UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE CIENCIAS EXACTAS Y NATURALES

CONICET
INQUIMAE

LiÔ

LIO User Guide

V 1.0

March, 2018

Version Developers

Contents

Chapter 1

Introduction

What is LIO?

Welcome to the LIO project! LIO is a library that can perform electronic structure calculations using density functional theory.

Instalation

If you are reading this manual, you probably already have a version of LIO ready to compile. If you don't, or if you want to make sure you have the most most up-to-date version of the code, all you need to do is either download it the git repository online or use git to clone a copy.

For the first option, go to ***https://github.com/MALBECC/lio*** and click on the green button that says *clone or download* and click on *Download ZIP*.

For the second one, you can directly run the following command:

```
git clone https://github.com/MALBECC/lio.git .
```

Pre-requisites

In addition to an UNIX-like OS, fortran and c++ compilers, LIO depends on LAPACK *LAPACK* and BLAS libraries for linear algebra calculations. In addition, *CUDA 6.5*

or higher is required for GPU calculations,which unleash LIO's true potential. As of the writing of this manual, LIO has not yet been tested with CUDA 9.0.
In addition, *libxc* is required for its usage, although said library is CPU-only. This is entirely optional as LIO can run without libxc, using only the PBE functional.

Compilation

By default, LIO compiles with GPU options enabled. It is highly recommended to specify the GPU architecture as a compilation option, since the compiler performs additional enhancements. After compilation, LIOHOME environment variable should be set to the current LIO installation directory:

```
export LIOHOME=/dir/to/lio/
```

For a CPU-only compilation, use:

```
make cuda=0
```

If INTEL compilers are present, they can be used by setting the *intel* option to 1, or to 2 if INTEL MKL usage is also desired.

```
make intel=2
```

The following is a list of available compilation and their meanings. They can be used in any combination possible (including several GPU architectures for greater compatibility). For example, the default LIO compilation could be written as:

```
make cuda=2 sm30=1 sm52=1 sm61=1
```

Variable	Description
cuda *0, 1, 2*	*default cuda=2* Sets the level of GPU dependencies. cuda=0 means a CPU-only compilation, =1 enables GPU, and =2 includes CUBLAS usage for linear algebra operations.
intel *0, 1, 2*	*default intel=0* Sets the usage of INTEL compilers. intel=0 means only GNU compilers, =1 means INTEL, and =2 uses INTEL MKL instead of BLAS/LAPACK routines.

Table 1.1: Compilation options

Variable	Description
precision *0, 1*	*default precision=0* Sets level of precision in XC calculations. By default LIO uses mixed precision; precision=1 sets everything in double precision. This is specially useful when attempting high-precision geometry optimizations.
analytics *0, 1, 2, 3*	*default analytics=0* Setting analytics = 1 specifies a profiling compilation, while analytics = 2 and 3 set higher levels of debugging information (for usage with gdb, for example).
smXX *0, 1*	*default sm30=1 sm52=1 sm61=1* Sets a specific GPU architecture for compilation. Available options are sm30, sm35 (Kepler, CUDA $\geq$ 5.0), sm50, sm52 (Maxwell and GeForce 980, CUDA $\geq$ 6.5), sm60 and sm61 (Pascal and GeForce 1080, CUDA $\geq$ 8.0).

Table 1.1: Compilation options

MD-Engine Interfacing

LIO can be linked with *AMBER*, our own *GROMACS* fork, our own *HYBRID* code for QM/MM calculations.

In all three cases, the LIOHOME environment variable should be set to the current LIO installation directory. In addition to this section, please refer to each of the software packages' installation manual for further clarifications.

