not installed in your system, an error message such as `cmake: command not found` will appear. If CMake is installed on your system, the version number will be shown. To build SALMON, CMake of version 3.0.2 or later is required. If you confirm that CMake of version 3.0.2 or later is installed in your system, proceed to *Build using CMake*. However, we realize that old versions of CMake are installed in many systems. If CMake is not installed or CMake of older versions is installed in your system, you need to install the new version by yourself. It is a simple procedure and explained below.

2.3.2 Installation of CMake

CMake is a cross-platform build tool. The simplest way to make CMake usable in your environment is to get the [binary distribution of CMake from the download page](#). (The file name of the binary distribution will be `cmake-<VERSION>-<PLATFORM>.tar.gz`). In standard Unix environment, a file for the platform of Linux `x86_64` will be appropriate.

To download the file, proceed as follows: We assume that you are in the directory that you extracted files from the downloaded file of SALMON, and that you will use the version 3.8.2. First get the URL of the download link from your browser, and use `wget` command in your Unix command-line:

```
$ wget https://cmake.org/files/v3.8/cmake-3.8.2-Linux-x86_64.tar.gz
```

Next, unpack the archive by:

```
$ tar -zxvf cmake-3.8.2-Linux-x86_64.tar.gz
```

and you will have the binary `make-3.8.2-Linux-x86_64/bin/cmake` in your directory.

To make the `cmake` command usable in your command-line, you need to modify the environment variable `$PATH` so that the executable of CMake are settled inside the directory specified in your `$PATH`. If you use the bash shell, you need to modify the file `~/.bashrc` that specifies the `$PATH` variable. It can be done by typing the following command in your login directory:

```
$ export PATH=<SALMON_INSTALLATION_DIRECTORY>/cmake-3.8.2-Linux-x86_64/bin:$PATH
```

and then reload the configuration by typing:

```
$ source ~/.bashrc
```

2.3.3 Build using CMake

Confirming that CMake of version 3.0.2 or later can be usable in your environment, proceed the following steps. We assume that you are in the directory SALMON.

1. Create a new temporary directory `build` and move to the directory:

```
$ mkdir build
$ cd build
```

2. Execute the python script "configure.py" and then make:

```
$ python ../configure.py --arch=ARCHITECTURE --prefix=../
$ make
$ make install
```

In executing the python script, you need to specify `ARCHITECTURE` that indicates the architecture of the CPU in your computer system such as `intel-avx`. The options of the `ARCHITECTURE` are as follows:

arch	Detail	Compiler	Numerical Library
intel-knl	Intel Knights Landing	Intel Compiler	Intel MKL
intel-knc	Intel Knights Corner	Intel Compiler	Intel MKL
intel-avx	Intel Processor (Ivy-, Sandy-Bridge)	Intel Compiler	Intel MKL
intel-avx2	Intel Processor (Haswell, Broadwell ..)	Intel Compiler	Intel MKL
intel-avx512	Intel Processor (Skylake-SP)	Intel Compiler	Intel MKL
fujitsu-fx100	FX100 Supercomputer	Fujitsu Compiler	SSL-II
fujitsu-k	Fujitsu FX100 / K-computer	Fujitsu Compiler	SSL-II

If the build is successful, you will get a file `salmon.cpu` at the directory `salmon/bin`. If you specify many-core architectures, `intel-knl` or `intel-knc`, you find a file `salmon.mic` or both files `salmon.cpu` and `salmon.mic`.

2.4 Files necessary to run SALMON

To run SALMON, at least two kinds of files are required for any calculations. One is an input file with the filename extension `*.inp` that should be read from the standard input `stdin`. This file should be prepared in the Fortran90 namelist format. Pseudopotential files of relevant elements are also required. Depending on your purpose, some other files may also be necessary. For example, coordinates of atomic positions of the target material may be either written in the input file or prepared as a separate file.

2.4.1 Pseudopotentials

SALMON utilizes norm-conserving pseudopotentials. You may find pseudopotentials of some elements in the samples prepared in *Exercises*. In SALMON, several formats of pseudopotentials may be usable. Pseudopotentials with an extension `.fhi` can be obtained from the website listed below. (This is a part of previous atomic data files for the ABINIT code.)

Pseudopotential	Website
Pseudopotentials for the ABINIT code	https://www.abinit.org/sites/default/files/PrevAtomicData/psp-links/psp-links/lda_fhi

Filenames of the pseudopotentials should be written in the input file.

2.4.2 input file

Input files are composed of several blocks of namelists:

```
&namelist1
  variable1 = int_value
  variable2 = 'char_value'
/
&namelist2
  variable1 = real8_value
  variable2 = int_value1, int_value2, int_value3
/
```

A block of namelists starts with `&namelist` line and ends with `/` line. The blocks may appear in any order.

Between two lines of `&namelist` and `/`, descriptions of variables and their values appear. Note that many variables have their default values so that it is not necessary to give values for all variables. Descriptions of the variables may appear at any position if they are between `&namelist` and `/`.

SALMON describes electron dynamics in systems with both isolated and periodic boundary conditions. The boundary condition is specified by the variable `iperiodic` in the namelist `&system`.

Calculations are usually achieved in two steps; first, the ground state calculation is carried out and then electron dynamics calculations in real time is carried out. A choice of the calculation mode is specified by the variable `calc_mode` in the namelist `&calculation`. For isolated systems, the ground state and the electron dynamics calculations should be carried out as two separate executions. First the ground state calculation is carried out specifying `calc_mode = 'GS'`. Then the real-time electron dynamics calculation is carried out specifying `calc_mode = 'RT'`. For periodic systems, two calculations should be carried out as a single execution specifying `calc_mode = 'GS_RT'`.

In *Exercises*, we prepare six exercises that cover typical calculations feasible by SALMON. We prepare explanations of the input files of the exercises that will help to prepare input files of your own interests.

There are more than 20 groups of namelists. A complete list of namelist variables is given in the file `SALMON/manual/input_variables.md`. Namelist variables that are used in our exercises are explained at *Inputs*.

2.5 Run SALMON

Before running SALMON, the following preparations are required as described above: The executable file of `salmon.cpu` and `salmon.mic` (if your system is the many-core machine) should be built from the source file of SALMON. An input file `inputfile.inp` and pseudopotential files should also be prepared.

The execution of the calculation can be done as follows: In single process environment, type the following command:

```
$ salmon.cpu < inputfile.inp > fileout.out
```

In multiprocess environment in which the command to execute parallel calculations using MPI is `mpiexec`, type the following command:

```
$ mpiexec -n NPROC salmon.cpu < inputfile.inp > fileout.out
```

where `NPROC` is the number of MPI processes that you will use. In many-core processor (e.g. intel-knl) environment, the execution command is:

```
$ mpiexec.hydra -n NPROC salmon.mic < inputfile.inp > fileout.out
```

The execution command and the job submission procedure depends much on local environment. We summarize general conditions to execute SALMON:

- SALMON runs in both single-process and multi-process environments using MPI.
- executable files are prepared as `/salmon/bin/salmon.cpu` and/or `/salmon/bin/salmon.mic` in the standard build procedure.
- to start calculations, `inputfile.inp` should be read through `stdin`.

2.6 Appendix

2.6.1 Additional options in `configure.py` script

Manual specifications of compiler and environment variables

In executing `configure.py`, you may manually specify compiler and environment variables instead of specifying the architecture, for example:

```
$ python ../configure.py FC=mpiifort CC=mpiicc FFLAGS="-xAVX" CFLAGS="-restrict -xAVX"
```

The major options of `configure.py` are as follows:

Commandline switch	Detail
<code>-a ARCH, --arch=ARCH</code>	Target architecture
<code>--enable-mpi, --disable-mpi</code>	enable/disable MPI parallelization
<code>--enable-scalapack, --disable-scalapack</code>	enable/disable computations with ScaLAPACK library
<code>--enable-libxc, --with-libxc</code>	see <i>To use Libxc</i>
<code>FC, FFLAGS</code>	User-defined Fortran Compiler, and the compiler options

To use Libxc

In SALMON, you may use [Libxc functional library](#). To use the Libxc library, some additional procedures are necessary. First you need to download the source files in your system as follows:

```
$ wget http://www.tddft.org/programs/octopus/down.php?file=libxc/4.2.1/libxc-4.2.1.
→tar.gz
$ tar -zxvf libxc-4.2.1.tar.gz
```

Then, enter the libxc source directory and make the library as follows:

```
$ ./configure --prefix=INSTALL/PATH/OF/LIBXC
$ make && make install
```

Finally, enter the SALMON directory and execute `configure.py` script specifying the Libxc directory:

```
$ configure.py --arch=ARCHITECTURE --prefix=PREFIX --with-libxc=INSTALL/PATH/OF/LIBXC
$ make && make install
```

Build for single process calculations

If you use a single processor machine, specify `--disable-mpi` in executing the python script:

```
$ python ../configure.py --arch=<ARCHITECTURE> --disable-mpi
```

Build in GCC/GFortran environemnt

If you use GCC/GFortran compiler, specify the following flags in executing the python script:

```
$ python ../configure.py FC=gfortran CC=gcc FFLAG=-O3 CFLAG=-O3
```

2.6.2 Build using GNU Makefile

If CMake build fails in your environment, we recommend you to try to use Gnu Make for the build process. First, enter the directory `makefiles`:

```
$ cd SALMON/makefiles
```

In the directory, Makefile files are prepared for several architectures:

- fujitsu
- gnu
- gnu-without-mpi
- intel
- intel-avx
- intel-avx2
- intel-knc
- intel-knl
- intel-without-mpi

Makefile files with `*-without-mpi` indicate that they are for single processor environment. Choose Makefile appropriate for your environment, and execute the make command:

```
$ make -f Makefile.PLATFORM
```

If the make proceeds successful, a binary file is created in the directory `SALMON/bin/`.

2.7 Troubleshooting of the Installation Process

2.7.1 Installation of CMake

The `CMake` is a cross-platform build tool. In order to build the SALMON from the source code, the CMake of version 3.0.2 or later is required. You may install it following one of the three instructions below.

Installation of pre-compiled binary

You can get the binary distribution from the [download page](#). First, move to the directory that you installed SALMON,

```
cd <SALMON_INSTALLATION_DIRECTORY>
```

and download the binary distribution (`cmake-<VERSION>-<PLATFORM>.tar.gz`) appropriate for your platform. You can do it by copy the URL of download link from the browser, and use `wget` command:


```
wget https://cmake.org/files/v3.8/cmake-3.8.2-Linux-x86_64.tar.gz
```

In this document, we will use version 3.8.2 as an example. You can unpack the downloaded archive

```
tar -zxvf cmake-3.8.2-Linux-x86_64.tar.gz
```

and you will have the binary `make-3.8.2-Linux-x86_64/bin/cmake`.

Next, to utilize the `cmake` command, it is required that the executable are settled inside the directory specified in your `$PATH`. If you use the bash shell, edit `~/.bashrc` and append the line:

```
export PATH=<SALMON_INSTALLATION_DIRECTORY>/cmake-3.8.2-Linux-x86_64/bin:$PATH
```

and reload the configuration:

```
source ~/.bashrc
```

Installation by package manager

If your system has a built-in package manager, you may conveniently install the CMake tools as below:

Debian/Ubuntu Linux

```
sudo apt-get install cmake
```

Fedora Linux/CentOS

```
sudo yum install cmake
```

openSUSE Linux

```
sudo zypper install cmake
```

Installation from source code

You can get the source code distribution from the [download page](#). In this time, we will use the `cmake` version 3.8.2 as an example. Download the archive by `wget` command and unpack it as below:

```
wget https://cmake.org/files/v3.8/cmake-3.8.2.tar.gz
tar -zxvf cmake-3.8.2.tar.gz
```

And, move to the unpacked directory and build.

```
cd cmake-3.8.2
./configure --prefix=INSTALLATION_DIRECTORY
make
make install
```

(replace `INSTALLATION_DIRECTORY` to your installation directory.)

Next, to utilize the `cmake` command, it is required that the executable are settled inside the directory specified in your `$PATH`. If you use the bash shell, edit `~/.bashrc` and append the line:

```
export PATH=INSTALLATION_DIRECTORY/bin:$PATH
```

and reload the configuration:

```
source ~/.bashrc
```

2.7.2 Installation of BLAS/LAPACK

We here summarize namelists that appear in this Tutorial. A thorough list of the namelist variables may be found in the downloaded file in 'SALMON/manual/input_variables.md'.

3.1 &units

Mandatory: none

```
&units
  unit_system='A_eV_fs'
/
```

This namelist specifies the unit system to be used in the input file. Options are 'A_eV_fs' for Angstrom, eV, and fs, and 'a.u.' or 'au' for atomic units. If you do not specify it, atomic unit will be used as default.

For isolated systems (specified by `iperiodic = 0` in `&system`), the unit of 1/eV is used for the output files of DOS and PDOS if `unit_system = 'A_eV_fs'` is specified, while atomic unit is used if not. For other output files, the Angstrom/eV/fs units are used irrespective of the namelist value.

For periodic systems (specified by `iperiodic = 3` in `&system`), the unit system specified by this namelist variable is used for most output files. See the first few lines of output files to confirm the unit system adopted in the file.

3.2 &calculation

Mandatory: `calc_mode`

```
&calculation
  calc_mode = 'GS'
/
```

The value of the `calc_mode` should be one of 'GS', 'RT', and 'GS-RT'. For isolated systems (specified by `iperiodic = 3` in `&system`), the ground state ('GS') and the real time ('RT') calculations should be done separately and sequentially. For periodic systems (specified by `iperiodic = 3` in `&system`), both ground state and real time calculations should be carried out as a single task (`calc_mode = 'GS_RT'`).

For Maxwell + TDDFT multi-scale calculation, add the following namelist, `use_ms_maxwell = 'y'`.

3.3 &control

Mandatory: none

```
&control
  sysname = 'C2H2'
/
```

'C2H2' defined by `sysname = 'C2H2'` will be used in the filenames of output files. If you do not specify it, the file name will start with 'default'.

3.4 &functional

```
&functional
  xc = 'PZ'
/
```

`xc = 'PZ'` indicates that (adiabatic) local density approximation is adopted (Perdew-Zunger: Phys. Rev. B23, 5048 (1981)). This is the default choice.

For isolated systems (specified by `iperiodic = 0` in `&system`), only the default choice of 'PZ' is available at present.

For periodic systems (specified by `iperiodic = 3` in `&system`), the following functionals may be available in addition to 'PZ', `xc = 'PZM'`

Perdew-Zunger LDA with modification to improve smooth connection between high density form and low density one, `xc = 'TBmBJ' cval = 1.0`: J. P. Perdew and Alex Zunger, Phys. Rev. B 23, 5048 (1981).

Tran-Blaha meta-GGA exchange with Perdew-Wang correlation. :Fabien Tran and Peter Blaha, Phys. Rev. Lett. 102, 226401 (2009). John P. Perdew and Yue Wang, Phys. Rev. B 45, 13244 (1992). This potential is known to provide a reasonable description for the bandage of various insulators. For this choice, the additional mixing parameter 'cval' may be specified. If cval is set to a minus value, the mixing-parameter will be computed following the formula in the original paper [Phys. Rev. Lett. 102, 226401 (2009)]. The default value for this parameter is 1.0.

Since version 1.1.0, exchange-correlation functionals in Libxc library (<http://www.tddft.org/programs/libxc/>) have been usable in SALMON. At present, usable functionals are limited to LDA and GGA. For periodic systems, meta-GGA functionals are usable as well. To specify the exchange-correlation potentials of Libxc library, there are two ways. If the exchange and correlation potentials are given separately, you need to specify both `alibx` and `alibc` separately. If the exchange and correlation potentials are given as a combined set, you need to specify `alibxc`. We show below an example:

```
&functional
alibx = 'LDA_X'
alibc = 'LDA_C_PZ'
/
```

Available sets of the functionals are listed at the website <http://www.tddft.org/programs/libxc/functionals/>.

Note that, the hybrid functionals (hybrid gga/mgga) are not supported in the current (version 1.2.0) of SALMON.

3.5 &system

Mandatory: `iperiodic`, `al`, `nstate`, `nelem`, `natom`

For an isolated molecule (Tutorial-1, 2, 3):

```
&system
  iperiodic = 0
  al = 16d0, 16d0, 16d0
  nstate = 5
  nelelem = 2
  natom = 4
  nelec = 10
/
```

`iperiodic = 0` indicates that the isolated boundary condition will be used in the calculation. `al = 16d0, 16d0, 16d0` specifies the lengths of three sides of the rectangular parallelepiped where the grid points are prepared. `nstate = 8` indicates the number of Kohn-Sham orbitals to be solved. `nelec = 10` indicate the number of valence electrons in the system. Since the present code assumes that the system is spin saturated, `nstate` should be equal to or larger than `nelec/2`. `nelem = 2` and `natom = 4` indicate the number of elements and the number of atoms in the system, respectively.

For a periodic system (Tutorial-4, 5):

```
&system
  iperiodic = 3
  al = 10.26d0,10.26d0,10.26d0
  nstate = 32
  nelec = 32
  nelelem = 1
  natom = 8
/
```

`iperiodic = 3` indicates that three dimensional periodic boundary condition (bulk crystal) is assumed. `al = 10.26d0, 10.26d0, 10.26d0` specifies the lattice constants of the unit cell. `nstate = 32` indicates the number of Kohn-Sham orbitals to be solved. `nelec = 32` indicate the number of valence electrons in the system. `nelem = 1` and `natom = 8` indicate the number of elements and the number of atoms in the system, respectively.

For Maxwell - TDDFT multi scale calculation (Tutorial-6):

```
&system
  iperiodic = 3
  al = 10.26d0,10.26d0,10.26d0
  isym = 8
  crystal_structure = 'diamond'
  nstate = 32
  nelec = 32
  nelelem = 1
  natom = 8
/
```

The difference from the above case is the variables, `isym = 8` and `crystal_structure = 'diamond'`, which indicates that the spatial symmetry of the unit cell is used in the calculation. Although the use of the symmetry substantially reduces the computational cost, it should be used very carefully. At present, the spatial symmetry has been implemented only for the case of the diamond structure.

3.6 &pseudo

Mandatory: `pseudo_file`, `izatom`

For C2H2 molecule:

```
&pseudo
  izatom(1)=6
  izatom(2)=1
  pseudo_file(1)='C_rps.dat'
  pseudo_file(2)='H_rps.dat'
  lmax_ps(1)=1
  lmax_ps(2)=0
  lloc_ps(1)=1
  lloc_ps(2)=0
/
```

Parameters related to atomic species and pseudopotentials. `izatom(1) = 6` specifies the atomic number of the element #1. `pseudo_file(1) = 'C_rps.dat'` indicates the filename of the pseudopotential of element #1. `lmax_ps(1) = 1` and `lloc_ps(1) = 1` specify the maximum angular momentum of the pseudopotential projector and the angular momentum of the pseudopotential that will be treated as local, respectively.

For crystalline Si:

```
&pseudo
  izatom(1)=14
  pseudo_file(1) = './Si_rps.dat'
  lloc_ps(1)=2
/
```

`izatom(1) = 14` indicates the atomic number of the element #1. `pseudo_file(1) = 'Si_rps.dat'` indicates the pseudopotential filename of element #1. `lloc_ps(1) = 2` indicate the angular momentum of the pseudopotential that will be treated as local.

3.7 &atomic_coor

Mandatory: `atomic_coor` or `atomic_red_coor` (they may be provided as a separate file)

For C₂H₂ molecule:

```
&atomic_coor
'C' 0.000000 0.000000 0.599672 1
'H' 0.000000 0.000000 1.662257 2
'C' 0.000000 0.000000 -0.599672 1
'H' 0.000000 0.000000 -1.662257 2
/
```

Cartesian coordinates of atoms. The first column indicates the element. Next three columns specify Cartesian coordinates of the atoms. The number in the last column labels the element.

3.8 &atomic_red_coor

Mandatory: `atomic_coor` or `atomic_red_coor` (they may be provided as a separate file)

For a crystalline silicon:

```
&atomic_red_coor
'Si' .0 .0 .0 1
'Si' .25 .25 .25 1
```

(continues on next page)

(continued from previous page)

```
'Si' .5 .0 .5 1
'Si' .0 .5 .5 1
'Si' .5 .5 .0 1
'Si' .75 .25 .75 1
'Si' .25 .75 .75 1
'Si' .75 .75 .25 1
/
```

Cartesian coordinates of atoms are specified in a reduced coordinate system. First column indicates the element, next three columns specify reduced Cartesian coordinates of the atoms, and the last column labels the element.

3.9 &rgrid

Mandatory: dl or num_rgrid

This namelist provides grid spacing of Cartesian coordinate system. dl (3) specify the grid spacing in three Cartesian coordinates. This is adopted for C₂H₂ calculation (Tutorial-1).

```
&rgrid
dl = 0.25d0, 0.25d0, 0.25d0
/
```

num_rgrid(3) specify the number of grid points in each Cartesian direction. This is adopted for crystalline Is calculation (Tutorial-4, 5, 6).

```
&rgrid
num_rgrid = 12,12,12
/
```

3.10 &kgrid

Mandatory: none

This namelist provides grid spacing of k-space for periodic systems.

```
&kgrid
num_kgrid = 4,4,4
/
```

3.11 &scf

Mandatory: nscf

This namelists specify parameters related to the self-consistent field calculation.

```
&scf
ncg = 4
nscf = 1000
convergence = 'norm_rho_dng'
threshold_norm_rho = 1.d-15
/
```

`ncg = 4` is the number of conjugate-gradient iterations in solving the Kohn-Sham equation. Usually this value should be 4 or 5. `nscf = 1000` is the number of scf iterations. For isolated systems specified by `&system/iperiodic = 0`, the scf loop in the ground state calculation ends before the number of the scf iterations reaches `nscf`, if a convergence criterion is satisfied. There are several options to examine the convergence. If the value of `norm_rho_dng` is specified, the convergence is examined by the squared difference of the electron density,

3.12 &hartree

Mandatory: none

```
&hartree
  meo = 3
  num_pole_xyz = 2,2,2
/
```

`meo` specifies the order of multipole expansion of electron density that is used to prepare boundary condition for the Hartree potential.

- `meo=1`: Monopole expansion (spherical boundary condition).
- `meo=2`: Multipole expansions around each atom.
- `meo=3`: Multipole expansion around the center of mass of electrons in cubits that are defined by `num_pole_xyz`.

`num_pole_xyz (3)` defines the division of space when `meo = 3` is specified.

A default for `meo` is 3, and defaults for `num_pole_xyz` are (0, 0, 0). When default is set for `num_pole_xyz`, the division of space is carried out using a prescribed method.

3.13 &tgrid

Mandatory: dt, Nt

```
&tgrid
  dt=1.25d-3
  nt=5000
/
```

`dt=1.25d-3` specifies the time step of the time evolution calculation. `nt=5000` specifies the number of time steps in the calculation.

3.14 &propagation

This namelist specifies the numerical method for time evolution calculations of electron orbitals.

```
&propagation
  propagator='etrs'
/
```

`propagator = 'etrs'` indicates the use of enforced time-reversal symmetry propagator. M.A.L. Marques, A. Castro, G.F. Bertsch, and A. Rubio, *Comput. Phys. Commun.*, 151 60 (2003).


```
&propagation
  propagator='middlepoint'
/
```

propagation='middlepoint' indicates that Hamiltonian at midpoint of two-times is used.

The default is *middlepoint*.

3.15 &emfield

This namelist specifies the pulse shape of an electric field applied to the system in time evolution calculations. We explain below separating two cases, *Linear response calculations* and *Pulsed electric field calculations*.

3.15.1 Linear response calculations

A weak impulsive field is applied at $t=0$. For this case, ae_shape1 = 'impulse' should be described.

Mandatory: ae_shape1

```
&emfield
  ae_shape1 = 'impulse'
  ekdir_rel = 0.d0,0.d0,1.d0
/
```

epdir_rel(3) specify a unit vector that indicates the direction of the impulse.

For a periodic system specified by iperiodic = 3, one may add trans_longi. It has the value, 'tr'(transverse) or 'lo'(longitudinal), that specifies the treatment of the polarization in the time evolution calculation. The default is 'tr'.

```
&emfield
  trans_longi = 'tr'
  ae_shape1 = 'impulse'
  ekdir_rel = 0.,0.,1.
/
```

The magnitude of the impulse of the pulse may be explicitly specified by, for example, e_impulse = 1d-2. The default is '1d-2' in atomic unit.

3.15.2 Pulsed electric field calculations

A Pulsed electric field of finite time duration is applied. For this case, as_shape1 should be specified. It indicates the shape of the envelope of the pulse. The options include 'Acos2' and 'Ecos2' (See below for other options).

Mandatory: ae_shape1, ekdir_rel, {rlaser_int1 or amplitude1}, omega1, pulse_tw1, phi_cep1

```
&emfield
  ae_shape1 = 'Ecos2'
  ekdir_rel = 0.d0,0.d0,1.d0
  rlaser_int_wcm2_1 = 1.d8
  omega1=9.28d0
  pulse_tw1=6.d0
  phi_cep1=0.75d0
/
```

`ae_shape1 = 'Ecos2'` specifies the envelope of the pulsed electric field, 'Ecos2' for the \cos^2 envelope for the electric field. If 'Acos2' is specified, this gives \cos^2 envelope for the vector potential. Note that 'phi_cep1' must be 0.75 (or 0.25) if one employs 'Ecos2' pulse shape, since otherwise the time integral of the electric field does not vanish. There is no such restriction for the 'Acos2' pulse shape.

`epdir_re1 = 0.d0, 0.d0, 1.d0` specifies the real part of the unit polarization vector of the pulsed electric field. If only the real part is specified, it describes a linearly polarized pulse. Using both real ('epdir_re1') and imaginary ('epdir_im1') parts of the polarization vector, circularly (and general ellipsoidal) polarized pulses may be described.

`laser_int_wcm2_1 = 1.d8` specifies the maximum intensity of the applied electric field in unit of W/cm^2 . It is also possible to specify the maximum intensity of the pulse by `amplitude1`.

`omega1=9.26d0` specifies the average photon energy (frequency multiplied with \hbar).

`pulse_tw1=6.d0` specifies the pulse duration. Note that it is not the FWHM but a full duration of the \cos^2 envelope.

`phi_cep1=0.75d0` specifies the carrier envelope phase of the pulse. As noted above, 'phi_cep1' must be 0.75 (or 0.25) if one employs 'Ecos2' pulse shape, since otherwise the time integral of the electric field does not vanish. There is no such restriction for the 'Acos2' pulse shape.

It is possible to use two pulses simultaneously to simulate pump-probe experiments, adding information for two pulses. To specify the second pulse, change from 1 to 2 in the namelist variables, like `ae_shape2`. The time delay between two pulses is specified by the variable 't1_t2'.

