5. Program BSR_CONF

5.1. Outline of the BSR_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**. First, the program reads the target configuration expansions from all **targ_nnn.c** files. For each partial wave, the corresponding close-coupling expansions are then generated in *LS* coupling, or in *LSJ*, *jK* or *jj* coupling in the case of semi-relativistic calculations, according to input parameter **coupling**. The continuum configurations are simply generated from target configurations by adding the continuum orbital, in accordance with the angular momentum coupling rules. All possible continuum orbitals are considered, if only the additional restrictions are not indicated by input parameters **max/min_ll**, **max/min_LT**, or **max/min_ST**. These parameters are mainly used only in the bound-state calculations, in order to get more compact expansions. Then, if there is a **pert_nnn.c** file for a pertuber, this expansion is also added without any change. The resulting configuration expansion for each partial wave is recorded in the file **cfg.nnn**, and the relevant channel information is added to the file **target**. Note that if some orthogonal conditions are imposed on the continuum orbitals, the program will not generate any additional (N+1)-electron configurations, which are intended to compensate for the orthogonal restraints implied on the continuum orbitals.

The continuum configurations in the close-coupling expansions in the **cfg.nnn** files have the same expansion coefficients, as the corresponding target states, whereas the expansion coefficients for the (N+1)-electron configurations are equal to unity for the standard perturbers. For perturbers, given through **kpert** option, the expansion coefficients are kept the same as in the corresponding input files. Further modification of the expansion coefficients is introduced in the case of intermediate coupling. The configurations in the *c*-files are written in consecutive *LS*-coupling of shells terms and have a definite total *LS* term. The intermediate coupling scheme is only applied to the coupling of the target with the continuum electron. So, in order to guarantee that the given configuration represents *jK* or *jj* coupling, we should apply the corresponding unitary transformation to the coupling between the target and the continuum electron. The transformation coefficients are given by following formulas:

$$\begin{aligned} T_{LS,jK} &= <[(l_1 l_2)L,(s_1 s_2)S]J \,|\, (l_1 s_1) j_1, l_2]K, s_2, J> \\ &= (-1)^{s_2+J-l_2-j_1}[L,S,j_1,K]^{1/2} \begin{Bmatrix} L & s_1 & K \\ s_2 & J & S \end{Bmatrix} \begin{Bmatrix} l_2 & l_1 & L \\ s_1 & K & j_1 \end{Bmatrix}, \end{aligned} \tag{5.1}$$

$$T_{LS,jj} = [L, S, j_1, j_2]^{1/2} \begin{Bmatrix} l_1 & l_2 & L \\ s_1 & s_2 & S \\ j_1 & j_2 & J \end{Bmatrix} , \qquad (5.2)$$

Here the indices 1 and 2 denote the target and continuum orbital, respectively; *LS* is the total term of the given configuration; *J* is the total angular momentum for the partial wave, and [*L*] stands for (2*L*+1). The expansion coefficients for the continuum configurations in the files **cfg.nnn** are multiplied by the corresponding transformation coefficients, depending on the intermediate coupling scheme under consideration. This procedure allows us to run all further calculations for generating and diagonalizing the Hamiltonian matrix in the same way for all coupling schemes.

The most complicated point concerns the orthogonal constraints imposed on the continuum orbitals. The present package allows the user to run calculations with minimum number of orthogonality constraints, which don't lead to any compensation configurations. BSR_CONF3 supports only such type of calculations and don't include the fully orthogonal mode as in the original version. The algorithm for imposing the orthogonal constraints is also changed and contains the following points.

(a) All continuum orbitals should be orthogonal to the closed-shell orbitals that are common to all states. Such shells will also be referred to as core shells. Since these shells are closed, this does not lead to any additional (*N*+1)-electron bound terms in the close-coupling expansion in order to compensate these restrictions. The orthogonality of the scattering orbitals to closed-shell core functions is physically justified. It avoids unphysical low-lying *R*-matrix poles corresponding to the capture of the incident electron into these shells. Orthogonality to the closed core shell is automatically imposed in the present package, without any additional input parameters.

(b) All bound orbitals are divided on spectroscopic and corelated. Orthogonality constraints are required only to the spectroscopic orbitals. Spectroscopic (or physical) orbitals are defined by their occupation numbers. The list of spectroscopic orbitals for each target and perturber is defined byn BSR_PREP program and recorded in the file **target_orb**

(c) All additional orthogonality conditions should be indicated as **<*kl*|*nl*>=0.** These conditions can be placed in the input file **bsr_par** and they will be effective for all partial waves, or they can be placed at the end of the **cfg.nnn** file and then will only be effective for a given partial wave. If we introduce some orthogonal constraint, we should generally also introduce the relevant (*N*+1)-electron bound states in order to compensate for this constraint in the full functional space. 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.