In order to compile AMBER with LIO, AMBER should be compiled after LIO with the following options set:

```
export AMBERHOME=/dir/to/amber/
./configure -lio -noX11 -netcdfstatic gnu
make clean
make install
```

For a GROMACS-LIO compilation, GROMACS should be compiled after LIO with the following options:

```
cd gromacs_compilation_directory/
cmake gromacs_src_dir/ -DGMX_QMMM=1
    -DGMX_QMMM_PROGRAM="lio" -DLIO_LINK_FLAGS="-L/usr/lib
    -L/usr/lib64 -L$LIOHOME/g2g -L$LIOHOME/lioamber -lg2g
    -llio-g2g" -DGMX_GPU=0 -DGMX_THREAD_MPI=0
make
make install
```

Running LIO

In order to run LIO as a QM stand-alone program, you must first set the LIOHOME environment variable to that of the current LIO installation path (if it was not set before), and add /liosolo folder to the current excecutable paths.

```
export LIOHOME=/dir/to/lio/
export PATH=$PATH:$LIOHOME/liosolo
```

Then, LIO can be excecuted as follows:

```
liosolo -i input_file -c coords.xyz
```

Where input_file contains the input options and coords.xyz contains the coordinates in XYZ format. For more options, please refer to Chapter 2.
When running LIO with AMBER, you only need to specify '***qm_theory=extern***' in the &qmmm section of the code (please refer to the AmberTools manual for further &qmmm options), in addition to specifying both the QM system and its charge. When running with GROMACS, you only need to specify the QM system and its charge. In both cases, additional LIO options will be read from a 'lio.in' file in the working directory.

Tips and tricks - Optimizing your runs

There are several ways to optimize your production runs, and it is highly recommended to fine tune these settings if the same system or very similar systems are going to run for extended periods of time. Please see the corresponding sections in this manual for a more detailed explanation of these variables.

The first set to tune is ***rmax*** and ***rmaxs***, which are related to the integral cut-offs used in both Coulomb and fitting set integrals. The *higher* the value of ***rmaxs*** and the *lower* the value of ***rmax***, the calculations will run faster. Keep in mind, however, that ***rmax*** should *never* be lower that ***rmaxs***, and that tuning those values for a faster calculation will certainly result in a loss of precision. Therefore, the idea is to decrease ***rmax*** and increase ***rmaxs*** until the difference in energies and forces stops being negligible.

Next is the set of options available for the GPU library which performs the exchange-correlation calculations. The option ***max_function_exponent*** indicates the maximum exponent considered for a fuction in a point of the grid; it should be tweaked taking into account the aforementioned criterium: the faster the calculation is performed, the lesser precision is achieved. There are also three other options whose optimal values depend on the GPU architecture available: ***little_cube_size***, ***min_points_per_cube***, and ***sphere_radius***. These can be tweaked for maximum speed without worrying about the resulting precision.

Chapter 2

General Settings

In this chapter we will describe the basic settings for running the code. These will be necessary or useful for any of the features of the code that you might be interested in using.

System Description

Properties Calculations

GPU Options

Chapter 3

Model Hamiltonian

Density Functional Theory

Tight Binding Interface

External Electric Fields

Single Atom Potentials

Effective Core Potentials

Chapter 4

Ground State Calculations

Single-point and BO-MD

Single-point calculations consist in finding the ground state density that minimizes the energy for a given nuclei distribution. Once found, the program can also calculate the force field of that density for the MD-engine to move the nuclei. This method of

Relevant Keywords

Examples

Geometry optimizations

Geometry optimizations or energy minimization is the process of finding an atomic arrangement in space where the force on each atom is acceptably close to zero.