For a periodic system specified by `iperiodic = 3`, one may add `trans_longi`. It has the value, 'tr'(transverse) or 'lo'(longitudinal), that specifies the treatment of the polarization in the time evolution calculation. The default is 'tr'. For a periodic system, it is also specify 'Acos3', 'Acos4', 'Acos6', 'Acos8' for `ae_shape1`.

3.16 &analysis

Mandatory: none

The following namelists specify whether the output files are created or not after the calculation. In the ground state calculation of isolated systems (Tutorial-1):

```
&analysis
  out_psi = 'y'
  out_dns = 'y'
  out_dos = 'y'
  out_pdos = 'y'
  out_elf = 'y'
/
```

In the time evolution calculation of isolated systems (Tutorial-3):

```
&analysis
  out_dns_rt = 'y'
  out_elf_rt = 'y'
  out_estatic_rt = 'y'
/
```

In the following namelists, variables related to time-frequency Fourier analysis are specified.

```
&analysis
  nenergy=1000
```

(continues on next page)

(continued from previous page)

```

de=0.001
/

```

nenergy=1000 specifies the number of energy steps, and de=0.001 specifies the energy spacing in the time-frequency Fourier transformation.

3.17 &multiscale

This namelist specifies information necessary for Maxwell - TDDFT multiscale calculations.

```

&multiscale
  ftdddim = '1D'
  twod_shape = 'periodic'
  nx_m = 4
  ny_m = 1
  hx_m = 250d0
  nxvac1_m = -2000
  nxvacr_m = 256
/

```

ftddim specifies the spatial dimension of the macro system. ftdddim='1D' indicates that one-dimensional equation is solved for the macroscopic vector potential.

nx_m = 4 specifies the number of the macroscopic grid points in for x-direction in the spatial region where the material exists.

hx_m = 250d0 specifies the grid spacing of the macroscopic grid in x-direction.

nxvac1_m = -2000 and nxvacr_m = 256 indicate the number of grid points in the vacuum region, nxvac1_m for the left and nxvacr_m for the right from the surface of the material.

3.18 ¶llel

When you execute a job with MPI parallelization, you are not required to specify any parameters that describe the assignment of the parallelization; the assignment is carried out automatically. You may also specify the parameters explicitly as below.

Mandatory: none

```

&parallel
  nproc_ob = 1
  nproc_domain = 1,1,1
  nproc_domain_s = 1,1,1
/

```

- nproc_ob specifies the number of MPI parallelization to divide the electron orbitals. The default value is 0 (automatic parallelization).
- nproc_domain(3) specifies the number of MPI parallelization to divide the spatial grids of the electron orbitals in three Cartesian directions. The default values are (0/0/0) (automatic parallelization).
- nproc_domain_s(3) ' specifies the number of MPI parallelization to divide the spatial grids related to the electron density in three Cartesian directions. The default values are (0/0/0) (automatic parallelization).

The following conditions must be satisfied.

- The total number of processors must be equal to both $nproc_ob * nproc_domain(1) * nproc_domain(2) * nproc_domain(3)$ and also $nproc_domain_s(1) * nproc_domain_s(2) * nproc_domain_s(3)$.
- $nproc_domain_s(1)$ is a multiple of $nproc_domain(1)$, and the same relations to the second and third components.

EXERCISES

4.1 Getting started

Welcome to SALMON Exercises!

In these exercises, we explain the use of SALMON from the very beginning, taking a few samples that cover applications of SALMON in several directions. We assume that you are in the computational environment of UNIX/Linux OS. First you need to download and install SALMON in your computational environment. If you have not yet done it, do it following the instruction, [download](#) and *Install and Run*.

As described in *Install and Run*, you are required to prepare at least an input file and pseudopotential files to run SALMON. In the following, we present input files for several sample calculations and provide a brief explanation of the input keywords that appear in the input files. You may modify the input files to execute for your own calculations. Pseudopotential files of elements that appear in the samples are also attached. We also present explanations of main output files.

We present 10 exercises.

First 3 exercises (Exercise-1 ~ 3) are for an isolated molecule, acetylene C₂H₂. If you are interested in learning electron dynamics calculations in isolated systems, please look into these exercises. In SALMON, we usually calculate the ground state solution first. This is illustrated in *Exercise-1*. After finishing the ground state calculation, two exercises of electron dynamics calculations are prepared. *Exercise-2* illustrates the calculation of linear optical responses in real time, obtaining polarizability and photoabsorption of the molecule. *Exercise-3* illustrates the calculation of electron dynamics in the molecule under a pulsed electric field.

Next 2 exercises (Exercise-4 ~ 5) are for a crystalline solid, silicon. If you are interested in learning electron dynamics calculations in extended periodic systems, please look into these exercises. Since ground state calculations of small unit-cell systems are not computationally expensive and a time evolution calculation is usually much more time-consuming than the ground state calculation, we recommend to run the ground and the time evolution calculations as a single job. The following two exercises are organized in that way. *Exercise-4* illustrates the calculation of linear response properties of crystalline silicon to obtain the dielectric function. *Exercise-5* illustrates the calculation of electron dynamics in the crystalline silicon induced by a pulsed electric field.

Exercise-6 is for an irradiation and a propagation of a pulsed light in a bulk silicon, coupling Maxwell equations for the electromagnetic fields of the pulsed light and the electron dynamics in the unit cells. This calculation is quite time-consuming and is recommended to execute using massively parallel supercomputers. *Exercise-6* illustrates the calculation of a pulsed, linearly polarized light irradiating normally on a surface of a bulk silicon.

Exercise-7 ~ 8 are for the linear response and the pulsed electromagnetic field calculation over the metallic nanosphere solving the time-dependent Maxwell equations, where the materials are expressed by dielectric function. The calculation method is the Finite-Difference Time-Domain (FDTD).

Final exercises (Exercise-9 ~ 10) are for geometry optimization and Ehrenfest molecular dynamics based on the TDDFT method for a single water molecule under periodic boundary condition. Currently, these are trial functions. We omit the explanations, but the input keywords are explained in *List of all input keywords*.

4.2 C2H2 (isolated molecules)

4.2.1 Exercise-1: Ground state of C2H2 molecule

In this exercise, we learn the calculation of the ground state solution of acetylene (C2H2) molecule, solving the static Kohn-Sham equation. This exercise will be useful to learn how to set up calculations in SALMON for any isolated systems such as molecules and nanoparticles. It should be noted that at present it is not possible to carry out the geometry optimization in SALMON. Therefore, atomic positions of the molecule are specified in the input file and are fixed during the calculations.

Input files

To run the code, following files are used:

file name	description
<i>C2H2_gs.inp</i>	input file that contains input keywords and their values
<i>C_rps.dat</i>	pseudopotential file for carbon atom
<i>H_rps.dat</i>	pseudopotential file for hydrogen atom

You may download the above 3 files (zipped file) from:

https://salmon-tddft.jp/webmanual/v_1_2_0/exercise_zip_files/C2H2_gs_input.zip

(zipped input and pseudopotential files)

In the input file *C2H2_gs.inp*, input keywords are specified. Most of them are mandatory to execute the ground state calculation. This will help you to prepare an input file for other systems that you want to calculate. A complete list of the input keywords that can be used in the input file can be found in *List of all input keywords*.

```
&units
  unit_system='A_eV_fs'
/
&calculation
  calc_mode = 'GS'
/
&control
  sysname = 'C2H2'
/
&system
  iperiodic = 0
  al = 16d0, 16d0, 16d0
  nstate = 5
  nelem = 2
  natom = 4
  nelec = 10
/
&pseudo
  izatom(1)=6
  izatom(2)=1
  pseudo_file(1)='C_rps.dat'
  pseudo_file(2)='H_rps.dat'
  lmax_ps(1)=1
  lmax_ps(2)=0
```

(continues on next page)

(continued from previous page)

```

lloc_ps(1)=1
lloc_ps(2)=0
/
&rgrid
dl = 0.25d0, 0.25d0, 0.25d0
/
&scf
ncg = 4
nscf = 1000
convergence = 'norm_rho_dng'
threshold_norm_rho = 1.d-15
/
&analysis
out_psi = 'y'
out_dos = 'y'
out_pdos = 'y'
out_dns = 'y'
out_elf = 'y'
/
&atomic_coor
'C' 0.000000 0.000000 0.599672 1
'H' 0.000000 0.000000 1.662257 2
'C' 0.000000 0.000000 -0.599672 1
'H' 0.000000 0.000000 -1.662257 2
/

```

We present their explanations below:

Required and recommended variables

&units

Mandatory: none

```

&units
unit_system='A_eV_fs'
/

```

This input keyword specifies the unit system to be used in the input file. If you do not specify it, atomic unit will be used. See *&units in Inputs* for detail.

For isolated systems (specified by `iperiodic = 0` in `&system`), the unit of 1/eV is used for the output files of DOS and PDOS if `unit_system = 'A_eV_fs'` is specified, while atomic unit is used if not. For other output files, the Angstrom/eV/fs units are used irrespective of the input keyword.

&calculation

Mandatory: `calc_mode`

```

&calculation
calc_mode = 'GS'
/

```

This indicates that the ground state (GS) calculation is carried out in the present job. See *&calculation in Inputs* for detail.

&control

Mandatory: none

```
&control
  sysname = 'C2H2'
/
```

'C2H2' defined by `sysname = 'C2H2'` will be used in the filenames of output files.

&system

Mandatory: `iperiodic`, `al`, `nstate`, `nelem`, `natom`

```
&system
  iperiodic = 0
  al = 16d0, 16d0, 16d0
  nstate = 5
  nelem = 2
  natom = 4
  nelec = 10
/
```

`iperiodic = 0` indicates that the isolated boundary condition will be used in the calculation. `al = 16d0, 16d0, 16d0` specifies the lengths of three sides of the rectangular parallelepiped where the grid points are prepared. `nstate = 8` indicates the number of Kohn-Sham orbitals to be solved. `nelec = 10` indicate the number of valence electrons in the system. Since the present code assumes that the system is spin saturated, `nstate` should be equal to or larger than `nelec/2`. `nelem = 2` and `natom = 4` indicate the number of elements and the number of atoms in the system, respectively. See [&system in Inputs](#) for more information.

&pseudo

Mandatory: `pseudo_file`, `izatom`

```
&pseudo
  izatom(1)=6
  izatom(2)=1
  pseudo_file(1)='C_rps.dat'
  pseudo_file(2)='H_rps.dat'
  lmax_ps(1)=1
  lmax_ps(2)=0
  lloc_ps(1)=1
  lloc_ps(2)=0
/
```

Parameters related to atomic species and pseudopotentials. `izatom(1) = 6` specifies the atomic number of the element #1. `pseudo_file(1) = 'C_rps.dat'` indicates the filename of the pseudopotential of element #1. `lmax_ps(1) = 1` and `lloc_ps(1) = 1` specify the maximum angular momentum of the pseudopotential projector and the angular momentum of the pseudopotential that will be treated as local, respectively.

&rgrid

Mandatory: `dl` or `num_rgrid`

```
&rgrid
  dl = 0.25d0, 0.25d0, 0.25d0
/
```

`dl = 0.25d0, 0.25d0, 0.25d0` specifies the grid spacings in three Cartesian directions. See [&rgrid in Inputs](#) for more information.

&scf

Mandatory: `nscf`


```
&scf
  ncg = 4
  nscf = 1000
  convergence = 'norm_rho_dng'
  threshold_norm_rho = 1.d-15
/
```

`ncg` is the number of CG iterations in solving the Khon-Sham equation. `nscf` is the number of scf iterations. For isolated systems specified by `&system/iperiodic = 0`, the scf loop in the ground state calculation ends before the number of the scf iterations reaches `nscf`, if a convergence criterion is satisfied. There are several options for the convergence check. If the value of `norm_rho_dng` is specified, the convergence is examined by the squared difference of the electron density,

&analysis

The following input keywords specify whether the output files are created or not after the calculation.

```
&analysis
  out_psi = 'y'
  out_dos = 'y'
  out_pdos = 'y'
  out_dns = 'y'
  out_elf = 'y'
/
```

&atomic_coor

Mandatory: `atomic_coor` or `atomic_red_coor` (it may be provided as a separate file)

```
&atomic_coor
'C'  0.000000  0.000000  0.599672  1
'H'  0.000000  0.000000  1.662257  2
'C'  0.000000  0.000000 -0.599672  1
'H'  0.000000  0.000000 -1.662257  2
/
```

Cartesian coordinates of atoms. The first column indicates the element. Next three columns specify Cartesian coordinates of the atoms. The number in the last column labels the element.

Output files

After the calculation, following output files are created in the directory that you run the code,

file name	description
<i>C2H2_info.data</i>	information on ground state solution
<i>dns.cube</i>	a cube file for electron density
<i>elf.cube</i>	electron localization function (ELF)
<i>psi1.cube, psi2.cube, ...</i>	electron orbitals
<i>dos.data</i>	density of states
<i>pdos1.data, pdos2.data, ...</i>	projected density of states
<i>C2H2_gs.bin</i>	binary output file to be used in the real-time calculation

You may download the above files (zipped file, except for the binary file *C2H2_gs.bin*) from:

https://salmon-tddft.jp/webmanual/v_1_2_0/exercise_zip_files/C2H2_gs_output.zip

(zipped output files)

Main results of the calculation such as orbital energies are included in *C2H2_info.data*. Explanations of the *C2H2_info.data* and other output files are below:

C2H2_info.data

Calculated orbital and total energies as well as parameters specified in the input file are shown in this file.

```
Total number of iteration = 49
Number of states = 5
Number of electrons = 5
Total energy (eV) = -339.7041368688747
1-particle energies (eV)
  1      -18.4492      2      -13.9884      3      -12.3935      4      -7.3310
  5      -7.3310
Size of the box (A) = 15.99999363 15.99999363 15.99999363
Grid spacing (A) = 0.24999990 0.24999990 0.24999990
Number of atoms = 4
iZatom( 1) = 6
iZatom( 2) = 1
Ref. and max angular momentum and pseudo-core radius of PP (A)
( 1) Ref, Max, Rps = 1 1 0.800
( 2) Ref, Max, Rps = 0 0 0.800
```

dns.cube

A cube file for electron density. For isolated systems (specified by `iperiodic = 0` in `&system`), atomic unit is adopted in all cube files.

elf.cube

A cube file for electron localization function (ELF).

psi1.cube, psi2.cube, ...

Cube files for electron orbitals. The number in the filename indicates the index of the orbital..

dos.data

A file for density of states. The units used in this file are affected by the input parameter, `unit_energy` in `&unit`.

```
# Density of States
# Energy[eV] DOS[1/eV]
#-----
-21.22853 0.00000000
-21.20073 0.00000000
-21.17294 0.00000000
-21.14514 0.00000000
-21.11735 0.00000000
...
-7.38656 13.67306519
-7.35876 15.35302960
-7.33097 15.95769122
-7.30317 15.35301925
-7.27538 13.67304675
...
```

(continues on next page)

(continued from previous page)

```

-4.66264    0.00000000
-4.63484    0.00000000
-4.60705    0.00000000
-4.57925    0.00000000
-4.55146    0.00000000

```

pdos1.data, pdos2.data, ...

Files for projected density of states. The units used in this file are affected by the input parameter, `unit_energy` in `&unit`. The number in the filename indicates the order of atoms specified in `&atomic_coor`.

```

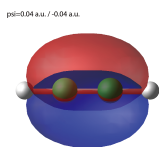
# Projected Density of States
# Energy[eV]  PDOS(l=0) [1/eV]  PDOS(l=1) [1/eV]
#-----
-21.22853    0.00000000    0.00000000
-21.20073    0.00000000    0.00000000
-21.17294    0.00000000    0.00000000
-21.14514    0.00000000    0.00000000
-21.11735    0.00000000    0.00000000
...
-7.38656    0.00000000    18.33035096
-7.35876    0.00000000    20.58254071
-7.33097    0.00000000    21.39316068
-7.30317    0.00000000    20.58252684
-7.27538    0.00000000    18.33032625
...
-4.66264    0.00000000    0.00000000
-4.63484    0.00000000    0.00000000
-4.60705    0.00000000    0.00000000
-4.57925    0.00000000    0.00000000
-4.55146    0.00000000    0.00000000

```

We show several image that are created from the output files.

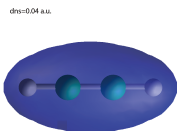
- **Highest occupied molecular orbital (HOMO)**

The output files *psi1.cube*, *psi2.cube*, ... are used to create the image.



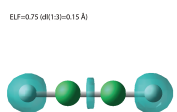
- **Electron density**

The output files *dns.cube*, ... are used to create the image.



- **Electron localization function**

The output files *elf.cube*, ... are used to create the image.



4.2.2 Exercise-2: Polarizability and photoabsorption of C₂H₂ molecule

In this exercise, we learn the linear response calculation in the acetylene (C₂H₂) molecule, solving the time-dependent Kohn-Sham equation. The linear response calculation provides the polarizability and the oscillator strength distribution of the molecule. This exercise should be carried out after finishing the ground state calculation that was explained in *Exercise-1*. In the calculation, an impulsive perturbation is applied to all electrons in the C₂H₂ molecule along the molecular axis which we take *z* axis. Then a time evolution calculation is carried out without any external fields. During the calculation, the electric dipole moment is monitored. After the time evolution calculation, a time-frequency Fourier transformation is carried out for the electric dipole moment to obtain the frequency-dependent polarizability. The imaginary part of the frequency-dependent polarizability is proportional to the oscillator strength distribution and the photoabsorption cross section.

Input files

To run the code, the input file *C2H2_rt_response.inp* that contains input keywords and their values for the linear response calculation is required. The binary file *C2H2_gs.bin* that is created in the ground state calculation and pseudopotential files are also required. The pseudopotential files should be the same as those used in the ground state calculation.

file name	description
<i>C2H2_rt_response.inp</i>	input file that contains input keywords and their values
<i>C_rps.dat</i>	pseudopotential file for carbon
<i>H_rps.dat</i>	pseudopotential file for hydrogen
<i>C2H2_gs.bin</i>	binary file created in the ground state calculation

You may download the *C2H2_rt_response.inp* file (zipped file) from:

https://salmon-tddft.jp/webmanual/v_1_2_0/exercise_zip_files/C2H2_rt_response_input.zip

(zipped input file)

In the input file *C2H2_rt_response.inp*, input keywords are specified. Most of them are mandatory to execute the linear response calculation. This will help you to prepare the input file for other systems that you want to calculate. A complete list of the input keywords that can be used in the input file can be found in the downloaded file *SALMON/manual/input_variables.md*.

```

&units
  unit_system='A_eV_fs'
/
&calculation
  calc_mode='RT'
/
&control
  sysname = 'C2H2'
/
&system
  iperiodic = 0
  al = 16d0, 16d0, 16d0
  nstate = 5
  nelelem = 2
  natom = 4
  nelecc = 10
/
&pseudo
  izatom(1)=6
  izatom(2)=1
  pseudo_file(1)='C_rps.dat'
  pseudo_file(2)='H_rps.dat'
  lmax_ps(1)=1
  lmax_ps(2)=0
  lloc_ps(1)=1
  lloc_ps(2)=0
/
&tgrid
  dt=1.25d-3
  nt=5000
/
&emfield
  ae_shape1 = 'impulse'
  epcdir_re1 = 0.d0,0.d0,1.d0
/
&atomic_coor
'C'    0.000000    0.000000    0.599672    1
'H'    0.000000    0.000000    1.662257    2
'C'    0.000000    0.000000   -0.599672    1
'H'    0.000000    0.000000   -1.662257    2
/

```

We present their explanations below:

Required and recommended variables

&units

Mandatory: none

```

&units
  unit_system='A_eV_fs'
/

```

This input keyword specifies the unit system to be used in the input file. If you do not specify it, atomic unit will be used. See *&units in Inputs* for detail.

&calculation

Mandatory: calc_mode

```
&calculation
  calc_mode = 'RT'
/
```

This indicates that the real time (RT) calculation is carried out in the present job. See *&calculation in Inputs* for detail.

&control

Mandatory: none

```
&control
  sysname = 'C2H2'
/
```

'C2H2' defined by `sysname = 'C2H2'` will be used in the filenames of output files.

&system

Mandatory: `iperiodic`, `al`, `nstate`, `nelem`, `natom`

```
&system
  iperiodic = 0
  al = 16d0, 16d0, 16d0
  nstate = 5
  nelem = 2
  natom = 4
  nelec = 10
/
```

These input keywords and their values should be the same as those used in the ground state calculation. See *&system in Exercise-1*.

&pseudo

Mandatory: `pseudo_file`, `izatom`

```
&pseudo
  izatom(1)=6
  izatom(2)=1
  pseudo_file(1)='C_rps.dat'
  pseudo_file(2)='H_rps.dat'
  lmax_ps(1)=1
  lmax_ps(2)=0
  lloc_ps(1)=1
  lloc_ps(2)=0
/
```

These input keywords and their values should be the same as those used in the ground state calculation. See *&pseudo in Exercise-1*.

&tgrid

Mandatory: `dt`, `Nt`

```
&tgrid
  dt=1.25d-3
  nt=5000
/
```

`dt=1.25d-3` specifies the time step of the time evolution calculation. `nt=5000` specifies the number of time steps in the calculation.

&emfield

Mandatory: ae_shape1

```
&emfield
  ae_shape1 = 'impulse'
  ekdir_re1 = 0.d0,0.d0,1.d0
/
```

ae_shape1 = 'impulse' indicates that a weak impulse is applied to all electrons at $t=0$ ekdir_re1 (3) specifies a unit vector that indicates the direction of the impulse. See *&emfield in Inputs* for details.

&atomic_coor

Mandatory: atomic_coor or atomic_red_coor (it may be provided as a separate file)

```
&atomic_coor
'C'    0.000000    0.000000    0.599672    1
'H'    0.000000    0.000000    1.662257    2
'C'    0.000000    0.000000   -0.599672    1
'H'    0.000000    0.000000   -1.662257    2
/
```

Cartesian coordinates of atoms. The first column indicates the element. Next three columns specify Cartesian coordinates of the atoms. The number in the last column labels the element. They must be the same as those in the ground state calculation.

Output files

After the calculation, following output files are created in the directory that you run the code,

file name	description
<i>C2H2_lr.data</i>	polarizability and oscillator strength distribution as functions of energy
<i>C2H2_p.data</i>	components of dipole moment as functions of time

You may download the above files (zipped file) from:

https://salmon-tddft.jp/webmanual/v_1_2_0/exercise_zip_files/C2H2_rt_response_output.zip

(zipped output files)

Explanations of the output files are below:

C2H2-p.data

For time steps from 1 to nt,

- 1 column: time
- 2-4 columns: x,y,z components of the dipole moment
- 5 column: total energy of the system

```
# time[fs],      dipoleMoment (x, y, z) [A],      Energy[eV]
0.12500E-02  0.20197641E-09  0.12143673E-09  0.27407578E-02  -0.33969042E+03
0.25000E-02  -0.23127543E-09  -0.38283389E-09  0.54651286E-02  -0.33969040E+03
0.37500E-02  -0.24342401E-08  -0.25180060E-08  0.81587485E-02  -0.33969039E+03
```

(continues on next page)

(continued from previous page)

```

0.50000E-02 -0.63429482E-08 -0.62611945E-08 0.10810857E-01 -0.33969038E+03
0.62500E-02 -0.11655064E-07 -0.11294666E-07 0.13413805E-01 -0.33969038E+03
...
0.62450E+01 -0.21648194E-05 -0.12589717E-05 -0.15217299E-02 -0.33969011E+03
0.62463E+01 -0.22246530E-05 -0.12919132E-05 -0.14111473E-02 -0.33969011E+03
0.62475E+01 -0.22836011E-05 -0.13244333E-05 -0.12951690E-02 -0.33969011E+03
0.62488E+01 -0.23416512E-05 -0.13565206E-05 -0.11738782E-02 -0.33969011E+03
0.62500E+01 -0.23987916E-05 -0.13881638E-05 -0.10473800E-02 -0.33969011E+03

```

C2H2_lr.data

For energy steps from 0 to nenergy,

- 1 column: energy
- 2-4 columns: x,y,z components of real part of the polarizability (time-frequency Fourier transformation of the dipole moment)
- 5-7 columns: x,y,z components of imaginary part of the polarizability (time-frequency Fourier transformation of the dipole moment)
- 8-10 columns: x,y,z components of power spectrum of the dipole moment

```

# energy[eV], Re[alpha](x,y,z)[A**3], Im[alpha](x,y,z)[A**3], S(x,y,z)[1/eV]
0.00000E+00 0.90041681E-02 0.42900323E-02 0.47230167E+01 0.00000000E+00 0.
↪00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00
0.10000E-01 0.89986618E-02 0.42874031E-02 0.47230192E+01 0.25932415E-03 0.
↪12379226E-03 0.18663776E-03 0.15045807E-07 0.71823406E-08 0.10828593E-07
0.20000E-01 0.89821593E-02 0.42795232E-02 0.47230267E+01 0.51808569E-03 0.
↪24731589E-03 0.37320742E-03 0.60117942E-07 0.28698192E-07 0.43306470E-07
0.30000E-01 0.89547084E-02 0.42664157E-02 0.47230393E+01 0.77572398E-03 0.
↪37030322E-03 0.55964230E-03 0.13502090E-06 0.64454205E-07 0.97410171E-07
0.40000E-01 0.89163894E-02 0.42481186E-02 0.47230569E+01 0.10316824E-02 0.
↪49248844E-03 0.74587862E-03 0.23942997E-06 0.11429535E-06 0.17310143E-06
0.50000E-01 0.88673137E-02 0.42246853E-02 0.47230796E+01 0.12854100E-02 0.
↪61360857E-03 0.93185683E-03 0.37289297E-06 0.17800571E-06 0.27032843E-06
...
0.99601E+01 0.15674984E-03 0.37403402E-04 -0.44437601E+00 -0.10631864E-03 -0.
↪14544171E-03 0.27060202E+01 -0.61438595E-05 -0.84046729E-05 0.15637340E+00
0.99701E+01 0.15448331E-03 0.37400902E-04 -0.14920113E+00 -0.10649714E-03 -0.
↪14698080E-03 0.25947889E+01 -0.61603535E-05 -0.85021406E-05 0.15009620E+00
0.99801E+01 0.15224601E-03 0.37478652E-04 0.14911900E+00 -0.10665066E-03 -0.
↪14847068E-03 0.24965858E+01 -0.61754213E-05 -0.85969375E-05 0.14456047E+00
0.99901E+01 0.15003254E-03 0.37632621E-04 0.45012407E+00 -0.10678183E-03 -0.
↪14990965E-03 0.24115316E+01 -0.61892122E-05 -0.86889561E-05 0.13977547E+00
0.10000E+02 0.14783807E-03 0.37858911E-04 0.75334591E+00 -0.10689373E-03 -0.
↪15129625E-03 0.23397373E+01 -0.62019000E-05 -0.87781030E-05 0.13574993E+00

```

4.2.3 Exercise-3: Electron dynamics in C2H2 molecule under a pulsed electric field

In this exercise, we learn the calculation of the electron dynamics in the acetylene (C2H2) molecule under a pulsed electric field, solving the time-dependent Kohn-Sham equation. As outputs of the calculation, such quantities as the total energy and the electric dipole moment of the system as functions of time are calculated. This tutorial should be

carried out after finishing the ground state calculation that was explained in *Exercise-1*. In the calculation, a pulsed electric field that has \cos^2 envelope shape is applied. The parameters that characterize the pulsed field such as magnitude, frequency, polarization direction, and carrier envelope phase are specified in the input file.

