5. Program DBSR_CONF

5.1. Outline of the DBSR_CONF calculations

This program generates the close-coupling expansions (2.5) for the case under consideration, based on the information given in the file **target_jj**. First, the program reads the target configuration expansions from all **targ_nnn.c** files. Then the continuum configurations are simply generated from target configurations by adding the continuum orbital, in accordance with the *JJ*-coupling rules. All possible continuum orbitals are considered, if only the additional restrictions are not indicated by input parameters **max_ll** or **max_ka**. These parameters are mainly used only in the bound-state calculations, in order to get more compact expansions. If there is a **pert_nnn.c** file for a pertuber, this expansion is also added without any change. The resulting configuration expansions for each partial wave is recorded in the files **cfg.nnn**, and the relevant channel information is added to the file **target**. The continuum configurations in the close-coupling expansions in the **cfg.nnn** files have the same expansion coefficients as in the corresponding target states or perturber expansions.

The most complicated point concerns the orthogonal constraints imposed on the continuum orbitals. By default, continuum orbital are supposed to be non-orthogonal to the bound orbitals in the open subshells. In this case, we don't need to add any (N+1)-electron bound terms in the close-coupling expansion in order to compensate the orthogonality restrictions. However, to work completely without orthogonality constraints is not possible.

First, all continuum orbitals should be orthogonal to the closed-shell orbitals. This requirement does not lead to any additional (N+1)-electron bound terms in the close-coupling expansion in order to compensate these restrictions. Orthogonality to the closed shell is automatically imposed in the present package, without any additional input parameters.

Another common situation is related to the case when different channels can generate the same configurations. Consider again, for example, electron scattering from xenon with channels $5p^6kl$ and $5p^5nlk$p. If the orbital kl has no orthogonal constrains, then the first channel can generate the $5p^6nl$ bound states. The same states will also be generated by a second channel if the kp orbital is not orthogonal to the 5p orbital. Such situation may lead to the 'overloaded' overlap matrix in the generalized eigenvalue problem (2.17). As a result, the overlap matrix may not be positive definite, and this prevents its diagonalization. To avoid such a situation, the program should impose the orthogonality constraint $<k\text{p}|5\text{p}>=0$ in the given example, without any compensation configurations, because these configurations can still can be generated in the first channel. As a general rule, program try to impose all orthogonality constraints which do not lead to compensation configurations.

The above procedure is easy to imply in case of simple one-configuration target states, however, it is not straightforward for the many-configurational expansions with big mixing coefficients. If we have a continuum configuration such as 'target,kl' and the continuum orbital kl is imposed to be orthogonal to the bound orbital nl,

then the compensation configurations have the structure 'target,*nl*', with proper recoupling for the equivalent electrons. As practical calculations show, not all target configurations here are of the same importance. The compensation configurations based on the target configurations with small expansion coefficients may lead to an overestimate of the compensation effects. These, in turn, may lead to underestimating the cross sections or to additional pseudo-resonance structure. The program first should find leading physical (spectroscopic) configurations. Here the program uses the information about the physical and correlation orbitals obtained in the DBSR_PREP run (see description of the **target_orb** file). Then from the leading configurations, the program choses only one configuration which represents given target states and don't coincide with any other target states. In case of strong configuration mixing it should not be the configuration with the leading coefficient. When all target states are represented by a single configuration, the procedure to find the orthogonality conditions which don't lead to (N+1)-electron compensation configurations is rather straightforward.

The generated orthogonality conditions are recorded in the end of the corresponding **cfg.nnn** file as **`<kl|nl>=0`**, one line for one constraint. If needed the user can add additional constraints, ether in the **cfg.nnn** file, or in the **dbsr_par** file. In the last case, these conditions will be applied to all partial waves. Additional constraints may be also added later by the DBSR_MAT program if the overlap matrix will contain big matrix element between different channels.

Another important point concerns the cases when orthogonality constraints should be imposed to non-orthogonal but very similar orbitals. For example, we can use the term-dependence radial functions for the same *nl* orbital, or we can introduce the relaxation effects in the core-excitation. In the above example of e–Xe^+ scattering, we can use two different radial functions for the 5p orbital in the $5p^5$ and $5p^6$ configurations to account for the relaxation effects. These functions are different but very similar. In this case we should apply the <*k*p|5p>=0 or <*k*p-|5p->=0 condition only to one of these orbitals. For these reasons, the orthogonality constraints are applied only to substitution orbitals introduced by DBSR_PREP program. Recall that substitution orbitals mimic all physical orbitals and they form orthogonal set of orbitals.

5.2. Data files

dbsr_par File type: formatted sequential input.
Prepared by user.
Description: input parameters for given run.
Format: see section 3.2 and examples in section 14.

target_jj File type: formatted sequential input.
Prepared by user.
Read and modified by programs DBSR_PREP and DBSR_CONF**.**
Description: contains list of target states and scattering channels.
Format: see examples in the present section.

targ_nnn.c File type: formatted sequential input.
Created by program DBSR_PREP.
Read by program DBSR_CONF.
Description: contains configuration expansion for target states **nnn.**

pert_nnn.c File type: formatted sequential input.
Created by program DBSR_PREP.
Read by program DBSR_CONF.
Description: contains configuration expansion for perturber **nnn.**

cfg_nnn.c File type: formatted sequential output.
Created by program DBSR_CONF.
Read by programs DBSR_BREIT and DBSR_MAT.
Description: contains configuration expansion for partial wave **nnn.**

dbsr_conf.log File type: formatted sequential output.
Written by program DBSR_CONF**.**
Read by user.
Description: running information.

