SPARC

Simulation Package for Ab-initio Real-space Calculations

User guide

Material Physics & Mechanics Group
PI: Phanish Suryanarayana
Main Developers: Qimen Xu, Abhiraj Sharma
Collaborators: J.E. Pask (LLNL), A.J. Medford (GT), E. Chow (GT)
Georgia Institute of Technology

Introduction

SPARC is an open-source software package for the accurate, efficient, and scalable solution of the Kohn-Sham density functional theory (DFT) problem. The main features of SPARC currently include

- Isolated systems such as molecules and clusters as well as extended systems such as crystals, surfaces, and wires.
- Calculation of ground state energy, atomic forces, and stress tensor.
- Unconstrained collinear magnetization via spin polarized calculations.
- Structural relaxation and quantum molecular dynamics (QMD).
- Local and semilocal exchange correlation functionals.
- ONCV and TM pseudopotentials in psp8 (ABINIT) format.

Additional details regarding the formulation and implementation of SPARC can be found in the accompanying paper. Please direct any questions and report any bugs to Prof. Phanish Suryanarayana.

Installation - Option 1

Prerequisite: C compiler, MPI.
There are several options to compile SPARC, depending on the available external libraries.

- Option 1 (default): Compile with BLAS and LAPACK.
  - Step 1: Install/Load OpenBLAS/BLAS and LAPACK.
  - Step 2: Go to `src/` directory, there is an available `makefile`.
  - Step 3 (optional): Edit `makefile`. If the BLAS library path and LAPACK library path are not in the search path, edit the `BLASROOT` and `LAPACKROOT` variables, and add them to `LDFLAGS`. If you are using BLAS instead of OpenBLAS, replace all `-lopenblas` flags with `-lblas`.
  - Step 4 (optional): To turn on `DEBUG` mode, set `DEBUG_MODE` to 1 in the `makefile`.
  - Step 5: Within the `src/` directory, compile the code by
    ```
    $ make clean; make
    ```

Installation - Option 2

- Option 2 (recommended): Compile with MKL.
  - Step 1: Install/Load MKL.
  - Step 2: Go to `src/` directory, there is an available `makefile`.
  - Step 3: Edit `makefile`. Set `USE_MKL` to 1 to enable compilation with MKL. If the MKL library path is not in the search path, edit the `MKLROOT` variable to manually set the MKL path.
  - Step 4 (optional): For the projection/subspace rotation step, to use SPARC routines for matrix data distribution rather than ScaLAPACK (through MKL), set `USE_DP_SUBEIG` to 1. We found on some machines this option is faster.
  - Step 5 (optional): To turn on `DEBUG` mode, set `DEBUG_MODE` to 1 in the `makefile`.
  - Step 6: Within the `src/` directory, compile the code by
    `$ make clean; make`

Installation - Option 3

- Option 3: Compile with BLAS, LAPACK, and ScaLAPACK.
  - Step 1: Install/Load OpenBLAS/BLAS, LAPACK, and ScaLAPACK.
  - Step 2: Go to `src/` directory, there is an available `makefile`.
  - Step 3: Edit `makefile`. Set `USE_SCALAPACK` to 1 to enable compilation with ScaLAPACK. If the BLAS library path, LAPACK library path, and/or ScaLAPACK library path are not in the search path, edit the `BLASROOT`, `LAPACKROOT`, and/or `SCALAPACKROOT` variables accordingly, and add them to `LDFLAGS`. If you are using BLAS instead of OpenBLAS, replace all `-lopenblas` flags with `-lblas`.
  - Step 4 (optional): For the projection/subspace rotation step, to use SPARC routines for matrix data distribution rather than ScaLAPACK, set `USE_DP_SUBEIG` to 1. We found on some machines this option is faster.
  - Step 5 (optional): To turn on `DEBUG` mode, set `DEBUG_MODE` to 1 in the `makefile`.
  - Step 6: Within the `src/` directory, compile the code by
    `$ make clean; make`

Installation - lib

Once compilation is done, a binary named `sparc` will be created in the `lib/` directory.

Input files

The required input files to run a simulation with SPARC are

- ".inpt" file – User options and parameters.
- ".ion" file – Atomic information.

It is required that the ".inpt" and ".ion" files are located in the same directory and share the same name. A detailed description of the input options is provided in this document. Examples of input files can be found in the directory `SPARC/tests`.

In addition, SPARC requires pseudopotential files of psp8 format which can be generated by D. R. Hamann's open-source pseudopotential code ONCVPSP. A large number of accurate and efficient pseudopotentials are already provided within the package. For access to more pseudopotentials, the user is referred to the SG15 ONCV potentials. Using the ONCVPSP input files included in the SG15 ONCV potentials, one can easily convert the SG15 ONCV potentials from upf format to psp8 format. Paths to the pseudopotential files are specified in the ".ion" file.