Implemenation

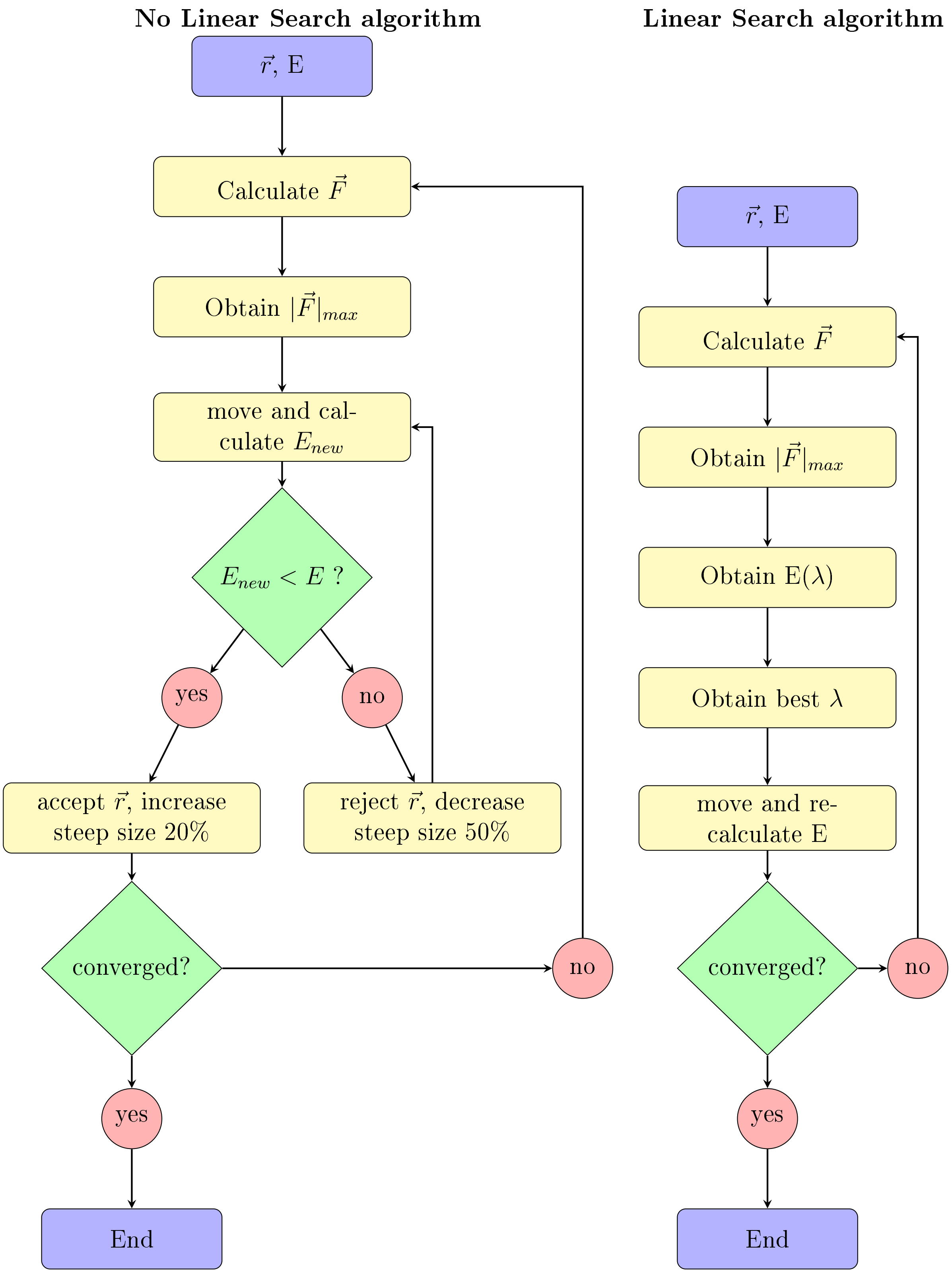
LIO has a simple steepest-descent algorithm. The idea is to move the system in the force direction, at a λ step value.

$$\vec{r^i}_{new} = \vec{r^i} + \lambda \vec{F}^i \tag{4.1}$$

Variable	**Description** ($Type = Default$)
timedep	$integer = 0$ Determines how the electronic structure will be calculated. A value of 0 indicates that an SCF will be performed with each call to LIO. Other values are reserved for electron dynamics.
told	$double\ precision = 1.0d-6$ Criteria for the maximal square deviation of the density matrix to consider that the convergence has been achieved.
etold	$double\ precision = 1.0d0$ Criteria for the maximal energy difference to consider that the con--vergence has been achieved.
DIIS	$logical = .true.$
nDIIS	$integer = 30$
gold	$double\ precision = 1.0d+1$ Determines the weight of the previous density matrix in the linear combination with the new one when using the damping convergence method (DIIS = .false. / hybrid_converg).
hybrid_converg	$logical = .false.$
good_cut	$double\ precision = 1.0d-3$
VCInp	$logical = .false.$ Reads a MO coefficient restart.
restart_freq	$integer = 1$ Writes a MO coefficient restart every restart_freq iterations.
frestart	$char * 20 =' restart.out'$ Output restart file.
frestartin	$char * 20 =' restart.in'$ Input restart file.

