8. Program BSR_HD (version 3)

8.1. Structure and data flow

This stage of the BSR complex deals with the final diagonalization of the Hamiltonian matrix. The eigenproblem (2.17) in the *B*-spline basis leads to a generalized eigenvalue problem (7.1). For the solution of this problem, we use the standard routines from the LAPACK library (http://www.netlib.org/lapack/index.html). If desired, the user can replace these routines with others.

The block diagram of the program BSR_HD, along with the data flow, is given in Fig.8.1. First, the program reads the data common to all partial waves. This includes the parameters of the calculation (routine **read_arg**), the *B*-spline parameters (**define_spline**), and the description of the target states **(read_target**). Then the program executes fully independent calculations for each partial wave under consideration (**sub1_hd**).

The main task of the BSR_HD is the diagonalization of the interaction matrix. The using the B-spline basis requires preliminary modification of the interaction matrix due to the boundary conditions. Since the first *B*-spline is the only spline with nonzero value at $r = 0$, the zero boundary condition at $r = 0$ is imposed by removing the first *B*-spline from all expansions. In the code it means that we remove in the Hamiltonian matrix all rows and columns related to the first spline. A further modification is related to the known dependence proportional to r^{l+1} for the one-electron radial functions at small r. Recall that the first k_s splines $B_i^{k_s}$ behave as r^{i-1} at small r. Hence we can guarantee the r^{l+1} behavior of the one-electron radial functions simply by removing the first $(l+1)$ splines from expansions. This can be done only for orbitals with angular momentum $l \leq k_s-2$. For the remaining orbitals the r^{l+1} behavior is only approximately reproduced by a suitable combination of the first splines. In bound-state calculations with the input parameter **itype** = –1, we additionally remove the last splines from the *B*-spline expansion, the only spline which has non-zero value at the border $r = a$.

When we diagonalize the Hamiltonian matrix in the *B*-spline basis, we get a very wide spectrum of eigenvalues, which spread from threshold energies up to big positive energies on the absolute energy scale. This is related to the effective completeness of the *B*-spline basis, and hence no Buttle correction is required in this case. On the other hand, the *B*-spline expansions are comparatively large, and in practical calculations it is not reasonable to treat all eigenvalues on equal footing. We may disregard the high-lying solutions as they provide only a small contribution to the final *R*-matrix. For this reason, we divide the diagonalization process on several steps. We first diagonalize only the diagonal blocks in the matrix (6.5) which are connected to the given scattering channels (using the LAPACK routine DSYGV) . It provides us with new one-electron basis. Then the interaction matrix is transformed to this new basis, and in the

final expansion we retain only those one-electron solutions with energies below some value **Edmax.** This parameter is an input parameter and defined by the user on the basis of the physical model under consideration. Then the new simplified matrix is diagonalized, and the resulting solutions are very close to the solution of the full matrix for energies below **Edmax**. Such simplified treatment can reduce the size of the interaction matrix by up to factor of 2, and hence considerably reduce the computational time.