Execution

SPARC can be executed in parallel using the `mpirun` command. Sample PBS script files are available in "SPARC/tests" folder. It is required that the ".inpt" and ".ion" files are located in the same directory and share the same name. For example, to run a simulation with 8 processes with input files as "filename.inpt" and "filename.ion" in the root directory (`SPARC/`), use the following command:

```
$ mpirun -np 8 ./lib/sparc -name filename
```

As an example, one can run a test located in `SPARC/tests`. First go to `SPARC/tests/MeshConvergence/Si8` directory:

```
$ cd tests/MeshConvergence/Si8
```

There are a few input files available. Run a DC silicon system with mesh = 0.4 Bohr by

Execution

```
$ mpirun -np 8 ../../../lib/sparc -name Si8-ONCV-0.4
```

The result is printed to output file "Si8-ONCV-0.4.out", located in the same directory as the input files. If the file "Si8-ONCV-0.4.out" is already present, the result will be printed to "Si8-ONCV-0.4.out_1" instead. The max number of ".out" files allowed with the same name is 100. Once this number is reached, the result will instead overwrite the "Si8-ONCV-0.4.out" file. One can compare the result with the reference out file named "Si8-ONCV-0.4.refout".

In the `tests/quick/` directory, we also provide a sample script file `quickrun.sh`, which launches four quick tests one by one. To run these quick tests, simply change directory to `tests/quick/` directory, and run:

```
$ chmod +x ./quickrun.sh
$ ./quickrun.sh
```

Output

Upon successful execution of the `sparc` code, depending on the calculations performed, some output files will be created in the same location as the input files.

Single point calculations

- ".out" file – General information about the test, including input parameters, SCF convergence progress, ground state properties and timing information.
- ".static" file – Atomic positions and atomic forces if the user chooses to print these information.

Structural relaxation calculations

- ".out" file – See above.

Output

- ".geopt" file – Atomic positions and atomic forces for atomic relaxation, cell lengths and stress tensor for volume relaxation, and atomic positions, atomic forces, cell dimensions, and stress tensor for full relaxation.
- ".restart" file – Information necessary to perform a restarted structural relaxation calculation. Only created if atomic relaxation is performed.

Quantum molecular dynamics (QMD) calculations

- ".out" file – See above.
- ".aimd" file – Atomic positions, atomic velocities, atomic forces, electronic temperature, ionic temperature and total energy for each QMD step.
- ".restart" file – Information necessary to perform a restarted QMD calculation.

Input file options

System

CELL | LATVEC_SCALE | LATVEC | FD_GRID | MESH_SPACING | ECUT | BC | FD_ORDER | EXCHANGE_CORRELATION | SPIN_TYP | KPOINT_GRID | KPOINT_SHIFT | ELEC_TEMP_TYPE | ELEC_TEMP | SMEARING | NSTATES ATOM_TYPE | PSEUDO_POT | N_TYPE_ATOM | COORD | COORD_FRAC | RELAX | SPIN

SCF

CHEB_DEGREE | CHEFSI_BOUND_FLAG | RHO_TRIGGER | MAXIT_SCF | MINIT_SCF | TOL_SCF | SCF_FORCE_ACC | SCF_ENERGY_ACC | TOL_LANCZOS | MIXING_VARIABLE | MIXING_HISTORY | MIXING_PARAMETER | MIXING_PARAMETER_SIMPLE | PULAY_FREQUENCY | PULAY_RESTART_FREQ | MIXING_PRECOND | TOL_PRECOND | PRECOND_KERKER_KTF | PRECOND_KERKER_THRESH | FIX_RAND

Input file options

Electrostatics

TOL_POISSON | MAXIT_POISSON | TOL_PSEUDOCHARGE | REFERENCE_CUTOFF

Stress calculation

CALC_STRESS | CALC_PRES

QMD

MD_FLAG | MD_METHOD | MD_NSTEP | MD_TIMESTEP | ION_TEMP | ION_TEMP_END | ION_VEL_DSTR | ION_VEL_DSTR_RAND | QMASS | RESTART_FLAG | TWTIME

Input file options

Structural relaxation

RELAX_FLAG | RELAX_METHOD | RELAX_NITER | TOL_RELAX | TOL_RELAX_CELL | RELAX_MAXDILAT | NLCG_SIGMA | L_HISTORY | L_FINIT_STP | L_MAXMOV | L_AUTOSCALE | L_LINEOPT | L_ICURV | FIRE_DT | FIRE_MASS | FIRE_MAXMOV | RESTART_FLAG

Print options

PRINT_ATOMS | PRINT_FORCES | PRINT_MDOUT | PRINT_RELAXOUT | PRINT_RESTART | PRINT_RESTART_FQ | PRINT_VELS | OUTPUT_FILE

Parallelization options

NP_SPIN_PARAL | NP_KPOINT_PARAL | NP_BAND_PARAL | NP_DOMAIN_PARAL | NP_DOMAIN_PHI_PARAL | EIG_SERIAL_MAXNS | EIG_PARAL_BLKSZ

System: .inpt file

CELL

Type

Double

Unit

Bohr

Default

None

Example

`CELL: 10.20 11.21 7.58`

Description

A set of three whitespace delimited values specifying the cell lengths in the lattice vector (LATVEC) directions, respectively.

