6. Program BSR_BREIT (version 3)

6.1. Outline of the BSR_BREIT calculations

The BSR_BREIT program performs the angular integrations necessary to express the matrix elements of the Breit-Pauli Hamiltonian as a linear combination of radial integrals. Any amount of non-orthogonality between the orbitals may be present, leading to overlap factors in the matrix elements. The program can effectively reuse data obtained previously by creating a databank for angular coefficients. BSR_BREIT is an optimized version of the general program BREIT_NO [24]. It can be used in many other aspects of atomic-structure calculations. The calculations of angular integrals follow the method based upon the representation of configuration wave functions through Slater determinants. More details of the present approach can be found in the long write-up of the program BREIT_NO. Below we only provide a short description of the present version, outlining the most significant modifications made in comparison to the original BREIT_NO.

The matrix element in the most general case of fully non-orthogonal orbitals has the form

$$< \Phi(\{nl\}\gamma LSJ) \,|\, \mathbf{O} \,|\, \Phi'(\{n'l'\}\gamma' L'S'J') >= \sum a_k R^k(i, j; i', j') \times D(\{nl\};\{n'l'\}), \qquad (6.1)$$

where a_k are the numerical coefficients, which depend only on the angular symmetry of the configuration state functions (CSFs) involved, R^k is the corresponding radial integral dependent on the operator **O** under consideration, and $D(\{nl\},\{n'l'\})$ is the overlap factor, which depends only on the sets of radial orbitals used in the construction of the wavefunctions Φ and Φ', respectively. Hereafter, the notation *angular symmetry* (AS) is used to denote the full set of angular momenta $(\{l\}\gamma LS(L'S'))$, which includes the momenta of spin-orbitals $\{l\}$, the terms of subshells $\alpha_i L_i S_i$, and the intermediate terms $L'_i S'_i$ as well as the total momenta LS. The symbol γ defines the coupling of momenta of all occupied subshells according to the given scheme

$$((((\text{core},L_1)L'_1,L_2)L'_2,L_3)L'_3 \ldots)L, \quad ((((\text{core},S_1)S'_1,S_2)S'_2,S_3)S'_3 \ldots)S, \quad \boldsymbol{L} + \boldsymbol{S} = \boldsymbol{J}. \qquad (6.2)$$

It is convenient to introduce the notation *generalized shell term* for $(\alpha LS)L'S'$, which also includes the corresponding intermediate momenta $L'S'$. It should be emphasized that the coefficients a_k do not depend directly on the set of the principal quantum numbers $\{n\}$ and $\{n'\}$. The overlap factor $D(\{nl\},\{n'l'\})$ is generally a product of determinants corresponding to the matrices of one-electron overlap integrals $<nl|n'l>$; it depends only on the orthogonality conditions imposed on the one-electron radial functions. For example, in the special case of orthogonal radial orbitals, the overlap factors are trivially reduced to 1 or 0, and we obtain the weighted sum of radial integrals R^k, which is ordinarily used in atomic structure calculations. For partial orthogonality conditions, some D-factors disappear while others are simplified. Thus, once we obtain the matrix element in the case of fully non-orthogonal orbitals, we can also obtain the matrix elements for all other configurations with the same AS and for all other

orthogonal conditions, only by analyzing the overlap factors $D(\{nl\},\{n'l'\})$ and replacing the set $\{n,n'\}$ by the new one. Such a scheme is realized in the present code. In large-scale calculations, it can lead to a considerable reduction of the execution time, because in practical calculations the set of configurations contains many configurations with the same angular symmetry.

On the other hand, rather than record the angular coefficients a_k for a given set of CSFs, we can save only the expansions for the matrix between ASs with full non-orthogonality conditions. This data may be considered as a *databank* for angular coefficients a_k. They allow us to obtain the expansions of matrix elements for any set of CSFs constructed with the ASs included in the databank. In case of new symmetries, we have to perform only the additional calculations and thus the databank can be easy extended.