Without a linear search algorithm λ is obtained as $\frac{steep\_size}{|\vec{F}_{max}|}$. If the energy decreases with the movement, the step is accepted; but if the energy increases with the step, the steep is rejected and λ is reduced. Each accepted move increases step size a 20% and each rejected move decreases step size a 50%.

In a linear search algorithm the system scans the energy as function of λ and predicts the best value of λ to move the system in the gradient direction.



No Linear Search algorithm **Linear Search algorithm**

Best λ in lineal search algorithm is obtained by a quadratic function ajusted using minimum energy of the scan and previous and next points.

Using geometry optimizations

Adding steep=t in LIO input enables geometry optimization (steepest descent, lineal search by default). Convergence criteria are set by Force_cut and Energy_cut (5E-4 Hartree/bohr and 1E-4 Hartree by Default). The number of minimization steeps is set by n_min_steps (500 by default) and initial distance steep is set by minimzation_steep (by default 0.05 bohr)

It is highly advisable to compile LIO in double precision in order to minimise the error in exchange-correlation forces (precision=1). Outputs of geometry optimizations are traj.xyz (atoms coordinates in each steepes descent movement) and optimization.out (steep, energy and others). If verbose=true optimization.out includes the energy of each linear search point.

Examples

Examples of geometry optimization are made in lio/test/13_geom_optim.

Restraints

LIO may add an extra potential term to the Hamiltonian in order to restrain the distance between specified pairs of atoms.

Implemenation

The implementation is a simple harmonic potential over a generalized coordinate r.

$$U = \frac{1}{2}k[r - l_0]^2 \tag{4.2}$$

r may be defined as a weighted combination of distances between pairs of atoms.

$$r = \sum_i \sum_{j>i} w_{ij} |\vec{r_i} - \vec{r_j}| \tag{4.3}$$

In this formulation the force over an atom l is:

$$\vec{F_l} = -k[r - l_0] \sum_i \sum_{j>i} w_{ij} \frac{\vec{r_{ij}}}{r_{ij}} \eta_{ijl} \tag{4.4}$$

Where η_{ijl} is defined as:

$$\eta_{ijl} = \begin{cases} 1 & \text{if } l = i \\ -1 & \text{if } l = j \\ 0 & \text{in other case} \end{cases}$$

Using Restraints

The number of pairs of atoms to be added in the restraint potential(s) is defined by setting the variable number_restr, and a list of distance restrains have to be added to in an additional lio.restrain file. For example:

a_i	a_j	index	k	w_{ij}	l_0
1	2	0	0.1	1.0	7.86
3	4	0	0.1	-1.0	7.86
7	9	1	0.4	2.0	-2.3
13	1	1	0.4	1.0	-2.3
14	3	1	0.4	-3.0	-2.3
14	2	2	0.2	1.0	0.5
8	5	3	0.3	1.0	3.2

Columns a_i and a_j contain the atom numbers in the QM system to be restrained, while the index number determines which distances contribute to a same generalized reaction coordinate. The remaining columns are the force constants (k), weights of that distance in the generalized coordinate (w_{ij}) and equilibrium positions in atomic units (l_0).

Examples

1)In lio.in:

number_restr = 1

in lio.restrain:

a_i	a_j	index	k	w_{ij}	l_0
1	2	0	0.1	1.0	7.86

Potential added to system:

$$U = \frac{1}{2}0.1\Big[1.0|\vec{r_1} - \vec{r_2}| - 7.86\Big]^2 \tag{4.5}$$

2)In lio.in:

number_restr = 2

in lio.restrain:

a_i	a_j	index	k	w_{ij}	l_0
1	2	0	0.1	1.0	7.86
3	4	0	0.1	-1.0	7.86

Potential added to system:

$$U = \frac{1}{2}0.1\Big[1.0|\vec{r_1} - \vec{r_2}| - 1.0|\vec{r_3} - \vec{r_4}| - 7.86\Big]^2 \tag{4.6}$$

