

14. Compilation and test runs

This section describes the main features concerning the distribution, installation and compilation, and running of the system. Several test runs are presented in order to verify that the system works as expected.

14.1. Compilation

The description below and the script files are prepared for a UNIX/LINUX user in the *c*-shell. Other users can use these files as a guide for compilation in a specific environment.

The source-code files for the BSR complex are distributed in the compressed **bsr_cpc.tar.gz** file. In order to uncompress the source files, the user may employ the system routines

```
gzip -d bsr_cpc.tar.gz
tar -xvf bsr_cpc.tar.
```

All source files are unpacked into the directory BSR_CPC. Each main program, libraries or utilities are presented by a single source file. The instructions for compiling the entire system are given in the **make_BSR** script file. Before running **make_BSR**, the user should check the assignment of two additional environmental variables in this file. First the user should define the name of the FORTRAN compiler and optimization level in the variable `FC`. For example:

```
setenv FC 'fort -fast'.
```

Then the user should indicate in the variable `LAPACK` the directory where the linker can find the LAPACK and BLAS libraries. Recall that these popular libraries were chosen to provide the linear-algebra calculations in the BSR complex, and the user should install the LAPACK and BLAS libraries before compiling the BSR complex. The LAPACK and BLAS libraries can be obtained without charge from the web-site <http://www.netlib.org/lapack/index.html>. Many FORTRAN compilers also provide these libraries as an additional option, so we decided not to provide them along with the present complex. If the user copies the `liblapack.a` and `libblas.a` files to the BSR_CPC directory, then the variable `LAPACK` is defined as

```
setenv LAPACK '-L. -llapack -lblas'.
```

Otherwise the corresponding path should be indicated after the parameter `-L`.

After setting the environment variables, the **make_BSR** file provides the recommended sequence of commands for compilation of the libraries and other BSR programs. The user can run this file as a command script file. As a result, all executable files for the main programs and utilities will be placed in the directory BSR_CPC. In order to use these programs in the test runs or in the production calculations,

the user should make the corresponding modification in the shell variable `path`. For example, as a command

```
set path = ($HOME/BSR_CPC $path)
```

or by putting

```
setenv PATH "$HOME/BSR_CPC: $PATH"
```

into the shell script file.

14.2. Test runs

We provide four test runs to illustrate the main modes of operation of the BSR complex, namely, *LS*-coupling scattering calculations, *JK*-coupling scattering calculations, photoionization, and bound-state calculations. In order to provide examples of realistic calculations and choices of appropriate physical models, the test runs were taken from published calculations for bound states [69] and electron scattering [70] from carbon atoms. Since these calculations were rather extensive, the test runs are given here in a simplified version. They only serve as a benchmark to ensure that the programs are running correctly, and as a template for users to construct their own runs.

Each scattering calculation begins with the choice and generation of the target states. In the framework of the BSR complex, this is the responsibility of the user. In principle, the user can employ any atomic-structure program, and then convert the results to the form acceptable in the BSR programs. The BSR programs themselves are adapted to use the target states from the MCHF complex of Froese Fischer [68]. Below we assume that all target states are generated in the framework of the MCHF approach, and we will show the run sequences in the different modes.

In the following test-run description we only discuss the input data files and the sequence of required commands. A complete set of input files and files with the final results is provided in the separate tar-file **bsr_test.tar.gz**, which contains the main directory `BSR_TEST` and four subdirectories `C_LS`, `C_JK`, `C_PHOT`, `C_BOUND` for each test run described below. Each subdirectory contains the file **run_script**, with a short description of the test and a required sequence of commands. Note that each target state is represented by a pair of *c*- and *frm*-files. The radial functions are given in the ASCII *frm*-file instead of unformatted *w*- or *bsw*-files, in order to simplify the test runs on different computer platforms, which may use a different default structure for unformatted files. As a consequence, the **run_script** file contains first a rather long sequence of preliminary commands for transformation of ASCII *frm*-files to the unformatted *bsw*-files required by the BSR complex. Each test-run subdirectory contains the file with the correct final

results. These files have the extension *.test*. In the end of each **run_script** file we give information what files with final results should be compared in order to verify that the system works correctly.