Input files

To run the code, following files are used. The *C2H2_gs.bin* file is created in the ground state calculation. Pseudopotential files are already used in the ground state calculation. Therefore, *C2H2_rt_pulse.inp* that specifies input keywords and their values for the pulsed electric field calculation is the only file that the users need to prepare.

file name	description
<i>C2H2_rt_pulse.inp</i>	input file that contain input keywords and their values.
<i>C_rps.dat</i>	pseudopotential file for Carbon
<i>H_rps.dat</i>	pseudopotential file for Hydrogen
<i>C2H2_gs.bin</i>	binary file created in the ground state calculation

You may download the *C2H2_rt_pulse.inp* file (zipped file) from:

https://salmon-tddft.jp/webmanual/v_1_2_0/exercise_zip_files/C2H2_rt_pulse_input.zip

In the input file *C2H2_rt_pulse.inp*, input keywords are specified. Most of them are mandatory to execute the calculation of electron dynamics induced by a pulsed electric field. This will help you to prepare the input file for other systems and other pulsed electric fields that you want to calculate. A complete list of the input keywords that can be used in the input file can be found in the downloaded file *SALMON/manual/input_variables.md*.

```
&units
  unit_system='A_eV_fs'
/
&calculation
  calc_mode='RT'
/
&control
  sysname = 'C2H2'
/
&system
  iperiodic = 0
  a1 = 16d0, 16d0, 16d0
  nstate = 5
  nelem = 2
  natom = 4
  nelec = 10
/
&pseudo
  izatom(1)=6
  izatom(2)=1
  pseudo_file(1)='C_rps.dat'
  pseudo_file(2)='H_rps.dat'
  lmax_ps(1)=1
  lmax_ps(2)=0
  lloc_ps(1)=1
  lloc_ps(2)=0
/
&tgrid
```

(continues on next page)

(continued from previous page)

```

dt=1.25d-3
nt=4800
/
&emfield
  ae_shape1 = 'Ecos2'
  ekdir_re1 = 0.d0,0.d0,1.d0
  r_laser_int_wcm2_1 = 1.d8
  omega1=9.28d0
  pulse_twl=6.d0
  phi_cep1=0.75d0
/
&atomic_coor
'C'    0.000000    0.000000    0.599672    1
'H'    0.000000    0.000000    1.662257    2
'C'    0.000000    0.000000   -0.599672    1
'H'    0.000000    0.000000   -1.662257    2
/

```

We present explanations of the input keywords that appear in the input file below:

required and recommended variables

&units

Mandatory: none

```

&units
  unit_system='A_eV_fs'
/

```

This input keyword specifies the unit system to be used in the input file. If you do not specify it, atomic unit will be used. See *&units in Inputs* for detail.

&calculation

Mandatory: calc_mode

```

&calculation
  calc_mode = 'RT'
/

```

This indicates that the real time (RT) calculation is carried out in the present job. See *&calculation in Inputs* for detail.

&control

Mandatory: none

```

&control
  sysname = 'C2H2'
/

```

'C2H2' defined by sysname = 'C2H2' will be used in the filenames of output files.

&system

Mandatory: iperiodic, al, nstate, nelemt, natom

```

&system
  iperiodic = 0
  al = 16d0, 16d0, 16d0

```

(continues on next page)

(continued from previous page)

```
nstate = 5
nelem = 2
natom = 4
nelec = 10
/
```

These input keywords and their values should be the same as those used in the ground state calculation. See *&system in Exercise-1*.

&pseudo

Mandatory: pseudo_file, izatom

```
&pseudo
  izatom(1)=6
  izatom(2)=1
  pseudo_file(1)='C_rps.dat'
  pseudo_file(2)='H_rps.dat'
  lmax_ps(1)=1
  lmax_ps(2)=0
  lloc_ps(1)=1
  lloc_ps(2)=0
/
```

These input keywords and their values should be the same as those used in the ground state calculation. See *&pseudo in Exercise-1*.

&tgrid

Mandatory: dt, Nt

```
&tgrid
  dt=1.25d-3
  nt=4800
/
```

dt=1.25d-3 specifies the time step of the time evolution calculation. Nt=4800 specifies the number of time steps in the calculation.

&emfield

Mandatory: ae_shape1, ekdir_re1, {rlaser_int1 or amplitude1}, omega1, pulse_tw1, phi_cep1

```
&emfield
  ae_shape1 = 'Ecos2'
  ekdir_re1 = 0.d0,0.d0,1.d0
  rlaser_int_wcm2_1 = 1.d8
  omega1=9.28d0
  pulse_tw1=6.d0
  phi_cep1=0.75d0
/
```

ae_shape1 = 'Ecos2' indicates that the envelope of the pulsed electric field has a \cos^2 shape.

ekdir_re1 = 0.d0,0.d0,1.d0 specifies the real part of the unit polarization vector of the pulsed electric field. Using the real polarization vector, it describes a linearly polarized pulse.

laser_int_wcm2_1 = 1.d8 specifies the maximum intensity of the applied electric field in unit of W/cm^2 .

omega1=9.26d0 specifies the average photon energy (frequency multiplied with \hbar).

pulse_tw1=6.d0 specifies the pulse duration. Note that it is not the FWHM but a full duration of the \cos^2 envelope.

phi_cep1=0.75d0 specifies the carrier envelope phase of the pulse. As noted above, 'phi_cep1' must be 0.75 (or 0.25) if one employs 'Ecos2' pulse shape, since otherwise the time integral of the electric field does not vanish.

See *&emfield in Inputs* for details.

&atomic_coor

Mandatory: atomic_coor or atomic_red_coor (it may be provided as a separate file)

```
&atomic_coor
'C'  0.000000  0.000000  0.599672  1
'H'  0.000000  0.000000  1.662257  2
'C'  0.000000  0.000000 -0.599672  1
'H'  0.000000  0.000000 -1.662257  2
/
```

Cartesian coordinates of atoms. The first column indicates the element. Next three columns specify Cartesian coordinates of the atoms. The number in the last column labels the element. They must be the same as those in the ground state calculation.

Output files

After the calculation, following output files are created in the directory that you run the code,

file name	description
<i>C2H2_p.data</i>	components of the electric dipole moment as functions of time
<i>C2H2_ps.data</i>	power spectrum that is obtained by a time-frequency Fourier transformation of the electric dipole moment

You may download the above files (zipped file) from:

https://salmon-tddft.jp/webmanual/v_1_2_0/exercise_zip_files/C2H2_rt_pulse_output.zip

Explanations of the files are described below:

C2H2_p.data

For time steps from 1 to nt,

- 1 column: time
- 2-4 columns: x,y,z components of the dipole moment
- 5 column: total energy of the system

```
# time[fs],      dipoleMoment (x, y, z) [A],      Energy[eV]
0.12500E-02  0.18257556E-09  0.11097584E-09  0.48217422E-09  -0.33970414E+03
0.25000E-02  0.91251666E-09  0.54016872E-09  0.19424475E-08  -0.33970414E+03
0.37500E-02  0.24945802E-08  0.14520397E-08  0.43921301E-08  -0.33970414E+03
0.50000E-02  0.50230110E-08  0.29055651E-08  0.78162260E-08  -0.33970414E+03
0.62500E-02  0.83018473E-08  0.48072377E-08  0.12178890E-07  -0.33970413E+03
```

...

(continues on next page)

(continued from previous page)

0.59950E+01	0.10101410E-04	0.55756362E-05	0.32250943E-03	-0.33970394E+03
0.59963E+01	0.10109316E-04	0.55775491E-05	0.38471398E-03	-0.33970394E+03
0.59975E+01	0.10115053E-04	0.55780512E-05	0.44680913E-03	-0.33970394E+03
0.59988E+01	0.10118632E-04	0.55771582E-05	0.50877609E-03	-0.33970394E+03
0.60000E+01	0.10120064E-04	0.55748807E-05	0.57059604E-03	-0.33970394E+03

C2H2_ps.data

For energy steps from 0 to nenergy,

- 1 column: energy
- 2-4 columns: x,y,z components of the real part of the time-frequency Fourier transformation of the dipole moment
- 5-7 columns: x,y,z components of imaginary part of the time-frequency Fourier transformation of the dipole moment
- 8-10 columns: x,y,z components of power spectrum of the dipole moment

```
# energy[eV], Re[alpha](x,y,z)[A*fs], Im[alpha](x,y,z)[A*fs], I(x,y,z)[A**2*fs**2]
0.00000E+00 0.12836214E-01 0.60771681E-02 -0.28240863E-02 0.00000000E+00 0.
↪00000000E+00 0.00000000E+00 0.16476838E-03 0.36931972E-04 0.79754632E-05
0.10000E-01 0.12829079E-01 0.60737829E-02 -0.28241953E-02 0.35253318E-03 0.
↪16719128E-03 -0.41437502E-04 0.16470954E-03 0.36918792E-04 0.79777964E-05
0.20000E-01 0.12807693E-01 0.60636364E-02 -0.28245142E-02 0.70436985E-03 0.
↪33405211E-03 -0.83009748E-04 0.16453313E-03 0.36879277E-04 0.79847710E-05
0.30000E-01 0.12772113E-01 0.60467557E-02 -0.28250177E-02 0.10548158E-02 0.
↪50025311E-03 -0.12484976E-03 0.16423951E-03 0.36813507E-04 0.79963126E-05
0.40000E-01 0.12722434E-01 0.60231857E-02 -0.28256644E-02 0.14031812E-02 0.
↪66546701E-03 -0.16708711E-03 0.16382925E-03 0.36721612E-04 0.80122973E-05
0.50000E-01 0.12658789E-01 0.59929893E-02 -0.28263966E-02 0.17487830E-02 0.
↪82936975E-03 -0.20984627E-03 0.16330319E-03 0.36603775E-04 0.80325532E-05
...
0.99601E+01 0.38757368E-03 0.19783358E-03 0.11087376E+01 -0.27465428E-03 -0.
↪29515838E-03 0.10183658E+01 0.22564833E-06 0.12625659E-06 0.22663679E+01
0.99701E+01 0.38446279E-03 0.19754997E-03 0.10416956E+01 -0.27241140E-03 -0.
↪29512921E-03 0.10381647E+01 0.22201960E-06 0.12612724E-06 0.21629157E+01
0.99801E+01 0.38136406E-03 0.19733388E-03 0.97519659E+00 -0.27017795E-03 -0.
↪29508231E-03 0.10542348E+01 0.21843467E-06 0.12601423E-06 0.20624194E+01
0.99901E+01 0.37827032E-03 0.19718146E-03 0.90943725E+00 -0.26795413E-03 -0.
↪29501502E-03 0.10666811E+01 0.21488785E-06 0.12591439E-06 0.19648847E+01
0.10000E+02 0.37517469E-03 0.19708886E-03 0.84460457E+00 -0.26574105E-03 -0.
↪29492512E-03 0.10756186E+01 0.21137435E-06 0.12582485E-06 0.18703122E+01
```

4.3 Crystalline silicon (periodic solids)

4.3.1 Exercise-4: Dielectric function of crystalline silicon

In this exercise, we learn the linear response calculation of the crystalline silicon of a diamond structure. Calculation is done in a cubic unit cell that contains eight silicon atoms. Since the ground state calculation costs much less computational time than the time evolution calculation, both calculations are successively executed. After finishing the ground state calculation, an impulsive perturbation is applied to all electrons in the unit cell along z direction. Since

the dielectric function is isotropic in the diamond structure, calculated dielectric function should not depend on the direction of the perturbation. During the time evolution, electric current averaged over the unit cell volume is calculated. A time-frequency Fourier transformation of the electric current gives us a frequency-dependent conductivity. The dielectric function may be obtained from the conductivity using a standard relation.

Input files

To run the code, following files are used:

file name	description
<i>Si_gs_rt_response.inp</i>	input file that contain input keywords and their values.
<i>Si_rps.dat</i>	pseudopotential file of silicon

You may download the above 2 files (zipped file) from:

https://salmon-tddft.jp/webmanual/v_1_2_0/exercise_zip_files/Si_gs_rt_response_input.zip

In the input file *Si_gs_rt_response.inp*, input keywords are specified. Most of them are mandatory to execute the calculation. This will help you to prepare the input file for other systems that you want to calculate. A complete list of the input keywords that can be used in the input file can be found in the downloaded file *SALMON/manual/input_variables.md*.

```
&calculation
  calc_mode = 'GS_RT'
/
&control
  sysname = 'Si'
/
&units
  unit_system = 'a.u.'
/
&system
  iperiodic = 3
  a1 = 10.26d0, 10.26d0, 10.26d0
  nstate = 32
  nelec = 32
  nelemp = 1
  natom = 8
/
&pseudo
  izatom(1) = 14
  pseudo_file(1) = './Si_rps.dat'
  lloc_ps(1) = 2
/
&functional
  xc = 'PZ'
/
&rgrid
  num_rgrid = 12, 12, 12
/
&kgrid
  num_kgrid = 4, 4, 4
/
&tgrid
```

(continues on next page)

(continued from previous page)

```

nt = 3000
dt = 0.16
/
&propagation
  propagator = 'etrs'
/
&scf
  ncg = 5
  nscf = 120
/
&emfield
  trans_longi = 'tr'
  ae_shape1 = 'impulse'
  ekdir_rel = 0., 0., 1.
/
&analysis
  nenergy = 1000
  de = 0.001
/
&atomic_red_coor
'Si'      .0      .0      .0      1
'Si'      .25     .25     .25     1
'Si'      .5      .0      .5      1
'Si'      .0      .5      .5      1
'Si'      .5      .5      .0      1
'Si'      .75     .25     .75     1
'Si'      .25     .75     .75     1
'Si'      .75     .75     .25     1
/

```

We present explanations of the input keywords that appear in the input file below:

&calculation

Mandatory: calc_mode

```

&calculation
  calc_mode = 'GS_RT'
/

```

This indicates that the ground state (GS) and the real time (RT) calculations are carried out sequentially in the present job. See *&calculation in Inputs* for detail.

&control

Mandatory: none

```

&control
  sysname = 'Si'
/

```

'Si' defined by sysname = 'C2H2' will be used in the filenames of output files.

&system

Mandatory: periodic, al, state, nelem, natom

```

&system
  iperiodic = 3

```

(continues on next page)

(continued from previous page)

```

al = 10.26d0,10.26d0,10.26d0
nstate = 32
nelec = 32
nelem = 1
natom = 8
/

```

`iperiodic = 3` indicates that three dimensional periodic boundary condition (bulk crystal) is assumed. `al = 10.26d0, 10.26d0, 10.26d0` specifies the lattice constants of the unit cell. `nstate = 32` indicates the number of Kohn-Sham orbitals to be solved. `nelec = 32` indicate the number of valence electrons in the system. `nelem = 1` and `natom = 8` indicate the number of elements and the number of atoms in the system, respectively. See *&system in Inputs* for more information.

&pseudo

```

&pseudo
  izatom(1)=14
  pseudo_file(1) = './Si_rps.dat'
  lloc_ps(1)=2
/

```

`izatom(1) = 14` indicates the atomic number of the element #1. `pseudo_file(1) = 'Si_rps.dat'` indicates the pseudopotential filename of element #1. `lloc_ps(1) = 2` indicate the angular momentum of the pseudopotential that will be treated as local.

&functional

```

&functional
  xc = 'PZ'
/

```

This indicates that the adiabatic local density approximation with the Perdew-Zunger functional is used. We note that meta-GGA functionals that reasonably reproduce the band gap of various insulators may also be used in the calculation of periodic systems. See *&functional in Inputs* for detail.

&rgrid

Mandatory: `dl` or `num_rgrid`

```

&rgrid
  num_rgrid = 12,12,12
/

```

`num_rgrid=12,12,12` specifies the number of the grids for each Cartesian direction. See *&rgrid in Inputs* for more information.

&kgrid

Mandatory: none

This input keyword provides grid spacing of k-space for periodic systems.

```

&kgrid
  num_kgrid = 4,4,4
/

```

&tgrid


```
&tgrid
  nt=3000
  dt=0.16
/
```

`dt=0.16` specifies the time step of the time evolution calculation. `nt=3000` specifies the number of time steps in the calculation.

&propagation

```
&propagation
  propagator='etrs'
/
```

`propagator = 'etrs'` indicates the use of enforced time-reversal symmetry propagator. See *&propagation in Inputs* for more information.

&scf

Mandatory: `nscf`

This input keywords specify parameters related to the self-consistent field calculation.

```
&scf
  ncg = 5
  nscf = 120
/
```

`ncg = 5` is the number of conjugate-gradient iterations in solving the Kohn-Sham equation. Usually this value should be 4 or 5. `nscf = 120` is the number of scf iterations.

&emfield

Mandatory: `ae_shape1`

```
&emfield
  trans_longi = 'tr'
  ae_shape1 = 'impulse'
  ekdir_rel = 0.,0.,1.
/
```

`ae_shape1 = 'impulse'` indicates that a weak impulsive field is applied to all electrons at $t=0$

`ekdir_rel(3)` specify a unit vector that indicates the direction of the impulse.

`trans_longi = 'tr'` specifies the treatment of the polarization in the time evolution calculation, transverse for 'tr' and longitudinal for 'lo'.

See *&emfield in Inputs* for detail.

&analysis

```
&analysis
  nenergy=1000
  de=0.001
/
```

`nenergy=1000` specifies the number of energy steps, and `de=0.001` specifies the energy spacing in the time-frequency Fourier transformation.

&atomic_red_coor

Mandatory: atomic_coor or atomic_red_coor (they may be provided as a separate file)

```
&atomic_red_coor
'Si'   .0   .0   .0   1
'Si'   .25  .25  .25  1
'Si'   .5   .0   .5   1
'Si'   .0   .5   .5   1
'Si'   .5   .5   .0   1
'Si'   .75  .25  .75  1
'Si'   .25  .75  .75  1
'Si'   .75  .75  .25  1
/
```

Cartesian coordinates of atoms are specified in a reduced coordinate system. First column indicates the element, next three columns specify reduced Cartesian coordinates of the atoms, and the last column labels the element.

Output files

After the calculation, following output files are created in the directory that you run the code,

file name	description
<i>Si_gs_info.data</i>	information of ground state calculation
<i>Si_eigen.data</i>	energy eigenvalues of orbitals
<i>Si_k.data</i>	information on k-points
<i>Si_rt.data</i>	electric field, vector potential, and current as functions of time
<i>Si_force.data</i>	force acting on atoms
<i>Si_lr.data</i>	Fourier spectra of the dielectric functions
<i>Si_gs_rt_response.out</i>	standard output file

You may download the above files (zipped file) from:

https://salmon-tddft.jp/webmanual/v_1_2_0/exercise_zip_files/Si_gs_rt_response_output.zip

Explanations of the output files are described below:

Si_gs_info.data

Results of the ground state as well as input parameters are provided.

```
#-----
#grid information-----
#aL = 10.2600000000000 10.2600000000000 10.2600000000000
#a1(1),a1(2),a1(3) = 10.2600000000000 10.2600000000000
10.2600000000000
#aLx,aLy,aLz = 10.2600000000000 10.2600000000000
10.2600000000000
#bLx,bLy,bLz = 0.612396228769940 0.612396228769940
0.612396228769940
#Nd = 4
#NLx,NLy,NLz= 12 12 12
#NL = 1728
#Hx,Hy,Hz = 0.855000000000000 0.855000000000000
0.855000000000000
#(pi/max(Hx,Hy,Hz))**2 = 13.5010490764192
```

(continues on next page)

(continued from previous page)

```

#(pi/Hx)**2+(pi/Hy)**2+(pi/Hz)**2 = 40.5031472292576
#Hxyz = 0.625026375000000
#NKx,NKy,NKz= 4 4 4
#NKxyz = 64
#Sym= 1
#NK = 64
#NEwald, aEwald = 4 0.500000000000000
#-----
#GS calc. option-----
#FSset_option =n
#Ncg= 5
#Nmemory_MB,alpha_MB = 8 0.750000000000000
#NFSset_start,NFSset_every = 75 25
#Nscf= 120
#Nscf_conv= 120
#NI,NE= 8 1
#Zatom= 14
#Lref= 2
#i,Kion(ia)(Rion(j,a),j=1,3)
# 1 1
# 0.000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000
# 2 1
# 2.565000000000000 2.565000000000000 2.565000000000000
# 3 1
# 5.130000000000000 0.000000000000000E+000 5.130000000000000
# 4 1
# 0.000000000000000E+000 5.130000000000000 5.130000000000000
# 5 1
# 5.130000000000000 5.130000000000000 0.000000000000000E+000
# 6 1
# 7.695000000000000 2.565000000000000 7.695000000000000
# 7 1
# 2.565000000000000 7.695000000000000 7.695000000000000
# 8 1
# 7.695000000000000 7.695000000000000 2.565000000000000
#-----
#GS information-----
#NB,Nelec= 32 32
#Eall = -31.2658878806236
#ddns(iter = Nscf_conv) 2.798849279746559E-010
#ddns_abs_1e(iter = Nscf_conv) 2.364732236264119E-010
#esp_var_ave(iter = Nscf_conv) 1.196976937606010E-009
#esp_var_max(iter = Nscf_conv) 4.031276129792963E-009
#NBoccmx is 16
#-----
#band information-----
#Bottom of VB -0.194802063980608
#Top of VB 0.216731478175047
#Bottom of CB 0.255681914576368
#Top of CB 0.533214678236198
#Fundamental gap 3.895043640132098E-002
#Fundamental gap[eV] 1.05990369517819
#BG between same k-point 3.895043648321342E-002
#BG between same k-point[eV] 1.05990369740661
#Physically upper bound of CB for DOS 0.454100922291231
#Physically upper bound of CB for eps(omega) 0.609752486428134
#-----

```

(continues on next page)

(continued from previous page)

#iter	total-energy	ddns/nelec	esp_var_ave	esp_var_max
1	-0.2059780903E+02	0.5134199377E+00	0.1332473220E-01	0.1986049398E-01
2	-0.2600097163E+02	0.3186108570E+00	0.1526707771E-01	0.2520724900E-01
3	-0.2866336088E+02	0.1363849859E+00	0.6359704895E-02	0.1247448390E-01
4	-0.3006244467E+02	0.1245614607E+00	0.5868323970E-02	0.1942874074E-01
5	-0.3096872596E+02	0.7495214064E-01	0.2566344769E-02	0.1102001262E-01
...				
115	-0.3126588788E+02	0.1355175468E-09	0.1208579378E-08	0.4031265522E-08
116	-0.3126588788E+02	0.1452261250E-09	0.1204317051E-08	0.4031272647E-08
117	-0.3126588788E+02	0.1419175726E-09	0.1198067051E-08	0.4031255783E-08
118	-0.3126588788E+02	0.1686476198E-09	0.1198945057E-08	0.4031251395E-08
119	-0.3126588788E+02	0.2159059511E-09	0.1200809994E-08	0.4666412657E-08
120	-0.3126588788E+02	0.2364732236E-09	0.1196976938E-08	0.4031276130E-08

Si_eigen.data

Orbital energies in the ground state calculation.

```
# Ground state eigenenergies
# ik: k-point index
# ib: Band index
# energy: Eigenenergy
# occup: Occupation
# 1:ik[none] 2:ib[none] 3:energy[a.u.] 4:occup[none]
  1      1 -1.38676447625070E-001  2.000000000000000E+000
  1      2 -1.10783431105032E-001  2.000000000000000E+000
  1      3 -1.10783428207470E-001  2.000000000000000E+000
  1      4 -1.10783427594037E-001  2.000000000000000E+000
  1      5 -1.57456296850928E-002  2.000000000000000E+000
...
  64     28  3.68051950109468E-001  0.000000000000000E+000
  64     29  4.91528586750629E-001  0.000000000000000E+000
  64     30  4.91528587785578E-001  0.000000000000000E+000
  64     31  4.91528588058071E-001  0.000000000000000E+000
  64     32  5.14831956233275E-001  0.000000000000000E+000
```

Si_k.data

Information on k-points.

```
# k-point distribution
# ik: k-point index
# kx,ky,kz: Reduced coordinate of k-points
# wk: Weight of k-point
# 1:ik[none] 2:kx[none] 3:ky[none] 4:kz[none] 5:wk[none]
  1 -3.750000000000000E-001 -3.750000000000000E-001 -3.750000000000000E-001  1.
↪000000000000000E+000
  2 -3.750000000000000E-001 -3.750000000000000E-001 -1.250000000000000E-001  1.
↪000000000000000E+000
  3 -3.750000000000000E-001 -3.750000000000000E-001  1.250000000000000E-001  1.
↪000000000000000E+000
  4 -3.750000000000000E-001 -3.750000000000000E-001  3.750000000000000E-001  1.
↪000000000000000E+000
```

(continues on next page)

(continued from previous page)

```

5 -3.750000000000000E-001 -1.250000000000000E-001 -3.750000000000000E-001 1.
↳000000000000000E+000
...
60 3.750000000000000E-001 1.250000000000000E-001 3.750000000000000E-001 1.
↳000000000000000E+000
61 3.750000000000000E-001 3.750000000000000E-001 -3.750000000000000E-001 1.
↳000000000000000E+000
62 3.750000000000000E-001 3.750000000000000E-001 -1.250000000000000E-001 1.
↳000000000000000E+000
63 3.750000000000000E-001 3.750000000000000E-001 1.250000000000000E-001 1.
↳000000000000000E+000
64 3.750000000000000E-001 3.750000000000000E-001 3.750000000000000E-001 1.
↳000000000000000E+000