3)In lio.in:

number_restr = 4

in lio.restrain:

a_i	a_j	index	k	w_{ij}	l_0
1	2	0	0.1	1.0	7.86
3	4	0	0.1	-1.0	7.86
1	3	1	0.3	3.5	-2.31
7	8	1	0.3	-2.2	-2.31

Potential added to system:

$$U = \frac{1}{2}0.1\Big[1.0|\vec{r_1} - \vec{r_2}| - 1.0|\vec{r_3} - \vec{r_4}| - 7.86\Big]^2 + \frac{1}{2}0.3\Big[3.5|\vec{r_1} - \vec{r_3}| - 2.2|\vec{r_7} - \vec{r_8}| + 2.31\Big]^2 \tag{4.7}$$

Chapter 5

Electron Dynamics

Real Time TD-DFT

Electronic transport

Ehrenfest Dynamics

Chapter 6

Post-Processing Tools

TD-Analize: Electronic Spectra

CubeGen: Orbital and Density Visualization

Chapter 7

Reference Section

This section contains a quick reference for all of LIO's input variables and commandline options. For more detailed descriptions, please refer to the previous chapters.

Command line options

Variable	Description
-i file_name *character*20*	*default = 'lio.in'* Name of the input file containing LIO options.
-c crd_file.xyz *character*20*	*default = 'qm.xyz'* Name of thte XYZ file containing coordinates.
-b basis_file *character*20*	*default = 'basis'* A file containing the basis set and fitting set data, only used when int_basis=f.
-v *logical*	*default = .false.* Sets verbose level to 4.

Table 7.1: Command Line

Keywords - General Setup

Variable	Description
natom *integer*	*default = 0* Number of QM atoms in the system (Ignored when running from AMBER/GROMACS).
nsol *integer*	*default = 0* Number of classical atoms in the system (Ignored when running from AMBER/GROMACS).
charge *integer*	*default = 0* Total charge of the QM system.
open *logical*	*default = .false.* Perform an open-shell calculation.
nunp *integer*	*default = 0* Number of unpaired electrons for open-shell calculations.
style *logical*	*default = .false.* Activates a formatted version of the output.
fcoord *character*20*	*default = 'qm.xyz'* Name of the output file for the coordinates of the QM system. Please check the formats section for further information.
writexyz *logical*	*default = .false.* Writes an xyz file containing the QM system coordinates at each step of the MD-engine (AMBER/GROMACS/Hybrid).
verbose *integer*	*default = 1* Determines de level of verbosity: that is, the ammount of information that will be returned by the program. 0 = only the end results required. 1 = basic information about how is running: description pending.
timers *integer*	*default = 0* Activates timers for several parts of the code. 0 = timers are off.

Table 7.2: General Setup

Variable	Description
dbug *logical*	*default = .false.* Activates debug options in the code.
writeDens *logical*	*default = .false.* Writes electronic density to an output file after having finished the calculations.
writeForces *logical*	*default = .false.* Writes final forces to output.
dipole *logical*	*default = .false.* Calculates and prints dipole moment.
mulliken *logical*	*default = .false.* Performs a Mulliken Population Analysis.
lowdin *logical*	*default = .false.* Performs a Lowdin Population Analysis.
fukui *logical*	*default = .false.* Calculates condensed-to-atoms Fukui function (Spin Polarized Fukui in open-shell systems).
gaussian_convert *logical*	*default = .false.* Reads a density matrix restart as formated by gaussian09.
print_coeffs *logical*	*default = .false.* Prints MO coefficients in AO basis.

Table 7.2: General Setup

Keywords - GPU Options

Variable	Description
gpu_level *integer*	*default = 4* Determines which calculations are performed by the GPU. 0 = none.
max_function_exponent *integer*	*default = 10* Ignore functions with $\|exponent\| > max\_function\_exponent$.
little_cube_size *double precision*	*default = 8.0d0* Small cube-type point group size.
min_points_per_cube *integer*	*default = 1* Minimum number of grid points in a cube.
assign_all_functions *logical*	*default = .false.* Calculate all functions (ignores $max\_function\_exponent$).
sphere_radius *double precision*	*default = 0.6d0* Radius of the sphere-type point groups. 0 means there are no sphere-type groups, 1 means all points are contained in sphere-type groups.
remove_zero_weights *logical*	*default = .true.* Discard functions for those whose weight is zero (.false. option only remains as a debug option).
energy_all_iterations *logical*	*default = .false.* Calculate Exc energy in all SCF iterations.
free_global_memory *double precision*	*default = 0.0d0* Fraction of global GPU memory available for the calculation (1 means 100%).