Remark

Note that CELL ignores the lengths of the lattice vectors specified in the `.inpt` file and only treats them as unit vectors. LATVEC_SCALE and CELL cannot be specified simultaneously.

LATVEC_SCALE

Type	Unit
Double	Bohr

Default	Example
None	LATVEC_SCALE: 10.20 11.21 7.58

Description

A set of three whitespace delimited values specifying the scaling factors in the lattice vectors (LATVEC), respectively.

Remark

The difference between LATVEC_SCALE and CELL is that CELL treats the lattice vectors as unit vectors, whereas LATVEC_SCALE scales the lattice vectors directly as specified by the user. LATVEC_SCALE and CELL cannot be specified simultaneously.

LATVEC

Type

Double array

Unit

No unit

Default

1.0 0.0 0.0
0.0 1.0 0.0
0.0 0.0 1.0

Example

LATVEC:
0.5 0.5 0.0
0.0 0.5 0.5
0.5 0.0 0.5

Description

A set of three vectors in row major order specifying the lattice vectors of the simulation domain (CELL).

FD_GRID

Type

Integer

Unit

No unit

Default

None

Example

FD_GRID: 26 26 30

Description

A set of three whitespace delimited values specifying the number of finite-difference intervals in the lattice vector (LATVEC) directions, respectively.

FD_GRID

Remark

The convergence of results with respect to spatial discretization needs to be verified. ECUT, MESH_SPACING, FD_GRID cannot be specified simultaneously.

MESH_SPACING

Type

Double

Unit

Bohr

Default

None

Example

MESH_SPACING: 0.4

Description

Mesh spacing of the finite-difference grid.

Remark

The exact mesh-size will be determined by the size of the domain.
MESH_SPACING, FD_GRID, ECUT cannot be specified simultaneously.

ECUT

Type

Double

Unit

Ha

Default

None

Example

```
ECUT: 30
```

Description

Equivalent plane-wave energy cutoff, based on which MESH_SPACING will be automatically calculated.

Remark

This is not exact, but rather an estimate. ECUT, MESH_SPACING, FD_GRID cannot be specified simultaneously.

BC

Type

Character

Unit

No unit

Default

None

Example

```
BC: P D D
```

Description

A set of three whitespace delimited characters specifying the boundary conditions in the lattice vector directions, respectively. `P` represents periodic boundary conditions and `D` represents Dirichlet boundary conditions.

FD_ORDER

Type

Integer

Unit

No unit

Default

12

Example

```
FD_ORDER: 12
```

Description

Order of the finite-difference approximation.

Remark

Restricted to even integers since central finite-differences are employed. The default value of 12 has been found to be an efficient choice for most systems.

EXCHANGE_CORRELATION

Type

String

Unit

No unit

Default

No Default

Example

```
EXCHANGE_CORRELATION: LDA_PW
```

Description

Choice of exchange-correlation functional. Options are `LDA_PW` (Perdew-Wang LDA), `LDA_PZ` (Purdew-Zunger LDA), `GGA_PBE` (PBE GGA), `GGA_RPBE` (revised PBE GGA), and `GGA_PBEsol` (PBE GGA revised for solids).

Remark

For spin-polarized calculation (`SPIN_TYP` = 1), `LDA_PZ` is not available.

SPIN_TYP

Type

Integer

Unit

No unit

Default

0

Example

SPIN_TYP: 1

Description

SPIN_TYP: 0 performs spin unpolarized calculation.

SPIN_TYP: 1 performs unconstrained collinear spin-polarized calculation.

Remark

SPIN_TYP can only take values 0 and 1.

KPOINT_GRID

Type

Integer array

Unit

No unit

Default

1 1 1

Example

KPOINT_GRID: 2 3 4

Description

Number of k-points in each direction of the Monkhorst-Pack grid for Brillouin zone integration.

Remark

Time-reversal symmetry is assumed to hold.

KPOINT_SHIFT

Type

Double array

Unit

No unit

Default

0.0 for odd k-point mesh
0.5 for even k-point mesh

Example

KPOINT_SHIFT: 0.5 0.5 0.5

Description

Shift of k-points with respect to k-point grid containing Γ-point.

Remark

The shift is in reduced coordinates.

ELEC_TEMP_TYPE

Type

String

Unit

No unit

Default

`gaussian`

Example

`ELEC_TEMP_TYPE: fd`

Description

Function used for the smearing (electronic temperature). Options are: `fermi-dirac` (or `fd`), `gaussian`.

Remark

Use ELEC_TEMP or SMEARING to set smearing value.

ELEC_TEMP

Type

Double

Unit

Kelvin

Default

2320.904 for `gaussian`
1160.452 for `fermi-dirac`

Example

`ELEC_TEMP`: 315.775

Description

Electronic temperature.