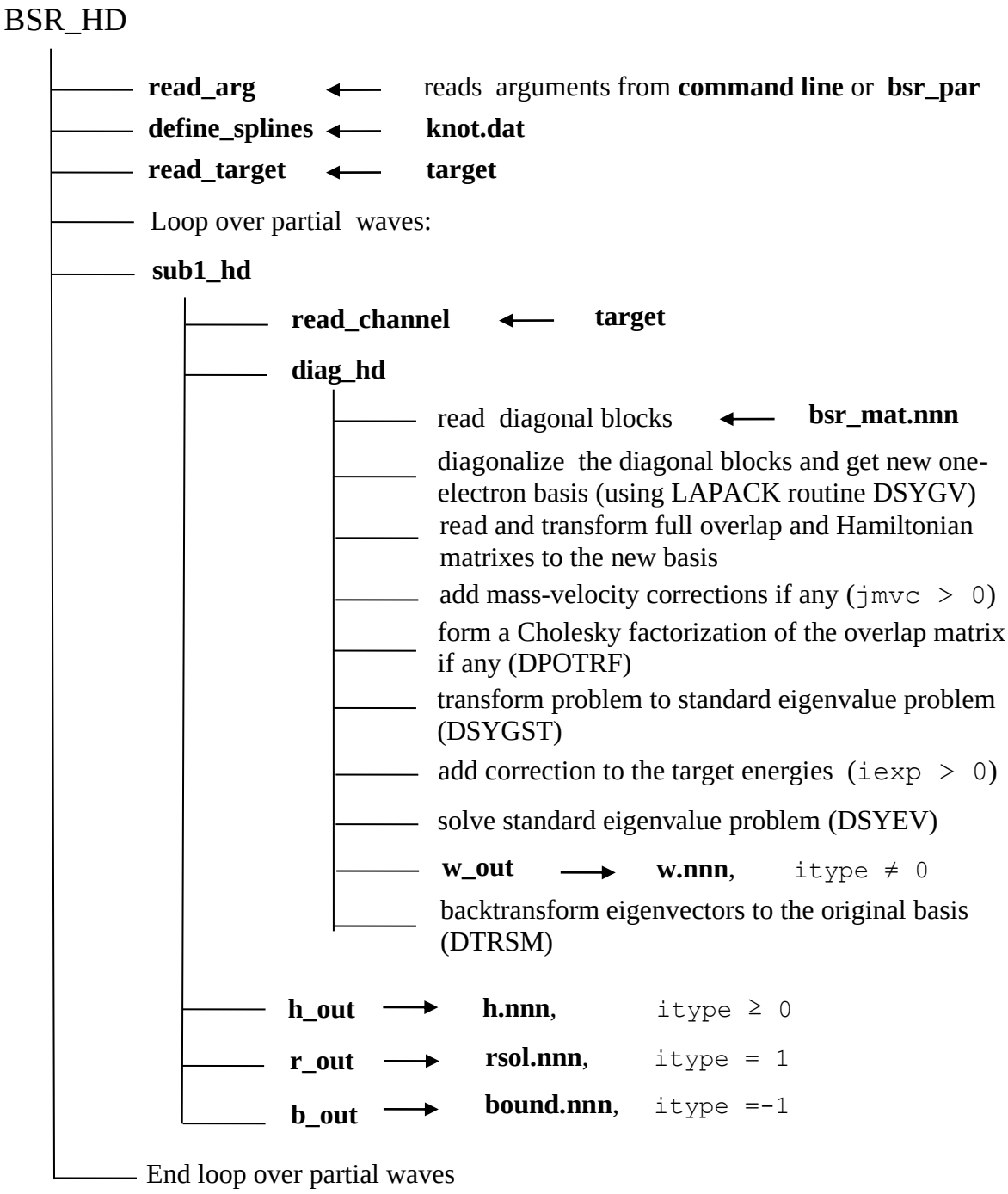

Fig. 8.1. Block diagram for the program BSR_HD and data flow (see text).

The two-step diagonalization of the interaction matrix also allows us to introduce additional corrections at the level of first-order perturbation theory. When we have zero-order solutions, $\boldsymbol{c}_i$, obtained from the diagonalization of the diagonal blocks, we can add to the Hamiltonian matrix any perturbation as $\langle \boldsymbol{c}_{i'} | V | \boldsymbol{c}_i \rangle$, where V is the perturbation potential. Such a possibility is realized in the present implementation in the case of the mass-velocity correction for scattering orbitals, which is controlled by the parameter **jmvc**. Note that the direct inclusion of the mass-velocity operator (2.41) in the Hamiltonian matrix leads to unphysical solutions with very low energies.

Another very important correction is the introduction the experimental target energies instead of theoretical values. This option is controlled by the input parameter **iexp**. If this parameter is greater 0, the program reads experimental threshold energies from the file **thresholds** and made the corresponding modification of the interaction matrix. This modification is not as straightforward as in case of orthogonal orbitals; in general, it needs an additional diagonalization procedure described in the original publication (Zatsarinny 2006), in particular, find the square root of the overlap matrix. In the present implementation we avoid it by transforming the general eigenvalue problem to standard one. To do that, we obtain the Cholesky factorization of the overlap matrix **S** in the form:

$$\mathbf{S} = \mathbf{L}\,\mathbf{L}^T$$

where **L** is a low triangular matrix. It is performed with LAPACK routine DPOTRF. Then we can transform our problem to standard eigenvalue problem with Hamiltonian matrix

$$\mathbf{H}' = \mathbf{L}^{-1}\mathbf{H}(\mathbf{L}^T)^{-1}.$$

It is performed with LAPACK routine DSYGST. Now we may introduce the correction to the target energies in the standard way, i.e. by modifying the corresponding diagonal element of matrix **H'**.