Table 7.3: GPU Module Options

Keywords - DFT Hamiltonian

Variable	Description
iexch *integer*	*default = 9* Identifies the exchange-correlation potential to use with the calculation when not using libxc. Iexch=9 is the only option currently available.
use_libxc *logical*	*default = .false.* Activates the use of libxc version of the XC potential.
ex_functional_id *integer*	*default = ?* Exchange functional to use with libxc.
ec_functional_id *integer*	*default = ?* Correlation functional to use with libxc.
int_basis *logical*	*default = .true.* If true, looks for the internal basis indicated in variables $basis\_set$ and $fitting\_set$ (defaults are $'DZVP'$ and$'DZVPCoulombFitting'$. If false, an external basis file must be provided in the keyword $basis$.
basis_set *character*20*	*default = 'DZVP'* Name of the basis set used in the calculation when *int_basis* is set to true.
fitting_set *character*20*	*default = 'DZVP Coulomb Fitting'* Name of the fitting set used in the calculation when *int_basis* is set to true.
basis *character*20*	*default = 'basis'* Name of the input file containing the information of the basis set (useful for custom or mixed basis sets). Please check the formats section for further information.
n_ghosts *integer*	*default = 0* TODO: Number of ghost atoms.

Table 7.4: DFT Hamiltonian

Variable	Description
ghost_atoms *integer*	*default = 0* TODO: ?
rmax *double precision*	*default = 16.0d0* Maximum exponent in 3-center integrals. If exponent is greater than *rmax*, the current term is ignored.
rmaxs *double precision*	*default = 5.0d0* If the exponent in 3-center integral is within *rmax* and *rmaxs*, calculation is performed using single precision. This is ignored when using the AINT module.
iGrid *integer*	*default = 2* Grid type when iterating through SCF.
iGrid2 *integer*	*default = 2* Grid type for final energy calculation in SCF.

Table 7.4: DFT Hamiltonian

Keywords - Effective Core Potentials

Variable	Description
ECPMode *logical*	*default = .false.* Activate effective core potentials.
ECPTypes *integer*	*default = 0* Number of atoms with ECP.
tipeECP *character*30*	*default = 'NOT-DEFINED'* Type of ECP used.
ZListECP *integer*	*default = 0* Array with Z of atoms with ECP enabled.
cutECP *logical*	*default = .true.* Enables cuts for ECP integrals.
cut2_0 *double precision*	*default = 15.d0* Cut value for 2-center ECP integrals.
cut3_0 *double precision*	*default = 12.d0* Cut value for 3-center ECP integrals.
ECP_debug *logical*	*default = .false.* Enables ECP debug mode.
local_nonlocal *integer*	*default = 0* Calculates only local terms (when = 1) or only non-local terms (when = 2).
ECP_full_range_int *logical*	*default = .false.* Enables full-range integral calculations.
verbose_ECP *integer*	*default = 0* Controls ECP verbose levels.
fock_ECP_read *logical*	*default = .false.* Enables restart read in ECP.

Table 7.5: Effective Core Potentials

Variable	Description
fock_ECP_write *logical*	*default = .false.* Enables restart write in ECP.
fullTimer_ECP *logical*	*default = .false.* Enables full timers in ECP.