The next idea concerns the structure of the angular coefficients a_k themselves. Because the calculations of angular integrals follow the method based upon the representation of configuration wavefunctions through Slater determinants, the coefficients a_k can be separated into a factor that depends only on the Slater determinants involved and a factor dependent on the coupling scheme (6.2). Since many atomic states contain the same Slater determinants, it seems to be much more efficient to first calculate the matrix elements between separate Slater determinants before multiplying them with the coupling-scheme factor depending on the AS under consideration. The Slater determinants involved are defined only by the sets $\{l^q\}$ where q is the shell occupation number. Hereafter, we will refer to the set $\{l^q\}$ as a *configuration symmetry* (CS). Hence, in order to avoid the recalculation of matrix elements between the same Slater determinants, it is most effective to first define the list of all possible Slater determinants and then to obtain the angular coefficients for the matrix elements between these Slater determinants.

The above ideas have been implemented in the present code. It requires additional sorting of the input configuration list according to their ASs as well as to CSs. The principal features of the present version are given below.

6.2. Structure and data flow

The block diagram of the program BSR_BREIT, along with the data flow, is shown in Fig. 6.1. The only required input-data file is a *c*-file, **name.c**, with a list of the configurations under consideration. The name of c-file is define by the first position parameter in the command line. If this parameter is absent, the default name for the c-file is **cfg.inp** or **cfg.nnn**, if a set of *c*-files for different partial waves **nnn** should be handled. The range of required partial waves is given by the input parameters **klsp1** and **klsp2**. BRS_BREIT requires only a limited number of input parameters (see section 6.4), which can be defined in the command line in the format '*name=value*', see section 6.4. All input parameters are read from the command line by the routine **read_arg.**

Then the program executes fully independent calculations for each partial wave. First, the r**ead_conf** routine reads the input configurations in spectroscopic notation, decodes them to the

internal 'integer' representation and tests the CSFs as to the correctness of angular coupling, the number of electrons, parity and AFTER conditions. **read_conf** also allocates (or reallocates) all the main arrays with dimensions specific for a given partial wave. Then the CSFs are sorted according to their angular and configuration symmetries. At this stage the program also checks if it is a continued calculation, i.e., if there exists a relevant databank from previous calculations. The program analyses information contained in the databank and defines the subset of configurations and operators that need additional consideration, or it decides that information in the databank is sufficient for the given input configuration list.

In the next block, **pre_det_exp**, the determinant expansions for all configurations are generated and recorded in a scratch file. This block is the main modification made in comparison to the previous version, BREIT_NO. It allows one to avoid a large amount of recalculations for repeating determinant coefficients. The determinant expansion for a given angular symmetry is generated in the routine **det_expn**, on the basis of angular coupling coefficients, calculated in **detc_sh**, and coefficients for determinant expansions of individual shells, stored in the library SHELLS.

The calculations of a full set of angular coefficients and overlap factors for input angular symmetries are running in the block **conf_loop,** in the loop over the configuration symmetries. First, the determinant coefficients for given CS are read from the scratch file and routine **det_breit** computes the angular coefficients for matrix elements between all Slater determinants involved. This information is accumulated in the module **zoef_list**. The cases of overlaps, one-electron and two-electron operators are treated separately by the routines **zno_0ee**, **zno_1ee**, and **zno_2ee**, respectively. Routine **zno_breit** contains expressions for all Breit-Pauli two-electron matrix elements in the *nlms*-representation. All these expressions are presented in the BREIT_NO write-up [24]. The corresponding angular coefficients for the integrals involved are accumulated in the module **boef_list**, in order to avoid their recalculation. It is controlled by the routine **check_boef**.

Then, in routine **term_loop**, the angular coefficients between two given angular symmetries are calculated based upon the matrix elements between the determinants involved, stored in module **zoef_list**, and the determinant expansion coefficients, which are read from scratch file **nud** (see block **pre_det_exp)**. These data are accumulated in the module **coef_list**. When all determinants for the given angular symmetries are exhausted, the final angular coefficients are recorded by the routine **add_res** in the scratch file **nui**, and module **coef_list** is initialized for accumulation of data from other matrix elements. The above calculations are repeated in the block **conf_loop** for all combinations of input angular symmetries.