```

Si_rt.data

Results of time evolution calculation. $Ac_ext_{x,y,z}$ are applied vector potential. For transverse calculation specified by `trans_longi = 'tr'`, $Ac_tot_{x,y,z}$ are equal to $Ac_ext_{x,y,z}$. For longitudinal calculation specified by `trans_longi = 'lo'`, $Ac_tot_{x,y,z}$ are the sum of $Ac_ext_{x,y,z}$ and the induced polarization. The same relation holds for electric fields of $E_ext_{x,y,z}$ and $E_tot_{x,y,z}$. $Jm_{x,y,z}$ are macroscopic current. $Eall$ and $Eall-Eall0$ are total energy and electronic excitation energy, respectively. 'Tion' is the kinetic energy of atoms. $Temperature_ion$ is the temperature estimated from the atomic motion.

```

# Real time calculation
# Ac_ext: External vector potential field
# E_ext: External electric field
# Ac_tot: Total vector potential field
# E_tot: Total electric field
# Jm: Matter current density
# Eall: Total energy
# Eall0: Initial energy
# Tion: Kinetic energy of ions
# 1:Time[a.u.] 2:Ac_ext_x[a.u.] 3:Ac_ext_y[a.u.] 4:Ac_ext_z[a.u.] 5:E_ext_x[a.u.] 6:E_
↳ext_y[a.u.] 7:E_ext_z[a.u.] 8:Ac_tot_x[a.u.] 9:Ac_tot_y[a.u.] 10:Ac_tot_z[a.u.]
↳11:E_tot_x[a.u.] 12:E_tot_y[a.u.] 13:E_tot_z[a.u.] 14:Jm_x[a.u.] 15:Jm_y[a.u.]
↳16:Jm_z[a.u.] 17:Eall[a.u.] 18:Eall-Eall0[a.u.] 19:Tion[a.u.] 20:Temperature_ion[K]
0.00000000 0.000000000000000E+000 0.000000000000000E+000 1.000000000000000E-002
↳0.000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 0.
↳000000000000000E+000 0.000000000000000E+000 1.000000000000000E-002 0.
↳000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 -8.
↳65860214541267E-013 1.04880923197437E-012 2.79610491078699E-004 -3.
↳12643773655041E+001 1.51051511945255E-003 0.000000000000000E+000 0.
↳000000000000000E+000
0.16000000 0.000000000000000E+000 0.000000000000000E+000 1.000000000000000E-002
↳0.000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 0.
↳000000000000000E+000 0.000000000000000E+000 1.000000000000000E-002 0.
↳000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 -7.
↳80220609595942E-013 1.25669598865900E-012 2.77640461612200E-004 -3.
↳12643780708603E+001 1.50980976327020E-003 0.000000000000000E+000 0.
↳000000000000000E+000
0.32000000 0.000000000000000E+000 0.000000000000000E+000 1.000000000000000E-002
↳0.000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 0.
↳000000000000000E+000 0.000000000000000E+000 1.000000000000000E-002 0.
↳000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 -6.
↳65469342838961E-013 1.44166600383436E-012 2.72256619397668E-004 -3.
↳12643780794812E+001 1.50980114240440E-003 0.000000000000000E+000 0.
↳000000000000000E+000

```

(continues on next page)

(continued from previous page)

```

0.48000000 0.000000000000000E+000 0.000000000000000E+000 1.000000000000000E-002
↳ 0.000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 0.
↳ 000000000000000E+000 0.000000000000000E+000 1.000000000000000E-002 0.
↳ 000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 -5.
↳ 07694047189471E-013 1.65330407801294E-012 2.65100129464106E-004 -3.
↳ 12643780384343E+001 1.50984218925032E-003 0.000000000000000E+000 0.
↳ 000000000000000E+000
0.64000000 0.000000000000000E+000 0.000000000000000E+000 1.000000000000000E-002
↳ 0.000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 0.
↳ 000000000000000E+000 0.000000000000000E+000 1.000000000000000E-002 0.
↳ 000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 -3.
↳ 21400178809861E-013 1.87627749522222E-012 2.57460045574299E-004 -3.
↳ 12643779799564E+001 1.50990066720169E-003 0.000000000000000E+000 0.
↳ 000000000000000E+000
...
479.36000000 0.000000000000000E+000 0.000000000000000E+000 1.000000000000000E-002
↳ 0.000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 0.
↳ 000000000000000E+000 0.000000000000000E+000 1.000000000000000E-002 0.
↳ 000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 -7.
↳ 94263263896610E-013 3.79557494087330E-012 -3.59285386087180E-006 -3.
↳ 12643819342307E+001 1.50594639281820E-003 0.000000000000000E+000 0.
↳ 000000000000000E+000
479.52000000 0.000000000000000E+000 0.000000000000000E+000 1.000000000000000E-002
↳ 0.000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 0.
↳ 000000000000000E+000 0.000000000000000E+000 1.000000000000000E-002 0.
↳ 000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 -5.
↳ 67828280529921E-013 3.78374121551490E-012 -2.90523320634650E-006 -3.
↳ 12643819351033E+001 1.50594552028593E-003 0.000000000000000E+000 0.
↳ 000000000000000E+000
479.68000000 0.000000000000000E+000 0.000000000000000E+000 1.000000000000000E-002
↳ 0.000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 0.
↳ 000000000000000E+000 0.000000000000000E+000 1.000000000000000E-002 0.
↳ 000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 -3.
↳ 61839313869103E-013 3.74173331529800E-012 -2.24958911411780E-006 -3.
↳ 12643819359872E+001 1.50594463632103E-003 0.000000000000000E+000 0.
↳ 000000000000000E+000
479.84000000 0.000000000000000E+000 0.000000000000000E+000 1.000000000000000E-002
↳ 0.000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 0.
↳ 000000000000000E+000 0.000000000000000E+000 1.000000000000000E-002 0.
↳ 000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 -1.
↳ 73847971134404E-013 3.66573716775167E-012 -1.63591499831827E-006 -3.
↳ 12643819368722E+001 1.50594375133295E-003 0.000000000000000E+000 0.
↳ 000000000000000E+000
480.00000000 0.000000000000000E+000 0.000000000000000E+000 1.000000000000000E-002
↳ 0.000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 0.
↳ 000000000000000E+000 0.000000000000000E+000 1.000000000000000E-002 0.
↳ 000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 3.
↳ 16688678319438E-016 3.55459629253500E-012 -1.06271326454723E-006 -3.
↳ 12643819377811E+001 1.50594284247063E-003 0.000000000000000E+000 0.
↳ 000000000000000E+000

```

Si_force.data

Force acting on each atom during time evolution.

```

# Force calculatio
# force: Force
# time[a.u.] force[a.u.]
  0.000000E+000 -0.663815E-008  0.381467E-008  0.178186E-002  0.280496E-008  0.
↪236613E-009  0.178187E-002  0.190620E-008  0.346038E-008  0.178186E-002 -0.
↪255965E-008  0.162582E-008  0.178187E-002 -0.713246E-009 -0.607621E-008  0.
↪178187E-002 -0.124821E-008  0.434748E-008  0.178187E-002 -0.932639E-008 -0.
↪112168E-007  0.178187E-002 -0.505708E-008 -0.289586E-008  0.178187E-002
  0.160000E+001 -0.131290E-008  0.165516E-008  0.339940E-002 -0.941496E-009 -0.
↪767670E-009  0.339940E-002  0.138786E-008  0.172143E-008  0.339940E-002 -0.
↪451825E-009 -0.106362E-008  0.339940E-002  0.298232E-009  0.383164E-009  0.
↪339940E-002 -0.296521E-009 -0.195556E-008  0.339940E-002  0.348404E-009 -0.
↪849494E-009  0.339940E-002 -0.297429E-009  0.578589E-009  0.339940E-002
  0.320000E+001  0.615410E-008 -0.278186E-008  0.457711E-002 -0.486320E-008 -0.
↪116861E-008  0.457711E-002 -0.112143E-008 -0.166802E-008  0.457711E-002  0.
↪253122E-008 -0.368112E-008  0.457710E-002  0.935799E-009  0.830658E-008  0.
↪457711E-002  0.621491E-009 -0.804263E-008  0.457710E-002  0.123310E-007  0.
↪130141E-007  0.457711E-002  0.636436E-008  0.330898E-008  0.457710E-002
  0.480000E+001  0.635332E-008 -0.357991E-008  0.446307E-002 -0.388157E-008 -0.
↪157542E-008  0.446307E-002 -0.193530E-008 -0.255271E-008  0.446308E-002  0.
↪230966E-008 -0.227850E-008  0.446306E-002 -0.341100E-009  0.746659E-008  0.
↪446307E-002  0.734950E-009 -0.635113E-008  0.446307E-002  0.943051E-008  0.
↪126831E-007  0.446307E-002  0.494958E-008  0.330406E-008  0.446306E-002
  0.640000E+001  0.407644E-009  0.406484E-010  0.320569E-002  0.134973E-008 -0.
↪648732E-009  0.320569E-002 -0.148635E-009 -0.650159E-009  0.320569E-002 -0.
↪231759E-009  0.163276E-008  0.320569E-002 -0.961535E-009 -0.941812E-009  0.
↪320569E-002  0.847442E-009  0.130553E-008  0.320569E-002 -0.264725E-008 -0.
↪351407E-009  0.320569E-002 -0.141512E-008  0.421806E-009  0.320569E-002
...
  0.473600E+003  0.246506E-009  0.251205E-009 -0.148216E-003 -0.416554E-011  0.
↪779853E-009 -0.148215E-003 -0.115879E-009  0.104374E-008 -0.148217E-003  0.
↪913004E-009 -0.465967E-009 -0.148217E-003  0.176729E-009 -0.270103E-009 -0.
↪148216E-003  0.962326E-009  0.799398E-009 -0.148218E-003  0.220066E-009 -0.
↪152063E-008 -0.148216E-003  0.571304E-009 -0.132336E-008 -0.148217E-003
  0.475200E+003 -0.504521E-009 -0.437234E-010 -0.316399E-003 -0.459509E-009  0.
↪105940E-008 -0.316398E-003  0.105290E-009  0.547364E-009 -0.316401E-003  0.
↪181887E-009 -0.343314E-009 -0.316399E-003 -0.804290E-010 -0.500340E-009 -0.
↪316400E-003  0.372911E-009  0.141733E-008 -0.316401E-003 -0.244574E-009 -0.
↪259207E-008 -0.316400E-003  0.202885E-009 -0.147976E-008 -0.316400E-003
  0.476800E+003 -0.475521E-009 -0.161693E-009 -0.415900E-003 -0.925954E-009  0.
↪240941E-009 -0.415900E-003  0.291237E-009 -0.453400E-009 -0.415902E-003 -0.
↪580783E-009 -0.751060E-010 -0.415900E-003 -0.683807E-009 -0.202391E-010 -0.
↪415902E-003 -0.618227E-011  0.138283E-008 -0.415902E-003 -0.274419E-009 -0.
↪218740E-008 -0.415901E-003  0.175364E-009 -0.657477E-009 -0.415900E-003
  0.478400E+003  0.303920E-009 -0.402101E-009 -0.439830E-003 -0.134116E-008 -0.
↪816066E-009 -0.439830E-003  0.318015E-009 -0.927198E-009 -0.439831E-003 -0.
↪150791E-008 -0.169799E-009 -0.439831E-003 -0.702142E-009  0.881452E-009 -0.
↪439831E-003 -0.618720E-009  0.779075E-009 -0.439831E-003  0.540736E-009  0.
↪352559E-009 -0.439830E-003  0.382572E-009  0.794098E-009 -0.439830E-003
  0.480000E+003  0.957060E-009 -0.635421E-009 -0.336591E-003 -0.873698E-009 -0.
↪134192E-008 -0.336592E-003 -0.660852E-010 -0.282862E-009 -0.336591E-003 -0.
↪156118E-008 -0.398368E-009 -0.336593E-003 -0.480887E-010  0.961042E-009 -0.
↪336592E-003 -0.121634E-008 -0.277887E-009 -0.336591E-003  0.104632E-008  0.
↪244269E-008 -0.336591E-003  0.412975E-009  0.133042E-008 -0.336591E-003

```

Si_Ir_data

In transverse calculation specified by `trans_longi = 'tr'`, time-frequency Fourier transformation of the macroscopic current gives the conductivity of the system. Then the dielectric function is calculated.

```
# Fourier-transform spectra
# sigma: Conductivity
# eps: Dielectric constant
# 1:Frequency[a.u.] 2:Re(sigma_x)[a.u.] 3:Re(sigma_y)[a.u.] 4:Re(sigma_z)[a.u.]
↪5:Im(sigma_x)[a.u.] 6:Im(sigma_y)[a.u.] 7:Im(sigma_z)[a.u.] 8:Re(eps_x)[none]
↪9:Re(eps_y)[none] 10:Re(eps_z)[none] 11:Im(eps_x)[none] 12:Im(eps_y)[none]
↪13:Im(eps_z)[none]
    0.00100000 -1.03308449903699E-010 -2.55685769383253E-011 3.36356888185559E-005
↪ 9.38757700305135E-010 2.38405472055867E-010 -1.31839196070590E-003 -1.
↪17967771791178E-005 -2.99589151834528E-006 1.75674019932220E+001 -1.
↪29821226908484E-006 -3.21304213888753E-007 4.22678531563230E-001
    0.00200000 -4.05463997396279E-010 -1.00459000515141E-010 1.32405016849080E-004
↪ 1.82449482725124E-009 4.64061580393162E-010 -2.62118275831395E-003 -1.
↪14636390916102E-005 -2.91578490355285E-006 1.74693769944707E+001 -2.
↪54760543103060E-006 -6.31202516010683E-007 8.31925256463007E-001
    0.00300000 -8.83952914849078E-010 -2.19401192737277E-010 2.90077713140610E-004
↪ 2.60896580505206E-009 6.65304400028214E-010 -3.89397682658909E-003 -1.
↪09284104088247E-005 -2.78682055403947E-006 1.73110519888816E+001 -3.
↪70269331121220E-006 -9.19025567056355E-007 1.21507468343022E+000
    0.00400000 -1.50380858485809E-009 -3.74177620806077E-010 4.96861248105049E-004
↪ 3.25293966934794E-009 8.32525470173577E-010 -5.12467699872510E-003 -1.
↪02194113677943E-005 -2.61545590102370E-006 1.70996476112154E+001 -4.
↪72435400259545E-006 -1.17551366466208E-006 1.56093564690028E+000
    0.00500000 -2.22112273174113E-009 -5.54404046892706E-010 7.40224957578435E-004
↪ 3.72943718693087E-009 9.58925096932178E-010 -6.30436402916416E-003 -9.
↪37309797478928E-006 -2.41004163189201E-006 1.68445949756623E+001 -5.
↪58229028540739E-006 -1.39336934467086E-006 1.86038823098578E+000

...

    0.99600000 -2.76735852669967E-009 -1.50791378263185E-009 4.18549443295463E-003
↪-3.48281730295103E-010 -2.38950132823120E-011 2.58042637047465E-002 4.
↪39421415772947E-009 3.01479510783703E-010 6.74431785999496E-001 -3.
↪49153141258183E-008 -1.90251038625021E-008 5.28077050691215E-002
    0.99700000 -2.79907084112808E-009 -1.43228946145853E-009 4.21502473100264E-003
↪-4.64190825344567E-010 -1.65916319932293E-010 2.58406831005378E-002 5.
↪85074618562197E-009 2.09123968629867E-009 6.74299297121799E-001 -3.
↪52800015701720E-008 -1.80528387158764E-008 5.31269437497179E-002
    0.99800000 -2.80549388829912E-009 -1.33123845334775E-009 4.22285528976820E-003
↪-5.93339164267705E-010 -2.85965452283521E-010 2.58784739372621E-002 7.
↪47106196212637E-009 3.60074935500759E-009 6.74149805180691E-001 -3.
↪53255270107077E-008 -1.67623605018579E-008 5.31723092405153E-002
    0.99900000 -2.78217278629315E-009 -1.21099840604532E-009 4.20947560905717E-003
↪-7.28526525583285E-010 -3.79100172729291E-010 2.59111098101567E-002 9.
↪16409842129228E-009 4.76868195243629E-009 6.74065456552766E-001 -3.
↪49968111569009E-008 -1.52330878716353E-008 5.29507813768946E-002
    1.00000000 -2.72693112746934E-009 -1.07872277288261E-009 4.17738539625698E-003
↪-8.61256547421816E-010 -4.42238226589537E-010 2.59324188318589E-002 1.
↪08228689689459E-008 5.55732945516107E-009 6.74123614032074E-001 -3.
↪42676271876120E-008 -1.35556301541920E-008 5.24945730883769E-002
```

Si_gs_rt_response.out

Standard output file.

4.3.2 Exercise-5: Electron dynamics in crystalline silicon under a pulsed electric field

In this exercise, we learn the calculation of electron dynamics in a unit cell of crystalline silicon of a diamond structure. Calculation is done in a cubic unit cell that contains eight silicon atoms. Since the ground state calculation costs much less computational time than the time evolution calculation, both calculations are successively executed. After finishing the ground state calculation, a pulsed electric field that has \cos^2 envelope shape is applied. The parameters that characterize the pulsed field such as magnitude, frequency, polarization, and carrier envelope phase are specified in the input file.

Input files

To run the code, following files are used:

file name	description
<i>Si_gs_rt_pulse.inp</i>	input file that contain input keywords and their values.
<i>Si_rps.dat</i>	pseudopotential file for Carbon

You may download the above 2 files (zipped file) from:

https://salmon-tddft.jp/webmanual/v_1_2_0/exercise_zip_files/Si_gs_rt_pulse_input.zip

In the input file *Si_gs_rt_pulse.inp*, input keywords are specified. Most of them are mandatory to execute the calculation. This will help you to prepare the input file for other systems that you want to calculate. A complete list of the input keywords that can be used in the input file can be found in the downloaded file *SALMON/manual/input_variables.md*.

```
&calculation
  calc_mode = 'GS_RT'
/
&control
  sysname = 'Si'
/
&units
  unit_system = 'a.u.'
/
&system
  iperiodic = 3
  a1 = 10.26d0, 10.26d0, 10.26d0
  nstate = 32
  nelec = 32
  nelem = 1
  natom = 8
/
&pseudo
  izatom(1) = 14
  pseudo_file(1) = './Si_rps.dat'
  lloc_ps(1) = 2
/
&functional
  xc = 'PZ'
/
&rgrid
  num_rgrid = 12, 12, 12
```

(continues on next page)

(continued from previous page)

```

/
&kgrid
  num_kgrid = 4, 4, 4
/
&tgrid
  nt = 3000
  dt = 0.16
/
&propagation
  propagator = 'etrs'
/
&scf
  ncg = 5
  nscf = 120
/
&emfield
  trans_longi = 'tr'
  ae_shape1 = 'Acos2'
  rlaser_int_wcm2_1 = 1d14
  pulse_twl = 441.195136248d0
  omegal = 0.05696145187d0
  ekdir_rel = 0., 0., 1.
/
&atomic_red_coor
  'Si'      .0      .0      .0      1
  'Si'      .25     .25     .25     1
  'Si'      .5      .0      .5      1
  'Si'      .0      .5      .5      1
  'Si'      .5      .5      .0      1
  'Si'      .75     .25     .75     1
  'Si'      .25     .75     .75     1
  'Si'      .75     .75     .25     1
/

```

We present explanations of the input keywords that appear in the input file below:

&calculation

Mandatory: calc_mode

```

&calculation
  calc_mode = 'GS_RT'
/

```

This indicates that the ground state (GS) and the real time (RT) calculations are carried out sequentially in the present job. See *&calculation in Inputs* for detail.

&control

Mandatory: none

```

&control
  sysname = 'Si'
/

```

'Si' defined by sysname = 'C2H2' will be used in the filenames of output files.

&system

Mandatory: `periodic`, `al`, `state`, `nelem`, `natom`

```
&system
  iperiodic = 3
  al = 10.26d0,10.26d0,10.26d0
  nstate = 32
  nelec = 32
  nelem = 1
  natom = 8
/
```

`iperiodic = 3` indicates that three dimensional periodic boundary condition (bulk crystal) is assumed. `al = 10.26d0, 10.26d0, 10.26d0` specifies the lattice constants of the unit cell. `nstate = 32` indicates the number of Kohn-Sham orbitals to be solved. `nelec = 32` indicate the number of valence electrons in the system. `nelem = 1` and `natom = 8` indicate the number of elements and the number of atoms in the system, respectively. See *&system Inputs* for more information.

&pseudo

```
&pseudo
  izatom(1)=14
  pseudo_file(1) = './Si_rps.dat'
  lloc_ps(1)=2
/
```

`izatom(1) = 14` indicates the atomic number of the element #1. `pseudo_file(1) = 'Si_rps.dat'` indicates the pseudopotential filename of element #1. `lloc_ps(1) = 2` indicate the angular momentum of the pseudopotential that will be treated as local.

&functional

```
&functional
  xc = 'PZ'
/
```

This indicates that the adiabatic local density approximation with the Perdew-Zunger functional is used. We note that meta-GGA functionals that reasonably reproduce the band gap of various insulators may also be used in the calculation of periodic systems. See *&functional in Inputs* for detail.

&rgrid

Mandatory: `dl` or `num_rgrid`

```
&rgrid
  num_rgrid = 12,12,12
/
```

`num_rgrid=12,12,12` specifies the number of the grids for each Cartesian direction. See *&rgrid in Inputs* for more information.

&kgrid

Mandatory: none

This input keyword provides grid spacing of k-space for periodic systems.

```
&kgrid
  num_kgrid = 4,4,4
/
```

&tgrid

```
&tgrid
  nt=3000
  dt=0.16
/
```

`dt=0.16` specifies the time step of the time evolution calculation. `nt=3000` specifies the number of time steps in the calculation.

&propagation

```
&propagation
  propagator='etrs'
/
```

`propagator = 'etrs'` indicates the use of enforced time-reversal symmetry propagator. See [&propagation in Inputs](#) for more information.

&scf

Mandatory: `nscf`

This input keywords specify parameters related to the self-consistent field calculation.

```
&scf
  ncg = 5
  nscf = 120
/
```

`ncg = 5` is the number of conjugate-gradient iterations in solving the Kohn-Sham equation. Usually this value should be 4 or 5. `nscf = 120` is the number of scf iterations.

&emfield

```
&emfield
  trans_longi = 'tr'
  ae_shape1 = 'Acos2'
  rlaser_int_wcm2_1 = 1d14
  pulse_tw1 = 441.195136248d0
  omega1 = 0.05696145187d0
  ekdir_re1 = 0.,0.,1.
/
```

This input keyword specifies the pulsed electric field applied to the system

`ae_shape1 = 'Acos2'` specifies the envelope of the pulsed electric field, \cos^2 envelope for the vector potential.

`epdir_re1 = 0., 0., 1.` specify the real part of the unit polarization vector of the pulsed electric field. Specifying only the real part, it describes a linearly polarized pulse.

`laser_int_wcm2_1 = 1d14` specifies the maximum intensity of the applied electric field in unit of W/cm^2 .

`omega1=0.05696145187d0` specifies the average photon energy (frequency multiplied with \hbar).

`pulse_tw1=441.195136248d0` specifies the pulse duration. Note that it is not the FWHM but a full duration of the \cos^2 envelope.

`trans_longi = 'tr'` specifies the treatment of the polarization in the time evolution calculation, 'tr' indicating transverse.

See [&emfield in Inputs](#) for detail.

&atomic_red_coor

Mandatory: atomic_coor or atomic_red_coor (they may be provided as a separate file)

```
&atomic_red_coor
'Si'   .0   .0   .0   1
'Si'   .25  .25  .25  1
'Si'   .5   .0   .5   1
'Si'   .0   .5   .5   1
'Si'   .5   .5   .0   1
'Si'   .75  .25  .75  1
'Si'   .25  .75  .75  1
'Si'   .75  .75  .25  1
/
```

Cartesian coordinates of atoms are specified in a reduced coordinate system. First column indicates the element, next three columns specify reduced Cartesian coordinates of the atoms, and the last column labels the element.