14.3. *LS-coupling test run*

This test corresponds to low-energy elastic and inelastic electron scattering from carbon atoms. The close-coupling expansion includes the seven lowest states of neutral carbon with configurations $1s^2 2s^2 2p^2$, $1s^2 2s 2p^3$, and $1s^2 2s^2 2p 3s$. These states exhibit a strong term-dependence of the one-electron orbitals, and it is difficult to obtain accurate wavefunctions with only one orthogonal set of orbitals. In order to illustrate the advantage of the present complex, the atomic wavefunctions for odd-parity and even-parity states were generated using two different non-orthogonal sets of orbitals. This allows us to obtain fairly accurate target wavefunctions with relatively small configuration expansions. The results of the MCHF calculations are presented in the *c*- and *w*-files, for the configuration expansions and the one-electron radial functions, respectively. These files are the input for the present BSR calculations. The *c*-file for the carbon ground state is given in section 12.5.

The first step of the BSR calculations concerns the choice of the *B*-spline knot sequence. The user should prepare the **knot.dat** file, indicating the *R*-matrix radius and the maximum size of subintervals, as discussed in section 12.2. (see also figure 12.2). The *w*-files are then transformed to *bsw*-files with the program-utility **w_bsr** (see section 13). At this stage the user can check the *R*-matrix radius by considering the value of the one-electron orbitals at the boundary. It is recommended that all orbitals should at least have fallen to magnitudes $< 10^{-3}$. Recall that the value of *R*-matrix radius also automatically defines the number of *B*-splines. Since the time of the calculations is approximately proportional to the square of the number of *B*-splines, the user should choose the minimal possible *R*-matrix radius with a minimum number of *B*-splines.

The next step involves the preparation of the file **target**, with the specification of the scattering model, as shown below.

```

e + C
-----
coupling = LS      !   LS coupling
nz = 6            !   nuclear charge
nelc = 6           !   number of electrons
-----
ntarg = 7          !   number of target states
-----
2p2_3P.c
2p2_1D.c
2p2_1S.c
2p3_5So.c
2p3s_3Po.c
```

```

2p3s_1Po.c
2p3_3Do.c
-----
nlsp =    12          !  number of partial waves
-----
2Se      0    2    1    no
4Se      0    4    1    no
2So      0    2   -1    no
4So      0    4   -1    p_4So.c
2Pe      1    2    1    no
4Pe      1    4    1    no
2Po      1    2   -1    p_2Po.c
4Po      1    4   -1    no
2De      2    2    1    no
4De      2    4    1    no
2Do      2    2   -1    p_2Do.c
4Do      2    4   -1    no
-----

```

The test run contains 12 partial waves, and for three partial waves there are additional *c*-files. These provide an example of (*N*+1)-electron configurations, which may be added to the close-coupling expansion to improve the description of short-range correlations. In the present example, the (*N*+1)-electron configurations were added for a more accurate description of the lowest resonance states with configuration $2p^3$. The orbitals in these files were optimized in separate MCHF calculations for each given term of the $2p^3$ configuration, and they are not orthogonal to the target orbitals. These additional multi-configuration expansions for the $2p^3$ states may be treated in two ways. First, they may be completely included in the close-coupling expansion, in which case we need to impose the orthogonality constraint for the corresponding continuum orbital according to $\langle kp|3p \rangle = 0$. Alternatively, we may remove from these perturber files the main configuration $2p^3$, leaving only the correlation configurations. We chose this way in the test run.

The last step in preparing the input data concerns the **bsr_par** file. Its parameters will control the specific run. This test run has only a few parameters that differ from the default values, namely:

```

# bsr_conf

< 2s| ks>=0

# bsr_hd:

itype = 0
idiag = 2
Emax = -0.1

```

The above orthogonality condition requires that all *ks* continuum orbitals should be orthogonal to the 2s orbital. By default, the continuum orbitals are not orthogonal to the bound target orbitals, but it is very important to keep orthogonality to the closed subshell as discussed in section 5.1. Orthogonality to the

closed core shells (here the $1s$ orbital) is imposed automatically. The orthogonality of the ks orbital to the $2s$ orbitals in the channel $2s^2 2p^2 ks$ is also imposed automatically by the BSR_CONF program. However, it is also important to keep the ks orbital orthogonal in all other channels (such as $2s 2p^3 ks$); otherwise different channels may generate the same $(N+1)$ -electron states ($2s^2 2p^3$ states in the present example). Note that no additional $(N+1)$ -electron states are used in this case to compensate for the orthogonality conditions.