Next step is the diagonalization of the standard eigenvalue problem

$$\mathbf{H}'\psi' = E\psi',$$

which is performed with the LAPACK routine DSYEV. At this point we also can find the composition of the R-matrix or bound solutions, subroutine **w_out.** It may be used in further analysis of solutions. Finally, we make back transformation of eigenvectors to the original one-electron basis as

$$\psi = (L^{-1})^T\psi',$$

using the LAPACK routine DSYGST.

After diagonalizing the Hamiltonian matrix, the results are treated in different ways, depending on the type of calculation specified by the input parameter **itype**, namely, scattering calculations, **itype**=0, photoionization calculations, **itype**=1, or bound-state calculations, **itype**= –1. For scattering calculations, including photoionization, the program first generates the corresponding *H* file in the routine **h_out**. The

output *H* file has a standard structure, first adopted in the Belfast *R*-matrix code. For a detailed description of this file, see section 8.4. Note that the surface amplitudes, which are needed to generate the final *R*-matrix, are simply defined in the present implementation by the expansion coefficients of the last *B*-spline, which is the only spline with nonzero value at the boundary. Another important difference from the earlier codes is that we do not provide any Buttle correction in the *H* file. The user should remember that the resulting *H* file can only be used with the parameter **ibuttle** = 0 in further calculations, for example, when running the FARM program.

In photoionization calculations, the program also records the full set of inner-region solutions in file **rsol.nnn**. These data are then used by program BSR_DMAT to generate the corresponding dipole matrix.

The last option in the BSR_HD program is a bound-state calculation. The resulting bound-state solutions are generated by the routine **b_out**. In general, we can get a huge number of solutions, not all of which are physical or even helpful in further calculations. A set of input parameters is therefore introduced in order to avoid the generation of a large number of output **bound.nnn** files with excess information. The parameter **msol** limits the total number of output solutions, the parameters **Emin** and **Emax** restrict the solutions according to their energies, the parameter **it_max** provides the highest target threshold for output solutions.

The routine **b_out** also tries to provide the spectroscopic assignment for the output solutions. To do this, the weights of different channels and configurations are needed. They are recoded in the **w.nnn** files.

8.3. Data files

The following is a summary of input and output data files of BSR_HD. The unit numbers and file names are defined within the program. Some files have a extension that depends upon the partial wave under consideration.

bsr_par
File type: formatted sequential input.
Prepared by user.
Read by routine **read_arg.**
Description: input parameters for given run if they differ from default values.

target
File type: formatted sequential input.
Prepared by user and modified by BSR_PREP and BSR_CONF programs.
Read by routine **read_target.**
Description: contains the description of the target states and scattering channels.

knot.dat
File type: formatted sequential input.
Prepared by user (only main parameters).
Read and modified by routine **define_grid** from the BSPLINE library.
Description: input parameters that define the B-spline grid.

cfg.nnn
File type: formatted sequential input

Created by program BSR_CONF.
Read by routine **read_conf**, used in routine **w_out** for solution notations
Description: list of configurations in spectroscopic notation for the partial wave **nnn**.

bsr_mat.nnn File type: unformatted sequential input.
Created by program BSR_MAT.
Read by routine **diag_hd**.
Description: Overlap/Hamiltonian matrixes and asymptotic coefficients.

thresholds File type: formatted sequential input.
Written by user.
Read by routine **read_arg.**
Description: experimental target energies, optional.

bsr_hd.nnn File type: formatted sequential output.
Written by program BSR_HD.
Read by user.
Description: running information.