Output files

After the calculation, following output files are created in the directory that you run the code,

file name	description
<i>Si_gs_info.data</i>	information of ground state calculation
<i>Si_eigen.data</i>	energy eigenvalues of orbitals
<i>Si_k.data</i>	information on k-points
<i>Si_rt.data</i>	electric field, vector potential, and current as functions of time
<i>Si_force.data</i>	force acting on atoms
<i>Si_lr.data</i>	Fourier transformations of various quantities
<i>Si_gs_rt_pulse.out</i>	standard output file

You may download the above files (zipped file) from:

https://salmon-tddft.jp/webmanual/v_1_2_0/exercise_zip_files/Si_gs_rt_pulse_output.zip

Explanations of the output files are described below:

Si_gs_info.data

Results of the ground state as well as input parameters are provided.

```
#-----
#grid information-----
#aL = 10.2600000000000 10.2600000000000 10.2600000000000
#a1(1), a1(2), a1(3) = 10.2600000000000 10.2600000000000 10.
↪260000000000000
#aLx, aLy, aLz = 10.2600000000000 10.2600000000000 10.2600000000000
#bLx, bLy, bLz = 0.612396228769940 0.612396228769940 0.612396228769940
#Nd = 4
#NLx, NLy, NLz = 12 12 12
#NL = 1728
#Hx, Hy, Hz = 0.855000000000000 0.855000000000000 0.855000000000000
#(pi/max(Hx, Hy, Hz)) **2 = 13.5010490764192
#(pi/Hx) **2 + (pi/Hy) **2 + (pi/Hz) **2 = 40.5031472292576
```

(continues on next page)

(continued from previous page)

```

#Hxyz = 0.6250263750000000
#NKx,NKy,NKz=          4          4          4
#NKxyz =          64
#Sym=          1
#NK =          64
#NEwald, aEwald =          4 0.5000000000000000
#-----
#GS calc. option-----
#FSset_option =n
#Ncg=          5
#Nmemory_MB,alpha_MB =          8 0.7500000000000000
#NFSset_start,NFSset_every =          75          25
#Nscf=          120
#Nscf_conv=          120
#NI,NE=          8          1
#Zatom=          14
#Lref=          2
#i,Kion(ia)(Rion(j,a),j=1,3)
#          1          1
# 0.0000000000000000E+000 0.0000000000000000E+000 0.0000000000000000E+000
#          2          1
# 2.5650000000000000 2.5650000000000000 2.5650000000000000
#          3          1
# 5.1300000000000000 0.0000000000000000E+000 5.1300000000000000
#          4          1
# 0.0000000000000000E+000 5.1300000000000000 5.1300000000000000
#          5          1
# 5.1300000000000000 5.1300000000000000 0.0000000000000000E+000
#          6          1
# 7.6950000000000000 2.5650000000000000 7.6950000000000000
#          7          1
# 2.5650000000000000 7.6950000000000000 7.6950000000000000
#          8          1
# 7.6950000000000000 7.6950000000000000 2.5650000000000000
#-----
#GS information-----
#NB,Nelec=          32          32
#Eall = -31.2658878806236
#ddns(iter = Nscf_conv) 2.798849279746559E-010
#ddns_abs_1e(iter = Nscf_conv) 2.364732236264119E-010
#esp_var_ave(iter = Nscf_conv) 1.196976937606010E-009
#esp_var_max(iter = Nscf_conv) 4.031276129792963E-009
#NBocccmax is          16
#-----
#band information-----
#Bottom of VB -0.194802063980608
#Top of VB 0.216731478175047
#Bottom of CB 0.255681914576368
#Top of CB 0.533214678236198
#Fundamental gap 3.895043640132098E-002
#Fundamental gap[eV] 1.05990369517819
#BG between same k-point 3.895043648321342E-002
#BG between same k-point[eV] 1.05990369740661
#Physically upper bound of CB for DOS 0.454100922291231
#Physically upper bound of CB for eps(omega) 0.609752486428134
#-----
#iter      total-energy      ddns/nelec      esp_var_ave      esp_var_max

```

(continues on next page)

(continued from previous page)

1	-0.2059780903E+02	0.5134199377E+00	0.1332473220E-01	0.1986049398E-01
2	-0.2600097163E+02	0.3186108570E+00	0.1526707771E-01	0.2520724900E-01
3	-0.2866336088E+02	0.1363849859E+00	0.6359704895E-02	0.1247448390E-01
4	-0.3006244467E+02	0.1245614607E+00	0.5868323970E-02	0.1942874074E-01
5	-0.3096872596E+02	0.7495214064E-01	0.2566344769E-02	0.1102001262E-01
...				
115	-0.3126588788E+02	0.1355175468E-09	0.1208579378E-08	0.4031265522E-08
116	-0.3126588788E+02	0.1452261250E-09	0.1204317051E-08	0.4031272647E-08
117	-0.3126588788E+02	0.1419175726E-09	0.1198067051E-08	0.4031255783E-08
118	-0.3126588788E+02	0.1686476198E-09	0.1198945057E-08	0.4031251395E-08
119	-0.3126588788E+02	0.2159059511E-09	0.1200809994E-08	0.4666412657E-08
120	-0.3126588788E+02	0.2364732236E-09	0.1196976938E-08	0.4031276130E-08

Si_eigen.data

Orbital energies in the ground state calculation.

```
# Ground state eigenenergies
# ik: k-point index
# ib: Band index
# energy: Eigenenergy
# occup: Occupation
# 1:ik[none] 2:ib[none] 3:energy[a.u.] 4:occup[none]
  1      1 -1.38676447625070E-001  2.000000000000000E+000
  1      2 -1.10783431105032E-001  2.000000000000000E+000
  1      3 -1.10783428207470E-001  2.000000000000000E+000
  1      4 -1.10783427594037E-001  2.000000000000000E+000
  1      5 -1.57456296850928E-002  2.000000000000000E+000
...
  64     28  3.68051950109468E-001  0.000000000000000E+000
  64     29  4.91528586750629E-001  0.000000000000000E+000
  64     30  4.91528587785578E-001  0.000000000000000E+000
  64     31  4.91528588058071E-001  0.000000000000000E+000
  64     32  5.14831956233275E-001  0.000000000000000E+000
```

Si_k.data

Information on k-points.

```
# k-point distribution
# ik: k-point index
# kx,ky,kz: Reduced coordinate of k-points
# wk: Weight of k-point
# 1:ik[none] 2:kx[none] 3:ky[none] 4:kz[none] 5:wk[none]
  1 -3.750000000000000E-001 -3.750000000000000E-001 -3.750000000000000E-001  1.
↪000000000000000E+000
  2 -3.750000000000000E-001 -3.750000000000000E-001 -1.250000000000000E-001  1.
↪000000000000000E+000
  3 -3.750000000000000E-001 -3.750000000000000E-001  1.250000000000000E-001  1.
↪000000000000000E+000
  4 -3.750000000000000E-001 -3.750000000000000E-001  3.750000000000000E-001  1.
↪000000000000000E+000
  5 -3.750000000000000E-001 -1.250000000000000E-001 -3.750000000000000E-001  1.
↪000000000000000E+000
```

(continues on next page)

(continued from previous page)

```

...
    60  3.750000000000000E-001  1.250000000000000E-001  3.750000000000000E-001  1.
↳000000000000000E+000
    61  3.750000000000000E-001  3.750000000000000E-001 -3.750000000000000E-001  1.
↳000000000000000E+000
    62  3.750000000000000E-001  3.750000000000000E-001 -1.250000000000000E-001  1.
↳000000000000000E+000
    63  3.750000000000000E-001  3.750000000000000E-001  1.250000000000000E-001  1.
↳000000000000000E+000
    64  3.750000000000000E-001  3.750000000000000E-001  3.750000000000000E-001  1.
↳000000000000000E+000

```

Si_rt.data

Results of time evolution calculation. $Ac_ext_{x,y,z}$ are applied vector potential. For transverse calculation specified by `trans_longi = 'tr'`, $Ac_tot_{x,y,z}$ are equal to $Ac_ext_{x,y,z}$. For longitudinal calculation specified by `trans_longi = 'l0'`, $Ac_tot_{x,y,z}$ are the sum of $Ac_ext_{x,y,z}$ and the induced polarization. The same relation holds for electric fields of $E_ext_{x,y,z}$ and $E_tot_{x,y,z}$. $Jm_{x,y,z}$ are macroscopic current. $Eall$ and $Eall0$ are total energy and electronic excitation energy, respectively. 'Tion' is the kinetic energy of atoms. $Temperature_ion$ is the temperature estimated from the atomic motion.

```

# Real time calculation
# Ac_ext: External vector potential field
# E_ext: External electric field
# Ac_tot: Total vector potential field
# E_tot: Total electric field
# Jm: Matter current density
# Eall: Total energy
# Eall0: Initial energy
# Tion: Kinetic energy of ions
# 1:Time[a.u.] 2:Ac_ext_x[a.u.] 3:Ac_ext_y[a.u.] 4:Ac_ext_z[a.u.] 5:E_ext_x[a.u.] 6:E_
↳ext_y[a.u.] 7:E_ext_z[a.u.] 8:Ac_tot_x[a.u.] 9:Ac_tot_y[a.u.] 10:Ac_tot_z[a.u.]
↳11:E_tot_x[a.u.] 12:E_tot_y[a.u.] 13:E_tot_z[a.u.] 14:Jm_x[a.u.] 15:Jm_y[a.u.]
↳16:Jm_z[a.u.] 17:Eall[a.u.] 18:Eall-Eall0[a.u.] 19:Tion[a.u.] 20:Temperature_ion[K]
    0.00000000  0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
↳0.000000000000000E+000  0.000000000000000E+000  3.77331308204139E-008  0.
↳000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000  0.
↳000000000000000E+000  0.000000000000000E+000  3.77331308204139E-008 -8.
↳70901886780464E-013  1.04477060852801E-012  2.05240902737187E-014 -3.
↳12658878806237E+001 -1.13686837721616E-013  0.000000000000000E+000  0.
↳000000000000000E+000
    0.16000000  0.000000000000000E+000  0.000000000000000E+000 -1.20746018625324E-008
↳0.000000000000000E+000  0.000000000000000E+000  2.89492697662796E-007  0.
↳000000000000000E+000  0.000000000000000E+000 -1.20746018625324E-008  0.
↳000000000000000E+000  0.000000000000000E+000  2.89492697662796E-007 -7.
↳85903272323908E-013  1.25651122959738E-012 -3.36584280927329E-010 -3.
↳12658878806202E+001  3.36797256750287E-012  0.000000000000000E+000  0.
↳000000000000000E+000
    0.32000000  0.000000000000000E+000  0.000000000000000E+000 -9.26376632520948E-008
↳0.000000000000000E+000  0.000000000000000E+000  9.25330085901344E-007  0.
↳000000000000000E+000  0.000000000000000E+000 -9.26376632520948E-008  0.
↳000000000000000E+000  0.000000000000000E+000  9.25330085901344E-007 -6.
↳72570067469639E-013  1.44513383464745E-012 -2.58015152519122E-009 -3.
↳12658878805859E+001  3.76836339910369E-011  0.000000000000000E+000  0.
↳000000000000000E+000

```

(continues on next page)

(continued from previous page)

```

0.48000000 0.000000000000000E+000 0.000000000000000E+000 -3.08180229350963E-007
↪ 0.000000000000000E+000 0.000000000000000E+000 1.97661471359977E-006 0.
↪ 000000000000000E+000 0.000000000000000E+000 -3.08180229350963E-007 0.
↪ 000000000000000E+000 0.000000000000000E+000 1.97661471359977E-006 -5.
↪ 16188379881674E-013 1.65999923818627E-012 -8.55700433990977E-009 -3.
↪ 12658878804485E+001 1.75052861095537E-010 0.000000000000000E+000 0.
↪ 000000000000000E+000
0.64000000 0.000000000000000E+000 0.000000000000000E+000 -7.25154371604021E-007
↪ 0.000000000000000E+000 0.000000000000000E+000 3.44304368307922E-006 0.
↪ 000000000000000E+000 0.000000000000000E+000 -7.25154371604021E-007 0.
↪ 000000000000000E+000 0.000000000000000E+000 3.44304368307922E-006 -3.
↪ 31706377656679E-013 1.88556925268305E-012 -2.00507097518009E-008 -3.
↪ 12658878800854E+001 5.38197042487809E-010 0.000000000000000E+000 0.
↪ 000000000000000E+000
...
479.36000000 0.000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000
↪ 0.000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 0.
↪ 000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 0.
↪ 000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 -5.
↪ 55195781926435E-013 -3.96119870066348E-012 8.12847438945111E-005 -5.
↪ 27018855348676E+000 2.59956993271368E+001 0.000000000000000E+000 0.
↪ 000000000000000E+000
479.52000000 0.000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000
↪ 0.000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 0.
↪ 000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 0.
↪ 000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 -3.
↪ 02810565250972E-013 -3.64518553366057E-012 5.63898060743298E-005 -5.
↪ 27028126908996E+000 2.59956066115336E+001 0.000000000000000E+000 0.
↪ 000000000000000E+000
479.68000000 0.000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000
↪ 0.000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 0.
↪ 000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 0.
↪ 000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 -7.
↪ 16671188540074E-014 -3.26226560875438E-012 2.80138979239849E-005 -5.
↪ 27037369552727E+000 2.59955141850963E+001 0.000000000000000E+000 0.
↪ 000000000000000E+000
479.84000000 0.000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000
↪ 0.000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 0.
↪ 000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 0.
↪ 000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 1.
↪ 25642145342234E-013 -2.83144546563755E-012 -1.41212701168313E-006 -5.
↪ 27046577678837E+000 2.59954221038352E+001 0.000000000000000E+000 0.
↪ 000000000000000E+000
480.00000000 0.000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000
↪ 0.000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 0.
↪ 000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 0.
↪ 000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 2.
↪ 93865192596554E-013 -2.36028795847010E-012 -2.94243971611215E-005 -5.
↪ 27055732569162E+000 2.59953305549319E+001 0.000000000000000E+000 0.
↪ 000000000000000E+000

```

Si_force.data

Force acting on each atom during time evolution.

```

# Force calculatio
# force: Force
# time[a.u.] force[a.u.]
  0.000000E+000 -0.663696E-008  0.381594E-008  0.147786E-006  0.280111E-008  0.
↪228130E-009  0.152372E-006  0.190603E-008  0.347020E-008  0.144454E-006 -0.
↪255639E-008  0.162309E-008  0.157887E-006 -0.715186E-009 -0.607023E-008  0.
↪150657E-006 -0.125193E-008  0.434347E-008  0.154377E-006 -0.932342E-008 -0.
↪112253E-007  0.150918E-006 -0.505492E-008 -0.289610E-008  0.158433E-006
  0.160000E+001 -0.131252E-008  0.164755E-008  0.796703E-004 -0.945892E-009 -0.
↪762583E-009  0.796709E-004  0.139448E-008  0.172324E-008  0.796683E-004 -0.
↪452537E-009 -0.106913E-008  0.796682E-004  0.302666E-009  0.384512E-009  0.
↪796698E-004 -0.295296E-009 -0.194915E-008  0.796704E-004  0.357413E-009 -0.
↪849344E-009  0.796713E-004 -0.289642E-009  0.582711E-009  0.796703E-004
  0.320000E+001  0.611719E-008 -0.279866E-008  0.299229E-003 -0.482069E-008 -0.
↪108492E-008  0.299224E-003 -0.113705E-008 -0.180657E-008  0.299232E-003  0.
↪249663E-008 -0.365775E-008  0.299214E-003  0.894693E-009  0.822775E-008  0.
↪299225E-003  0.674776E-009 -0.800682E-008  0.299222E-003  0.122632E-007  0.
↪130955E-007  0.299227E-003  0.634943E-008  0.330482E-008  0.299216E-003
  0.480000E+001  0.632236E-008 -0.360316E-008  0.625960E-003 -0.380678E-008 -0.
↪149542E-008  0.625956E-003 -0.195847E-008 -0.271155E-008  0.625964E-003  0.
↪224789E-008 -0.225876E-008  0.625950E-003 -0.416507E-009  0.737796E-008  0.
↪625958E-003  0.788341E-009 -0.633137E-008  0.625955E-003  0.936024E-008  0.
↪128155E-007  0.625960E-003  0.491902E-008  0.331005E-008  0.625950E-003
  0.640000E+001  0.402103E-009  0.205061E-010  0.102016E-002  0.136956E-008 -0.
↪668489E-009  0.102016E-002 -0.133472E-009 -0.624741E-009  0.102016E-002 -0.
↪272159E-009  0.161549E-008  0.102016E-002 -0.993343E-009 -0.896691E-009  0.
↪102016E-002  0.819106E-009  0.130318E-008  0.102016E-002 -0.262966E-008 -0.
↪314962E-009  0.102016E-002 -0.140244E-008  0.427951E-009  0.102016E-002
...
  0.473600E+003  0.597367E-009  0.387050E-009 -0.348389E-002  0.605905E-009 -0.
↪465705E-009 -0.348389E-002 -0.641806E-009 -0.394485E-009 -0.348389E-002 -0.
↪964505E-009  0.116781E-008 -0.348389E-002 -0.589312E-009  0.786468E-009 -0.
↪348389E-002 -0.237343E-009  0.194589E-009 -0.348389E-002 -0.120897E-008 -0.
↪693231E-009 -0.348389E-002  0.229921E-009 -0.733008E-009 -0.348389E-002
  0.475200E+003 -0.168428E-009 -0.115247E-008  0.473981E-002 -0.265453E-009 -0.
↪760895E-010  0.473981E-002  0.442756E-009 -0.106813E-009  0.473981E-002  0.
↪875680E-009  0.104699E-008  0.473981E-002 -0.870002E-009  0.234294E-009  0.
↪473982E-002  0.694922E-009  0.543690E-009  0.473981E-002  0.694120E-009  0.
↪135771E-009  0.473981E-002 -0.233819E-009  0.444395E-009  0.473981E-002
  0.476800E+003  0.102600E-008  0.751831E-009 -0.162791E-002  0.826826E-009  0.
↪101675E-008 -0.162791E-002  0.656143E-009 -0.777523E-009 -0.162791E-002  0.
↪379311E-010  0.158618E-008 -0.162791E-002 -0.375430E-009  0.123075E-008 -0.
↪162791E-002  0.363301E-009 -0.428326E-009 -0.162792E-002  0.297257E-009 -0.
↪713355E-009 -0.162791E-002 -0.253648E-010  0.813094E-010 -0.162791E-002
  0.478400E+003  0.101572E-008  0.116909E-008 -0.663462E-002 -0.435127E-010 -0.
↪480843E-009 -0.663462E-002  0.122317E-008  0.313588E-009 -0.663462E-002 -0.
↪951952E-010 -0.156395E-008 -0.663462E-002  0.528331E-009 -0.200449E-009 -0.
↪663462E-002 -0.593208E-009  0.104932E-009 -0.663462E-002  0.293176E-009 -0.
↪106265E-008 -0.663462E-002 -0.144531E-009  0.662959E-010 -0.663462E-002
  0.480000E+003  0.378024E-009 -0.244626E-009  0.169685E-002  0.792446E-009 -0.
↪137457E-008  0.169684E-002 -0.267886E-009  0.231108E-009  0.169684E-002 -0.
↪568326E-009  0.242847E-011  0.169684E-002  0.803904E-010 -0.153003E-009  0.
↪169684E-002 -0.698906E-009 -0.493838E-009  0.169684E-002 -0.201697E-009 -0.
↪515273E-009  0.169684E-002 -0.182781E-009 -0.848598E-009  0.169684E-002

```

Si_Ir_data

Fourier transformations of various quantities.

```

# Fourier-transform spectra
# Jm: Matter current density
# E_ext: External electric field
# E_tot: Total electric potential field
# 1:Frequency[a.u.] 2:Re(Jm_x)[a.u.] 3:Re(Jm_y)[a.u.] 4:Re(Jm_z)[a.u.] 5:Im(Jm_x)[a.u.
↪] 6:Im(Jm_y)[a.u.] 7:Im(Jm_z)[a.u.] 8:Re(E_ext_x)[a.u.] 9:Re(E_ext_y)[a.u.] 10:Re(E
↪ext_z)[a.u.] 11:Im(E_ext_x)[a.u.] 12:Im(E_ext_y)[a.u.] 13:Im(E_ext_z)[a.u.] 14:Re(E
↪tot_x)[a.u.] 15:Re(E_tot_y)[a.u.] 16:Re(E_tot_z)[a.u.] 17:Im(E_tot_x)[a.u.] 18:Im(E
↪tot_y)[a.u.] 19:Im(E_tot_z)[a.u.]
0.00036749 -2.84895463680408E-013 5.15173051622877E-014 -3.52107345483375E-005
↪ 4.48903005613079E-012 -2.32288812334531E-012 6.75568020279512E-004 0.
↪0000000000000000E+000 0.0000000000000000E+000 4.97953467812365E-004 0.
↪0000000000000000E+000 0.0000000000000000E+000 3.58501823355665E-004 0.
↪0000000000000000E+000 0.0000000000000000E+000 4.97953467812365E-004 0.
↪0000000000000000E+000 0.0000000000000000E+000 3.58501823355665E-004
0.00073499 -1.13540774534206E-012 2.07937929182977E-013 -1.40798553922957E-004
↪ 8.90861768500183E-012 -4.65722890058464E-012 1.34678502951777E-003 0.
↪0000000000000000E+000 0.0000000000000000E+000 4.63469055289272E-004 0.
↪0000000000000000E+000 0.0000000000000000E+000 7.13528955721859E-004 0.
↪0000000000000000E+000 0.0000000000000000E+000 4.63469055289272E-004 0.
↪0000000000000000E+000 0.0000000000000000E+000 7.13528955721859E-004
0.00110248 -2.53907962392519E-012 4.74814744878412E-013 -3.16626882537289E-004
↪ 1.31903398564115E-011 -7.01379574946040E-012 2.00927257456496E-003 0.
↪0000000000000000E+000 0.0000000000000000E+000 4.07013867440069E-004 0.
↪0000000000000000E+000 0.0000000000000000E+000 1.06169736359997E-003 0.
↪0000000000000000E+000 0.0000000000000000E+000 4.07013867440069E-004 0.
↪0000000000000000E+000 0.0000000000000000E+000 1.06169736359997E-003
0.00146997 -4.47536487471274E-012 8.61226149017987E-013 -5.62456776380136E-004
↪ 1.72677949815665E-011 -9.40201922497097E-012 2.65859850138268E-003 0.
↪0000000000000000E+000 0.0000000000000000E+000 3.30095592579552E-004 0.
↪0000000000000000E+000 0.0000000000000000E+000 1.39980301776817E-003 0.
↪0000000000000000E+000 0.0000000000000000E+000 3.30095592579552E-004 0.
↪0000000000000000E+000 0.0000000000000000E+000 1.39980301776817E-003
0.00183747 -6.91594776469286E-012 1.37951536625165E-012 -8.77930263458079E-004
↪ 2.10775706012377E-011 -1.18293542969418E-011 3.29025303683963E-003 0.
↪0000000000000000E+000 0.0000000000000000E+000 2.34784059348594E-004 0.
↪0000000000000000E+000 0.0000000000000000E+000 1.72490861730070E-003 0.
↪0000000000000000E+000 0.0000000000000000E+000 2.34784059348594E-004 0.
↪0000000000000000E+000 0.0000000000000000E+000 1.72490861730070E-003
...

0.36602326 -1.34089639696033E-011 -1.39416295506675E-010 1.78132697555994E-003
↪ -4.82428771517963E-011 2.46762847283247E-012 -1.61143602449332E-004 0.
↪0000000000000000E+000 0.0000000000000000E+000 4.45170812023321E-006 0.
↪0000000000000000E+000 0.0000000000000000E+000 -3.65224569772520E-004 0.
↪0000000000000000E+000 0.0000000000000000E+000 4.45170812023321E-006 0.
↪0000000000000000E+000 0.0000000000000000E+000 -3.65224569772520E-004
0.36639076 -1.24360432096437E-011 -1.36245370206982E-010 1.73864190682382E-003
↪ -5.21816299182567E-011 -3.50916159867163E-012 -7.77023817604767E-005 0.
↪0000000000000000E+000 0.0000000000000000E+000 5.11905824950650E-006 0.
↪0000000000000000E+000 0.0000000000000000E+000 -3.63173718411360E-004 0.
↪0000000000000000E+000 0.0000000000000000E+000 5.11905824950650E-006 0.
↪0000000000000000E+000 0.0000000000000000E+000 -3.63173718411360E-004
0.36675825 -1.12274531733429E-011 -1.32788988742883E-010 1.69384196276721E-003
↪ 5.60444962150072E-011 9.12174813252230E-012 2.15275586453434E-006 0
↪0000000000000000E+000 0.0000000000000000E+000 5.62963567554736E-006 0.
↪0000000000000000E+000 0.0000000000000000E+000 -3.61035199699344E-004 0.
↪0000000000000000E+000 0.0000000000000000E+000 5.62963567554736E-006 0.
↪0000000000000000E+000 0.0000000000000000E+000 -3.61035199699344E-004

```

(continues on next page)

(continued from previous page)

```

0.36712574 -9.79019347225448E-012 -1.29082073143664E-010 1.64792403884542E-003
↪-5.98140679681705E-011 -1.43564301457731E-011 6.57920339739901E-005 0.
↪00000000000000E+000 0.00000000000000E+000 5.97107609420323E-006 0.
↪00000000000000E+000 0.00000000000000E+000 -3.58835797250056E-004 0.
↪00000000000000E+000 0.00000000000000E+000 5.97107609420323E-006 0.
↪00000000000000E+000 0.00000000000000E+000 -3.58835797250056E-004
0.36749324 -8.13273340472761E-012 -1.25159803753324E-010 1.60180907923576E-003
↪-6.34737299655122E-011 -1.92028837998549E-011 1.26529960616246E-004 0.
↪00000000000000E+000 0.00000000000000E+000 6.13554270900561E-006 0.
↪00000000000000E+000 0.00000000000000E+000 -3.56603849256161E-004 0.
↪00000000000000E+000 0.00000000000000E+000 6.13554270900561E-006 0.
↪00000000000000E+000 0.00000000000000E+000 -3.56603849256161E-004

```

Si_gs_rt_pulse.out

Standard output file.

4.4 Maxwell + TDDFT multiscale simulation

4.4.1 Exercise-6: Pulsed-light propagation through a silicon thin film

In this exercise, we learn the calculation of the propagation of a pulsed light through a thin film of crystalline silicon. We consider a silicon thin film of 53 nm thickness, and an irradiation of a few-cycle, linearly polarized pulsed light normally on the thin film. First, to set up initial orbitals, the ground state calculation is carried out. The pulsed light locates in the vacuum region in front of the thin film. The parameters that characterize the pulsed light such as magnitude and frequency are specified in the input file. The calculation ends when the reflected and transmitted pulses reach the vacuum region.

Input files

To run the code, following files are used:

file name	description
<i>Si_gs_rt_multiscale.inp</i>	input file that contain input keywords and their values.
<i>Si_rps.dat</i>	pseudopotential file for silicon

You may download the above two files (zipped file) from:

https://salmon-tddft.jp/webmanual/v_1_2_0/exercise_zip_files/Si_gs_rt_multiscale_input.zip

In the input file *Si_gs_rt_multiscale.inp*, input keywords are specified. Most of them are mandatory to execute the calculation. This will help you to prepare the input file for other systems that you want to calculate. A complete list of the input keywords that can be used in the input file can be found in the downloaded file *SALMON/manual/input_variables.md*.

```

&calculation
  calc_mode = 'GS_RT'
  use_ms_maxwell = 'y'
/
&control

```

(continues on next page)

(continued from previous page)

```

    sysname = 'Si'
/
&system
  iperiodic = 3
  a1 = 10.26d0, 10.26d0, 10.26d0
  nstate = 32
  nelec = 32
  nelem = 1
  natom = 8
/
&pseudo
  izatom(1) = 14
  pseudo_file(1) = './Si_rps.dat'
  lloc_ps(1) = 2
/
&functional
  xc = 'PZ'
/
&rgrid
  num_rgrid = 12, 12, 12
/
&kgrid
  num_kgrid = 2, 2, 2
/
&tgrid
  nt = 4000
  dt = 0.08
/
&propagation
  propagator = 'middlepoint'
/
&scf
  ncg = 5
  nscf = 100
/
&emfield
  ae_shape1 = 'Acos2'
  rlasers_int_wcm2_1 = 1d12
  pulse_tw1 = 441.195136248d0
  omegal = 0.05696145187d0
  ekdir_re1 = 0., 0., 1.
/
&multiscale
  ftdddim = '1d'
  twod_shape = 'periodic'
  nx_m = 4
  ny_m = 1
  hx_m = 250d0
  nxvacl_m = -2000
  nxvacr_m = 256
/
&atomic_red_coor
  'Si'      .0      .0      .0      1
  'Si'      .25     .25     .25     1
  'Si'      .5      .0      .5      1
  'Si'      .0      .5      .5      1
  'Si'      .5      .5      .0      1

```

(continues on next page)

(continued from previous page)

```
'Si'      .75    .25    .75    1
'Si'      .25    .75    .75    1
'Si'      .75    .75    .25    1
/
```

We present explanations of the input keywords that appear in the input file below:

&calculation

Mandatory: calc_mode

```
&calculation
  calc_mode = 'GS_RT'
  use_ms_maxwell = 'y'
/
```

calc_mode = 'GS_RT' indicates that the ground state (GS) and the real time (RT) calculations are carried out sequentially in the present job. use_ms_maxwell = 'y' indicates the multi-scale Maxwell - TDDFT calculation. See *&calculation in Inputs* for detail.

&control

Mandatory: none

```
&control
  sysname = 'Si'
/
```

'Si' defined by sysname = 'C2H2' will be used in the filenames of output files.