Finally, the new data and data from previous calculations are stored in file **int_res.nnn**. This file is then renamed to the initial name **int_bnk.nnn**. Such a procedure allows us to prevent loosing the data in the initial databank if an error occurs when running BSR_BREIT. The format of the output databank is given in section 6.5. For specific input *c*-file **name.c**, the output databank has name **name**.**bnk.** However, if exists the databank with generic name **int_bnk**, this

file will be used to accumulate coefficients if any. This allows one to use one files for many different c-files and avoid the repeating calculations.

All calculations in BSR_BREIT are carried out in full non-orthogonal mode, without specifications of principal quantum numbers for the orbitals involved. Instead, the positions of the corresponding shells in the configurations are used to specify the one-electron orbitals involved. Consequently, the data obtained can be used further for all atomic states with the given angular symmetries. In the present version we also abandoned the creation of an additional list of angular coefficients for the given set of one-electron orbitals (as the file **int.lst** in BREIT_NO). Instead, the matrix elements for specific orbitals are calculated dynamically at the stage when the Hamiltonian matrix is generated in the program BSR_MAT.

6.3. Data files

name.c	File type: formatted sequential input.
cfg.nnn	Created by program BSR_CONF. Read by programs BSR_BREIT and BSR_MAT. Description: contains configuration expansion for specific atonic state or for partial wave **nnn**.
name.bnk	File type: unformatted sequential output.
int_bnk.nnn	Written by program BSR_BREIT**.**
int_bnk	Read by BRS_MAT. Description: databank for angular coefficients.
bsr_breit.log	File type: formatted sequential output. Written by program BSR_BREIT**.** Read by user. Description: running information.