h.nnn File type: unformatted sequential output.
Created by routine **h_out.**
Read by program FARM, STG4, BSR_PHOT or other programs that deal with the outer region**.**
Description: diagonalized Hamiltonian matrix data for given partial wave.

bound.nnn File type: formatted sequential output.
Created by routine **b_out.**
Read by program BSR_DMAT and utilities **bound_tab, bound_bsw**.
Description: bound-state solutions for given partial wave in the *B*-spline close-coupling representation (8.2).

ubound.nnn File type: unformatted sequential output.
Created by routine **b_out.**
Read by program BSR_DMAT and utilities **bound_tab, bound_bsw**.
Description: copy of **bound.nnn** in unformatted style.

rsol.nnn File type: unformatted sequential output.
Created by routine **r_out.**
Read by program BSR_DMAT.
Description: full set of *R*-matrix solutions in the inner region.

w.nnn File type: unformatted sequential output.
Created by routine **w_out.**
Read by routine **b_out** and program BSR_PHOT.
Description: weights of different channels in the inner-region solutions. Used for classification and sorting of bound-state solutions.

8.4. Input data

Input data can be provided in the command line or in the input file **bsr_par** (the data from the command line overwrite the data from the input file). Below we describe those data from **bsr_par** which are read by the program **bsr_hd** (all input parameters are read only in the routine **read_arg**). All data have the default values indicated in the brackets. All default values, along with unit numbers and default file names, are placed in the module **bsr_hd**.

klsp1 [1]	first partial wave under consideration
klsp2 [klsp1]	last partial wave under consideration
itype [0]	mode of calculations: **`itype= 0`** – scattering calculations; the **h.nnn** file is generated **`itype= 1`** – photoionization calculations; **h.nnn** and **rsol.nnn** files are generated **`itype=-1`** – bound-state calculations; **bound.nnn** file is generated
iexp [0]	controls the treatment of experimental thresholds
jmvc [0]	pointer to the inclusion of the mass-velocity term as a first-order perturbation
Edmax [0]	if >0, only R-matrix solutions with E < Emax are considered
msol [100]	if > 0, restricts the number of bound-state solutions for output in **bound.nnn** file
Emax [0]	if > 0, only solutions with E < Emax will be output
cwt [0.0]	if > 0, additional information about the channel composition of solutions will be output. Only the components with coefficients > **cwt** will be shown
iwt [0]	if > 0, the **w.nnn** file will be saved, otherwise it will be deleted
itrm [0]	if > 0, the additional output for time-dependence R-matrix calculations is provided in **hs.nnn** files
ihout [0]	if > 0, the energy order of R-matrix solutions in **h.nnn** is change from descending to ascending
iub [-1]	if = 0, both **bound.nnn** and **ubound.nnn** are created if < 0, only **bound.nnn** if > 0, only **ubound.nnn**
eps_o [0.5]	tolerance for overlap matrix elements to provide warning messages
debug [0]	if > 0, additional information in **bsr_hd.nnn** is provided

The corrected (experimental) target energies, if any, are given in the file **thresholds** as one column. By default, energies are given in atomic units (absolute values). There is also option to provide the relative energies in eV or cm units, by indicating **unit=eV** or **unit=cm** on the separate line in the end of file.

8.5. Structure of the H file

We keep the same structure for the *H* files as in the Belfast *R*-matrix code [4]. The **h.nnn** files from the BSR_HD program contain information only for one partial wave. In order to generate the full H.DAT file, the user should run the utility program SUM_HH (see section 13). Below is a summary of the output records in the **h.nnn** files.

1. `nelc,nz,lrange,km,ntarg,RA,BSTO`

   `nelc` – number of electrons in the target
   `nz` – atomic number
   `lrange` – maximum value of (l+1) for the continuum orbitals
   `km` – maximum multipole index for the asymptotic coefficients.
   `RA` – *R*-matrix radius
   `BSTO {0.0}` – the value of the logarithmic derivative to be imposed on the continuum orbitals.
   The variables `lrange` and `BSTO` were included only for consistency with earlier versions.