Remark

This is equivalent to setting SMEARING (0.001 Ha = 315.775 Kelvin).

SMEARING

Type

Double

Unit

Ha

Default

0.007350 for `gaussian`
0.003675 for `fermi-dirac`

Example

`SMEARING`: 0.001

Description

Value of smearing.

Remark

Equivalent to setting `ELEC_TEMP` (0.001 Ha = 315.775 Kelvin).

NSTATES

Type

Integer

Unit

No unit

Default

$N_e/2 \times 1.2 + 5$

Example

NSTATES: 24

Description

The number of Kohn-Sham states/orbitals.

Remark

This number should not be smaller than half of the total number of valence electrons (N_e) in the system. Note that the number of additional states required increases with increasing values of ELEC_TEMP/SMEARING.

System: .ion file

ATOM_TYPE

Type

String

Unit

No unit

Default

None

Example

```
ATOM_TYPE: Fe
```

Description

Atomic type symbol.

Remark

The atomic type symbol can be attached with a number, e.g., Fe1 and Fe2. This feature is useful if one needs to provide two different pseudopotential files (PSEUDO_POT) for the same element.

PSEUDO_POT

Type

String

Unit

No unit

Default

None

Example

```
PSEUDO_POT: ../psp/Fe.psp8
```

Description

Path to the pseudopotential file.

Remark

The default directory for the pseudopotential files is the same as the input files. For example, if a pseudopotential Fe.psp8 is put in the same directory as the input files, one can simply specify `PSEUDO_POT: Fe.psp8`.

N_TYPE_ATOM

Type

Integer

Unit

No unit

Default

None

Example

```
N_TYPE_ATOM: 2
```

Description

The number of atoms of a ATOM_TYPE specified immediately before this variable.

Remark

For a system with different types of atoms, one has to specify the number of atoms for every type.

COORD

Type

Double

Unit

Bohr

Default

None

Example

```
COORD:
0.0 0.0 0.0
2.5 2.5 2.5
```

Description

The Cartesian coordinates of atoms of a ATOM_TYPE specified before this variable. If the coordinates are outside the fundamental domain (see CELL and LATVEC) in the periodic directions (see BC), it will be automatically mapped back to the domain.

COORD

Remark

For a system with different types of atoms, one has to specify the coordinates for every ATOM_TYPE. One can also specify the coordinates of the atoms using COORD_FRAC.

COORD_FRAC

Type

Double

Unit

None

Default

None

Example

```
COORD_FRAC:
0.5 0.5 0.0
0.0 0.5 0.5
```

Description

The fractional coordinates of atoms of a ATOM_TYPE specified before this variable. COORD_FRAC(i, j) $\times$ CELL(j), $(j = 1, 2, 3)$ gives the coordinate of the i^{th} atom along the j^{th} LATVEC direction. If the coordinates are outside the fundamental domain (see CELL and LATVEC) in the periodic directions (see BC), it will be automatically mapped back to the domain.

COORD_FRAC

Remark

For a system with different types of atoms, one has to specify the coordinates for every ATOM_TYPE. One can also specify the coordinates of the atoms using COORD.

RELAX

Type

Integer

Unit

No unit

Default

1 1 1

Example

```
RELAX:
1 0 1
0 1 0
```

Description

Atomic coordinate with the corresponding RELAX value 0 is held fixed during relaxation/QMD.

SPIN

Type

Double

Unit

No unit

Default

0.0

Example

```
SPIN:
1.0
-1.0
```

Description

Specifies the net initial spin on each atom for a spin-polarized calculation.

SCF

CHEB_DEGREE

Type

Integer

Unit

No unit

Default

Automatically set.

Example

CHEB_DEGEE: 25

Description

Degree of polynomial used for Chebyshev filtering.

Remark

For larger mesh-sizes, smaller values of CHEB_DEGREE are generally more efficient, and vice-versa.

CHEFSI_BOUND_FLAG

Type

Integer

Unit

No unit

Default

0

Example

```
CHEFSI_BOUND_FLAG: 1
```

Description

Flag to recalculate the bounds for Chebyshev filtering. If set to 0, then only for the very first SCF will the upper bound be evaluated based on the maximum eigenvalue using Lanczos algorithm, and the upper bound will be the same for the rest steps. If set to 1, the upper bound will be reevaluated for every SCF.

RHO_TRIGGER

Type

Integer

Unit

No unit

Default

4

Example

```
RHO_TRIGGER: 5
```

Description

The number of times Chebyshev filtering is repeated before updating the electron density in the very first SCF iteration.

Remark

Values smaller than the default value of 4 can result in a significant increase in the number of SCF iterations. Larger values can sometimes reduce the number of SCF iterations.

MAXIT_SCF

Type

Integer

Unit

No unit

Default

100

Example

```
MAXIT_SCF: 50
```

Description

Maximum number of SCF iterations.