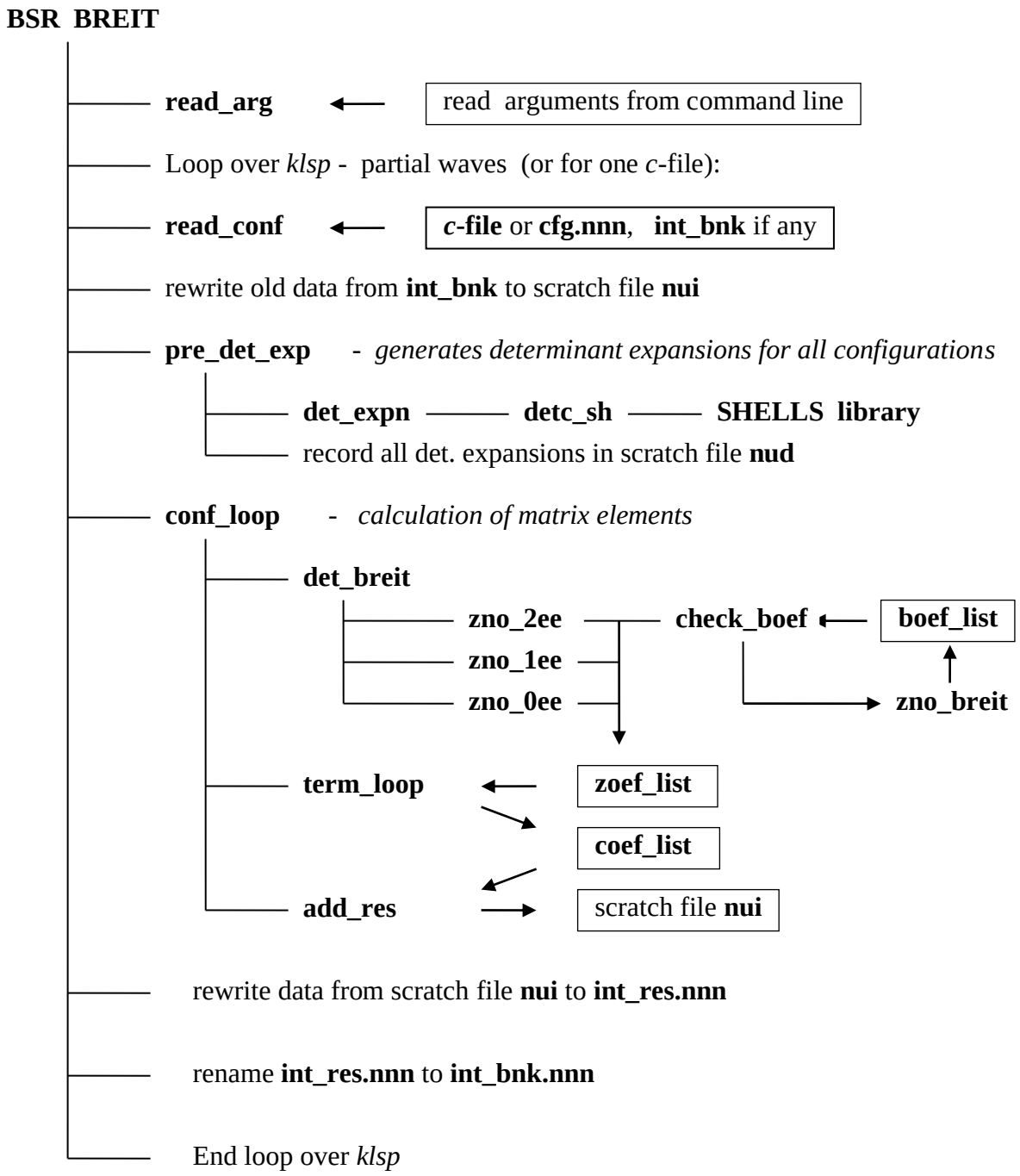

Fig. 6.1. Block diagram for the program BSR_BREIT and data flow (see text).

6.4. Input parameters

Input parameters can be provided only in the command line in the format '*name*=*value*'. All parameters have the default values indicated in the brackets. The name for the specific c-file should be indicated as first position parameter as **name.c** or just **name**.

klsp1 [0]	first partial wave under consideration; default value **klsp1**=0 means that input configuration file is **cfg.inp** and output data will be placed in file **int_bnk**.
klsp2 [klsp1]	last partial wave under consideration.
oper [1110000]	character*7, where each position can be 0 or 1, and indicate the operator under consideration: oper(1) - overlaps oper(2) - kinetic energy oper(3) - two-electron electrostatic oper(4) - spin orbit oper(5) - spin-other-orbit oper(6) - spin-spin oper(7) - orbit-orbit Default value **oper=**1110000 stands for non-relativistic calculations.
mk [9]	maximum multipole index.

Examples for calling:

```
bsr_breit3   klsp1=1  klsp2=5  oper=1111000
bsr_breit3   2p2_3P.c  mk=5
```

6.5. Structure of the angular-coefficients databank

Summary of the output records in the **int_bnk** file.

1. Description of configuration symmetries (see **MOD_symc.f90**)

```
nsymc,lsymc
(LT_conf(i),i=1,nsymc)
(ST_conf(i),i=1,nsymc)
(no_conf(i),i=1,nsymc)
(ip_conf(i),i=1,nsymc)
(iq_conf(i),i=1,lsymc)
(ln_conf(i),i=1,lsymc)
```

2. Description of angular symmetries (see **MOD_symt.f90**)

```
nsymt,lsymt
(IT_conf(i),i=1,nsymt)
(ip_term(i),i=1,nsymt)
(LS_term(i,:),i=1,lsymt)
```

3. Description of done matrix elements

```
n = nsymt×(nsymt+1)/2
IT_OPER(1:7,1:n)
```

`IT_OPER(.,ij)` indicates the operators that have already been considered by the program for angular symmetries `i,j` such that `ij = i×(i+1)/2+j, i > j.`
`IT_OPER(1:7,.)` indicates the operators under consideration, in the same way as the input parameter **oper** (see section 6.3)

4. Description of overlap determinants

```
ndet,kdet
for i = 1,ndet
kpd(i),ipd(i),jpd(i),npd(ipd(i)+1:ipd(i)+kpd(i))
```

`ndet` – total number of overlap determinants.
`kdet` – size of the npd array, containing the description of all overlap determinants.
`kpd(i)` – dimension of the `i`-th overlap determinant, `kd`.
`ipd(i)` – pointer on the `i`-th overlap determinant in array `NPD`, `ip`.
`npd(ip+1:ip+kd)` – `contains` description of the *i*-th determinant as `NPD(.) =` $i_1 * b_d + i_2$, where $b_d = 2^{15}$ – packing basis for overlap determinants; i_1, i_2 – pointers to shells in the involved in the involved configurations.