(d) Even when we using the non-orthogonal mode, we cannot avoid orthogonal constraints completely. It is is related to the case when different channels can generate the same configurations. Consider, for example, electron scattering from neon, with channels $2p^6kl$ and $2p^5nlk$p. If the orbital kl has no orthogonal constrains, then this channel can generate the $2p^6nl$ states. The same states will also be generated by a second channel if the kp orbital is not orthogonal to the 2p orbital. Such situation may lead to the 'overloaded' overlap matrix in the generalized eigenvalue problem (2.17). As a result, the overlap matrix will no longer be positive definite, and this prevents its diagonalization. To avoid such a situation, the user should impose the orthogonality constraint $<kp|2p>=0$ in the given example, without any compensation configurations, because these configurations still can be generated in the first channel.

(e) The main idea of the present algorithm is to find all needed orthogonal constraints (and to avoid the situation in the previous example) is to generate all possible (N+1)-electron configurations if the continuum orbital is substituted by one of the spectroscopic orbital. Such procedure, of course, should follow all angular momentum restrictions. Here we consider only main physical configurations from the target expansions and take in to account also their mixing coefficients. The physical configurations are defined as consisting from the spectroscopic orbitals. If coefficient for the (N+1)-electron configuration becomes more than **c_comp** parameter, the orthogonality constraint is introduced for the channel orbital under consideration. The **c_comp** parameter is usually chosen close to unity; however, user may increase or decrease it in order to make the orthogonal constraints weaker or stronger, respectively. The bound-type calculations is advised to carry out with weaker orthogonality constrains. Note that the overlap matrix further analyzed in program BSR_MAT, based on the actual values of the overlap matrix elements. If these matrix elements are big, additional constraints may be applied.

(f) Another important remark concerns the cases when orthogonality constraints are 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–Ne scattering, we can use two different radial functions for the 2p orbital in the $2p^5$ and $2p^6$ configurations to account for the relaxation effects. These functions are different but very similar. In this case we should apply the $<kp|2p>=0$ condition only to one of these orbitals. Besides, the orthogonal procedure (…) for several constraints for one continuum orbital is working only for the orthogonal subset of bound orbitals. The list of orthogonal orbitals used in the orthogonal constraints (called as substitution orbitals) are define by BSR_PREP program and recorded in the **target_orb** file, along with physical bound orbitals.

5.2. Data files

bsr_par File type: formatted sequential input (optional).
Written by user.
Description: input parameters for given run.
Format: see section 3.2 and examples in section 14.

target File type: formatted sequential input.
Written by user.
Read and modified by programs BSR_PREP and BSR_CONF**.**
Description: contains list of target states and scattering channels.
Format: see section 4.3 and section 5.4 below.

target.bsw File type: unformatted sequential input.
Created by program BSR_PREP.
Read by programs BSR_CONF, BSR_MAT, BSR_HD**.**
Description: contains list of target one-electron orbitals.
Format: see section 4.3?..

target_orb File type: formatted sequential input.
Created by program BSR_PREP.
Read by program BSR_CONF**.**
Description: contains list of physical and substitution orbitals used to assign orthogonal conditions..
Format: see section 4.3?.

target_del File type: formatted sequential input (optional).
Created by user from file target.
Read by program BSR_CONF**.**
Description: indicate the channels which will be deleted from further consideration.
Format: see section 5.3?.

targ_nnn.c File type: formatted sequential input.
Created by program BSR_PREP.
Read by program BSR_CONF.
Description: contains configuration expansion for target state **nnn.**

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

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

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

pert_comp.c File type: formatted sequential input (optional, **debug** = 1).

Created by program BSR_CONF.
Read by user for additional information about orthogonal constraints.
Description: contains list configurations resulting from capture of the scattering orbitals to the bound target orbitals. This information is used to assign minimum orthogonal constraints.

5.3. Input parameters

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

min_ll [-1]	restriction on the minimum orbital angular momentum for the continuum orbitals (the default value of -1 means no restrictions)
max_ll [-1]	restriction on the maximum orbital angular momentum for the continuum orbitals (the default value of -1 means no restrictions)
max_LT [-1]	restriction on the maximum total orbital angular momentum for the partial waves in LS coupling (the default value of -1 means no restrictions).
max_ST [-1]	restriction on the maximum total spin for the partial waves in LS coupling (the default value of -1 means no restrictions).
c_comp [1.1]	tolerance for compensation configurations when we need to put orthogonal conditions.
< *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*. By default, all continuum orbitals are supposed to be non-orthogonal to the bound orbitals.
kort [0]	if **kort**=1, program reads from **cfg.nnn** file the orthogonal constraints specific for the given partial wave.
iread_targ [0]	if **iread_trag**=1, program is forced to read spectroscopic configurations from **targ_nnn.c** files.
debug [0]	if **debug**=1,2,3, program output more and more detailed information about assignment of the orthogonal constraints.