&system

```
&system
  iperiodic = 3
  a1 = 10.26d0,10.26d0,10.26d0
  isym = 8
  crystal_structure = 'diamond'
  nstate = 32
  nelec = 32
  nelelem = 1
  natom = 8
/
```

iperiodic = 3 indicates that three dimensional periodic boundary condition (bulk crystal) is assumed. a1 = 10.26d0, 10.26d0, 10.26d0 specifies the lattice constants of the unit cell. nstate = 32 indicates the number of Kohn-Sham orbitals to be solved. nelec = 32 indicate the number of valence electrons in the system. nelelem = 1 and natom = 8 indicate the number of elements and the number of atoms in the system, respectively. isym = 8 and crystal_structure = 'diamond', which indicates that the spatial symmetry of the unit cell is used in the calculation. Although the use of the symmetry substantially reduces the computational cost, it should be used very carefully. At present, the spatial symmetry has been implemented only for the case of the diamond structure. See *&system in Inputs* for more information.

&pseudo

```
&pseudo
  izatom(1)=14
  pseudo_file(1) = './Si_rps.dat'
```

(continues on next page)

(continued from previous page)

```
lloc_ps(1)=2
/
```

`izatom(1) = 14` indicates the atomic number of the element #1. `pseudo_file(1) = 'Si_rps.dat'` indicates the pseudopotential filename of element #1. `lloc_ps(1) = 2` indicate the angular momentum of the pseudopotential that will be treated as local.

&functional

```
&functional
xc='PZ'
/
```

This indicates that the adiabatic local density approximation with the Perdew-Zunger functional is used. We note that meta-GGA functionals that reasonably reproduce the band gap of various insulators may also be used in the calculation of periodic systems. See *&functional in Inputs* for detail.

&rgrid

Mandatory: `dl` or `num_rgrid`

```
&rgrid
num_rgrid = 12,12,12
/
```

`num_rgrid=12,12,12` specifies the number of the grids for each Cartesian direction. See *&rgrid in Inputs* for more information.

&kgrid

Mandatory: none

This input keyword provides grid spacing of k-space for periodic systems.

```
&kgrid
num_kgrid = 2,2,2
/
```

&tgrid

```
&tgrid
nt=4000
dt=0.08
/
```

`dt=0.08` specifies the time step of the time evolution calculation. `nt=4000` specifies the number of time steps in the calculation.

&propagation

```
&propagation
propagator='middlepoint'
/
```

`propagator = 'middlepoint'` indicates that Hamiltonian at midpoint of two-times is used. See *&propagation in Inputs* for more information.

&scf

Mandatory: `nscf`

This input keywords specify parameters related to the self-consistent field calculation.

```
&scf
  ncg = 5
  nscf = 120
/
```

`ncg = 5` is the number of conjugate-gradient iterations in solving the Kohn-Sham equation. Usually this value should be 4 or 5. `nscf = 120` is the number of scf iterations.

&emfield

```
&emfield
  ae_shape1 = 'Acos2'
  rlaser_int_wcm2_1 = 1d12
  pulse_tw1 = 441.195136248d0
  omega1 = 0.05696145187d0
  ekdir_re1 = 0.,0.,1.
/
```

This input keyword specifies the pulsed electric field applied to the system

`ae_shape1 = 'Acos2'` specifies the envelope of the pulsed electric field, \cos^2 envelope for the vector potential.

`ekdir_re1 = 0., 0., 1.` specify the real part of the unit polarization vector of the pulsed electric field. Specifying only the real part, it describes a linearly polarized pulse.

`laser_int_wcm2_1 = 1d12` specifies the maximum intensity of the applied electric field in unit of W/cm^2 .

`omega1=0.05696145187d0` specifies the average photon energy (frequency multiplied with \hbar).

`pulse_tw1=441.195136248d0` specifies the pulse duration. Note that it is not the FWHM but a full duration of the \cos^2 envelope.

See *&emfield in Inputs* for detail.

&multiscale

This input keyword specifies information necessary for Maxwell - TDDFT multiscale calculations.

```
&multiscale
  ftdddim = '1D'
  twod_shape = 'periodic'
  nx_m = 4
  ny_m = 1
  hx_m = 250d0
  nxvac1_m = -2000
  nxvacr_m = 256
/
```

`ftdddim` specifies the spatial dimension of the macro system. `ftdddim='1D'` indicates that one-dimensional equation is solved for the macroscopic vector potential.

`nx_m = 4` specifies the number of the macroscopic grid points in for x-direction in the spatial region where the material exists.

`hx_m = 250d0` specifies the grid spacing of the macroscopic grid in x-direction.

`nxvac1_m = -2000` and `nxvacr_m = 256` indicate the number of grid points in the vacuum region, `nxvac1_m` for the left and `nxvacr_m` for the right from the surface of the material.

&atomic_red_coor

Mandatory: atomic_coor or atomic_red_coor (they may be provided as a separate file)

```
&atomic_red_coor
'Si'   .0   .0   .0   1
'Si'   .25  .25  .25  1
'Si'   .5   .0   .5   1
'Si'   .0   .5   .5   1
'Si'   .5   .5   .0   1
'Si'   .75  .25  .75  1
'Si'   .25  .75  .75  1
'Si'   .75  .75  .25  1
/
```

Cartesian coordinates of atoms are specified in a reduced coordinate system. First column indicates the element, next three columns specify reduced Cartesian coordinates of the atoms, and the last column labels the element.

Output files

After the calculation, new directory *multiscale/* is created, then, following output files are created in the directory,

file name	description
<i>Si_gs_info.data</i>	results of the ground state as well as input parameters
<i>Si_eigen.data</i>	orbital energies in the ground state calculation
<i>Si_k.data</i>	information on k-points
<i>RT_Ac/Si_Ac_XXXXXX.data</i>	various quantities at a time as functions of macroscopic position
<i>RT_Ac/Si_Ac_vac.data</i>	vector potential at vacuum position adjacent to the medium
<i>MXXXXXX/Si_Ac_M.data</i>	various quantities at a macroscopic point as functions of time
<i>Si_gs_rt_multiscale.out</i>	standard output file

You may download the above files (zipped file) from:

https://salmon-tddft.jp/webmanual/v_1_2_0/exercise_zip_files/Si_gs_rt_multiscale_output.zip

Explanations of the output files are described below:

Si_gs_info.data

Results of the ground state as well as input parameters are provided.

```
#-----
#grid information-----
#aL = 10.2600000000000 10.2600000000000 10.2600000000000
#a1(1),a1(2),a1(3) = 10.2600000000000 10.2600000000000 10.
->2600000000000
#aLx,aLy,aLz = 10.2600000000000 10.2600000000000
10.2600000000000
#bLx,bLy,bLz = 0.612396228769940 0.612396228769940 0.612396228769940
#Nd = 4
#NLx,NLy,NLz= 12 12 12
#NL = 1728
#Hx,Hy,Hz = 0.855000000000000 0.855000000000000 0.855000000000000
#(pi/max(Hx,Hy,Hz))**2 = 13.5010490764192
#(pi/Hx)**2+(pi/Hy)**2+(pi/Hz)**2 = 40.5031472292576
#Hxyz = 0.625026375000000
```

(continues on next page)

(continued from previous page)

```

#NKx,NKy,NKz=          2          2          2
#NKxyz =              8
#Sym=                1
#NK =                8
#NEwald, aEwald =          4  0.5000000000000000
#-----
#GS calc. option-----
#FSset_option =n
#Ncg=                5
#Nmemory_MB,alpha_MB =          8  0.7500000000000000
#NFSset_start,NFSset_every =          75          25
#Nscf=               100
#Nscf_conv=          100
#NI,NE=              8          1
#Zatom=              14
#Lref=                2
#i,Kion(ia)(Rion(j,a),j=1,3)
#          1          1
# 0.0000000000000000E+000  0.0000000000000000E+000  0.0000000000000000E+000
#          2          1
# 2.5650000000000000  2.5650000000000000  2.5650000000000000
#          3          1
# 5.1300000000000000  0.0000000000000000E+000  5.1300000000000000
#          4          1
# 0.0000000000000000E+000  5.1300000000000000  5.1300000000000000
#          5          1
# 5.1300000000000000  5.1300000000000000  0.0000000000000000E+000
#          6          1
# 7.6950000000000000  2.5650000000000000  7.6950000000000000
#          7          1
# 2.5650000000000000  7.6950000000000000  7.6950000000000000
#          8          1
# 7.6950000000000000  7.6950000000000000  2.5650000000000000
#-----
#GS information-----
#NB,Nelec=           32          32
#Eall = -31.2444435912435
#ddns(iter = Nscf_conv)  1.054470043491702E-009
#ddns_abs_le(iter = Nscf_conv)  7.414743076744689E-010
#esp_var_ave(iter = Nscf_conv)  1.020334316849951E-008
#esp_var_max(iter = Nscf_conv)  2.402374610033353E-008
#NBoccmx is          16
#-----
#band information-----
#Bottom of VB -0.171685135011666
#Top of VB  0.200444750626413
#Bottom of CB  0.279085896060740
#Top of CB  0.467391510426397
#Fundamental gap  7.864114543432713E-002
#Fundamental gap[eV]  2.13995139310074
#BG between same k-point  7.864114544171730E-002
#BG between same k-point[eV]  2.13995139330183
#Physically upper bound of CB for DOS  0.467391509327423
#Physically upper bound of CB for eps(omega)  0.639076644315190
#-----
#iter      total-energy      ddns/nelec      esp_var_ave      esp_var_max
  1      -0.2036656319E+02      0.7386703002E+00      0.1069029083E+00      0.1685663917E+00

```

(continues on next page)

(continued from previous page)

2	-0.2528602134E+02	0.4434674022E+00	0.1251761861E+00	0.1947159558E+00
3	-0.2862371415E+02	0.2037034214E+00	0.5065455837E-01	0.8653325118E-01
4	-0.3011102231E+02	0.1582114006E+00	0.4920903464E-01	0.1181905324E+00
5	-0.3087795790E+02	0.1077328223E+00	0.1964285309E-01	0.4585050374E-01
...				
96	-0.3124444359E+02	0.4299999378E-09	0.1045606891E-07	0.2402425601E-07
97	-0.3124444359E+02	0.3271696693E-09	0.1042233851E-07	0.2402417900E-07
98	-0.3124444359E+02	0.2672131350E-09	0.1041933599E-07	0.3916492026E-07
99	-0.3124444359E+02	0.7129545698E-09	0.1020262421E-07	0.2402357654E-07
100	-0.3124444359E+02	0.7414743077E-09	0.1020334317E-07	0.2402374610E-07

Si_eigen.data

Orbital energies in the ground state calculation.

```

# Ground state eigenenergies
# ik: k-point index
# ib: Band index
# energy: Eigenenergy
# occup: Occupation
# 1:ik[none] 2:ib[none] 3:energy[a.u.] 4:occup[none]
  1      1 -1.71685134987767E-001  2.00000000000000E+000
  1      2 -9.95580252948176E-002  2.00000000000000E+000
  1      3 -9.95580233898604E-002  2.00000000000000E+000
  1      4 -9.95580220480095E-002  2.00000000000000E+000
  1      5  2.73787456875120E-003  2.00000000000000E+000
...
  8     28  4.34674205363501E-001  0.00000000000000E+000
  8     29  4.40615204659137E-001  0.00000000000000E+000
  8     30  4.40615205419465E-001  0.00000000000000E+000
  8     31  4.40615206907897E-001  0.00000000000000E+000
  8     32  4.67391509348806E-001  0.00000000000000E+000

```

Si_k.data

Information on k-points. Note that diamond symmetry is used to reduce the k-point in this calculation.

```

# k-point distribution
# ik: k-point index
# kx,ky,kz: Reduced coordinate of k-points
# wk: Weight of k-point
# 1:ik[none] 2:kx[none] 3:ky[none] 4:kz[none] 5:wk[none]
  1 -2.50000000000000E-001 -2.50000000000000E-001 -2.50000000000000E-001  1.
↪00000000000000E+000
  2 -2.50000000000000E-001 -2.50000000000000E-001  2.50000000000000E-001  1.
↪00000000000000E+000
  3 -2.50000000000000E-001  2.50000000000000E-001 -2.50000000000000E-001  1.
↪00000000000000E+000
  4 -2.50000000000000E-001  2.50000000000000E-001  2.50000000000000E-001  1.
↪00000000000000E+000
  5  2.50000000000000E-001 -2.50000000000000E-001 -2.50000000000000E-001  1.
↪00000000000000E+000
  6  2.50000000000000E-001 -2.50000000000000E-001  2.50000000000000E-001  1.
↪00000000000000E+000

```

(continues on next page)

(continued from previous page)

```

7  2.500000000000000E-001  2.500000000000000E-001  -2.500000000000000E-001  1.
↪000000000000000E+000
8  2.500000000000000E-001  2.500000000000000E-001  2.500000000000000E-001  1.
↪000000000000000E+000

```

RT_Ac/Si_Ac_000000.data

The number in the file name specifies the iteration number. Various quantities at a time are shown as function of macroscopic position.

- column 1-3: grid number of macroscopic coordinate in 3D format.
- column 4-6: macroscopic vector potential
- column 7-9: macroscopic electric field
- column 10-12: macroscopic magnetic field
- column 13-15: macroscopic current
- column 16: electronic excitation energy per unit cell
- column 17: energy absorbed by electrons per unit cell
- column 18: energy of macroscopic electromagnetic fields per unit cell

```

# Macroscopic field distribution
# IX,IY,IZ: Coordinate
# Ac: Vector potential field
# E: Electric field
# B: Magnetic field
# Jm: Matter current density
# Eex: Electron excitation energy
# Eabs: Absorbed energy
# Eemf: Total EM field energy
# 1:Ix[none] 2:Iy[none] 3:Iz[none] 4:Ac_x[a.u.] 5:Ac_y[a.u.] 6:Ac_z[a.u.] 7:Ex[a.u.]
↪8:Ey[a.u.] 9:Ez[a.u.] 10:B_x[a.u.] 11:B_y[a.u.] 12:B_z[a.u.] 13:Jm_x[a.u.] 14:Jm_
↪y[a.u.] 15:Jm_z[a.u.] 16:Eex[a.u./unitcell] 17:Eabs[a.u./unitcell] 18:Eemf[a.u./
↪unitcell]
-2000  1  1  0.000000000000000E+000  0.000000000000000E+000  0.
↪000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000  0.
↪000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000  0.
↪000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000  0.
↪000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000  0.
↪000000000000000E+000
-1999  1  1  0.000000000000000E+000  0.000000000000000E+000  0.
↪000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000  0.
↪000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000  0.
↪000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000  0.
↪000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000  0.
↪000000000000000E+000
-1998  1  1  0.000000000000000E+000  0.000000000000000E+000  0.
↪000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000  0.
↪000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000  0.
↪000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000  0.
↪000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000  0.
↪000000000000000E+000
-1997  1  1  0.000000000000000E+000  0.000000000000000E+000  0.
↪000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000  0.
↪000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000  0.
↪000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000  0.
↪000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000  0.
↪000000000000000E+000

```

(continues on next page)

(continued from previous page)

```

-1996      1      1  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000

...

 252      1      1  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000

 253      1      1  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000

 254      1      1  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000

 255      1      1  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000

 256      1      1  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000
    
```

RT_Ac/Si_Ac_vac.data

Vector potentials at vacuum points adjacent to the medium are shown, L for left and R for right adjacent point.

```

# Ac vacuum region
# Data of Ac field at the end of media
# L:      0      1      1
# R:      5      1      1
# 1:Time[a.u.] 2:Ac_x(L)[a.u.] 3:Ac_y(L)[a.u.] 4:Ac_z(L)[a.u.] 5:Ac_x(R)[a.u.] 6:Ac_
↪y(R)[a.u.] 7:Ac_z(R)[a.u.]
  0.00000000  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000

  0.08000000  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000

  0.16000000  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000  0.00000000000000E+000  0.00000000000000E+000  0.
↪00000000000000E+000
    
```

(continues on next page)

(continued from previous page)

```

0.24000000 0.000000000000000E+000 1.02802304891111E-015 3.34045640443765E-
↪016 0.000000000000000E+000 1.02802304891111E-015 3.34045640443765E-016
0.32000000 0.000000000000000E+000 5.19696825069140E-015 1.73169501167652E-
↪015 0.000000000000000E+000 5.19696825069140E-015 1.73169724492077E-015

...

319.68000000 0.000000000000000E+000 -8.56815797183038E-007 -1.74158413235687E-
↪001 0.000000000000000E+000 -8.00468477308030E-007 -1.58038091464266E-001
319.76000000 0.000000000000000E+000 -8.57513903130219E-007 -1.74393964495627E-
↪001 0.000000000000000E+000 -8.01174071107012E-007 -1.58187516982663E-001
319.84000000 0.000000000000000E+000 -8.58202516138042E-007 -1.74627725814573E-
↪001 0.000000000000000E+000 -8.01876269144167E-007 -1.58336335855095E-001
319.92000000 0.000000000000000E+000 -8.58881726038604E-007 -1.74859695029779E-
↪001 0.000000000000000E+000 -8.02575101929425E-007 -1.58484539522173E-001
320.00000000 0.000000000000000E+000 -8.59551627002296E-007 -1.75089870177989E-
↪001 0.000000000000000E+000 -8.03270602482416E-007 -1.58632119495344E-001

```

M000001/Si_Ac_M.data

The number in the file name specifies the macroscopic grid point in a medium. Various quantities at a macroscopic point are shown as functions of time.

```

# Local variable at macro point
# Macropoint: 1 1 1
# Jm: Matter current density
# Ac: External vector potential field
# 1:Time[a.u.] 2:Ac_x[a.u.] 3:Ac_y[a.u.] 4:Ac_z[a.u.] 5:Jm_x[a.u.] 6:Jm_y[a.u.] 7:Jm_
↪z[a.u.]
0.00000000 0.000000000000000E+000 0.000000000000000E+000 0.
↪000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000 0.
↪000000000000000E+000
0.08000000 0.000000000000000E+000 0.000000000000000E+000 0.
↪000000000000000E+000 -2.75496322261169E-012 -6.64693329538576E-012 -2.
↪15985341184321E-012
0.16000000 2.21567288859314E-013 5.34578094327944E-013 1.73705718053823E-
↪013 -2.44154670033013E-012 -7.05287750131598E-012 -2.56975702246339E-012
0.24000000 6.39495414253759E-013 1.63535423011393E-012 5.53749512919215E-
↪013 -2.13170597168112E-012 -7.46016650418711E-012 -2.97426987823802E-012
0.32000000 1.22886550624652E-012 3.33297163342029E-012 1.17193467844178E-
↪012 -1.81810297340216E-012 -7.87600103950396E-012 -3.35839167619253E-012

...

319.68000000 1.23027707032365E-003 -8.70482807641121E-007 -1.71017066989709E-
↪001 1.02881666841023E-007 4.92374979404618E-010 -3.32578810347408E-005
319.76000000 1.23227860920852E-003 -8.70992162244069E-007 -1.71172549338950E-
↪001 9.87865048046008E-008 4.85608114795546E-010 -3.45948031169888E-005
319.84000000 1.23427220321126E-003 -8.71495026996183E-007 -1.71326503499639E-
↪001 9.45989194632210E-008 4.78870087348624E-010 -3.59262231464884E-005
319.92000000 1.23625818911749E-003 -8.71991570086865E-007 -1.71478932689850E-
↪001 9.03230551511057E-008 4.72163059159496E-010 -3.72519512786483E-005
320.00000000 1.23823691081260E-003 -8.72481961985393E-007 -1.71629840326064E-
↪001 8.59631416251532E-008 4.65489301170927E-010 -3.85718005737764E-005

```

Si_gs_rt_multiscale.out

Standard output file.

4.5 FDTD simulation

4.5.1 Exercise-7: Polarizability and photoabsorption of a metallic nanosphere in classical electromagnetism

In this exercise, we learn the linear response calculation in the metallic nanosphere, solving the time-dependent Maxwell equations. The linear response calculation provides the polarizability and the oscillator strength distribution of the metallic nanosphere. In the calculation, an impulsive perturbation is applied to all electrons in the metallic nanosphere along the z axis, where the electrons are described by the Newtonian equation in the Drude model. Then a time evolution calculation is carried out without any external fields. During the calculation, the electric dipole moment is monitored. After the time evolution calculation, a time-frequency Fourier transformation is carried out for the electric dipole moment to obtain the frequency-dependent polarizability. The imaginary part of the frequency-dependent polarizability is proportional to the oscillator strength distribution and the photoabsorption cross section.

Input files

To run the code, the input file *classicEM_rt_response.inp* that contains input keywords and their values for the linear response calculation is required. The shape file of the metallic nanosphere *shape.cube* is also required. Those are included in SALMON(SALMON/examples/exercise_07_classicEM_lr/).

The shape file can be generated by program `FDTD_make_shape` in SALMON utilities:

<https://salmon-tddft.jp/utilities.html>

In the input file *classicEM_rt_response.inp*, input keywords are specified. Most of them are mandatory to execute the linear response calculation. This will help you to prepare the input file for other systems that you want to calculate. A complete list of the input keywords that can be used in the input file can be found in *List of all input keywords*.

```
!The read-in file 'shape_file' in &maxwell category can be generated by program 'FDTD_
↳make_shape' in SALMON utilities(https://salmon-tddft.jp/utilities.html).
&calculation
  Theory = 'Maxwell'
/

&control
  sysname = 'classicEM'
/

&units
  unit_system = 'A_eV_fs'
/

&system
  iperiodic = 0
/

&emfield
  ae_shape1 = 'impulse'
  ekdir_re1 = 0.0d0, 0.0d0, 1.0d0
/

&analysis
```

(continues on next page)

(continued from previous page)

```

nenergy = 1000
de      = 0.01d0
/

&maxwell
!grid and time-step information
al_em = 1000.0d0, 1000.0d0, 1000.0d0
dl_em = 10.0d0, 10.0d0, 10.0d0
nt_em = 5000
dt_em = 1.90d-3

!media information
shape_file   = 'shape.cube'
imedia_num   = 1
type_media(1) = 'drude'
omega_p_d(1) = 9.03d0
gamma_d(1)   = 0.53d0-1
/

```

We present their explanations below:

Required and recommended variables

&calculation

Mandatory: Theory

```

&calculation
  Theory = 'Maxwell'
/

```

This indicates that the real time classical electromagnetism calculation is carried out in the present job.

&control

Mandatory: none

```

&control
  sysname = 'classicEM'
/

```

classicEM defined by sysname = 'classicEM' will be used in the filenames of output files.

&units

Mandatory: none

```

&units
  unit_system = 'A_eV_fs'
/

```

This input keyword specifies the unit system to be used in the input file. If you do not specify it, atomic unit will be used.

&system

Mandatory: iperiodic


```
&system
  iperiodic = 0
/
```

`iperiodic = 0` indicates that the isolated boundary condition will be used in the calculation.

&emfield

Mandatory: `ae_shape1`, `epdir_re1`

```
&emfield
  ae_shape1 = 'impulse'
  epdir_re1 = 0.0d0, 0.0d0, 1.0d0
/
```

`ae_shape1 = 'impulse'` indicates that a weak impulse is applied to all electrons at $t=0$.

`epdir_re1(3)` specifies a unit vector that indicates the direction of the impulse.

&analysis

Mandatory: none

```
&analysis
  nenergy = 1000
  de      = 0.01d0
/
```

`nenergy = 1000` specifies the number of energy steps, and `de = 0.01d0` specifies the energy spacing in the time-frequency Fourier transformation.

&maxwell

Mandatory: `al_em`, `dl_em`, `nt_em`

```
&maxwell
!grid and time-step information
al_em = 1000.0d0, 1000.0d0, 1000.0d0
dl_em = 10.0d0, 10.0d0, 10.0d0
nt_em = 5000
dt_em = 1.90d-3

!media information
shape_file = 'shape.cube'
imedia_num = 1
type_media(1) = 'drude'
omega_p_d(1) = 9.03d0
gamma_d(1) = 0.53d0-1
/
```

`al_em = 1000.0d0, 1000.0d0, 1000.0d0` specifies the lengths of three sides of the rectangular parallelepiped where the grid points are prepared.

`dl_em = 10.0d0, 10.0d0, 10.0d0` specifies the grid spacings in three Cartesian directions.

`nt_em = 5000` specifies the number of time steps in the calculation.

`dt_em = 1.90d-3` specifies the time step of the time evolution calculation. If you do not input, this is automatically specified by the Courant-Friedrichs-Lewy Condition.

`shape_file = 'shape.cube'` indicates the filename of the shape file.

`imedia_num = 1` specifies the number of the types of media described by the shape file('shape.cube').

`type_media(1) = 'drude'` specifies the type of media as the Drude model.

`omega_p_d(1) = 9.03d0` and `gamma_d(1) = 0.53d0-1` specify the plasma- and collision-frequencies, respectively. See `&maxwell` in *List of all input keywords* for more information.

Output files

After the calculation, following output files are created in the directory that you run the code.

classicEM_p.data

For time steps from 1 to `nt_em`,

- 1 column: time
- 2-4 columns: x,y,z components of the dipole moment

classicEM_lr.data

For energy steps from 0 to `nenergy`,

- 1 column: energy
- 2-4 columns: x,y,z components of real part of the polarizability (time-frequency Fourier transformation of the dipole moment)
- 5-7 columns: x,y,z components of imaginary part of the polarizability (time-frequency Fourier transformation of the dipole moment)
- 8-10 columns: x,y,z components of power spectrum of the dipole moment

4.5.2 Exercise-8: Pulsed electric field response of a metallic nanosphere in classical electromagnetism

In this exercise, we learn the pulsed electric field response in the metallic nanosphere, solving the time-dependent Maxwell equations. As outputs of the calculation, the time response of the electromagnetic field is calculated. A pulsed electric field that has \cos^2 envelope shape is applied. The parameters that characterize the pulsed field such as magnitude, frequency, polarization direction, and carrier envelope phase are specified in the input file.

Input files

To run the code, the input file *classicEM_rt_pulse.inp* that specifies name list variables and their values for the pulsed electric field calculation is required. The shape file of the metallic nanosphere *shape.cube* is also required. Those are included in SALMON(SALMON/examples/exercise_08_classicEM_rt/).