Remark

Larger values than the default of 100 may be required for highly inhomogeneous systems, particularly when small values of SMEARING/ELEC_TEMP are chosen.

MINIT_SCF

Type

Integer

Unit

No unit

Default

3

Example

```
MAXIT_SCF: 5
```

Description

Minimum number of SCF iterations.

TOL_SCF

Type

Double

Unit

No unit

Default

Automatically set for 10^{-3} Ha/Bohr accuracy in force

Example

TOL_SCF: 1e-5

Description

The tolerance on the normalized residual of the effective potential or the electron density for convergence of the SCF iteration.

Remark

Only one of TOL_SCF, SCF_ENERGY_ACC, or SCF_FORCE_ACC can be specified.

SCF_FORCE_ACC

Type

Double

Unit

Ha/Bohr

Default

None

Example

SCF_FORCE_ACC: 1e-4

Description

The tolerance on the atomic forces for convergence of the SCF iteration.

Remark

Only one of SCF_FORCE_ACC, TOL_SCF or SCF_ENERGY_ACC can be specified.

SCF_ENERGY_ACC

Type

Double

Unit

Ha/atom

Default

None

Example

SCF_ENERGY_ACC: 1e-5

Description

The tolerance on the free energy for the convergence of the SCF iteration.

Remark

Only one of SCF_ENERGY_ACC, SCF_FORCE_ACC, or TOL_SCF can be specified.

TOL_LANCZOS

Type

Double

Unit

No unit

Default

1e-2

Example

TOL_LANCZOS: 1e-3

Description

The tolerance within the Lanczos algorithm for calculating the extremal eigenvalues of the Hamiltonian, required as part of the CheFSI method.

Remark

Typically, the Lanczos tolerance does not need to be very strict.

MIXING_VARIABLE

Type

String

Unit

No unit

Default

`density`

Example

```
MIXING_VARIABLE:
potential
```

Description

This specifies whether potential or density mixing is performed in the SCF iteration. Available options are: `potential` and `density`.

MIXING_HISTORY

Type

Integer

Unit

No unit

Default

7

Example

```
MIXING_HISTORY: 40
```

Description

The mixing history used in Pulay mixing.

Remark

Too small values of MIXING_HISTORY can result in poor SCF convergence.

MIXING_PARAMETER

Type

Double

Unit

No unit

Default

0.3

Example

```
MIXING_PARAMETER: 0.1
```

Description

The value of the relaxation parameter used in Pulay/simple mixing.

Remark

Values larger than the default value of 0.3 can be used for insulating systems, whereas smaller values are generally required for metallic systems, particularly at small values of SMEARING or ELEC_TEMP.

MIXING_PARAMETER_SIMPLE

Type

Double

Unit

No unit

Default

Automatically set to the same as MIXING_PARAMETER

Example

```
MIXING_PARAMETER_SIMPLE:
0.1
```

Description

The value of the relaxation parameter used in the simple mixing step in the periodic Pulay scheme.

PULAY_FREQUENCY

Type

Integer

Unit

No unit

Default

1

Example

```
PULAY_FREQUENCY: 4
```

Description

The frequency of Pulay mixing in Periodic Pulay.

Remark

The default value of 1 corresponds to Pulay mixing.

PULAY_RESTART

Type

Integer

Unit

No unit

Default

0

Example

```
PULAY_RESTART: 1
```

Description

The flag for restarting the ‘Periodic Pulay’ mixing. If set to 0, the restarted Pulay method is turned off.

MIXING_PRECOND

Type

String

Unit

No unit

Default

`kerker`

Example

`MIXING_PRECOND: none`

Description

This specifies the preconditioner used in the SCF iteration. Available options are: `none`, `kerker`.

TOL_PRECOND

Type

Double

Unit

No unit

Default

$h^2 \times 0.001$

Example

TOL_PRECOND: 1e-4

Description

The tolerance on the relative residual for the linear systems arising during the real-space preconditioning of the SCF.

Remark

The linear systems do not need to be solved very accurately. h is the mesh spacing.

PRECOND_KERKER_KTF

Type

Double

Unit

Bohr^{-1}

Default

1.0

Example

```
PRECOND_KERKER_KTF: 0.8
```

Description

The Thomas-Fermi screening length appearing in the `kerker` preconditioner (MIXING_PRECOND).

PRECOND_KERKER_THRESH

Type

Double

Unit

No unit

Default

0.1

Example

PRECOND_KERKER_THRESH: 0.0

Description

The threshold for the kerker preconditioner (MIXING_PRECOND).

Remark

This threshold will be scaled by the MIXING_PARAMETER. If the threshold is set to 0, the original kerker preconditioner is recovered.

FIX_RAND

Type

Integer

Unit

No unit

Default

0

Example

```
FIX_RAND: 1
```

Description

Flag to fix the random seeds for setting initial guesses. Once set to 1, the random seeds will be fixed for different runs and for different numbers of processors. This option will make sure the answers will be exactly the same (up to machine precision) when SPARC is executed with different numbers of processors.

