Lio

NaN

Departamento de Química Inorgánica, Analítica y Química Física/INQUIMAE, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, Ciudad Universitaria, Pab. II, Buenos Aires (C1428EHA) Argentina

E-mail:

Abstract

proyecto de manual de lio, hay q ponerle el formato adecuado

1 LIO - Quick Reference

This section provides a quick reference for LIO input variables, also providing the default values.

1.1 File I/O

Variable	Type	Default	Description
basis	char*20	'basis'	Filename for the basis set file (for custom basis sets).
output	char*20	'output'	Output file.
fCoord	char*20	'qm.xyz'	xyz file (without the header) containing the QM System coordinates.
writexyz	logical	false	Writes an xyz file containing the QM system coordinates, useful when using LIO in tandem with AMBER/GROMACS.
verbose	integer	1	Verbose level.
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.

1.2 Electronic Properties

Variable	Type	Default	Description
writeDens	logical	false	Writes electronic density to an output file after calculation.
writeForces	logical	false	Writes final forces to output.
print_coeffs	logical	false	Prints MO coefficients in AO basis.
mulliken	logical	false	Performs a Mulliken Population Analysis.
lowdin	logical	false	Performs a Lowdin Population Analysis.
fukui	logical	false	Calculates condensed-to-atoms Fukui function (Spin Polarized Fukui in open-shell systems).

1.3 Theory Level Options

Variable	Type	Default	Description
natom	integer	0	Number of QM atoms (ignored from AMBER/GROMACS).
nsol	integer	0	Number of classical atoms (ignored from AMBER/GROMACS).
charge	integer	0	QM system total charge.
open	logical	false	Perform an open-shell calculation.
nunp	integer	0	Number of unpaired electrons.
int_basis	logical	true	If set to false, an external basis file must be provided.
basis_set	char*20	'DZVP'	Name of the basis set used in the calculation (if int_basis is set to true).
fitting_set	char*40	'DZVP Coulomb Fitting'	Name of the fitting set used in the calculation (if int_basis is set to true).
nMax	integer	100	Maximum number of SCF steps.
number_restr	integer	0	Amount of distance restraints used.

1.4 Theory Level Options - Advanced

Variable	Type	Default	Description
DIIS	logical	true	Use DIIS convergence.
nDIIS	integer	30	Number of DIIS convergence iterations.
hybrid_converg	logical	false	Use Hybrid convergence.
Rho_LS	integer	0	0 use lineal search in Rho when SCF didnt converge, 1 use LS only i
told	double	1.0D-6	Tolerance threshold for density matrix convergence.
ETold	double	1.0D0	Tolerance threshold for energy convergence.
good_cut	double	1.0D-5	Tolerance threshold for damped convergence, switch to DIIS afterwards.
rmax	double	16	Maximum exponent in 3-center integrals. If exponent is greater than rmax, the current term is ignored.
rmaxs	double	5	If the exponent in 3-center integral is within rmax and rmaxs, calculation is performed in single precision (ignored when using AINT module).
iGrid	integer	2	Grid type when iterating through SCF.
iGrid2	integer	2	Grid type for final energy calculation in SCF.

1.5 RT-TD-DFT Options

Variable	Type	Default	Description
timeDep	integer	0	Use RT-TD-DFT when timeDep = 1.
tdStep	double	2.D-5	Timestep for TD-DFT (in atomic units).
ntdStep	integer	0	Maximum number of TD-DFT steps.
propagator	integer	1	RT-TD-DFT propagator (1 = Verlet, 2 = Magnus)
NBCH	integer	10	Number of [ρ. Fock^{n}] commutators in Magnus.
tdrestart	logical	false	Reads an input restart for TD (named td_in.restart) .
writedens	logical	false	Writes and output restart for TD during current run.
td_rst_freq	integer	500	Write the TD restart every td_rst_freq steps.

1.6 External Field Options

Variable	Type	Default	Description
field	logical	false	Use an external field (perturbation in TD).
epsilon	double	1.D0	Relative permitivity of the medium.
Fx, Fy, Fz	double	0.05D0	The strength of the external electric field.
nfields_iso	integer	0	Number of shape-isotropic fields. If nfields_iso > 0, the above inputs are ignored.
nfields_aniso	integer	0	Number of shape-anisotropic fields. If nfields_aniso > 0, the inputs above nfields_iso are ignored.
field_iso_file	char*20	'field.in'	Isotropic fields input file.
field_aniso_file	char*20	'fieldaniso.in'	Anisotropic fields input file.