Table 7.5: Effective Core Potentials

Keywords - DFTB Embedding

Variable	Description
dftb_calc *logical*	*default = .false.* Activates the TB embedding of the system.
MTB *integer*	*default = 0* TODO Size of the two tight-binding subatrices.
end_bTB *integer*	*default = 0* TODO Index matrix size.
start_tdtb *integer*	*default = 0* TODO Initial time step for evolution of diagonal TB terms (????).
end_tdtb *integer*	*default = 0* TODO Final time step for evolution of diagonal TB terms (????).
alfaTB *double precision*	*default = UNSET* Manually sets the on-site energies (diagonal values) for the TB part of the Hamiltonian.
betaTB *double precision*	*default = UNSET* Manually sets the hopping terms for the TB part of the Hamiltonian (ie, the non-diagonal nearest neighbour terms for TB - TB interactions).
gammaTB *double precision*	*default = UNSET* Manually sets the hopping terms for the interaction between TB atoms and DFT atoms.
Vbias_TB *double precision*	*default = UNSET* Sets a bias for the on-site energies to simulate electrodes.
TBload *logical*	*default = .false.* TODO.
TBsave *logical*	*default = .false.* TODO.

Table 7.6: DFTB Embedding

Keywords - Fields and Biases

Variable	Description
field *logical*	*default = .false.* Use an external field (perturbation in TD).
a0 *double precision*	*default = 1.0d3* TODO: WTF IS THIS?.
epsilon *double precision*	*default = 1.0d0* Relative permitivity of the medium.
Fx, Fy, Fz *double precision*	*default = 0.05d0* The value of the external electric field in the x, y and z directions.
nfields_iso *integer*	*default = 0* Number of shape-isotropic fields. If $nfields\_iso > 0$, the inputs in $field\_iso\_file$ are ignored..
field_iso_file *character*20*	*default = 'field.in'* Isotropic fields input file.
nfields_aniso *integer*	*default = 0* Number of shape-anisotropic fields. If $nfields\_aniso > 0$, the inputs in $field\_aniso\_file$ are ignored..
field_aniso_file *character*20*	*default = 'field.in'* Anisotropic fields input file.
fockbias_is_active *logical*	*default = .false.* TODO.
fockbias_is_shaped *logical*	*default = .false.* TODO.
fockbias_readfile *character*80*	*default = 'atombias.in'* Atomic bias input file.
fockbias_timeamp0 *double precision*	*default = UNSET* TODO.

Table 7.7: Fields and Biases

Variable	Description
fockbias_timefall *double precision*	*default = UNSET* TODO.
fockbias_timegrow *double precision*	*default = UNSET* TODO.

Table 7.7: Fields and Biases

Keywords - Self Consistent Field

Variable	Description
initial_guess *integer*	*default = 0* Method for generating the initial guess for the SCF.
nMax *integer*	*default = 100* Maximum number of SCF steps.
told *double precision*	*default = 1.0d-6* Tolerance threshold for density matrix convergence.
Etold *double precision*	*default = 1.0d0* Tolerance threshold for energy convergence.
DIIS *logical*	*default = .true.* Use DIIS convergence accelerator if true, or damping convergence accelerator if false.
nDIIS *integer*	*default = 30* Number of DIIS convergence iterations.
gold *double precision*	*default = 1.0d1* Proportion of old matrix to use when using the damping mixture (gold = X means that it will use an 1:X new to old proportion).
hybrid_converg *logical*	*default = .false.* Use Hybrid convergence accelerator.
good_cut *double precision*	*default = 1.0d-5* Tolerance threshold for damped convergence, switch to DIIS afterwards.
vcinp *logical*	*default = .false.* Reads the molecular orbital coefficients from *frestart* and uses that as the starting guess for the first SCF cycle.

Table 7.8: Self Consistent Field

Variable	Description
rst_dens *integer*	*default = 0* Restart option for how to write/read restart files. 0 = reads/writes coefficients and builds density. 1 = reads density but writes coefficients. 2 = reads and writes density.
frestartin *character*20*	*default = 'restart.in'* Filename for the input containing the molecular orbital coefficients to be used as starting guess when *vcinp* = *.true.*.
frestart *character*20*	*default = 'restart.out'* Filename for the output containing the molecular orbital coefficients.
restart_freq *integer*	*default = 0* Indicates the frequency for writing the restart: it will do so every set number of calls to LIO (that is, number of steps of nuclear moves performed by the MD-engine.