Electrostatics

TOL_POISSON

Type

Integer

Unit

No unit

Default

TOL_SCF×0.01

Example

TOL_POISSON: 1e-6

Description

The tolerance on the norm of the relative residual for the Poisson equation.

Remark

The tolerance for poisson solver should not be worse than TOL_SCF, otherwise it might seriously affect the convergence of the SCF iteration.

MAXIT_POISSON

Type

Integer

Unit

No unit

Default

3000

Example

`MAXIT_POISSON: 1000`

Description

The maximum number of iterations for solving the Poisson equation using an iterative linear solver.

TOL_PSEUDOCHARGE

Type

Double

Unit

No unit

Default

TOL_SCF×0.01

Example

TOL_PSEUDOCHARGE: 1e-6

Description

The error in the net enclosed charge for the pseudocharge density of each atom.

REFERENCE_CUTOFF

Type

Double

Unit

Bohr

Default

0.5

Example

```
REFERENCE_CUTOFF: 1.0
```

Description

The cutoff radius of the reference potential used for evaluating the electrostatic correction arising from overlapping pseudocharge densities.

Remark

This number should be smaller than half the smallest interatomic distance.

Stress calculation

CALC_STRESS

Type

Integer

Unit

No unit

Default

0

Example

```
CALC_STRESS: 1
```

Description

Flag for calculation of the Hellmann-Feynman stress tensor (in cartesian coordinates).

CALC_PRES

Type

Integer

Unit

No unit

Default

0

Example

```
CALC_PRES: 1
```

Description

Flag for calculation of the pressure.

Remark

Pressure is directly calculated, without calculation of the stress tensor.

QMD

MD_FLAG

Type

Integer

Unit

No unit

Default

0

Example

```
MD_FLAG: 1
```

Description

QMD simulations are performed if the flag is set to 1.

Remark

MD_FLAG and RELAX_FLAG both cannot be set to a value greater than 0.

MD_METHOD

Type

String

Unit

No unit

Default

NVT_NH

Example

`MD_METHOD`: NVE

Description

Type of QMD to be performed. Currently, NVE (microcanonical ensemble), NVT_NH (canonical ensemble with Nose-Hoover thermostat), and NVK_G (isokinetic ensemble with Gaussian thermostat) are supported.

MD_NSTEP

Type

Integer

Unit

No unit

Default

1e7

Example

`MD_NSTEP: 100`

Description

Specifies the number of QMD steps.

Remark

If MD_NSTEP = N, the QMD runs from 0 to $(N-1)\times$ `MD_TIMESTEP` fs.

MD_TIMESTEP

Type

Double

Unit

Femtosecond

Default

1

Example

MD_TIMESTEP: 0.1

Description

QMD time step.

Remark

Total QMD time is given by: MD_TIMESTEP × (MD_NSTEP - 1).

ION_TEMP

Type

Double

Unit

Kelvin

Default

No Default

Example

```
ION_TEMP: 315
```

Description

Starting ionic temperature in QMD, used to generate initial velocity distribution.

Remark

Must be specified if MD_FLAG is set to 1.

ION_TEMP_END

Type

Double

Unit

Kelvin

Default

ION_TEMP

Example

```
ION_TEMP_END: 100
```

Description

Specifies the final temperature of the thermostat. Thermostat temperature is varied linearly from ION_TEMP to ION_TEMP_END with respect to time.

Remark

Available for NVT_NH quantum molecular dynamics only.

ION_VEL_DSTR

Type

Integer

Unit

No unit

Default

2

Example

```
ION_VEL_DSTR: 1
```

Description

Specifies the type of distribution for the initial velocity of atoms based on their initial temperature. Choose 1 for uniform velocity distribution and 2 for Maxwell-Boltzmann distribution.

Remark

Currently, the code supports only two options for the variable.

ION_VEL_DSTR_RAND

Type

Integer

Unit

No unit

Default

0

Example

```
ION_VEL_DSTR_RAND: 1
```

Description

Flag to reseed the initial velocities of atoms in a QMD simulation. Set this flag to 1 to shuffle (change the random seed for) the initial velocities for different runs. Set this flag to 0 to maintain the same initial velocities.

Remark

This option is convenient for parallel statistics calculations.

QMASS

Type

Double

Unit

atomic unit

Default

1653.654933459720

Example

QMASS: 100000

Description

Gives the inertia factor for Nose Hoover thermostat.

Remark

Applicable to NVT_NH MD_METHOD only.

RESTART_FLAG

Type

Integer

Unit

No unit

Default

0

Example

```
RESTART_FLAG: 0
```

Description

Flag for restarting quantum molecular dynamics and structural relaxation. Stores last three histories for quantum molecular dynamics simulations in .restart, .restart-0 and .restart-1 files, respectively.

RESTART_FLAG

Remark

Restarts from the previous configuration which is stored in a .restart file. Currently, code provides restart feature for atomic relaxation and QMD only.