Additional input parameter, **coupling,** is reading from file **target**. This parameter has three possible values **LS**, **JK** and **JJ**, and defines the coupling for continuum states in the case of relativistic calculations. In this case, the input target states should be provided for each *J* level, with *J*-value being indicated in the first line of the corresponding *c*-file as ***2J = value***.

*5.4. Modification of file **target**.*

BSR_CONF modifies the file **target** by adding information about the close-coupling expansions generated. An example of such information is given in Fig.5.1. First, 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, for each channel in a separate line, we have the spectroscopic notation for the continuum orbital, its orbital momentum l, a pointer to the corresponding target state, № of the channel, a pointer to the last configuration for the given channel in the file **cfg.nnn**, and the value of K, if we use the jK intermediate coupling scheme. The spectroscopic notation for the continuum orbital is given only for the user's information. As additional information, the end of **target** 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.

5.5. Close-coupling configuration expansions **cfg.nnn**

One of the main purposes of the BSR_CONF program is to generate the close-coupling configuration expansions **cfg.nnn**. These are the text file, which contains the list of configurations in the spectroscopic notation. It is convinient for the small l and L values, which are tipical for the bound-states calculations, however, can take rather weird appearance for the high L values in case of scattering calculations. Based on the available character map, spectroscopic symbols are for L values are available for $L \leq 153$. Example of **cfg.nnn** file with added orthogonality constraints is given in **Fig.5.3.** At this stage, the user may add or remove any ortogonal constrain from the cfg-file by using text editor if needed.

The BSR_conf program also provide the option to remove individual channels. It is helpful for bound-state calculations where many channels provode very low contributions and it is important to simplify the resulting expansions for further using in the scattering calculations. In order to remove some cahnnels, user just copies the **target** file in the **target_del** file and put the zero in the configuration-pointer position for the deleted channel. Configuration pointer indicate the last configurations for the given channel and placed in the fifth column (see Fig.5.1). Then the re-run the program BSR_CONF create the **target** and cfg.nn files without indicated cahnnels.

```
-------------------------------------
channels:
-------------------------------------
  1.  001  nch =  22  nc =  5437    31
 kd4    2    2    1        263    0
 ks4    0    3    2        395    0
 kp4    1    5    3        713    0
 kp5    1    6    4        974    0
.................................
 kpf    1   35   22       5406    0
-------------------------------------
  2.  002  nch =  11  nc =  3065     0
 kp3    1    5    1        318    0
 kd3    2    9    2        600    0
 ks3    0   10    3        704    0
.................................
 kpa    1   35   11       3065    0
-------------------------------------
  3.  003  nch =  13  nc =  2655     0
 kp3    1    1    1        258    0
 kd3    2    7    2        491    0
 kp4    1    8    3        660    0
.................................
 kd9    2   36   13       2655    0
-------------------------------------
.................................
-------------------------------------
 64.  064  nch =  37  nc =  9689     0
 ku1   15    1    1        258    0
 kt1   14    5    2        576    0
 kv1   16    5    3        894    0
 kt2   14    7    4       1127    0
.................................
 kve   16   36   37       9689    0
-------------------------------------
max_ch  =       66
max_nc  =    15921
max_wf  =       90
```

Fig. 5.1. Channel information added to **target** by the program BSR_CONF.

```
           1S    -676.59058592
  1s  2s  2p  3s
 3p1( 6) 4s1( 1) ks4( 1)                                          0.99417882
 1S0 2S1 2S1 2S  1S
 3p1( 5) 7p1( 1) 7d1( 1) ks4( 1)                                  0.08990789
 2P1 2P1 2D1 2S1 3D  2S  1S
 3p1( 5) 7p1( 1) 7d1( 1) ks4( 1)                                  0.03520611
 2P1 2P1 2D1 2S1 1D  2S  1S
 3p1( 5) 7s2( 1) 7p1( 1) ks4( 1)                                 -0.03469795
 2P1 2S1 2P1 2S1 1P  2S  1S
 3p1( 5) 7s2( 1) 7p1( 1) ks4( 1)                                 -0.02552832
 2P1 2S1 2P1 2S1 3P  2S  1S
 3p1( 5) 7f1( 1) 8d1( 1) ks4( 1)                                  0.01189443
 2P1 2F1 2D1 2S1 1D  2S  1S
.....................................................................
 2P1 3P2 2P1 2P  1S
 3p1( 5) 7s3( 2) kp6( 1)                                          0.00496383
 2P1 1S0 2P1 2P  1S
 3p1( 6) 4p3( 1) kp6( 1)                                         -0.00359696
 1S0 2P1 2P1 2P  1S
*

Imposed orth. conditions:

< kp5| 3p1>=0
< kp6| 3p1>=0

Derived orth. conditions:

< ks5| 4s1>=0       targ_004     5s

< kd6| 3d2>=0       targ_005     3d

< kp6| 4p3>=0       targ_006     5p
```