1.7 Geometry Optimizations

Variable	Type	Default	Description
steep	logical	false	Activate steepest descent algorithm.
Force_cut	real*8	$5 * 10^{-4}$	Convergence criteria in forces (Hartree/bohr).
Energy_cut	real*8	$1 * 10^{-4}$	Convergence criteria in energy (Hartree).
minimzation_steep	real*8	0.05	Initial distance steep (bohr).
n_min_steeps	integer	500	Maximum number of optimization steps.
lineal_search	logical	true	Enable lineal search algorithm.
n_points	integer	5	Number of points scaned for lineal search.

1.8 Effective Core Potential Options

Variable	Type	Default	Description
ECPMode	logical	false	Activate the ECP mode.
ECPTypes	integer	0	Amount of atoms with ECP.
tipeECP	char*30	'NOT-DEFINED'	Type of ECP used.
ZListECP	integer	0	Z of atoms with ECP enabled.
cutECP	logical	true	Enables cuts for ECP integrals.
cut2_0	double	15.D0	Cut value for 2-center ECP integrals.
cut3_0	double	12.D0	Cut value for 3-center ECP integrals.
ECP_debug	logical	false	Enables ECP debug mode.
local_nonlocal	integer	0	Calculates only local terms (when = 1) or non-local terms (when = 2).
ECP_full_range_int	logical	false	Enables full-range integral calculations.
verbose_ECP	integer	0	Controls ECP verbose levels.
fock_ECP_read	logical	false	Enables restart read in ECP.
fock_ECP_write	logical	false	Enables restart write in ECP.
fullTimer_ECP	logical	false	Enables full timers in ECP.

1.9 CubeGen Options

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

1.10 GPU Options - Exchange-Correlation only

Variable	Type	Default	Description
max_function_exponent	integer	10	Ignore functions with $\|exponent\| >$ max_function_exponent
little_cube_size	double	8.0	Small cube-type point group size.
min_points_per_cube	integer	1	Minimum number of grid points in a cube.
assign_all_functions	logical	false	Calculate all functions (ignores max_function_exponent).
sphere_radius	double	0.6	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	true	Discard functions for those whose weight is zero.
energy_all_iterations	logical	false	Calculate Exc energy in all SCF iterations.
free_global_memory	double	0.0	Percentage (1 means 100%) of GPU memory available for storage.

2 What is LIO?

3 LIO Installation

4 Single-point and MD calculations

5 RT-TD-DFT calculations

6 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.