5. Description of overlap factors

```
ndef,kdef
for i = 1,ndef
kpf(i),ipf(i),jpf(i),npf(ipf(i)+1:ipf(i)+kpf(i))
```

`ndef` – number of different overlap factors.
`kdef` – size of the `NPF` array, containing the description of all overlap factors.
`kpf(i)` – number of determinants in the `i`-th overlap factor, `kd`.
`ipf(i)` – pointer on the `i`-th overlap factor in array `NPF`, `ip`.
`npf(ip+1:ip+kd)` – contains description of the `i`-th overlap factor as `NPF(.) = ipd*`b_f `+ nd`, where b_f `= 16` – packing basis for overlap factors; `ipd` – pointer to the overlap determinant; `nd` – its power.

6. `C,it,jt,int,idf` repeat up to the end of file

`C` – angular coefficient for matrix element between angular symmetries `it` and `jt`
`int` – pointer on the relevant radial integral in the packing form
`int = m`×b^8+`k`×b^4+i_1×b^3+i_2×b^2+i_3×`b`+i_4, where `m` – type of integral; `k` – multipole index; i_1,i_2,i_3,i_4 – pointer on the involved orbitals as position of the corresponding subshell in the configuration; `b = 10` – packing basis.
`idf` – pointer on the corresponding overlap factor, see item 5.

6.6. MPI version

The MPI version, BSR_BREIT3_MPI, employs the same algorithm for the calculation of angular coefficients as the serial version, and used the same input parameters and the same structure for the databank. Parallelization consists in the parallel calculations of the matrix elements between two specified configuration symmetries. The determinant expansions for different configurations can contain very different number of Slater determinants, from 1 to millions, resulting in very different time for the calculations. It is impossible in advance to spread all matrix element over processors in such way that they would take the approximately equal time. In BSR_BREIT3_MPI, we use one processor as the master which organizes the calculations and records the results. The master contains the loop over all needed matrix elements and assign, one by one, configuration symmetries for the slave processors to be considered. When the processor finish calculations for the given configuration symmetries, it sends the result to the master to record the coefficients in the databank and obtains the next configuration symmetries to processed. In such way we guarantee that all processors will be equally loaded, except the last part of calculations when the number of required matrix less than the number of processor. Block diagram for BSR_BREIT_MPI is given in Fig.6.2. All operations are carried out by the master, except the central computational block, beginning from the **conf_calc** subroutine. In this block, each processor performs calculations for one pair of configurational symmetries.