In this test we also chose the two-step diagonalization procedure (`idiag=2`, see section 8.1). It is more effective, because it reduces the size of the interaction matrix to be diagonalized.

After preparing the target and input files, the run of the BSR programs is described by the following script file:

```
bsr_prep
bsr_conf
bsr_breit klsp1=1 klsp2=12
bsr_mat klsp1=1 klsp2=12
bsr_hd klsp1=1 klsp2=12
sum_hh 1 12
```

The result of this run will be the H.DAT file, which may be used in subsequent cross-section calculations with some external-region program. If additional partial waves are needed (or we want to recalculate some partial waves), we may run additional calculations only for specific partial waves and then again the **sum_hh** utility to obtain the combined H.DAT file. The user may control the calculations using the information from the BSR_MAT and BSR_HD programs in the corresponding log-fails. For example, the BSR_MAT program compares the input and the calculated target energies. If they do not agree within some tolerance, it means that the input *c*- or *w*-files contain errors. If the BSR_HD program fails to diagonalize the interaction matrix, this means that the total overlap matrix is not positively definite. This may indicate that not all important orthogonal constraints are imposed, or that the input target states are not orthogonal or not normalized. In order to verify that the system works correctly, user should only compare the final *R*-matrix eigenvalues given by system in file **sum_hh.log** and provided in file **sum_hh.test** for comparison. If these final eigenvalues do not agree with each other, we suggest to check all *log*-files for possible warnings.

14.4. *jK*-coupling test run

This test corresponds to low-energy elastic and inelastic scattering of electron by atomic carbon in the same approximation as the above *LS*-coupling test run, but with the inclusion of relativistic corrections.

The **target** file now has the following structure:

```

e + C
-----
coupling = JK      !   coupling
nz = 6            !   nuclear charge
nelc = 6          !   number of electrons
-----
ntarg = 13        !   number of target states
-----
2p2_1D2.c
2p2_1S0.c
2p2_3P0.c
2p2_3P1.c
2p2_3P2.c
2p3s_1Po1.c
2p3s_3Po0.c
2p3s_3Po1.c
2p3s_3Po2.c
2p3_3Do1.c
2p3_3Do2.c
2p3_3Do3.c
2p3_5So2.c
-----
nlsp = 12         !   number of partial waves
-----
1+   1   0   1   no
1-   1   0  -1  p_1o.c
3+   3   0   1   no
3-   3   0  -1  p_3o.c
5+   5   0   1   no
5-   5   0  -1  p_5o.c
7+   7   0   1   no
7-   7   0  -1  no
9+   9   0   1   no
9-   9   0  -1  no
11+  11   0   1   no
11-  11   0  -1  no
-----

```

The main difference from the *LS*-case is that target *c*-files are now provided for each *J*-level. In the MCHF package, the Breit-Pauli corrections are included in the framework of the CI calculations, and the solutions are recorded in the *j*-files, which contain the configuration expansions for all *J*-values. In order to obtain the *c*-file for a specific *J*-level, the user may use the utility **cfile** (see section 13). For example, the command

```
cfile even.j 1 4 2p2_3P2.c 0.0000001
```

will generate the *c*-file for the state $2s^22p^2\ ^3P_2$, provided that CI results are recorded in the file *even.j*.

The **bsr_par** file contains only one additional parameter, **mrel**=2, indicating that the Hamiltonian now includes also all one-electron Breit-Pauli terms. The sequence of programs in the script file is the same as in the *LS* test run, with one correction for the BSR_BREIT program, namely:

```
bsr_breit oper=1111000 klsp1=1 klsp2=12
```

The additional parameter **oper** indicates that the calculations of the angular coefficients also include the spin-orbit interaction. Recall that the angular coefficients are recorded in the databank files **int_bnk.nnn**, which can be used in repeated or corrected calculations.

