Description of utility-programs dealing with the configuration expansions in LS-coupling (folder UTILS/CONF_LS)

Quick-references to the utilities

cfile	extracts the state expansion from CI *l*(*j*)-file
cfile_all	extracts the set of states from the *l*- or *j*-file in separate *c*-files
ci_list	creates list of states from one or set of *j*(*l*)-files
lfile	find the maximum contributions for given configurations from the list of solutions in *l*-file
merge	mergs a set of (name.c, name.w) files to one pair (merge.c, merge.w) with consistent set-indexes if orbitals are not-orthogonal
order_c	orders c-file according to configuration weights
order_cc	orders c-file according to configuration weights + defines the most important configurations
zf_tab	creates tables for f-values and tau based on the **zf_res** data
zgenconf	prepares the list of configurations (without terms indication) from the list of electron occupations and allowed promotions
zgenterm	prepares the list of CSF's from a list of the configurations

CFILE

Description:	extracts the given state from *l*- or *j*-file in separate *c*-files
Input files:	**name.l (j) + name.w**
Output files:	**state.c + state.w**
Call as:	**`cfile  name.l(j) nn 2J state eps_c`**

In the BSR complex, each target state is described by a pair **name.c** and **name.w**. The CI program from the MCHF complex produces the results in the *l*(*j*)-files as a list of states expansions, plus the corresponding w-file with one-electron radial functions. The CFILE utility allows one to extract the expansion coefficients for a given state and create the corresponding *c*- + *w*- files. The user should indicate in the command line the name of the input *l*- or *j*-file, the pointer for the given state, **`nn`**, the value of 2*J* if it is a *j*-file, the name for the output c-file, **state**, and the tolerance for the expansion coefficients, **eps_c**. If **eps_c** > 0, the configuration expansion in c-file will be ordered.

CFILE_ALL

Description:	extracts the set of states from the *l*- or *j*-file in separate *c*-files
Input files:	**name.l (j) + name.w**
Output files:	set of **{name_nnn.c + name_nnn.w}**
Call as:	**cfile_all name.l(j) msol=… JJ=… eps_c=…**
	msol – number of solutions to extract **JJ** - 2J-values **eps_c** - tolerance for the expansion coefficients

CI_LIST

Description:	creates list of states in set of *j*(*l*)-files
Input files:	**name.c + name.j +** [**ci_list.inp**]
Output files:	**name.j_list**
Call as:	**ci_list a.j** **ci_list inp=b.inp** **ci_list a.j b.j c.j ... unit=eV msol=10 out=list**
Parameters	**mco** - number of leading configurations to show [1] **eps_c** - tolerance for weights [0.2] **unit** - au, Ry, eV, cm [au] **shift** - overall energy shift [0.d0] (shift=1 means shift=-E1) **msol** - max.number of solutions for output [0->all] **awt** - atomic weight **out** - name for the resulting list

Example of input file [ci_list.inp]:

```
name1.j
name2.j
.........
nameN.j

mco = 2
eps_c = 0.1
unit = eV
msol = 10
shift = 1000
```

All parameters, except **name.j**, are optional

LFILE

Description: finds the average contributions for given configurations (c-file) from the subset of solutions in given *l*-file

Input files: **name.l + name.c**

Output files: **name.ca + name.cb**

Call as: **lfile name.l nsol [eps_c]**

nsol – number of solutions

eps_c [0] - optional cut-off parameter

name.ca - list of CAS in order of importance
with c = sqrt[(c_1^2 +c_2^2 +...)/ nsol]

name.cb - list of configurations in order of importance with
coefficients summed over all terms

This utility is used to find most important CAS or configurations when we want to reduce the expansions to minimum

MERGE

Description: Merging a set of (name.c, name.w) files to one pair (merge.c, merge.w) with consistent set-indexes if orbitals are not-orthogonal

Input files: set of c- and w-files

Output files: **merge.c + merge.bsw**

Call as: **merge name1 name2 ...** [**merge=… jmin=… jmax=… eps=…**] or
merge inp=...

merge - name for resulting **merge.c + merge.w** files **[merge]**

jmin - minimum 2J value [-1]

jmax - maximum 2J value [-1]

eps - tolerance for one-electron overlaps [1.d-6]

List of merging c-files are defined as arguments without '=' sign. This list also can be given in input file defined through **inp**=... parameter, one filename per line.