5.3.Input parameters

Input parameters can be provided in the command line or in the input file **dbsr_par** (data from the command line overwrite data from the input file). Below we describe the data from **dbsr_par** that are read by program DBSR_CONF. The default values for all data are indicated in the brackets.

c_comp [1.01]	tolerance for generation of orthogonality constraint based on the coefficient of the compensation configuration. By changing this tolerance, we may strengthen or weaken the orthogonality constraints.
min_ll [-1]	restriction on the minimum value of orbital angular momentum *l* for the continuum orbitals (the default value of -1 means no restrictions)
max_ll [-1]	restriction on the maximum value of orbital angular momentum for the continuum orbitals (the default value of -1 means no restrictions)
min_ka [-1]	restriction on the minimum *kappa*-value for the continuum orbital (the default value of -1 means no restrictions).
max_ka [-1]	restriction on the maximum *kappa*-value for the continuum orbital (the default value of -1 means no restrictions).
<*kl*|*nl*>=0	additional orthogonal condition for the continuum orbital *kl*. If a set index indicated, this orthogonal condition will be applied to specified orbitals; otherwise, it will be applied to all orbitals with the given *l* or κ. By default, all continuum orbitals are supposed to be non-orthogonal to the bound orbitals.

5.4. Modification of file ***target_jj***

DBSR_CONF modifies the file **target_jj** by adding information about the close-coupling expansions generated by the program. An example of such information is given in Fig.5.1. For each partial wave, the program provides the number of channels, **nch**, and the number of configurations, **nc**, in the corresponding **cfg.nnn** file. The number of configurations is represented by two values, corresponding to the continuum and the (N+1)-electron parts of the given close-coupling expansion. Then, in a separate line for each channel, we have index of the channel, the spectroscopic notation for the continuum orbital, its *kappa*-value, a pointer to the corresponding target state, a pointer to the last configuration for the given channel in the file **cfg.nnn.** Historically, we use the spectroscopic notation for the continuum orbital, though for big l-values it may contain some weird symbols. As additional overall information, **target_jj** also provides the maximum number of channels, the maximum number of configurations, and the maximum number of different one-electron orbitals used in the close-coupling expansion.

When the user need to remove some specific channels, the program provides the following option. First, we need to copy the **target_jj** in the **target_del** file. Then we nullify the configuration pointers for the channels to be removed. The repeating run of DBSR_CONF will then create the **target_jj** file with the indicated channel removed.

```
-----------------------------------------------------------------
channels:
-----------------------------------------------------------------
  1.   nch =   3    nc =       4       1
-----------------------------------------------------------------
  1.   kp 4   -2    1          1
  2.   kp-4    1    2          2
  3.   ks 4   -1    3          3
-----------------------------------------------------------------
  2.   nch =   3    nc =       3       0
-----------------------------------------------------------------
  1.   kd-1    2    1          1
  2.   ks 3   -1    2          2
  3.   kp-3    1    3          3
-----------------------------------------------------------------
  3.   nch =   7    nc =       7       0
-----------------------------------------------------------------
  1.   kp-3    1    1          1
  2.   kp 3   -2    1          2
.................................. ....
-----------------------------------------------------------------
max_ch  =        8  -  max.number of channels
max_nc  =       11  -  max.number of configurations
max_wf  =       28  -  max.number of orbitals
-----------------------------------------------------------------
```

Fig. 5.1. Channel information added to **target_jj** by the program DBSR_CONF.

```
Core subshells:  -7446.48480535
  1s   2s   2p-  2p   3s   3p-  3p   3d-  3d   4s   4p-  4p   4d-  4d
Peel subshells:
 5s 1 5p-1 5p 1 5s 2 5p-2 5p 2 5s 3 5p-3 5p 3 kp 4 kp-4 ks 4
CSF(s):
 5s 1( 2) 5p-1( 2) 5p 1( 3) kp 4( 1)                          1.000000000000000
                       3/2      3/2
                                  0+
 5s 1( 2) 5p-1( 1) 5p 1( 4) kp-4( 1)                          1.000000000000000
               1/2                1/2
                                  0+
 5s 2( 1) 5p-2( 2) 5p 2( 4) ks 4( 1)                          1.000000000000000
     1/2                          1/2
                                  0+
 5s 3( 2) 5p-3( 2) 5p 3( 4)                                   1.000000000000000

                           0+
***

Imposed orth. conditions:


Derived orth. conditions:

< kp 4| 5p 1>=0       targ_001     5p5_j3

< kp-4| 5p-1>=0       targ_002     5p5_j1

< ks 4| 5s 1>=0       targ_003     5s5p6_j1
```

Fig. 5.2. Example of orthogonal constraints added by program DBSR_CONF in the cfg.001 file for Xe-bound calculations.