

Autocip : User's manual

November 3, 2016

1 General purpose

This Program executes the CIPSI chain of Toulouse for atoms and molecules. The main output are :

- Potential curves : file vccind
 - For a series of interatomic distance, either on a grid or with LHS method in an hypercube depending on the grid_method tag (='grid' or 'LHS' or 'LIST')
 - For a series of NSYM symmetries SYM(NSYM)
- Transition dipole moment : files dx-, dy-, dz-

The Sorting is automatically done including the removal of parasites states for CINFV and DINFH group symmetries. Potentials can include the core-core term 'vccind_sym', or not 'v_sym'

General procedure (option a or b depending if spin symmetrisation done or not) :

1. pshf rpol eval pshf ijkl fock : for every symmetry
2. cipsym moy bd : for every symmetry, loop to get the initial active space
3. ciro : for every combinaison of symmetry asked

The run can stop after each step depending on input :

- L_MONO=T: monoelectronic energies are stored
- L_CIP=F: stop at step 1 (monoelectronic energies)
- L_DIP=F: stop at step 2 (molecular energies)
- L_DIP=T: also compute properties (dipole moment)

Execution : autocip<auto.in

NOTE: AUTOCIP has to be executed in a "root" directory (like /tlse/exemple); final results are written in the same directory. Direct output of the CIPSI chain are stored in an automatically created subdirectory : "Results". AUTOCIP also create a subdirectory work inside which each processor create his own work subdirectory. CIPSI binary are excuted in those work subdirectories, temporary unformatted files are deleted at the end of each geometry run.

Input files are created automatically at each step starting from a model file. The model files have to be in the initial root directory along the autocip binary :

- na2.dat, for step 1 (F_PSHF)
- na2sym.dat, for step 2 (F_SYM)
- na2dip.dat, for step 3 (F_MDIP)

See file sym_check.f90 for notation of the available symmetries : point group 'DINFH', 'CINFV', 'CNV', 'CNH', 'CS', 'C1', 'DNH'.

2 auto.in

The input file must start with the 12 lines explained in the subsection 2.1. Autocip will make sequential call to 'read(5,*)', each line have to start with the correct type and correct number of variables (ex : real, integer, logical, string). A format mismatch will probably crash the program. Anything written on a line after the requested variables won't be read by the program. After the first fixed lines, the input files should also contained several block depending on the calculation performed (:grid:, :method:, :symmetries:, :electricfield:, :dipolemoments:, :diabatization:). Each block start by the line with the name of the block followed by a fixed number of line explained in the following subsections.

In the tables, variables in paranthese in the Type column correspond to the dimension of the array, multiplication indicate that the type is repeated several times.

The environment variable \$CIPSI_ROOT must be included in the PATH of the user. Binary files directory (and Pseudopotential) are given relative to this root directory.

2.1 initialization

Type	Name	Explanation
Logical	l_calc	if false the program will only sort preexisting result (grid definition must be identical between the 'calc' and the 'sort' run !)
String	bindir	subdirectory of the CIPSI binary
String	bash_command	additional command used to call the CIPSI bin (for example a valgrind call)
Null		This variable is only used by the script calling autocip. The line can be left empty but still must be here.
Integer, Integer	ncore	number of core
	nelac	number of electron
String(ncore)	atom	name of each nucleus
String	f_pshf	name of the file containing the namelist used by pshf/rcut/cval/ijkl/fock
String	opt_rc	'RPOL' or 'RCUT'
Logical, Integer, Integer	l_mono	if true extract the mono-electronic energies from the raw output
	l_max	energies are given for Lambda from 0 to l_max
	nvec	number of mono-electronic energies for each Lambda symmetry
Logical*3, Integer*3	l_cip	if true run the CI calculation
	l_dip	if true compute the dipole moments
	l_vcc	if true add a short range and a long range term to the energies (deprecated : option should not be used)
	a_rep	parameter for the short range term
	b_rep	
	c_rep	
Logical, Logical	l_efield	if true do the computation with an external electric field, :electricfield: block needed
	l_efield_1	?
Logical, Logical	l_diab	option 'True' not available
	l_calcref	–