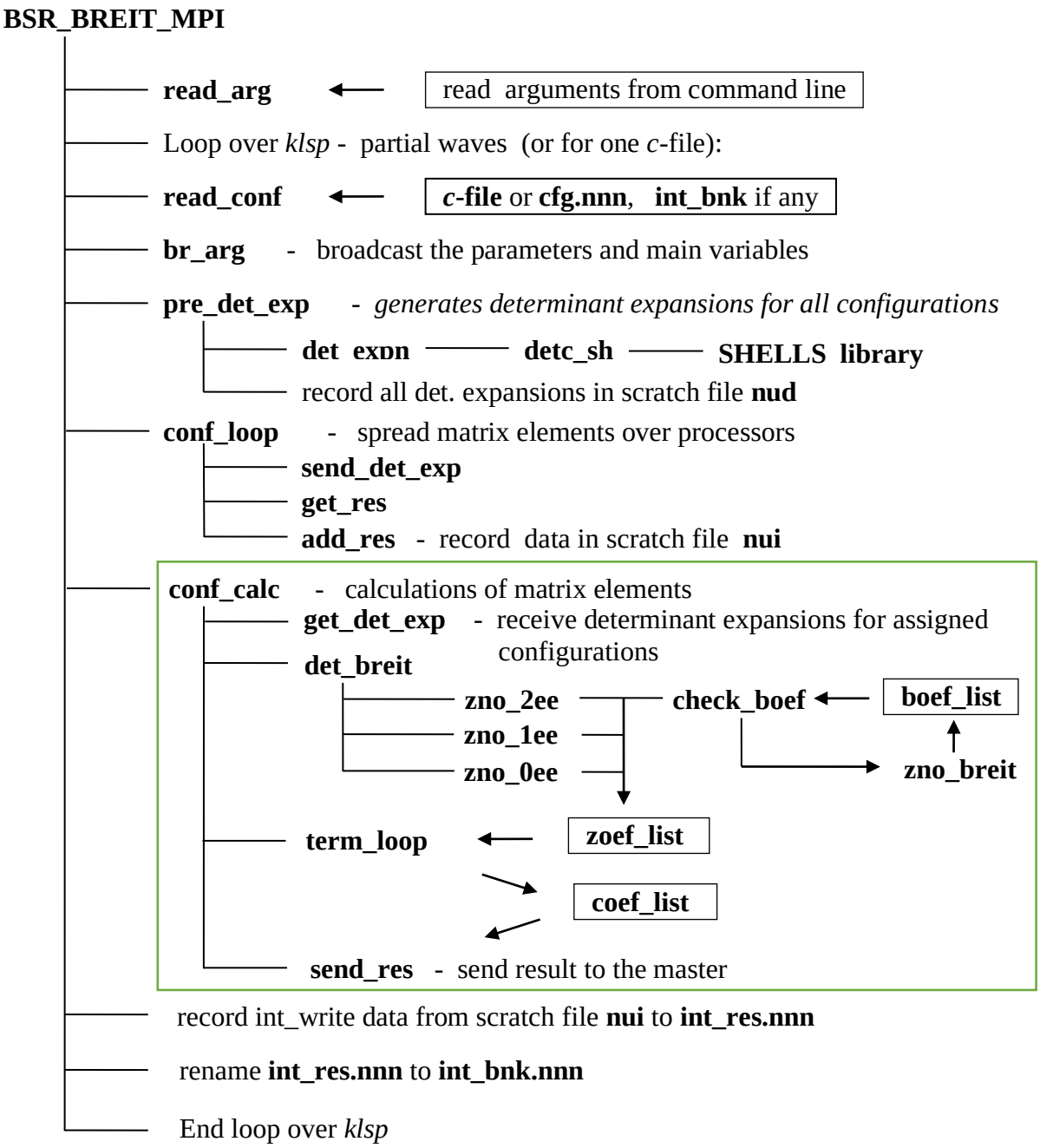

Fig. 6.2. Block diagram for the program BSR_BREIT_MPI. The green-border box indicate the calculations in the slave processors.

```
                    ======================
                      B S R _ B R E I T
                    ======================

 Operators included:

 OVERLAPS
 KINATIC ENERGY
 TWO-ELECTRON ELECTROSTATIC

 Max.multipole index =  7

--------------------------------------------------------------------------------

 Partial wave:     1

 It is new calculations

 input c-fail contains:

 number of atomic states   =   149
 number of conf.symmetries =    48
 number of ang. symmetries =    99

 Determinant expansions:

 ic_case, nterm, ndet           1           1           9
 ic_case, nterm, ndet           2           1           9
 ic_case, nterm, ndet           3           2         100
 ic_case, nterm, ndet           4           2          54
.....................................................
 ic_case, nterm, ndet          48           2          45

 Calculations for given symmetry:

 is=     1/    48  nterm=     1  ndet=     9      0.00 sec    s( 2) p( 2) d( 1)
 is=     2/    48  nterm=     1  ndet=     9      0.00 sec    p( 4) d( 1)
 is=     3/    48  nterm=     2  ndet=   100      0.02 sec    s( 1) p( 2) d( 1) d( 1)
 is=     4/    48  nterm=     2  ndet=    54      0.02 sec    s( 1) p( 1) s( 1) p( 1) d( 1)
 is=     5/    48  nterm=     1  ndet=    19      0.00 sec    s( 2) d( 2) d( 1)
 is=     6/    48  nterm=     1  ndet=    30      0.02 sec    s( 2) p( 1) f( 1) d( 1)
.....................................................................................
 is=    46/    48  nterm=     4  ndet=   117      0.12 sec    s( 1) p( 2) f( 2)
 is=    47/    48  nterm=     2  ndet=    81      0.11 sec    s( 1) p( 2) p( 1) f( 1)
 is=    48/    48  nterm=     2  ndet=    45      0.05 sec    s( 2) p( 1) d( 1) f( 1)

 Results for new calculations:

 number of overlap determinants =        90       1.8
 number of overlap factors      =      2827       3.1
 total number of coefficient    =     93851

 Partial wave:        0.04 min
 Total time:          0.04 min
```