Table 7.8: Self Consistent Field

Keywords - Geometry Optimization

Variable	Description
steep *logical*	*default = .false.* Activate steepest descent algorithm for geometry optimization.
Force_cut *double precision*	*default = 5.0d-4* Convergence criteria in forces (Hartree/bohr) for geometry optimization.
Energy_cut *double precision*	*default = 1.0d-4* Convergence criteria in energy (Hartree) for geometry optimization.
minimzation_steep *double precision*	*default = 0.05d0* Initial distance steep (bohr).
n_min_steeps *integer*	*default = 500* Maximum number of geometry optimization steps.
lineal_search *logical*	*default = .true.* Enable lineal search algorithm.
n_points *integer*	*default = 5* Number of points scaned for lineal search.
number_restr *integer*	*default = 0* Number of distance restraints used.

Table 7.9: Geometry Optimization

Keywords - Real Time TD-DFT

Variable	Description
timeDep *integer*	*default = 0* Use RT-TD-DFT when timeDep = 1.
tdStep *double precision*	*default = 2.0d-5* Timestep for TD-DFT (in atomic units).
ntdStep *integer*	*default = 0* Total number of TD-DFT steps.
propagator *integer*	*default = 1* RT-TD-DFT propagator (1 = Verlet, 2 = Magnus).
NBCH *integer*	*default = 10* Number of [ρ. Fock$^{\mathrm{n}}$] commutators in Magnus.
tdrestart *logical*	*default = .false.* Reads an input restart for TD (named td_in.restart).
td_rst_freq *integer*	*default = 500* Write the TD restart every td_rst_freq steps.
td_do_pop *integer*	*default = 0* Number of step stride in which the pop will be written. (0 means it is never written).

Table 7.10: Real Time TD-DFT

Keywords - Transport

Variable	Description
transport_calc *logical*	*default = .false.* TODO.
generate_rho0 *logical*	*default = .false.* TODO.
gate_field *logical*	*default = .false.* TODO.
driving_rate *double precision*	*default = UNSET* TODO.
pop_drive *integer*	*default = UNSET* TODO.
save_charge_freq *integer*	*default = UNSET* TODO.
nbias *integer*	*default = UNSET* TODO.

Table 7.11: Transport

Keywords - Ehrenfest

Variable	Description
ndyn_steps *integer*	*default = 0* Number of nuclear movement steps.
rsto_nfreq *integer*	*default = 0* Frequency (in steps) in which the restart is printed. (A value of 0 means only written in the end)
rsto_saves *logical*	*default = .false.* TODO.
rsti_loads *logical*	*default = .false.* TODO.
nullify_forces *logical*	*default = .false.* Returns 0 for all forces to the MD engine.
nullify_forces *logical*	*default = .false.* Returns 0 for all forces to the MD engine.
eefld_on *logical*	*default = .false.* TODO.
eefld_timegih *logical*	*default = .false.* TODO.
eefld_timegfh *logical*	*default = .false.* TODO.
eefld_ampx/ampy/ampz *double precision*	*default = 2.D-5* TODO.
eefld_timeamp *double precision*	*default = 2.D-5* TODO.
eefld_timepos *double precision*	*default = 2.D-5* TODO.

Table 7.12: Ehrenfest

Variable	Description
eefld_wavelen *double precision*	*default = 2.D-5* TODO.

Table 7.12: Ehrenfest

Keywords - CubeGen

Variable	Description
cube_dens *logical*	*default = .false.* Prints the electronic density.
cubeGen_only *logical*	*default = .false.* Avoid running SCF, only do cubeGen from a restart.
cube_res *integer*	*default = 40* Number of voxels per dimension (resolution).
cube_sel *integer*	*default = 0* Select only a particular orbital for printing (0 = all).
cube_dens_file *character*20*	*default = 'dens.cube'* File containing the electronic density.
cube_orb *logical*	*default = .false.* Prints orbital shapes.
cube_sqrt_orb *logical*	*default = .false.* Prints the orbitals' square root.
cube_orb_file *character*20*	*default = 'orb.cube'* File containing the orbital shapes.
cube_elec *logical*	*default = .false.* Prints the electric field.
cube_elec_file *character*20*	*default = 'field.cube'* File containing the electrical field.

Table 7.13: CubeGen