14.5. Photoionization calculations

As an example of a photoionization calculation, we will consider the photodetachment of the negative carbon ion from its ground state $2p^3\ ^4S^o$. The final term can be only 4P , and hence the **target** file only contains expansions for two partial waves, corresponding to the initial bound and the final continuum channels:

```

hv + C-
-----
coupling = LS      !   LS coupling
nz = 6             !   nuclear charge
nelc = 6           !   number of electrons
-----
ntarg = 7          !   number of target states
-----
2p2_3P.c
2p2_1D.c
2p2_1S.c
2p3_5So.c
2p3s_3Po.c
2p3s_1Po.c
2p3_3Do.c
-----
nlsp =      2      !   number of partial waves
-----
4So   0   4   -1   p_4So.c
4Pe   1   4    1   no
-----

```

Note that the description of the initial state through the close-coupling expansion is optional, because the BSR complex also allows to represent the initial state as a real bound state obtained separately from independent MCHF calculations. A typical run of the BSR program for the case of photoionization calculations is described by the following script file:

```

bsr_prep
bsr_conf

bsr_breit klsp1=1
bsr_mat klsp1=1
bsr_hd klsp1=1 itype=-1 msol=1

bsr_breit klsp1=2
bsr_mat klsp1=2
bsr_hd klsp1=2 itype=1 idiag=1

```

```

mult cfg.001 cfg.002 E1

bsr_dmat cfg.001 cfg.002 b p

bsr_phot

```

We generate the bound state in the first partial wave, and the results are stored in the file **bound.001**. Then we generate the **h.002** and **rsol.002** files in second partial wave, with the latter file containing all R -matrix solutions in the internal region. In comparison to scattering calculations, described in section 14.1 above, we have two additional runs in order to generate the dipole matrix. The MULT program generates the databank for the dipole angular coefficients, and the BSR_DMAT program generates the dipole matrix file **d.002** itself. The **h.002** and **d.002** files are then used by the BSR_PHOT program for the photoionization calculations. The parameters and input energies are provided in the file **bsr_phot.inp** (see section 11). The user can run the BSR_PHOT program several times for different energy intervals and with different energy steps. The results are accumulated in the files **photo.002** and **bsr_photo.002**. The file **photo.002** contains the total photoionization cross sections and can be used for a quick check of calculations. The more detailed data recorded in the file **bsr_photo.002** may then be used for the calculation of channel cross sections or asymmetry parameters (see the description of the program utility **photo_tab** in section 13). If we have several final continuum states with different total terms, the calculations are supposed to be carried out for each partial wave separately, and the total cross sections or asymmetry parameters may be generated again with the utility **photo_tab**. In order to verify that the system works correctly, user should only compare the final photoionization cross sections in file **photo.002** with data provided in file **photo.test**.

14.6. Bound-state calculations

This test is a simplified version of the calculation of oscillator strengths in carbon, presented in detail in [69]. The task was to obtain the oscillator strengths for transitions from the ground state to the Rydberg states $2s^2 2pns$ and $2s^2 2pnd$ for intermediate values of $n = 4-8$. In this range traditional methods, such as MCHF or the quantum defect method, are difficult to apply. The Rydberg series $2s^2 2pns$ and $2s^2 2pnd$ strongly interact with each other through both electrostatic and spin-orbit interaction. Besides, there are also strong corrections from core-valence and inner-core correlations. The test presents the simplest model of describing the $2s^2 2pns$ and $2s^2 2pnd$ series through the close-coupling expansion with five target states: $2s^2 2p \ ^2P^o$, $2s2p^2 \ ^2S$, $\ ^2P$, $\ ^2D$, and $2p^3 \ ^2P^o$. The $2s2p^2$ states are included to simulate the core-valence

correlation due to the strong 2s-2p transition, whereas the inclusion of the $2p^3$ state accounts, to some extent, for the inner-core correlation. The target file for this case has the following form:

```

C - bound
-----
coupling = LS      !   LS coupling
nz = 6            !   nuclear charge
nelc = 5          !   number of electrons
-----
ntarg = 5          !   number of target states
-----
2s2_2p_2Po.c
2s_2p2_2D.c
2s_2p2_2P.c
2s_2p2_2S.c
2p3_2Po.c
-----
nlsp = 4          !   number of partial waves
-----
0-  0  0  -1  p0.c
1-  2  0  -1  p1.c
2-  4  0  -1  p2.c
3-  6  0  -1  p3.c
-----