2.2 :grid: block

Type	Name	Explanation
String, Logical	group	Point group for the calculation
	l.com	if true origin = center-of-mass else origin = center-of-charge
String	grid_method	'grid' or 'lhs' or 'list'. Method used to define the geometry of the system : take points on a grid, or take random points in an hypercube using the Latin Hypercube Sampling method, or read a file with each geometry given
String	coordinate	unused for dimers. For trimers : 'jacobi' (adapted to the A + BC system), 'hyperspherical' (from Smith-Whitten), 'internuclear distance' For tetramers : 'jacobi' (adapted to the AB + CD system), dnh-jacobi (adapted to the AC + BD system), 'internuclear distance', not implemented yet : hyperspherical (note the program only read the first character of the string)
Real(ndim)	gridmin	lowest values of the box for each dimension
Real(ndim)	gridmax	maximal values of the box for each dimension
option dependent		
Real(ndim)	grid_step	for 'grid' : the step of the grid for each dimension, a step bigger than the size of the box can be used for single point calculation at R=R_min
Real	nz	'LHS' : number of points of the design
String	grid_file	'list' : name of the file containing the geometries. The file must start with the N number of points followed by N line describing each one geometry
Real	lower_bound	points with one or more internuclear distances below this value are excluded from the calculation

2.3 :method: block

Type	Name	Explanation
Logical	l_fci	true : Full-CI, false : CI-PSI
Integer, Integer	noac	number of orbitals of the starting active space. -1 : all orbitals taken, else the program try to add orbitals allowing construction of mono-excited determinants of the required symmetry
	nexmax	determinants with excitation higher than nexmax are excluded from the initial active space (for uneven number of electron the program automatically adjust the excitation to take into account the virtual electron used by CIPSI)
Integer, Real	netat	use convergence criteria envp_max for the first netat state, higher states will have lower precision (check vthresh output file)
(not used by FCI)	envp_max	maximum value of the contribution of the 'perturbative' space (ie of the determinant not included in the calculation), can be taken as an approximate error bar for the CIPSI run compared to the FCI
Real, Real	tau_init	initial value of TAU and TEST in the cip namelist
(not used by FCI)	tau_step	step value for TAU, must be lower than 1 : after each loop TAU=TAU*TAU_STEP
Integer*4	det_bounds	minimum and maximum number of determinant in the reference space, min. and max. number of determinants in the total space
(not used by FCI)		the minimal size of the ref. space is needed by the algorithm, the other values are constraints in case of limited computer time and can be set to zero/infinity if needed
		the program will keep iterating until the minimum values are reached. If the maximum values are reached before convergence, an error message will be displayed and further calculation on this geometry will be skipped
Logical	ysort	automatic detection of parasite state for CINFV/DINFH :
Real	trec	initial value for TREC in ijkl namelist, the program will try to adjust the value automatically if the run encounter errors

2.4 :symmetries: block

Type	Name	Explanation
String	f_sym	name of the file containing cip, moy and bd namelists
Integer, Integer	nsym	number of symmetry asked
	metat	number of state asked for each symmetry
String(nsym)	sym	name of each symmetry, the program will checked that each symmetry is compatible with the point group. See sym_check.f90 source file for a list of point group and their symmetry
Integer*6	isymat	the six integers correspond resp. to singlet/doublet/triplet/quartet/quintet/sextet. Creation and diagonalization of the corresponding spin-symmetrized CI matrix is done if isymat=1 and not done if isymat=0. Additionally the first integer can have the value -1, for a diagonalisation of the full matrix with no symmetrisation (in this case the spin of each state is determined afterwards).

2.5 :electricfield: block

Type	Name	Explanation
String	efield_dirn	axis of the electric field : 'x', 'y' or 'z'
Integer	nfield	number
Real(nfield)	efield_vals	values of the amplitude of the electric field
Logical	l_ffield	if true, compute PDM and polarisability with the finite field method (not fully implemented, currently just gather all energy output)

2.6 :dipolemoments: block

Type	Name	Explanation
String	f_mdip	name of the file with the ciro namelist
Integer	ncom	number of transition dipole moment combinaison to be computed
String, String	com1, com2	each line contain the name of the symmetry for the starting state, and the one for the final state.
-		the block must have ncom lines of symmetry combinaison

2.7 :diabatization: block

Not available in the current version

2.8 example

example of input :

```
t
'bin'
"
tmp
2,2
'Rb','Rb'
'rb2.dat'
'ropol'
t,3,20
t,t,f,0,0,0
f,f,
f,f,
```

```
# configuration block #
```

```
:grid:
'DINFH',t
'grid'
'jacobi'
11.45
30.0
100.
5.0
```

```
:method:
t
-1,2
1,1d-5
```

0.1,0.5
2000,20000,0,1000000
T
1d-4

:symmetries:
'rb2sym.dat'
3,15
'spu','spg','pxg'
1,0,1,0,0,0

:electricfield:
'y'
6
0.0,0.0001,0.0002,0.0003,0.0004,0.0005
f

:dipolemoments:
'rb2dip.dat'
2
'spu','spg'
'spu','pxg'