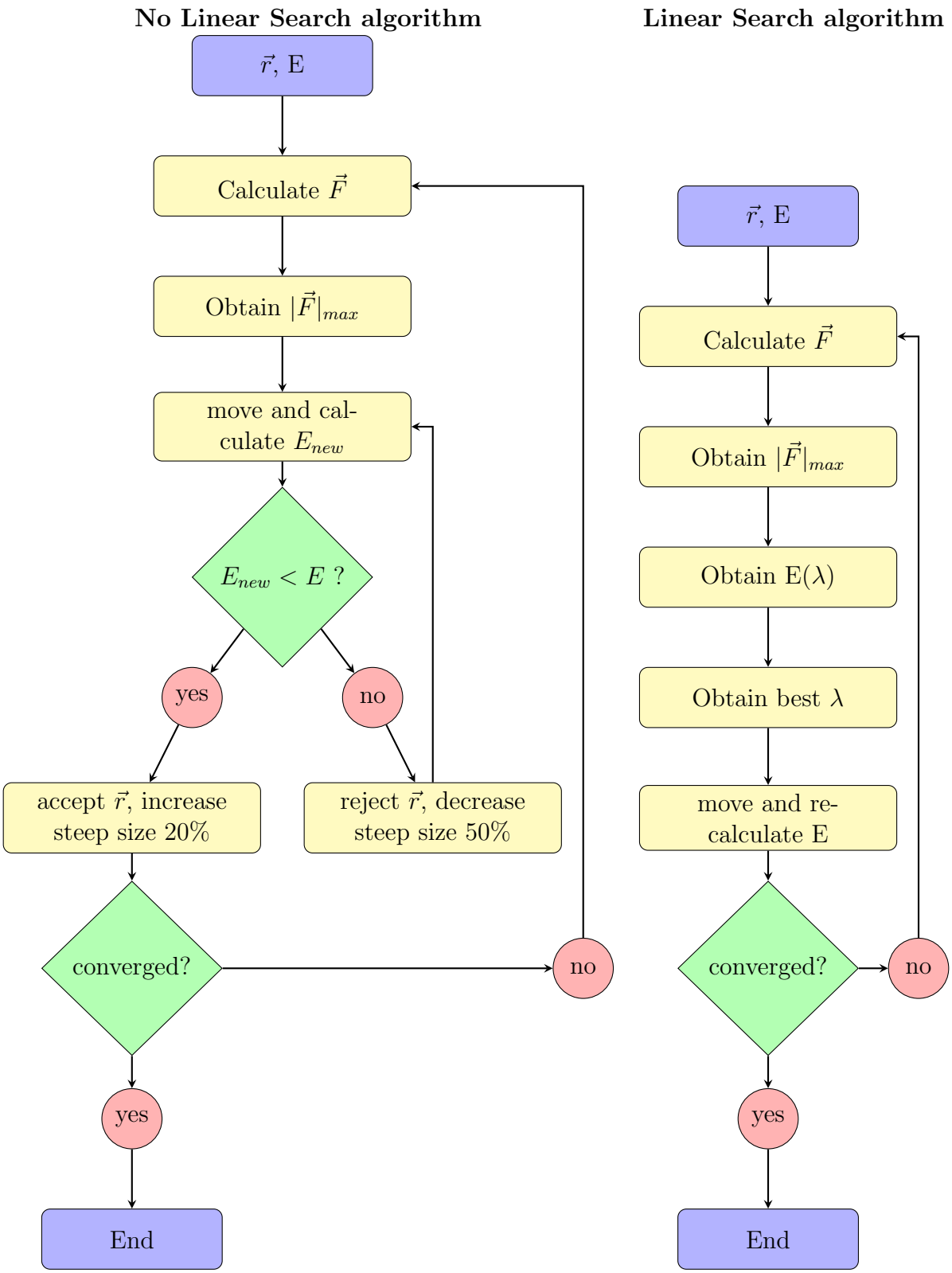
6.1 Implemenation

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

$$\vec{r}^{\boldsymbol{i}}_{new} = \vec{r}^{\boldsymbol{i}} + \lambda \vec{F}^{i} \tag{1}$$

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.



Best λ in lineal search algorithm is obtained by a quadratic function ajusted using min-

imum energy of the scan and previous and next points.

6.2 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.

6.3 Examples

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

7 Restraints

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

7.1 Implemenation

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

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

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

$$r = \sum_i \sum_{j>i} w_{ij} |\vec{r_i} - \vec{r_j}| \tag{3}$$

In this formulation the force over an atom l is:

$$\vec{F_l} = -k[r - l_0] \sum_i \sum_{j>i} w_{ij} \frac{\vec{r_{ij}}}{r_{ij}} \eta_{ijl} \tag{4}$$

Where η_{ijl} is defined as:

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

7.2 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).

7.3 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:

$$U = \frac{1}{2} 0.1 \Big[1.0 |\vec{r_1} - \vec{r_2}| - 7.86 \Big]^2 \tag{5}$$

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:

$$U = \frac{1}{2}0.1\Big[1.0|\vec{r_1} - \vec{r_2}| - 1.0|\vec{r_3} - \vec{r_4}| - 7.86\Big]^2 \quad (6)$$

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:

$$U = \frac{1}{2}0.1\Big[1.0|\vec{r_1} - \vec{r_2}| - 1.0|\vec{r_3} - \vec{r_4}| - 7.86\Big]^2 + \frac{1}{2}0.3\Big[3.5|\vec{r_1} - \vec{r_3}| - 2.2|\vec{r_7} - \vec{r_8}| + 2.31\Big]^2 \quad (7)$$

8 Effective Core Potentials

9 Using external fields

10 Ehrenfest Dynamics

11 Doing electron transport calculations

12 Using CubeGen for orbital and density printing

13 Tips and tricks - Optimizing your runs