2. `etarg (1:ntarg)`
3. `ltarg (1:ntarg)`
4. `starg (1:ntarg),ptarg(1:ntarg)`

   `etarg` – target energies, in a.u.
   `ltarg` – total L for target states (2J in case of intermediate coupling).
   `starg` – total S for target states (0 in case of intermediate coupling).
   `ptarg` – parity of the target states.

5. `(0.d0,0.d0,0.d0,i=1,lrange)`

   This record in the standard H.DAT file should contain the Buttle corrections, which are not used in the present implementation.

6. `lpar,spar,ipar,nch,nhm,more`

   `lpar` – total L for given partial wave (2J in case of intermediate coupling)
   `spar` – total S for given partial wave (0 in case of intermediate coupling)
   `ipar` – total parity for given partial wave
   `nch` – number of scattering channels
   `nhm` – number of inner-region solutions
   `more {0}` – indicates no other information, except for the given partial wave

7. `NCONAT(1:ntarg)` – indicates the number of channels associated with each target state.
8. `lch(1:nch)` – channel orbitals angular momenta
9. `CF(1:nch,1:nch,1:mk)` – matrix of asymptotic coefficients (2.26).
10. `eval(1:nhm)` – *R*-matrix poles in descending order (values E_k in eq. (2.20)).
11. `wmat(1:nch,1:nhm)` – corresponding surface amplitudes (values w_{ik} in eq. (2.20)).

8.6. Output of the R-matrix solutions - **rsol.nnn** files

```
1. nhm,khm,nch,ncp,ns
2. eval(1:khm)
3. a(1:nhm,:)
```
repeat record 3 `khm` times

`nhm` – dimension of the Hamiltonian matrix in B-spline basis.
`khm` – number of solutions.
`nch` – number of channels.
`npert` – number of (N+1) configurations.
`ns` – number of B-splines for each channel.
`eval(1:khm)` – R-matrix eigenvalues.
`a(1:nhm,1:khm)` – corresponding eigenvectors in B-spline representation.

8.7. *Output of bound solutions* - **bound.nnn** files

```
1. ns,nch,npert,nhm,nbound,L,S,P
2. i,LABEL
3. eval(i),Ebind(i),eff_n(i),n_eff(1:nch)
4. a(1:nhm,i)
```
repeat records 2-4 for each bound state.

`ns` – number of B-splines.
`nch` – number of channels.
`npert` – number of (N+1)-electron configurations (perturber)
`nhm` – dimension of solution vector in the B-spline representation.
`LABEL` – spectroscopic notation for solution i.
`eval` – eigen energy in a.u.
`Ebind` – binding energy.
`eff_n` – effective principal number.
`a` – solution vector in the B-spline representation, see Eq. (7.2).

8.8. *Output of weights* - the **w.nnn** files.

```
1. nch, npert, khm
2. W(1:nch+npert)
```
repeat record 2 for each solution

`nch` – number of channels.
`npert` – number of (N+1)-electron perturbers.
`khm` – number of solutions.
`W` – array of weights for each channel and (N+1)-electron pertuber.

8.9. *Additional output for TRM calcultions* - the **hx.nnn** files.

```
1. nch, khm
2. nx, x(1:nx)
3. WX(1:nch,1:nx,1:khm)
```

`nch`	–	number of channels.
`khm`	–	number of solutions.
`nx[17]`	–	number of points (in *r*-scale) for output
`x`	–	array of position points chosen for output
`WX`	–	array of solution values in the `nx` points for each channel

x-array is define by the nx-1[10] intervals before the border radius with step hx=0.08
The nx and hx parameters can be re-define from input arguments.

REMARKS:

1. User can use any semiempirical corrections to the target states, including those which change the order the target levels. In this case, the **h.nnn** file will contain new target energies and the order the cannels will change accordingly. If the user need to use **target** file in further calculations, he need to update it for new level order. The user can obtain this new target file directly from the H.DAT file with utility **h_targb**, or directly get new target file from the old one by using **convert_target** utility.