TWTIME

Type

Double

Unit

min

Default

1e9

Example

```
TWTIME: 1000
```

Description

Gives the upper bound on the wall time for quantum molecular dynamics.

Structural relaxation

RELAX_FLAG

Type

Integer

Unit

No unit

Default

0

Example

```
RELAX_FLAG: 1
```

Description

Flag for performing structural relaxation. 0 means no structural relaxation. 1 represents relaxation of atom positions. 2 represents optimization of volume with the fractional coordinates of the atoms fixed. 3 represents full optimization of the cell i.e., both atoms and cell volume are relaxed

Remark

This flag should not be specified if MD_FLAG is set to 1.

RELAX_METHOD

Type

String

Unit

No unit

Default

LBFGS

Example

`RELAX_METHOD`: NLCG

Description

Specifies the algorithm for structural relaxation. The choices are ‘LBFGS’ (limited-memory BFGS), ‘NLCG’ (Non-linear conjugate gradient), and ‘FIRE’ (Fast inertial relaxation engine).

Remark

LBFGS is typically the best choice.

RELAX_NITER

Type

Integer

Unit

No unit

Default

300

Example

RELAX_NITER: 25

Description

Specifies the maximum number of iterations for the structural relaxation (RELAX_FLAG).

Remark

If RESTART_FLAG is set to 1, then relaxation will restart from the last atomic configuration and run for maximum of RELAX_NITER iterations.

TOL_RELAX

Type

Double

Unit

Ha/Bohr

Default

5e-4

Example

```
TOL_RELAX: 1e-3
```

Description

Specifies the tolerance for termination of the structural relaxation. The tolerance is defined on the maximum force component (in absolute sense) over all atoms.

TOL_RELAX_CELL

Type

Double

Unit

GPa

Default

1e-2

Example

TOL_RELAX: 1e-3

Description

Specifies the tolerance for termination of the cell relaxation. The tolerance is defined on the maximum principle stress component.

RELAX_MAXDILAT

Type

Double

Unit

No unit

Default

1.06

Example

```
RELAX_MAXDILAT: 1.4
```

Description

The maximum scaling of the volume allowed with respect to the initial volume defined by CELL and LATVEC. This will determine the upper-bound and lower-bound in the bisection method (Brent's method) for the volume optimization.

NLCG_SIGMA

Type

Double

Unit

No unit

Default

0.5

Example

```
NLCG_SIGMA: 1
```

Description

Parameter in the secant method used to control the step length in NLCG (RELAX_METHOD).

Remark

Default value works well in most cases.

L_HISTORY

Type

Integer

Unit

No unit

Default

20

Example

```
L_HISTORY: 15
```

Description

Size of history in LBFGS (RELAX_METHOD).

Remark

Default value works well in most cases.

L_FINIT_STP

Type

Double

Unit

Bohr

Default

5e-3

Example

L_FINIT_STP: 0.01

Description

Step length for line optimizer in LBFGS (RELAX_METHOD).

Remark

Default value works well in most cases.

L_MAXMOV

Type

Double

Unit

Bohr

Default

0.2

Example

```
L_MAXMOV: 1.0
```

Description

The maximum allowed step size in LBFGS (RELAX_METHOD).

Remark

Default value works well in most cases.

L_AUTOSCALE

Type

Integer

Unit

No unit

Default

1

Example

L_AUTOSCALE: 0

Description

Flag for automatically determining the inverse curvature that is used to determine the direction for next iteration in LBFGS (RELAX_METHOD).

Remark

Default works well in most cases.

L_LINEOPT

Type

Integer

Unit

No unit

Default

1

Example

```
L_LINEOPT: 0
```

Description

Flag for atomic force based line minimization in LBFGS (RELAX_METHOD).

Remark

Required only if L_AUTOSCALE is 0.

L_ICURV

Type

Double

Unit

No unit

Default

1.0

Example

```
L_ICURV: 0.1
```

Description

Initial inverse curvature, used to construct the inverse Hessian matrix in LBFGS (RELAX_METHOD).

Remark

Needed only if L_AUTOSCALE is 0. Default value works well in most cases.

FIRE_DT

Type

Double

Unit

Femto second

Default

1

Example

FIRE_DT: 0.1

Description

Time step used in FIRE (RELAX_METHOD).

Remark

Default value works well in most cases.

FIRE_MASS

Type

Double

Unit

Atomic mass unit

Default

1.0

Example

FIRE_MASS: 2.5

Description

Pseudomass used in FIRE (RELAX_METHOD).

Remark

Default value works well in most cases.

FIRE_MAXMOV

Type

Double

Unit

Bohr

Default

0.2

Example

```
FIRE_MAXMOV: 1.0
```

Description

Maximum movement for any atom in FIRE (RELAX_METHOD).

Remark

Default value works well in most cases.