ORDER_C

Description:	orders c-file according to configuration weights
Input files:	**input.c** [**cfg.out**]
Output files:	**output.c** [**cfg.inp**]
Call as:	**order_c** [**input.c output.c eps_c**]
	eps_c [0] - cut-off parameter (if > 0)

ORDER_CC

Description:	orders c-file according to configuration weights + defines most important configurations
Input files:	**name.c**
Output files:	**name.c** (reordered) + **name.conf** (no terms)
Call as:	**order_cc name.c**

This routine is used to get the most important configurations

ZF_TAB

Description:	creates tables for f-values and tau based on the **zf_res** data
Input files:	**zf_res**
Output files:	**zf_tab** + **zf_tau**
Call as:	**zf_tab** [**eps_e=... eps_f=...**]
	eps_e - tolerance for equal energies [1.d-8]
	eps_f - tolerance for output of f-values [1.d-6]

ZGENCONF

Description:	prepares the list of possible configurations (without terms indication) from the list of electron occupations and allowed promotions.
Input files:	**electron.inp** (will be created as example if absent)
Output files:	**conf.inp**
Call as:	**zgenconf** [**mo=...**] **mo** - maximum number of subshells

If input file is absent, the example will be created for the user to fill it in.

```
Ti
  1s  2s  2p
   7  11   1   4   0   2   n_orb, n_elec, par, n_ref, k_min, k_max
  3s  3p  3d  4s  4p  4d  4f
   2   6   0   0   0   0   0
   2   6   2   2   2   2   2


n_orbitals  - number of orbitals in following list
n_electrons - number of electrons above common core
parity      - -/+ 1
n_ref       - reference orbitals for promotions
k_min       - minumum promotion, e.g,   1 - single
k_max       - minumum promotion, e.g.,  2 - double
```

Figure 1. Example of the **electron.inp** file.

```
Ti
  1s  2s  2p
  3s( 2)  3p( 6)  3d( 2)  4s( 1)
  3s( 2)  3p( 6)  3d( 1)  4s( 2)
  3s( 2)  3p( 6)  3d( 2)  4d( 1)
  3s( 2)  3p( 6)  3d( 1)  4s( 1)  4d( 1)
  3s( 2)  3p( 6)  3d( 1)  4p( 2)
  3s( 2)  3p( 6)  3d( 1)  4p( 1)  4f( 1)
  3s( 2)  3p( 6)  3d( 1)  4d( 2)
  3s( 2)  3p( 6)  3d( 1)  4f( 2)
*
```

Figure 2. Example of the **conf.inp** file obtained on base of the information in the **electron.inp,** Fig.1.

ZGENTERM

Description:	prepares the list of CSF's from a list of the configurations
Input files:	**conf.inp**
Output files:	**cfg.inp**
Call as:	**zgenterm** with interactive input or **genjterm conf-file J_min J_max L_min L_max S_min S_max c-file** All angular momentum values are given in (2J+1) representation, where zero value means no restrictions in this respect
Example:	**zgenterm a.conf 0 0 3 3 1 3 a.c**

```
Ti
  1s  2s  2p
  3s( 2)  3p( 6)  3d( 2)  4s( 1)
 1S0 1S0 1D2 2S1 1S  1D  2D
  3s( 2)  3p( 6)  3d( 2)  4d( 1)
 1S0 1S0 1S0 2D1 1S  1S  2D
  3s( 2)  3p( 6)  3d( 2)  4d( 1)
 1S0 1S0 3P2 2D1 1S  3P  2D
  3s( 2)  3p( 6)  3d( 2)  4d( 1)
 1S0 1S0 1D2 2D1 1S  1D  2D
  3s( 2)  3p( 6)  3d( 2)  4d( 1)
 1S0 1S0 3F2 2D1 1S  3F  2D
  3s( 2)  3p( 6)  3d( 2)  4d( 1)
 1S0 1S0 1G2 2D1 1S  1G  2D
  3s( 2)  3p( 6)  3d( 1)  4s( 2)
 1S0 1S0 2D1 1S0 1S  2D  2D
  3s( 2)  3p( 6)  3d( 1)  4s( 1)  4d( 1)
 1S0 1S0 2D1 2S1 2D1 1S  2D  1D  2D
  3s( 2)  3p( 6)  3d( 1)  4s( 1)  4d( 1)
 1S0 1S0 2D1 2S1 2D1 1S  2D  3D  2D
  3s( 2)  3p( 6)  3d( 1)  4p( 2)
 1S0 1S0 2D1 1S0 1S  2D  2D
  3s( 2)  3p( 6)  3d( 1)  4p( 2)
 1S0 1S0 2D1 3P2 1S  2D  2D
  3s( 2)  3p( 6)  3d( 1)  4p( 2)
 1S0 1S0 2D1 1D2 1S  2D  2D
  3s( 2)  3p( 6)  3d( 1)  4p( 1)  4f( 1)
 1S0 1S0 2D1 2P1 2F1 1S  2D  1P  2D
.......................................
*
```

Figure 3. Example of the **conf.inp** file obtained on base of the information in the **electron.inp,** Fig.1.