Fig. 6.3. Example for the **bsr_breit.log** file.

Example of the running information presented in the bsr_breit.log file is given in Fig. 6.3. It contains the operator includes, the configuration-expansion parameter, main sizes for the determinant expansions, and the timing of calculations for given configuration symmetry. This information is useful in large-scale calculations and allows the user to adjust the computations in the similar cases.

6.6. Further development (**version 4**)

The item 6 in **int_bnk** (see 6.5) contains the main results and is the most extended part of the databank. The recording of each coefficient separately is convenient and simple for handling, however, in case of large-scale calculations, with big number of coefficient, this recording becomes time consuming. The recording all information in one file is also convenient from computational point of view, however, it would requires too much time and efforts for re-writing all old information for updating the databank. It is the reason to develop new version, BSR_BREIT4, where the databank is divided on two parts, **int_inf** and **int_int**, where first part will be contain only information about configurations and operators included (items 1 to 5) and second part contains only coefficients (item 6). Part 6 now is recording **in the blocks** for different pairs of angular symmetries, `it` amd `jt`. In case of additional calculations, we now need only update and re-write only items 1-5, file **int_inf**, and new coefficient, if any, just are appended to the second file, **int_int**.

Another modification:

1. Restrictions on the number of angular symmetries considered simultaneously in the **conf_loop** routine.

It may happen that some configuration symmetries have extensive determinant expansions (up to 1000,000 determinants) and many different angular symmetries. Usually, it happens in case configurations with several open d- of f-shells. The calculations in this case are extremely time-consuming. In this case, a few processors continue to work whereas other processors are sleeping. To avoid such situation, we introduce new parameters which restrict the number of terms and determinants considered by individual processors.

mkt [10000] - maximum of the terms in one block of computations
mkdt [1000000] - maximum size of determinant expansion.

If these parameters are violated, the given case are divided on several cases with smaller dimensions.

2. Time restrictions.

time [0] - limit on time of calculations.

If parameter time is not zero, the program stops the calculations and create databank which contain only results for the matrix element proceeded. The user can re-run the BSR_BREIT4 to calculate the rest of matrix elements and adjust parameters or number of processors. The time parameter allow to avoid lost of data when program was terminated by the system due the time limits (what frequently happens in the supercomputer environment)

3. Tolerance for the configuration expansion coefficients.

eps_c [1.d-8] - the configurations with expansions coefficient less than **eps_c** will be ignored

eps_soo [1.d-8] - the configurations with expansions coefficient less than **eps_soo** will be ignored for the Breit-Pauly operators.

6.7. Structure of the angular-coefficients databank (**int_inf** and **int_int**)

The **int_inf** file has the same structure as first 5 items of **int_bnk**

The **int_int** file contains the angular coefficients recorded in blocks. Each block has following records:

`nbuf`	–	number of coefficients
`C(1:nbuf)`	–	angular coefficients
`it(1:nbuf)`	–	angular symmetry for the initial state
`jt(1:nbuf)`	–	angular symmetry for the final state
`int(1:nbuf)`	–	radial integral in the packing form
`idf(1:nbuf)`	–	pointer to the corresponding overlap factor

New structure of databank reduce the read/write operations and considerably reduce the computational time in the case of large configuration expansions