The shape file can be generated by program `FDTD_make_shape` in SALMON utilities:

<https://salmon-tddft.jp/utilities.html>

In the input file *classicEM_rt_pulse.inp*, input keywords are specified. Most of them are mandatory to execute the calculation of the pulsed electric field response in the metallic nanosphere. This will help you to prepare the input file for other systems and other pulsed electric fields that you want to calculate. A complete list of the input keywords that can be used in the input file can be found in *List of all input keywords*.

```

!The read-in file 'shape_file' in &maxwell category can be generated by program 'FDTD_
↳make_shape' in SALMON utilities(https://salmon-tddft.jp/utilities.html).
!Results can be visualized by program 'FDTD_make_figani' in SALMON utilities.
&calculation
  Theory = 'Maxwell'
/

&control
  directory = 'result'
/

&units
  unit_system = 'A_eV_fs'
/

&system
  iperiodic = 0
/

&emfield
  ae_shapel      = 'Ecos2'
  rlaser_int_wcm2_1 = 1.0d8
  pulse_twl      = 4.75d0
  omegal         = 5.0d0
  ekdir_re1     = 0.0d0, 0.0d0, 1.0d0
  phi_cep1      = 0.75d0
/

&maxwell
  !grid and time-step information
  al_em = 1000.0d0, 1000.0d0, 1000.0d0
  dl_em = 10.0d0, 10.0d0, 10.0d0
  nt_em = 2500
  dt_em = 1.90d-3

  !media information
  shape_file      = 'shape.cube'
  imedia_num      = 1
  type_media(1)   = 'drude'
  omega_p_d(1)    = 9.03d0
  gamma_d(1)      = 0.53d0-1

  !source information
  wave_input      = 'source'
  source_loc1(:)  = -3.15d2, 0.0d0, 0.0d0
  ek_dir1(:)      = 1.0d0, 0.0d0, 0.0d0

  !observation information
  iobs_num_em     = 1
  iobs_samp_em    = 100
  obs_loc_em(1,:) = 0.0d0, 0.0d0, 0.0d0
/

```

We present their explanations below:

Required and recommended variables

&calculation

Mandatory: Theory

```
&calculation
  Theory = 'Maxwell'
/
```

This indicates that the real time classical electromagnetism calculation is carried out in the present job.

&control

Mandatory: none

```
&control
  directory = 'result'
/
```

result defined by `directory = 'result'` will be used in the directory name that contains output files.

&units

Mandatory: none

```
&units
  unit_system = 'A_eV_fs'
/
```

This input keyword specifies the unit system to be used in the input file. If you do not specify it, atomic unit will be used.

&system

Mandatory: `iperiodic`

```
&system
  iperiodic = 0
/
```

`iperiodic = 0` indicates that the isolated boundary condition will be used in the calculation.

&emfield

Mandatory: `ae_shape1`, { `rlaser_int_wcm2_1` or `amplitude1` }, `pulse_tw1`, `omega1`, `epdir_re1`, `phi_cep1`

```
&emfield
  ae_shape1      = 'Ecos2'
  rlaser_int_wcm2_1 = 1.0d8
  pulse_tw1      = 4.75d0
  omega1         = 5.0d0
  epdir_re1      = 0.0d0, 0.0d0, 1.0d0
  phi_cep1       = 0.75d0
/
```

`ae_shape1 = 'Ecos2'` indicates that the envelope of the pulsed electric field has a \cos^2 shape.

`rlaser_int_wcm2_1 = 1.0d8` specifies the maximum intensity of the applied electric field in unit of W/cm^2 .

`pulse_tw1 = 4.75d0` specifies the pulse duration. Note that it is not the FWHM but a full duration of the \cos^2 envelope.

`omega1 = 5.0d0` specifies the average photon energy (frequency multiplied with \hbar).

`epdir_re1 = 0.0d0, 0.0d0, 1.0d0` specifies the real part of the unit polarization vector of the pulsed electric field. Using the real polarization vector, it describes a linearly polarized pulse.

$\text{phi_cep1} = 0.75\text{d0}$ specifies the carrier envelope phase of the pulse. As noted above, 'phi_cep1' must be 0.75 (or 0.25) if one employs 'Ecos2' pulse shape, since otherwise the time integral of the electric field does not vanish.

&maxwell

Mandatory: al_em, dl_em, nt_em

```
&maxwell
!grid and time-step information
al_em = 1000.0d0, 1000.0d0, 1000.0d0
dl_em = 10.0d0, 10.0d0, 10.0d0
nt_em = 5000
dt_em = 1.90d-3

!media information
shape_file   = 'shape.cube'
imedia_num   = 1
type_media(1) = 'drude'
omega_p_d(1) = 9.03d0
gamma_d(1)   = 0.53d0-1

!source information
wave_input   = 'source'
source_loc1(:) = -3.15d2, 0.0d0, 0.0d0
ek_dir1(:)   = 0.0d0, 1.0d0, 0.0d0

!observation information
iobs_num_em  = 1
iobs_samp_em = 100
obs_loc_em(1,:) = 0.0d0, 0.0d0, 0.0d0
/
```

$\text{al_em} = 1000.0\text{d0}, 1000.0\text{d0}, 1000.0\text{d0}$ specifies the lengths of three sides of the rectangular parallelepiped where the grid points are prepared.

$\text{dl_em} = 10.0\text{d0}, 10.0\text{d0}, 10.0\text{d0}$ specifies the grid spacings in three Cartesian directions.

$\text{nt_em} = 5000$ specifies the number of time steps in the calculation.

$\text{dt_em} = 1.90\text{d-}3$ specifies the time step of the time evolution calculation. If you do not input, this is automatically specified by the Courant-Friedrichs-Lewy Condition.

$\text{shape_file} = \text{'shape.cube'}$ indicates the filename of the shape file.

$\text{imedia_num} = 1$ specifies the number of the types of media described by the shape file('shape.cube').

$\text{type_media}(1) = \text{'drude'}$ specifies the type of media as the Drude model.

$\text{omega_p_d}(1) = 9.03\text{d0}$ and $\text{gamma_d}(1) = 0.53\text{d0-}1$ specify the plasma- and collision-frequencies, respectively.

$\text{wave_input} = \text{'source'}$ specifies a current source that is used for generating the pulse.

$\text{source_loc1}(:) = -3.15\text{d2}, 0.0\text{d0}, 0.0\text{d0}$ specifies the coordinate of the current source.

$\text{ek_dir1}(:) = 0.0\text{d0}, 1.0\text{d0}, 0.0\text{d0}$ specifies the propagation direction of the pulse (x,y,z).

$\text{iobs_num_em} = 1$ specifies the number of the observation point.

$\text{iobs_samp_em} = 100$ specifies the sampling number for time steps. In this case, output files are generated every 100 steps.

$\text{obs_loc_em}(1,:) = 0.0\text{d0}, 0.0\text{d0}, 0.0\text{d0}$ specifies the coordinate of the current source.

See `&maxwell` in *List of all input keywords* for more information.

Output files

After the calculation, following output files are created into 'result' in the directory that you run the code.

obs1_at_point.data

For time steps from 1 to `nt_em`,

- 1 column: time
- 2-4 columns: x,y,z components of the electric field in the observation point
- 5-7 columns: x,y,z components of the magnetic field in the observation point

obs1_@@_+_-.data and obs0_info.data

`obs1_@@_+_-.data` is the spatial distribution of @@ component(ex, ey, ez, hx, hy, or hz) on ++ plane(xy, yz, or xz) in -- time step.

`obs0_info.data` is the basic information of the calculation.

The time response figure and animation of the electromagnetic fields at the observation point can be generated by program `FDTD_make_figani` in SALMON utilities:

<https://salmon-tddft.jp/utilities.html>

Appendix:

LIST OF ALL INPUT KEYWORDS

- *&calculation*
- *&control*
- *&units*
- *¶llel*
- *&system*
- *&atomic_red_coor*
- *&atomic_coor*
- *&pseudo*
- *&functional*
- *&rgrid*
- *&kgrid*
- *&tgrid*
- *&propagation*
- *&scf*
- *&emfield*
- *&multiscale*
- *&maxwell*
- *&analysis*
- *&hartree*
- *&ewald*
- *&opt* (Trial)
- *&md* (Trial)
- *&misc*
- **'&group_fundamental'**_ (Trial)
- **'&group_parallel'**_ (Trial)
- **'&group_hartree'**_ (Trial)
- **'&group_file'**_ (Trial)

- **'&group_others'** (Trial)

5.1 &calculation

- **theory (character, 0d/3d)** Choice of Calculation theories. 'TDDFT' and 'Maxwell' can be chosen. Default is 'TDDFT'.
- **calc_mode (character, 0d/3d)** Choice of Calculation modes. 'GS' and 'RT' can be chosen. If `&system/iperiodic=3`, 'GS_RT' can be chosen.
- **use_ms_maxwell (character, 3d)** Enable('y')/disable('n') Multi-scale Maxwell-Kohn-Sham coupling. Default is 'n'
- **use_force (character, 0d)** Enable('y')/disable('n') force calculation. Default is 'n'.
- **(Trial) use_adiabatic_md (character, 3d)** Enable('y')/disable('n'). Adiabatic ground-state molecular dynamics option. Default is 'n'.
- **(Trial) use_ehrenfest_md (character, 0d/3d)** Enable('y')/disable('n'). Ehrenfest molecular dynamics option. Default is 'n'.
- **(Trial) use_geometry_opt (character, 0d/3d)** Enable('y')/disable('n'). Geometry optimization option. Default is 'n'.

5.2 &control

- **sysname (character, 0d/3d)** Name of calculation. This is used for a prefix of output files. Default is `default`.
- **directory (character, 0d/3d)** Name of a default directory, where the basic results will be written down. Default is the current directory, `./`.
- **(Trial) restart_option (character, 3d)** Flag for restart, 'new' or 'restart'. 'new' is default.
- **(Trial) backup_frequency (integer, 3d)** Frequency of backup during the time-propagation. If 0 is set, the backup is not performed. Default is 0.
- **(Trial) time_shutdown (real(8), 3d)** Timer for automatic shutdown. The unit is always second. If negative time is chosen, the automatic shutdown is not performed. Default is `-1` sec.
- **(Trial) dump_filename (character, 3d)** Name of a filename for the restart calculation.
- **(Trial) read_gs_wfn_k (character, 3d)** Read ground state wave function as initial guess (from pre-calculated "gs_wfn_k" directory printed by `calc_mode=GS`) if this option is `y`. The option is available for restart of ground state SCF calculation, geometrical optimization, etc. But the data is automatically read when `calc_mode=RT`. Default is `n`.
- **(Trial) write_gs_wfn_k (character, 3d)** Write ground state wave function into "gs_wfn_k" directory if this option is `y`, which is usually used for `calc_mode=GS_RT` calculation. (But the data is always printed in the case of `calc_mode=GS` calculation.) Default is `n`.
- **(Trial) modify_gs_wfn_k (character, 3d)** Option to modify initial guess wave function (in the pre-calculated "gs_wfn_k" directory) used in combination with `read_gs_wfn_k = y`. If `copy_1stk_to_all` is set, the first k-point data file, `wfn_gs_k0000001.wfn` (supposed to be obtained by gamma-point calculation), is copied to all other points. Default is `n`.
- **(Trial) read_rt_wfn_k (character, 3d)** Read RT wave function (from pre-calculated "rt_wfn_k" directory printed by `calc_mode=RT` or `calc_mode=GS_RT` with option of `write_rt_wfn_k=y`) if this is `y`. This is used for restart in combination with `calc_mode=RT` (if `use_ehrenfest_md=y`, coordinates

and velocities of atoms for restart must be included, too), then, "gs_wfn_k" directory is also necessary (even though this is actually not used if any analysis options are specified (but used for some analysis options)). Note that, currently, field is taken over after restarting only if `ae_shape1=AcosX` type is used. Default is `n`.

- **(Trial) write_rt_wfn_k (character, 3d)** Write RT wave function at the last time step into "rt_wfn_k" directory if this is `y`. (if `use_ehrenfest_md=y`, coordinates and velocities of atoms are also printed.) Default is `n`.
- **(Trial) read_gs_wfn_k_ms (character, 3d)** Read ground state wave function at each macro-grid point as initial state for multiscale calculation. This should be used together with `use_ms_maxwell='y'`, `calc_mode='RT'` and `set_ini_coor_vel='y'`. The ground state wave function data ('gs_wfn_k') must be pre-calculated for each macro-grid point (configuration or atomic coordinate can be different from macro-grid point to macro-grid point) and be put into the specific directories: `directory/multiscale/MXXXXXX/` where `XXXXXX` is the index number of the macro-grid point of the material region usually starting from '000001' up to the number of macro-grid point ('ini_coor_vel.dat' used by the option `set_ini_coor_vel` must be put in the same place). Default is `n`.
- **(Trial) read_rt_wfn_k_ms (character, 3d)** Read RT wave function and field information as initial state in multiscale calculation. These are the printed data at the last time step in the previous calculation generated by using `write_rt_wfn_k_ms='y'`. If you give incident pulse from input file option, the field is added. Default is `n`.
- **(Trial) write_rt_wfn_k_ms (character, 3d)** Write RT wave function and field information at the last step in multiscale calculation. It is used for restarting by using `read_rt_wfn_k_ms='y'`. Default is `n`.

5.3 &units

- **unit_system (character, 0d/3d)** Unit for input variables. If 'au' or 'a.u.', atomic unit system is used. If 'A_eV_fs', Angstrom-eV-fs unit system is used.

5.4 ¶llel

- **(Trial) domain_parallel (character, 3d)** If specified `domain_parallel='y'` and `&system/iperiodic=3`, program codes for domain parallel version run in periodic system calculations.
- **nproc_k/nproc_ob/nproc_domain(3)/nproc_domain_s(3) (integer, 0d)** Followings are explanation of each variable.

- `nproc_k`: Number of MPI parallelization for orbitals that related to the wavefunction calculation.
- `nproc_ob`: Number of MPI parallelization for orbitals that related to the wavefunction calculation.
- `nproc_domain(3)'`: Number of MPI parallelization for each direction in real-space that related to the wavefunction calculation.
- `nproc_domain_s(3)'`: Number of MPI parallelization for each direction in real-space that related to the electron density calculation.

Defaults are 0 for `nproc_k/nproc_ob` and (0/0/0) for `nproc_domain/nproc_domain_s`. If users use the defaults, automatic process assignment is done. Users can also specify `nproc_k`, `nproc_ob`, `nproc_domain`, and `nproc_domain_s` manually. In that case, `nproc_k` must be set to 1 for isolated system calculations. In addition, followings must be satisfied.

- $nproc_k * nproc_ob * nproc_domain(1) * nproc_domain(2) * nproc_domain(3) = \text{total number of processors}$

- `nproc_domain_s(1) * nproc_domain_s(2) * nproc_domain_s(3)` = total number of processors
- `nproc_domain_s(1)` is a multiple of `nproc_domain(1)`
- `nproc_domain_s(2)` is a multiple of `nproc_domain(2)`
- `nproc_domain_s(3)` is a multiple of `nproc_domain(3)`
- **num_datafiles_in/num_datafiles_out (integer, 0d)** Number of input/output files for wavefunction. Defaults are 1. If `num_datafiles_in/num_datafiles_out` are 1, wave functions are read from/ written in a regular intermediate file. If `num_datafiles_in/num_datafiles_out` are larger than or equal to 2, the wave functions are read from/ written in separated intermediate data files, and number of files are equal to `num_datafiles_in/num_datafiles_out`. These variables must be equal to nth power of 2. (n: 0 or positive integer)

5.5 &system

- **iperiodic (integer, 0d/3d)** Dimension for periodic boundary condition. 0 is for isolated systems, and 3 is for solids. Default is 0.
- **ispin (integer, 0d)** Variable for classification of closed shell systems and open shell systems. 0 is for closed shell systems, and 1 is for open shell systems. Default is 0
- **al(3) (real(8), 0d/3d)** Lattice constants. Unit of the length can be chosen by `&units/unit_system`.
- **isym (integer, 3d)** Number of symmetries that can be used for reduction of k-points. Default is 0.
- **crystal_structure (character, 3d)** Name of symmetry that can be used for the reduction of # of k-points. Default is 'none'.
- **nstate (integer, 0d/3d)** Number of states/bands.
- **nstate_spin(2) (integer, 0d)** Number of states/bands can be specified independently by `nstate_spin(1) / nstate_spin(2)`. This option is incompatible with `nstate`
- **nelec (integer, 0d/3d)** Number of valence electrons.
- **nelec_spin(2) (integer, 0d)** Number of up/down-spin electrons can be specified independently by `nelec_spin(1) / nelec_spin(2)`. This option is incompatible with `nelec`
- **temperature (real(8), 3d)** Temperature of electrons. When you calculate a system of zero band-gap energy like metals, zero or positive number of the temperature should be given. Unit of the energy can be chosen `&units/unit_system`. Default is -1.0 (this is for system which has a band gap energy).
- **(Trial) temperature_k (real(8), 0d)** Temperature of electrons [K]. Default is -1.d0.
- **nelem (integer, 0d/3d)** Number of elements that will be used in calculations.
- **natom (integer, 0d/3d)** Number of atoms in a calculation cell.
- **(Trial) file_atom_red_coor (character, 3d)** File name of atomic positions. In this file, the atomic coordinates can be written in reduced coordinates. This option is incompatible with `&system/file_atom_coor`, `&atomic_coor`, and `&atomic_red_coor`.
- **(Trial) file_atom_coor (character, 0d)** File name of atomic positions. In this file, the atomic coordinates can be written in Cartesian coordinates. The unit of the length can be chosen by `&units/unit_system`. This option is incompatible with `&system/file_atom_red_coor`, `&atomic_coor`, and `&atomic_red_coor`.

5.6 &atomic_red_coor

In &atomic_red_coor, positions of atoms can be written in reduced coordinates as follows:

```
'Si' 0.00 0.00 0.00 1
'Si' 0.25 0.25 0.25 1
...
```

Here, the information of atoms is ordered in row. For example, the first row gives the information of the first atom. The number of rows must be equal to &system/natom. The first column can be any characters and does not affect calculations. The second, third and fourth columns are reduced coordinates for the first, second and third directions, respectively. The fifth column is a serial number of the atom species, which is used in &pseudo. This option is incompatible with &system/file_atom_red_coor, &system/file_atom_coor, and &atomic_coor.

5.7 &atomic_coor

In &atomic_coor, positions of atoms can be written in Cartesian coordinates. The structure is same as &atomic_red_coor. The unit of the length can be chosen by &units/unit_length. This option is incompatible with &system/file_atom_red_coor, &system/file_atom_coor, and &atomic_red_coor.

5.8 &pseudo

Input for pseudopotentials. Size of array (:) is equal to &system/nelem.

- **pseudo_file(:)** (character, 0d/3d) Name of pseudopotential files.
- **lmax_ps(:)** (integer, 0d/3d) Maximum angular momentum of pseudopotential projectors.
- **lloc_ps(:)** (integer, 0d/3d) Angular momentum of pseudopotential that will be treated as local.
- **izatom(:)** (integer, 0d/3d) Atomic number.
- **(Trial) psmask_option(:)** (character, 0d/3d) Enable('y')/disable('n') Fourier filtering for pseudopotentials. Default is 'n'.
- **(Trial) alpha_mask(:)** (real(8), 0d/3d) Parameter for the Fourier filtering for pseudopotential. Default is '0.8'.
- **(Trial) gamma_mask(:)** (real(8), 0d/3d) Parameter for the Fourier filtering for pseudopotential. Default is '1.8'.
- **(Trial) eta_mask(:)** (real(8), 0d/3d) Parameter for the Fourier filtering for pseudopotential. Default is '15.0'.

5.9 &functional

- **xc** (character, 0d/3d) Exchange-correlation functionals. At present version, the functional 'PZ', 'PZM' and 'TBmBJ' is available for both 0d/3d calculations, and the functionals 'TPSS' and 'VS98' are available for 3d calculations.

- 'PZ': Perdew-Zunger LDA :Phys. Rev. B 23, 5048 (1981).
- 'PZM': Perdew-Zunger LDA with modification to improve sooth connection between high density form and low density one. :J. P. Perdew and Alex Zunger, Phys. Rev. B 23, 5048 (1981).
- 'TBmBJ': Tran-Blaha meta-GGA exchange with Perdew-Wang correlation. :Fabien Tran and Peter Blaha, Phys. Rev. Lett. 102, 226401 (2008). John P. Perdew and Yue Wang, Phys. Rev. B 45, 13244 (1992).
- 'TPSS': Tao, Perdew, Staroverov and Scuseria meta-GGA exchange correlation. :J. Tao, J. P. Perdew, V. N. Staroverov, and G. E. Scuseria, Phys. Rev. Lett. 91, 146401 (2003).
- 'VS98': van Voorhis and Scuseria exchange with Perdew-Wang correlation: T. Van Voorhis and G. E. Scuseria, J. Chem. Phys. 109, 400 (1998).
- **alibxc, alibx, alibc (character, 0d/3d)** By specifying `alibxc`, the functionals prepared in libxc package are available. They can be set indivisually by specifying `alibx` and `alibc`. To use libxc libraries, `--with-libxc` option must be added in executing `configure`. The available option of the exchange-correlation functionals are listed in the LibXC website. [See <http://www.tddft.org/programs/libxc/functionals/>]
- **cval (real(8), 3d)** Mixing parameter in Tran-Blaha meta-GGA exchange potential. If `cval` is set to a minus value, the mixing-parameter computed by the formula in the original paper [Phys. Rev. Lett. 102, 226401 (2008)]. Default is estimated from $\langle |\nabla\rho(\mathbf{r};t)|/\rho(\mathbf{r};t) \rangle$.
- **(Trial) no_update_func character (1); 3d** Option not to update functional (or Hamiltonian) in RT calculation, i.e., keep ground state Hamiltonian during time-evolution. Default is 'n'.

5.10 &rgrid

- **dl(3) (real(8), 0d/3d)** Spacing of real-space grids. Unit of length can be chosen by `&units/unit_system`. This variable cannot be set with `&rgrid/num_rgrid` simultaneously. If `&system/iperiodic` is set to 3, the actual grid spacing is automatically refined in calculations so that the size of the simulation box `&system/al(3)` becomes divisible by the spacing.
- **num_rgrid(3) (integer, 3d)** Number of real-space grids. This variable cannot be set with `&rgrid/dl` simultaneously.

5.11 &kgrid

- **num_kgrid(3) (integer, 3d)** Number of k-points (grid points of k-vector) discretizing the Brillouin zone. Each component must be even.
- **file_kw (character, 3d)** Name of a file for flexible k-point sampling. This file will be read if `num_kgrid` is smaller than 1.

5.12 &tgrid

- **nt (integer, 0d/3d)** Number of total time steps for real-time propagation.
- **dt (real(8), 0d/3d)** Time step. Unit of time can be chosen by `&units/unit_system`.

5.13 &propagation

- **propagator (character, 3d)** Choice of Propagator. `middlepoint` is an propagator with the Hamiltonian at midpoint of two-times. `etrs` is enforced time-reversal symmetry propagator. [M.A.L. Marques, A. Castro, G.F. Bertsch, and A. Rubio, *Comput. Phys. Commun.*, 151 60 (2003)]. Default is `middlepoint`.
- **(Trial) n_hamil (integer, 0d)** Order of Taylor expansion of a propagation operator. Default is 4.

5.14 &scf

- **nscf (integer, 0d/3d)** Number of maximum scf cycle.
- **amin_routine (character, 0d)** Minimization routine for the ground state calculation. `'cg'`, `'diis'`, and `'cg-diis'` can be chosen. Default is `'cg'`.
- **ncg (integer, 0d/3d)** Number of iteration of Conjugate-Gradient method for each scf-cycle. Default is 5.
- **amixing (character, 0d)** Methods for density/potential mixing for scf cycle. `simple` and `broyden` can be chosen. Default is `broyden`.
- **rmixrate (real(8), 0d)** Mixing ratio for simple mixing. Default is 0.5.
- **nmemory_mb (integer, 0d/3d)** Number of stored densities at previous scf-cycles for the modified-Broyden method. Default is 8. If `&system/iperiodic` is 0, `nmemory_mb` must be less than 21.
- **alpha_mb (real(8), 0d/3d)** Parameter of the modified-Broyden method. Default is 0.75.
- **(Trial) fsset_option (character, 3d)** Probably, we should remove this function since we can replace it with occupation smoothing with temperature.
- **(Trial) nfsset_start (integer, 3d)** Probably, we should remove this function since we can replace it with occupation smoothing with temperature.
- **(Trial) nfsset_every (integer, 3d)** Probably, we should remove this function since we can replace it with occupation smoothing with temperature.
- **subspace_diagonalization (character, 0d)** Enable('y')/disable('n') subspace diagonalization during scf cycle.
- **convergence (character, 0d/3d)** Choice of quantity that is used for convergence check in a scf calculation. Default is `'rho_dne'`.
 - `'rho_dne'`: Convergence is checked by $\sum_{ix} |\rho(ix, iter) - \rho(ix, iter-1)| dx / N$, where `iter` is an iteration number of the scf calculation and `N` is `&system/nelec`, the number of the valence electrons.
For isolated systems, the followings can also be chosen.
 - `'norm_rho'`: Convergence is checked by the square of the norm of difference of density, $\|\rho_{iter}(ix) - \rho_{iter-1}(ix)\|^2 = \sum_{ix} |\rho(ix, iter) - \rho(ix, iter-1)|^2$.
 - `'norm_rho_dng'`: Convergence is checked by $\|\rho_{iter}(ix) - \rho_{iter-1}(ix)\|^2 / (\text{number of grids})$. "dng" means "divided by number of grids".
 - `'norm_pot'`: Convergence is checked by $\|V_{local_iter}(ix) - V_{local_iter-1}(ix)\|^2$, where `Vlocal` is `Vh + Vxc + Vps_local`.
 - `'pot_dng'`: Convergence is checked by $\|V_{local_iter}(ix) - V_{local_iter-1}(ix)\|^2 / (\text{number of grids})$.
- **threshold (real(8), 0d/3d)** Threshold for convergence check that is used when `'rho_dne'` is specified. Default is `1d-17`.

- **threshold_norm_rho (real(8), 0d)** Threshold for convergence check that is used when either 'norm_rho' or 'norm_rho_dng' is specified. threshold_norm_rho must be set when either 'norm_rho' or 'norm_rho_dng' is specified. Default is -1d0 a.u. (1 a.u.= 45.54 A⁻⁶)
- **threshold_norm_pot (real(8), 0d)** Threshold for convergence check that is used when either 'norm_pot' or 'norm_pot_dng' is specified. threshold_norm_pot must be set when either 'norm_pot' or 'norm_pot_dng' is specified. Default is -1d0 a.u. (1 a.u.= 33.72x10⁴A⁻⁶eV²)
- **omp_loop (character, 3d)** Loop for OpenMP parallelization in the ground state SCF if periodic boundary system is used.
 - k: parallelization for k-point loop (Default).
 - b: parallelization mainly for band orbital loop (sometimes space grid loop too). This works efficiently if the number of k-point treated in each node is small (e.x. the case of single k-point for each node)
- **(Trial) skip_gsortho (character, 3d)** Flag to skip Gram-Schmidt orthogonalization in CG loop if periodic boundary system is used. If this is skipped the more iteration number is necessary to get convergence but each iteration step gets faster. If omp_loop=b, this flag is always applied. Default is n
- **(Trial) ngeometry_opt (integer, 0d)** Number of iteration of geometry optimization.