2. Fig. 8.2 provide typical example of the **bsr_hd.001** file. The output contains main parameters of the case, in particular, the output contains the first five eigen-energies, Eval. If they are unphysically too small, it indicate the problems, connected first of all with overlap matrix. In this case, the user should check the warnings, if any, about the big overlap matrix elements. In the given example, there is no problems with energies, however, it is worth to eliminate this big overlaps by imposing additional orthogonal conditions. In the example, we need to impose additional orthogonal condition for the ksV orbital to the one from the substitution list. To find this orbital, the user should check the ns substitution orbitals for the target 11 or 18, and then add the corresponding orthogonal condition to the end of the **cfg.001** and re-run BSR_MAT and BSR_HD for the given partial wave.

```
B S R _ H D
**********

calculations for partial wave:  klsp =  1

itype =    0  -  scattering calculations

nch   =   22  -  number of channels
npert =    1  -  number of pertubers
nhm   = 3345  -  full size of matrix

iexp  =    1  -  exp.target energies are read from file thrsholds

jmvc  =    0  -  number of channel solutions for inclusion of
                 mass-velocity corrections

Edmax =  0.00E+00  - if /=0, all one-channel solutions with E > Edmax will be ignored

Problem is a GENERALIZED eigenvalue problem

New basis: khm =  3208

Warning: channel-channel overlap = 0.527  ksV  channel  21  target  21  kp1J channel  11 target  11
Warning: channel-channel overlap = 0.624  ksV  channel  21  target  21  kp1L channel  18 target  18

Eval(1:5) =    -5869.36613    -5869.36349    -5869.35974    -5869.35509    -5869.35310

time =      1.97 min.
```

Fig. 8.2. Typical example of the **bsr_hd.001** file.

8.10. *MPI version – BSR_HDB*

The MPI version BDR_HDB is based on the BLACS and SCALAPACK libraries. The BLACS (Basic Linear Algebra Communication Subprograms, www.netlib.org/blacs) is a linear algebra oriented message passing interface that can be implemented across a large range of distributed memory platforms. In particular, BLACS are used as the communication layer for the SCALAPACK project (www.netlib.org/scalapack), which involves implementing the LAPACK library on distribute memory machines. The BACS and SCALAPACK projects are used two-dimensional block-cyclic distribution, Fig 8.3. The main parameters which we will use to describe the particular distribution are the size of block, **nblock**, number of rows, **p**, and number of columns, **q**, in the distributed matrices, presented in the right part of figure. The colored block represented part of distributed matrix connected with one processor. Overall, number of processors involved is **p×q**.

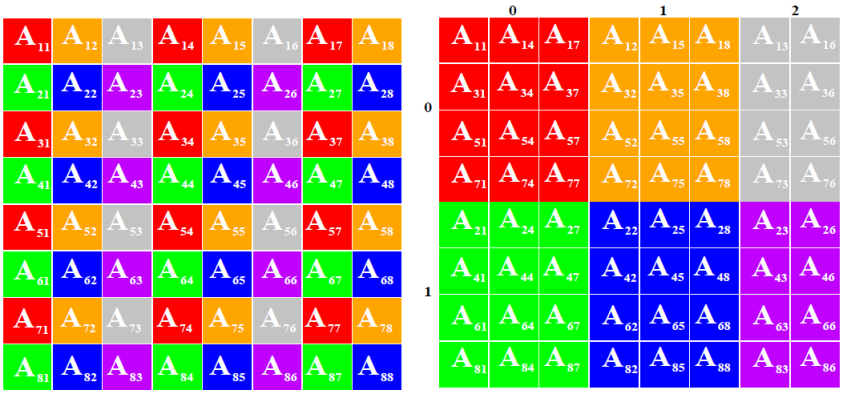

Fig.8.3. Two-dimensional block-cyclic distribution.

Overall, the BSR_HDB program follows the same logic as serial version BSR_HD, however, all main matrixes are distributed over p*q processor. The block scheme of the BSR_HDB program is given in Fig. 8.4. All MPI operation are performed through the BLACS routines, and all read/write operations are governed by the master processor only. In Fif.8.4 we use blue color for subroutines which are run only on the master processor. The SCALAPACK routines have the same names but with first letter P. The new SCALAPACK routine PDGEADD turned out to be very useful to fill in and transform of the distributed arrays.