```

All target states in this example are represented by simple single-configuration wavefunctions, but with term-dependent 2s and 2p one-electron orbitals. We are interested in the final J -level splitting, so the partial waves are constructed for total J -values from 0 to 3. Note that the target states here are represented in LS -coupling. Hence this test is an example of using the LSJ coupling in the framework of the present BSR complex. The **bsr_par** input file for this test has the form:

```

# bsr_conf

< 2s| ks>=0
< 2p| kp>=0

max_LT = 7
max_ST = 3
max_ll = 3

# bsr_mat:

mrel = 2

# bsr_hd:

itype = -1
jmvc = 2
idiag = 2
Emax = -0.1
msol = 30
n_max= 12
qd_0 = 1.5
qd_1 = 0.6
qd_2 = 0.2

```

```
qd_3 = 0.1
```

The most important difference from the previous examples is the orthogonality condition imposed on both the 2s and 2p bound orbitals. It is due to the fact that in this case the same $2s2p^3$ bound states can be generated in different channels. This, in turn, may lead to an ‘overloading’ of the total overlap matrix as discussed in chapter 5. The corresponding $2s2p^3$ states are included in the close-coupling expansion through the additional perturber expansion p0.c, p1.c, etc.. For a more accurate representation of the $2s2p^3$ states, their wavefunctions were also obtained with a full re-optimization of the 2s and 2p orbitals. The **mrel** parameter in the above list indicates that the interaction matrix will include all one-electron Breit-Pauli terms, and **itype** = -1 means that it is a bound-state calculation with zero boundary condition. In order to obtain the states with high principal quantum numbers we use a border radius of 300 a.u. However, this does not lead to an increase in the number of *B*-splines, because we can use the exponential radial grid (see subsection 12.2) for a bound calculation. The input parameters also contain the values for the expected quantum defect in the $2s^22pnl$ series. These parameters have no effect on the actual calculations; they are used only for a more appropriate labeling of the resulting states.

The run of the BSR programs in this test is described by the following script file:

```
bsr_prep
bsr_conf

bsr_breit klsp1=1 klsp2=4 oper=1111000
bsr_mat klsp1=1 klsp2=4
bsr_hd klsp1=1 klsp2=4

bound_tab

mult 2s2_2p2_3P.c cfg.001 E1
mult 2s2_2p2_3P.c cfg.002 E1
mult 2s2_2p2_3P.c cfg.003 E1
mult 2s2_2p2_3P.c cfg.004 E1

bsr_dmat 2s2_2p2_3P0.c cfg.002 c b
bsr_dmat 2s2_2p2_3P1.c cfg.001 c b
bsr_dmat 2s2_2p2_3P1.c cfg.002 c b
bsr_dmat 2s2_2p2_3P1.c cfg.003 c b
bsr_dmat 2s2_2p2_3P2.c cfg.002 c b
bsr_dmat 2s2_2p2_3P2.c cfg.003 c b
bsr_dmat 2s2_2p2_3P2.c cfg.004 c b
```

After running the BSR_HD program we obtained the set of **bound.nnn** files with solutions for a given *J*-value. The final list of bound states in the **bound.nnn** files may be obtained with the program utility **bound_tab**. The initial states $2s^22p^2\ ^3P_{0,1,2}$ in this test are supposed to have been obtained in separate configuration-interaction calculations, with a list of configurations being placed in the file **2s2_2p2_3P.c** and the *J*-dependent expansion coefficients in the files **2s2_2p2_3Pj.c**. First, we run the MULT program

to obtain all the required dipole angular coefficients in the databank **mult_bnk**. Then we run the program BSR_DMAT to calculate the oscillator strengths for each combination of initial and final J -value. Finally, the oscillator strengths from different runs are accumulated in the file **zf_res**. File **zf_res.test** contains the correct results provided for comparison.