5.15 &emfield

- **trans_longi (character, 3d)** Geometry of solid-state calculations. Transverse 'tr' and longitudinal 'lo' can be chosen. Default is 'tr'.
- **ae_shape1/ae_shape2 (character, 0d/3d)** Shape of the first/second pulse.
 - 'impulse': Impulsive fields.
 - 'Acos2': Envelope of cos² for a vector potential.
 - 'Ecos2': Envelope of cos² for a scalar potential.
 If &system/iperiodic is 3, following can be also chosen,
 - 'Acos3', 'Acos4', 'Acos6', and 'Acos8': Envelopes of cos³, cos⁴, cos⁶, and cos⁸ for vector potentials.
 - (Trial) 'Esin2sin', 'Asin2cos', 'Asin2cw', 'input', and 'none' can be also chosen.
- **e_impulse (real(8), 0d/3d)** Momentum of impulsive perturbation. This variable has the dimension of momentum, energy*time/length. Default value is 1d-2 a.u.
- **amplitude1/amplitude2 (real(8), 0d/3d)** Maximum amplitude of electric fields for the first/second pulse. This variable has the dimension of electric field, energy/(length*charge). This variable cannot be set with &emfield/rlaser_int_wcm2_1 (rlaser_int_wcm2_2) simultaneously.
- **rlaser_int_wcm2_1/rlaser_int_wcm2_2 (real(8), 0d/3d)** Peak laser intensity (W/cm²) of the first/second pulse. This variable cannot be set with &emfield/amplitude1 (amplitude2) simultaneously.
- **pulse_tw1/pulse_tw2 (real(8), 0d/3d)** Duration of the first/second pulse. Unit of time can be chosen by &units/unit_time.
- **omega1/omega2 (real(8), 0d/3d)** Mean photon energy (average frequency multiplied by the Planck constant) of the first/second pulse. Unit of energy can be chosen by &units/unit_energy.
- **epdir_re1(3)/epdir_re2(3) (real(8), 0d/3d)** Real part of polarization vector for the first/second pulse.
- **epdir_im1(3)/epdir_im2(3) (real(8), 0d/3d)** Imaginary part of polarization vector for the first/second pulse.
- **phi_cep1/phi_cep2 (real(8), 0d/3d)** Carrier envelope phase of the first/second pulse. Default is 0d0/0d0.

- **t1_delay (real(8), 3d)** Time-delay of the first pulse. Unit of time can be chosen by `&units/unit_time`. (this is not available for multiscale option). Default is `0d0`.
- **t1_t2 (real(8), 0d/3d)** Time-delay between the first and the second pulses. Unit of time can be chosen by `&units/unit_time`.
- **quadrupole (character, 0d)** Quadrupole potential can be employed if `quadrupole` is set to `'y'`. Default is `'n'`.
- **quadrupole_pot (character, 0d)** Form of a quadrupole potential.
- **(Trial) alocal_laser (character, 0d)** The pulse is applied to a specific domain. Default is `'n'`.
- **(Trial) rlaserboun_sta(3)/rlaserboun_end(3) (real(8), 0d)** The edge of the domain where the pulse is applied. These parameters are effective only when `alocal_laser` is `'y'`. Default is `-1d7/1d7` in atomic unit. Unit of length can be chosen by `&units/unit_length`.
- **nump (integer, 0d)** Number of radiation sources for optical near fields. Maximum number is 2.
- **vecp(3,2) (real(8), 0d)** Dipole vectors of the radiation sources for the optical near fields. Unit of length can be chosen by `&units/unit_length`.
- **coop(3,2) (real(8), 0d)** Central coordinates of the dipole vectors of the radiation sources. Unit of length can be chosen by `&units/unit_length`.
- **radp_diele (real(8), 0d)** Radii of dielectric spheres for the radiation sources. Unit of length can be chosen by `&units/unit_length`.

5.16 &multiscale

- **(Trial) ftdddim (character, 3d)** Dimension of FDTD calculation for multi-scale Maxwell-Kohn-Sham method. Default value is `'1D'`.
- **(Trial) twod_shape (character, 3d)** Boundary condition of the second dimension for FDTD calculation with multi-scale Maxwell-Kohn-Sham method. Default value is `'periodic'`.
- **nx_m (integer, 3d)** Number of macroscopic grid points inside materials for x-direction.
- **(Trial) ny_m/nz_m (integer, 3d)** Number of macroscopic grid points inside materials for (y/z)-direction.
- **hx_m (real(8), 3d)** Spacing of macroscopic grid points inside materials for (x)-direction. Unit of length can be chosen by `&units/unit_length`.
- **(Trial) hy_m/hz_m (real(8), 3d)** Spacing of macroscopic grid points inside materials for (y/z)-direction. Unit of length can be chosen by `&units/unit_length`.
- **nxvac_l_m/nxvac_r_m (integer, 3d)** Number of macroscopic grid points for vacuum region. `nxvac_l_m` gives the number for negative x-direction in front of material, while `nxvac_r_m` gives the number for positive x-direction behind the material.
- **(Trial) nx_origin_m/ny_origin_m/nz_origin_m (integer, 3d)** Origin coordinate of the grid points. Default value is `'1'`.
- **(Trial) set_ini_coor_vel (character, 3d)** Set initial atomic coordinates and velocities for each macro-grid point. This must be given with specific directories and files: Prepare `directory/multiscale/MXXXXXX/ini_coor_vel.dat`, where `'XXXXXX'` is the index number of the macro-grid point of the material region usually starting from `'000001'` up to the number of macro-grid point. The format of the file `'ini_coor_vel.dat'` is just `Rx, Ry, Rz, Vx, Vy, Vz` (with space separation) for each atom (i.e. for each line), where the unit of the coordinates, `Rx, Ry, Rz`, is angstrom or a.u. specified by `unit_system` but that of velocities is always a.u.. This option should be used together with

`read_gs_wfn_k_ms` which is the option to read the ground state wave function for each macro-grid point. Default value is 'n'.

- **(Trial) nmacro_write_group (integer, 3d)** If the number of macroscopic grids are very large, computers can be unstable by writing all information of all macroscopic grid points at the same time. To avoid that, the writings are divided by specifying this option. Writings will be done by each `nmacro_write_group` macroscopic grid points. (this number must be aliquot part of the total number of macroscopic grid points) Default value is '-1'.
- **(Trial) file_macropoint (character, 3d)** If file name is specified in the option, the coordinates of the macropoints are set from the file. Default value is ''.

5.17 &maxwell

- **al_em(3) (real(8), 0d/3d)** Size of simulation box in electromagnetic analysis. Unit of the length can be chosen by `&units/unit_system`.
- **dl_em(3) (real(8), 0d/3d)** Spacing of real-space grids in electromagnetic analysis. Unit of length can be chosen by `&units/unit_system`.
- **dt_em(3) (real(8), 0d/3d)** Time step in electromagnetic analysis. Unit of time can be chosen by `&units/unit_system`.
- **nt_em (integer, 0d/3d)** Number of total time steps for real-time propagation in electromagnetic analysis.
- **iboundary(3,2) (integer, 0d/3d)** Boundary condition in electromagnetic analysis. The first index(1-3 rows) corresponds to x, y, and z axes. The second index(1-2 columns) corresponds to bottom and top of the axes. If `&system/iperiodic=0`, 0 is absorbing boundary(PML) and 1 is perfect electric conductor(PEC). If `&system/iperiodic=3`, 0 is periodic boundary and 1 is absorbing boundary(PML). Default is 0.
- **shape_file (character, 0d/3d)** Name of shape file in electromagnetic analysis. The shape files can be generated by using SALMON utilities (<https://salmon-tddft.jp/utilities.html>).
- **imedia_num (integer, 0d/3d)** Number of media in electromagnetic analysis. Default is 0.
- **type_media(:) (character, 0d/3d)** Type of media in electromagnetic analysis. 'vacuum', 'constant media', 'pec', and 'drude' can be chosen. Default is 'vacuum'. If 'drude' is chosen, linear response calculation can be done by `&emfield/ae_shape1` or `ae_shape2='impulse'`.
- **epsilon(:) (real(8), 0d/3d)** Relative permittivity of the media in electromagnetic analysis. Default is 1d0.
- **rmu(:) (real(8), 0d/3d)** Relative permeability of the media in electromagnetic analysis. Default is 1d0.
- **sigma(:) (real(8), 0d/3d)** Conductivity of the media in electromagnetic analysis. Default is 0d0.
- **omega_p_d(:) (real(8), 0d/3d)** Plasma frequency of the media for the case of `type_media='drude'` in electromagnetic analysis. Default is 0d0.
- **gamma_d(:) (real(8), 0d/3d)** Collision frequency of the media for the case of `type_media='drude'` in electromagnetic analysis. Default is 0d0.
- **smooth_d (character, 0d/3d)** Enable('y')/disable('n'). Smoothing of the media for the case of `type_media='drude'` in electromagnetic analysis. Default is 'n'.
- **weight_d (real(8), 0d/3d)** Weight of smoothing for the case of `type_media='drude'` in electromagnetic analysis. Default is '0.5d0'.
- **wave_input (character, 0d/3d)** If 'source', the incident pulse in electromagnetic analysis is generated by the incident current source. Default is 'none'.
- **ek_dir1(3)/ek_dir2(3) (real(8), 0d/3d)** Propagation direction of the first/second pulse.

- **source_loc1(3)/source_loc2(3) (real(8), 0d/3d)** Location of the incident current source of the first/second pulse. Note that the coordinate system ranges from $-a_{em}/2$ to $a_{em}/2$ for `&system/iperiodic=0` while ranges from 0 to a_{em} for `&system/iperiodic=3`.
- **iobs_num_em (integer, 0d/3d)** Number of observation point in electromagnetic analysis. Default is 0. From the obtained results, figure and animation files can be generated by using SALMON utilities (<https://salmon-tddft.jp/utilities.html>).
- **iobs_samp_em (integer, 0d/3d)** Sampling time-step of the observation in electromagnetic analysis. Default is 1.
- **obs_loc_em(:,3) (integer, 0d/3d)** Location of the observation point in electromagnetic analysis. Note that the coordinate system ranges from $-a_{em}/2$ to $a_{em}/2$ for `&system/iperiodic=0` while ranges from 0 to a_{em} for `&system/iperiodic=3`.
- **(Trial) wf_em (character, 0d/3d)** Enable('y')/disable('n'). Applying a window function for linear response calculation when `&calculation/theory=Maxwell`. Default is 'y'.

5.18 &analysis

- **projection_option (character, 3d)** Methods of projection.
 - 'no': no projection.
 - 'gs': projection to eigenstates of ground-state Hamiltonian.
 - 'rt': projection to eigenstates of instantaneous Hamiltonian.
- **(Trial) projection_decomp (character, 3d)** If 'atom' combined with `projection_option='gs'`, the number of excited electron is decomposed into each atom (this is printed in `SYSname_nex_atom.data`). Default is 'n'.
- **out_projection_step (integer, 3d)** Interval time step of projection analysis if `projection_option` is not 'no'. Default is 100.
- **nenergy (integer, 0d/3d)** Number of energy grids for frequency-domain analysis. This parameter is required when 'impulse' is chosen in `&emfield/ae_shape1|2`.
- **de (real(8), 0d/3d)** Energy spacing for analysis. Unit of energy can be chosen by `&units/unit_energy`. This parameter is required when 'impulse' is chosen in `&emfield/ae_shape1|2`.
- **out_psi (character, 0d/3d)** If 'y', wavefunctions are output. For periodic system (`iperiodic=3`), it works only for ground state calculation. The converged wave functions of all orbitals with all k-points are printed in `gs_wfn_cube` or `gs_wfn_vtk` directory. The format is specified by `format3d`. Default is 'n'.
- **out_dos (character, 0d/3d)** If 'y', density of state is output. Default is 'n'.
- **out_dos_start (real(8), 0d/3d)** Lower bound (energy) of the density of state spectra. If this value is lower than a specific value near the lowest energy level, this value is overwritten by that value. Default value is $-1.d10$ eV.
- **out_dos_end (real(8), 0d/3d)** Upper bound (energy) of the density of state spectra. If this value is higher than a specific value near the highest energy level, this value is overwritten by that value. Default value is $1.d10$ eV.
- **iout_dos_nenergy (integer, 0d/3d)** Number of energy points sampled in the density of state spectra. Default is 601.
- **out_dos_smearing (real(8), 0d/3d)** Smearing width used in the density of state spectra.. Default is 0.1 eV.

- **out_dos_method (character, 0d/3d)** Choise of smearing method for the density of state spectra.. gaussian and lorentzian function are available. Default is gaussian.
- **out_dos_fshift (character, 0d/3d)** If 'y', the electron energy is shifted to fix the Fermi energy as zero point. For &system/iperiodic is 0, “ out_dos_fshift“ is not used if &system/nstate is equal to &system/nelec/2. Default is 'n'.
- **out_pdos (character, 0d)** If 'y', projected density of state is output. Default is 'n'.
- **out_dns (character, 0d/3d)** If 'y', the spatial electron density distribution at the ground state is output. Default is 'n'.
- **out_dns_rt/out_dns_rt_step Character/Integer; 0d/3d)** If 'y', the spatiotemporal electron density distribution during real-time time-propagation is output every outdns_rt_step time steps. Default is 'n'.
- **(Trial) out_dns_trans/out_dns_trans_energy Character/Real (8); 3d)** If 'y', transition in different density from the ground state at specified field frequency omega(given by out_dns_trans_energy) is calculated by $d\rho(r,\omega)=FT(\rho(r,t)-\rho_{gs}(r))/T$. Default is 'n'/1.55eV.
- **out_elf (character, 0d)** If 'y', electron localization function is output. Default is 'n'.
- **out_elf_rt/out_elf_rt_step Character/Integer; 0d)** If 'y', electron localization function during real-time time-propagation is output every out_elf_rt_step time steps. Default is 'n'.
- **out_estatic_rt/out_estatic_rt_step Character/Integer; 0d)** If 'y', static electric field during real-time time-propagation is output every out_estatic_rt_step time steps. Default is 'n'.
- **(Trial) out_rvf_rt/out_rvf_rt_step Character/Integer; 3d)** If 'y', coordinates[A], velocities[au], forces[au] on atoms during real-time time-propagation are printed in SYSname_trj.xyz every out_rvf_rt_step time steps. If use_ehrenfest_md='y', the printing option is automatically turned on. Defaults are 'n'/10.
- **(Trial) out_tm (character, 3d)** If 'y', transition moments between occupied and virtual orbitals are printed into SYSname_tm.data after the ground state calculation. Defaults are 'n'.
- **format3d (character, 0d/3d)** File format for three-dimensional volumetric data. 'avs', 'cube', and 'vtk' can be chosen. Default is 'cube'.
- **numfiles_out_3d (integer, 0d)** Number of separated files for three dimensional data. Effective only when format3d is 'avs'. numfiles_out_3d must be less than or equal to number of processes. Default is 1.
- **(Trial) timer_process (character, 0d)** Basically, elapsed times are written in the output file. But if timer_process is 'y', files of elapsed times for every process are also generated. This variable is effective only for the real-time caulation. Default is 'n'.

5.19 &hartree

- **meo (integer, 0d)** A variable to determine how to put multipoles in the Hartree potential calculation. Default is 3.
 - 1: A single pole is put at the center.
 - 2: Multipoles are put at the center of atoms.
 - 3: Multipoles are put at the center of mass of electrons in prepared cuboids.
- **num_pole_xyz(3) (integer, 0d)** Number of multipoles when meo is 3. Default is 0, 0, 0. When default is set, number of multipoles is calculated automatically.

5.20 &ewald

- **newald (integer, 3d)** Parameter for Ewald method. Short-range part of Ewald sum is calculated within `newald` th nearest neighbor cells. Default is 4.
- **aewald (real(8), 3d)** Square of range separation parameter for Ewald method in atomic unit. Default is 0.5.

5.21 &opt (Trial)

- **(Trial) cg_alpha_up (real(8), 3d)** Parameter for up-rate of step length in line search in conjugated gradient method. Default is 1.3.
- **(Trial) cg_alpha_down (real(8), 3d)** Parameter for down-rate of step length in line search in conjugated gradient method. Default is 0.5.
- **(Trial) cg_alpha_ini (real(8), 3d)** Parameter for initial step length in line search in conjugated gradient method. (currently not available) Default is 0.8.
- **(Trial) convrg_scf_ene (real(8), 3d)** Convergence threshold of ground state SCF calculation in energy difference at each optimization step. If negative number no threshold (SCF loop is up to `Nscf`). The other SCF thresholds such as `threshold` in `&scf` are also applied (if you do not want to use it, set very small number). Default is -1.0.
- **(Trial) convrg_scf_force (real(8), 3d)** Convergence threshold of ground state SCF calculation in force (average over atoms) difference. If negative number no threshold (SCF loop is up to `Nscf`). The other SCF thresholds such as `threshold` in `&scf` are also applied (if you do not want to use it, set very small number). Default is -1.0.
- **(Trial) convrg_opt_fmax (real(8), 3d)** Convergence threshold of optimization in maximum force. Default is 1d-3.
- **(Trial) convrg_opt_ene (real(8), 3d)** Convergence threshold of optimization in energy difference. (currently not available) Default is 1d-6.

5.22 &md (Trial)

- **(Trial) ensemble (character, 3d)** Ensemble in MD option: "NVE" or "NVT". Default is "NVE".
- **(Trial) thermostat (character, 3d)** Thermostat in "NVT" option: (currently only `nose-hoover`). Default is "nose-hoover".
- **(Trial) step_velocity_scaling (integer, 3d)** Time step interval for velocity-scaling. Velocity-scaling is applied if this is set to positive. Default is -1.
- **(Trial) step_update_ps/step_update_ps2 Integer/Integer; 3d)** Time step interval for updating pseudopotential (Larger number makes calculation time reduce greatly, but gets inaccurate) in case of `use_ehrenfest_md=y`. `step_update_ps` is for full update and `step_update_ps2` is for update without changing grid points array. Default is 10/1.
- **(Trial) temperature0_ion (real(8), 3d)** Setting temperature [K] for NVT ensemble, velocity scaling and generating initial velocities. Default is 298.15.
- **(Trial) set_ini_velocity (character, 3d)** Initial velocities are set. Default is `n`.
 - `y`: Generate initial velocity with Maxwell-Boltzmann distribution.

- **r**: Read initial velocity from file specified by keyword of `file_ini_velocity`. This is, for example, used for restarting MD from the previous run. The last atomic coordinates and velocities are printed in `SYSname_trj.xyz`. (atomic coordinate also should be copied from the previous output and put in the next input file for restart)
- (Trial) **file_ini_velocity (character, 3d)** File name for initial velocities. This is read when `set_ini_velocity` is 'r'. The format is simply `vx(iatom) vy(iatom) vz(iatom)` in each line. The order of atoms must be the same as the given coordinates in the main input file. In case of using nose-hoover thermostat, a thermostat variable should be put at the last line (all atomic unit). Default is none.
- (Trial) **seed_ini_velocity (integer, 3d)** Random seed (integer number) to generate initial velocity if `set_ini_velocity` is set to y. Default is 123.
- (Trial) **thermostat_tau (real(8), 3d)** Parameter in Nose-Hoover method: controlling time constant for temperature. Default is `41.34[au]` or `1.0[fs]`.
- (Trial) **stop_system_momt (character, 3d)** Center of mass is stopped every time step. Default is n.

5.23 &misc

- **fourier (character)** Method of Fourier transformation. 'ft', 'FT', 'ffte' or 'FFTE' can be chosen. Default is 'ft'. This variable is effective only when `domain_parallel='y'` and `&system/iperiodic=3`.

Following variables are moved from the isolated part. Some of them may be added to common input, be combined to it, and be removed.

5.24 &group_fundamental (Trial)

- (Trial) **iditerybcg (integer, 0d)** Iterations for which ybcg is calculated if `&scf/amin_routine` is 'cg-diis'. Default is 20.
- (Trial) **iditer_nosubspace_diag (integer, 0d)** Iterations for which subspace diagonalization is not done if `&scf/subspace_diagonalization` is 'y'. Default is 10.
- (Trial) **ntmg (integer, 0d)** Number of multigrid calculation for gs. At the moment, there is a malfunction in this variable, and recovery is needed. Default is 1.
- (Trial) **idisnum(2) (integer, 0d)** Label numbers for two atoms which are measured the distance. Default is (1, 2/).
- (Trial) **iwrite_projection (integer, 0d)** A variable for projection. Default is 0.
- (Trial) **itwproj (integer, 0d)** The projection is calculated every `itwproj` time steps. Default is -1.
- (Trial) **iwrite_projnum (integer, 0d)** There is a malfunction in this variable.
- (Trial) **itcalc_ene (integer, 0d)** Total energy is calculated every `itcalc_ene` time steps. There may be a malfunction in this variable. Default is 1.

5.25 &group_parallel (Trial)

- (Trial) **isequential (integer, 0d)** A variable to determine the way of assignment of processes. Default is 2.

- (Trial) **imesh_s_all (integer, 0d)** A variable to determine how to use processes if total number of processes and number of processes for Hartree/Exc calculation differ. There may be a malfunction in this variable. Default is 1.
- (Trial) **iflag_comm_rho (integer, 0d)** This variable may be removed.

5.26 &group_hartree (Trial)

- (Trial) **hconv (real(8), 0d)** A convergence value for the Hartree-cg calculation. The convergence is checked by $\|tVh(i)-tVh(i-1)\|^2/(\text{number of grids})$. Default is $1d-15$ a.u. ($= 1.10d-13 \text{ A}^3\text{eV}^2$)
- (Trial) **lmax_meo (integer, 0d)** A maximum angular momentum for multipole expansion in the Hartree-cg calculation. Default is 4.

5.27 &group_file (Trial)

- (Trial) **ic (integer, 0d)** A variable to check whether reentrance is done or not in the ground state calculation. Default is 0.
- (Trial) **oc (integer, 0d)** A variable to check whether intermediate files are generated in the ground state calculation. Default is 1.
- (Trial) **ic_rt (integer, 0d)** A variable to check whether reentrance is done or not in the time propagation calculation. Default is 0.
- (Trial) **oc_rt (integer, 0d)** A variable to check whether intermediate files are generated in the time propagation calculation. Default is 0.

5.28 &group_others (Trial)

- (Trial) **iparaway_ob (integer, 0d)** A variable to determine the way of division for orbitals. 1 is block division, and 2 is cyclic division. Default is 2.
- (Trial) **iscf_order (integer, 0d)** A variable to determine the order of the calculation for the ground state one. Default is 1.
- (Trial) **iswitch_orbital_mesh (integer, 0d)** A variable to apply descending order for orbitals in the ground state calculation. Default is 0.
- (Trial) **iflag_pscube (integer, 0d)** A variable to generate cube files for wave functions. This variable will be removed.
- (Trial) **lambda1_diis/lambda2_diis (real(8), 0d)** Parameters for the diis calculation. Default is 0.5/0.3.
- (Trial) **file_ini (character, 0d)** A input file to align wavefunctions. Default is 'file_ini'.
- (Trial) **num_projection Integer; 0d)** Number of orbitals for projections. Default is 1.
- (Trial) **iwrite_projection_ob(200) Integer; 0d)** Orbital number to be written as projections. Default is (1/2/3/.../200).
- (Trial) **iwrite_projection_k(200) Integer; 0d)** This variable will be removed.
- (Trial) **filename_pot (character, 0d)** Name of file to be written local potentials. Default is 'pot'.
- (Trial) **iwrite_external (integer, 0d)** A variable to generate file to be written local potentials. Default is 0.

- **(Trial) iflag_dip2 (integer, 0d)** A variable to determine whether dipole moments are calculated in divided area. Default is 0.
- **(Trial) iflag_intelectron (integer, 0d)** A variable related to the quadrupole calculation. Default is 0.
- **(Trial) num_dip2 (integer, 0d)** Number of area where dipole moments are calculated. Default is 1.
- **(Trial) dip2boundary(100) (real(8), 0d)** Boundary position of area where dipole moments are calculated. Default is 0 a.u.
- **(Trial) dip2center(100) (real(8), 0d)** Origin in the dipole moment calculation. Default is 0 a.u.
- **(Trial) iflag_fourier_omega integer; 0d)** A variable to determine whether Fourier transformation of 3d data for difference of density is calculated. Default is 0.
- **(Trial) num_fourier_omega (integer, 0d)** Number of energies for which the Fourier transformation is calculated. Default is 1.
- **(Trial) fourier_omega(200) (real(8), 0d)** Energies for which the Fourier transformation is calculated. Default is 0 a.u.
- **(Trial) itotntime2 (integer, 0d)** Number of time steps in the reentrance for real-time calculation. There may be a malfunction in this variable. Default is 0.
- **(Trial) iwdenoption (integer, 0d)** A variable to determine whether 3d output is generated in real-time calculation. This variable will be removed.
- **(Trial) iwdenstep (integer, 0d)** 3d output is generated every iwdenstep time steps. This variable will be removed.
- **(Trial) iflag_estatic (integer, 0d)** A variable to determine whether 3d output for the static electric field is generated in real-time calculation. This variable will be removed.

BIBLIOGRAPHY

- [1] Masashi Noda, Kazuya Ishimura, Katsuyuki Nobusada, Kazuhiro Yabana, and Taisuke Boku. Massively-parallel electron dynamics calculations in real-time and real-space: toward applications to nanostructures of more than ten-nanometers in size. *Journal of Computational Physics*, 265:145–155, 2014.
- [2] George F Bertsch, J-I Iwata, Angel Rubio, and Kazuhiro Yabana. Real-space, real-time method for the dielectric function. *Physical Review B*, 62(12):7998, 2000.
- [3] K. Yabana and G. F. Bertsch. Time-dependent local-density approximation in real time. *Phys. Rev. B*, 54:4484–4487, Aug 1996. URL: <https://link.aps.org/doi/10.1103/PhysRevB.54.4484>, doi:10.1103/PhysRevB.54.4484.
- [4] Kazuhiro Yabana, T Sugiyama, Y Shinohara, T Otobe, and GF Bertsch. Time-dependent density functional theory for strong electromagnetic fields in crystalline solids. *Physical Review B*, 85(4):045134, 2012.
- [5] Shunsuke A. Sato and Kazuhiro Yabana. Maxwell + tddft multi-scale simulation for laser-matter interactions. *Journal of Advanced Simulation in Science and Engineering*, 1(1):98–110, 2014. doi:10.15748/jasse.1.98.
- [6] Yuta Hirokawa, Taisuke Boku, Shunsuke A Sato, and Kazuhiro Yabana. Electron dynamics simulation with time-dependent density functional theory on large scale symmetric mode xeon phi cluster. In *Parallel and Distributed Processing Symposium Workshops, 2016 IEEE International*, 1202–1211. IEEE, 2016.