Print options

PRINT_ATOMS

Type

0 or 1

Unit

No unit

Default

0

Example

```
PRINT_ATOMS: 1
```

Description

Flag for writing the atomic positions. For ground-state calculations, atom positions are printed to a '.static' output file. For structural relaxation calculations, atom positions are printed to a '.geopt' file. For QMD calculations, atom positions are printed to a '.aimd' file.

PRINT_FORCES

Type

0 or 1

Unit

No unit

Default

0

Example

```
PRINT_FORCES: 1
```

Description

Flag for writing the atomic forces. For ground-state calculations, forces are printed to a '.static' output file. For structural relaxation calculations, forces are printed to a '.geopt' file. For QMD calculations, forces are printed to a '.aimd' file.

PRINT_MDOUT

Type

0 or 1

Unit

No unit

Default

1

Example

```
PRINT_MDOUT: 0
```

Description

Flag for printing the the QMD output into the .aimd file.

PRINT_RELAXOUT

Type

0 or 1

Unit

No unit

Default

1

Example

PRINT_RELAXOUT: 0

Description

Flag for printing the structural relaxation data in a .geopt file.

Remark

Required only if RELAX_FLAG is greater than 0.

PRINT_RESTART

Type

0 or 1

Unit

No unit

Default

1

Example

```
PRINT_RESTART: 0
```

Description

Flag for writing the .restart file, used to restart QMD and structural relaxation simulations.

Remark

Relevant only if either MD_FLAG is 1 or RELAX_FLAG is 1.

PRINT_RESTART_FQ

Type

Integer

Unit

No unit

Default

1

Example

PRINT_RESTART_FQ: 10

Description

Frequency at which .restart file is written in QMD and structural optimization simulations.

Remark

Relevant only if either MD_FLAG is 1 or RELAX_FLAG is 1.

PRINT_VELS

Type

0 or 1

Unit

No unit

Default

1

Example

PRINT_VELS: 0

Description

Flag for printing the ion velocities in an QMD simulation into the .aimd file.

Remark

Relevant only if MD_FLAG is set to 1.

OUTPUT_FILE

Type

String

Unit

No unit

Default

Same as the input file name

Example

OUTPUT_FILE: myfname

Description

The name of the output files. The output files are attached with a suffix (‘.out’,‘.static’,‘.geopt’ and ‘.aimd’).

OUTPUT_FILE

Remark

If an output file with the same name already exist, the results will be written to a file with a number attached, e.g., 'myfname.out_1'. The maximum number of output files with the same name allowed is 100. After that the output files will be overwritten in succession.

Parallelization options

NP_SPIN_PARAL

Type

Integer

Unit

No unit

Default

Automatically optimized

Example

NP_SPIN_PARAL: 2

Description

Number of spin groups.

Remark

This option is for development purpose. It's better to let SPARC choose the parallization parameters in practice.

NP_KPOINT_PARAL

Type

Integer

Unit

No unit

Default

Automatically optimized

Example

```
NP_KPOINT_PARAL: 5
```

Description

Number of k-point groups.

Remark

This option is for development purpose. It's better to let SPARC choose the parallization parameters in practice.

NP_BAND_PARAL

Type

Integer

Unit

No unit

Default

Automatically optimized

Example

NP_BAND_PARAL: 5

Description

Number of band groups.

Remark

This option is for development purpose. It's better to let SPARC choose the parallization parameters in practice.

NP_DOMAIN_PARAL

Type

Integer

Unit

No unit

Default

Automatically optimized

Example

`NP_DOMAIN_PARAL: 4`

Description

Number of processors in the Cartesian topology embedded in each domain group.

Remark

This option is for development purpose. It's better to let SPARC choose the parallization parameters in practice.

NP_DOMAIN_PHI_PARAL

Type

Integer

Unit

No unit

Default

Automatically optimized

Example

`NP_DOMAIN_PHI_PARAL: 4`

Description

Number of processors in the Cartesian topology embedded in the global communicator.

Remark

This option is for development purpose. It's better to let SPARC choose the parallization parameters in practice.

EIG_SERIAL_MAXNS

Type

Integer

Unit

No unit

Default

2000

Example

EIG_SERIAL_MAXNS: 1000

Description

Maximum NSTATES value up to which a serial algorithm will be used to solve the subspace eigenproblem.

Remark

If one wants to use a parallel algorithm to solve the subspace eigenproblem for all cases, simply set EIG_SERIAL_MAXNS to 0. Alternatively, set EIG_SERIAL_MAXNS to a very large value to always use serial algorithm.

EIG_PARAL_BLKSZ

Type

Integer

Unit

No unit

Default

128

Example

EIG_PARAL_BLKSZ: 64

Description

Block size for the distribution of matrix in block-cyclic format in a parallel algorithm for solving the subspace eigenproblem.

Acknowledgments

The work was supported by the grant DE-SC0019410 funded by the U.S. Department of Energy, Office of Science. Initial development efforts were supported by the grants 1333500 and 1553212 funded by the U.S. National Science Foundation.