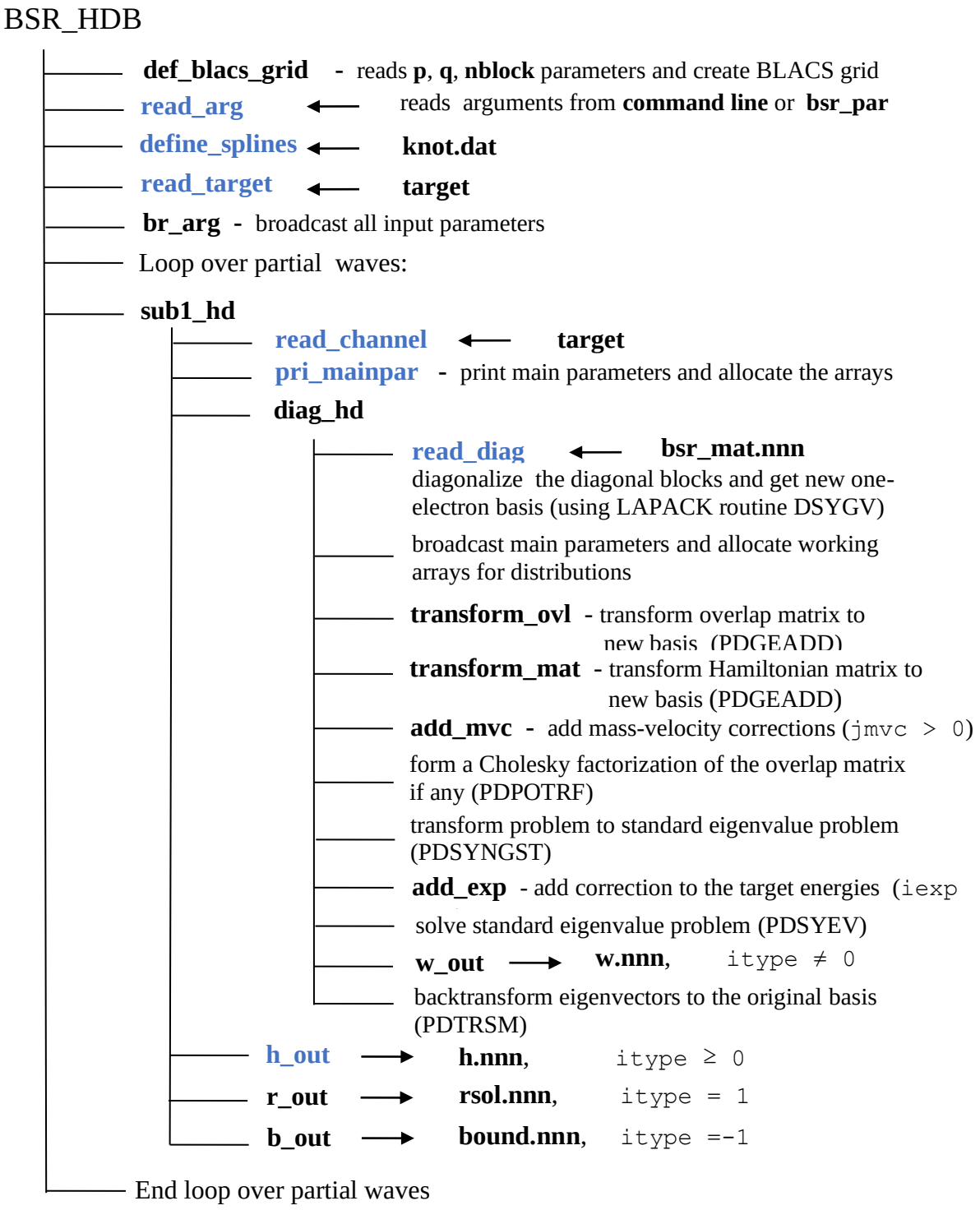

Fig. 8.4. Block diagram for the program BSR_HDB and data flow (see text).

8.11. *Futher developments*

The programs BSR_HD4 and BSR_HDB4 are slight modification of BSR_HD3 and BSR_HDB3 programs, where used new format of **bsr_mat.nnn** files provided by BSR_MAT4.