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

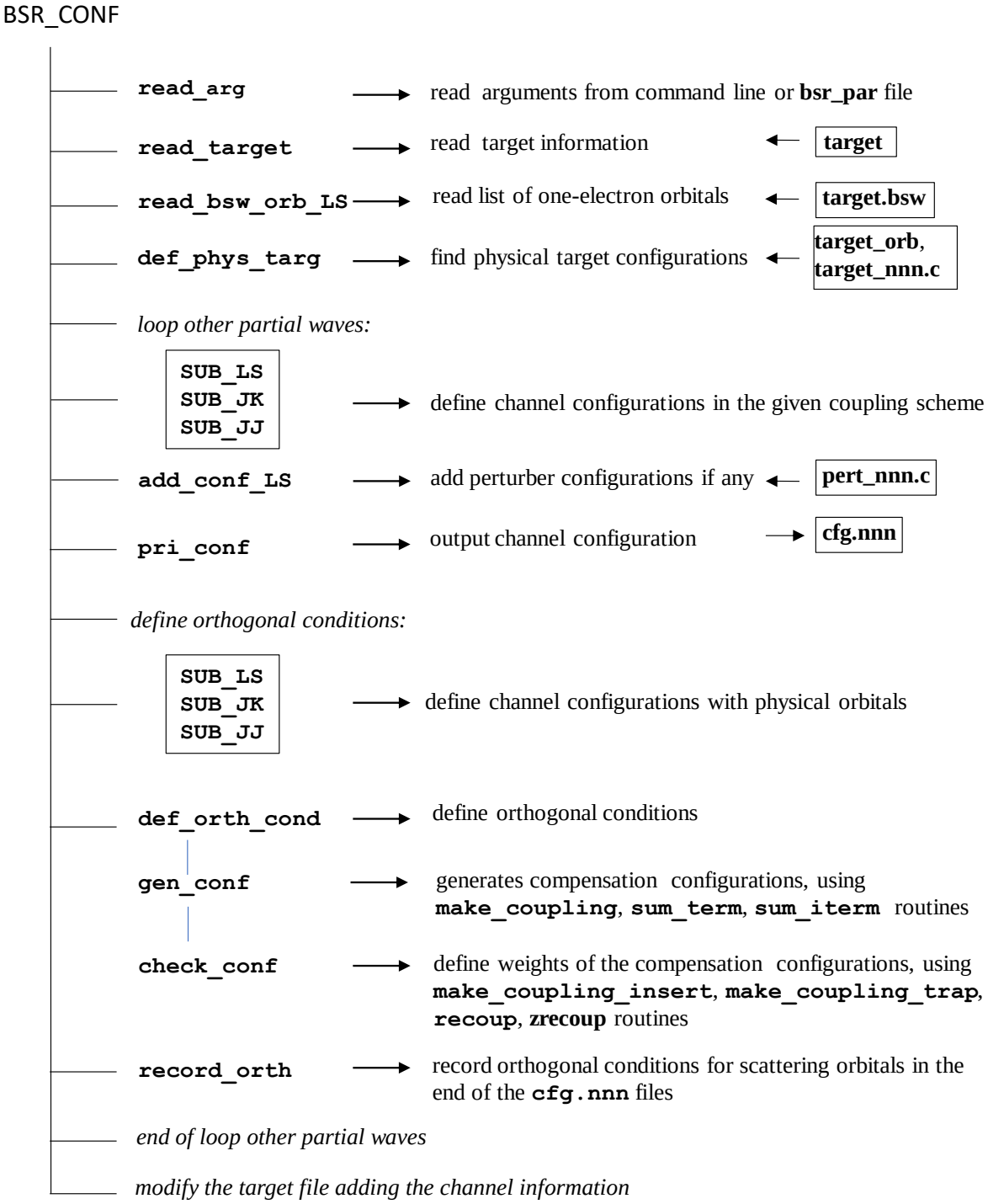

Fig.5.3. Block-scheme for the BSR_CONF program.