9. Program DBSR_MULT

9.1. Outline of the DBSR_MULT calculations

The DBSR_MULT program performs the angular integrations necessary to express the matrix elements of the multipole transition operators 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 and can be used in many other aspects of atomic-structure calculations. DBSR_MULT can also be considered as a straightforward extension of the program BSR_MULT [BSR] to the case of *jj*-coupling. The calculations of the angular integrals follow the method based upon the representation of configuration wavefunctions through Slater determinants.

9.1.1. Definitions

The program deals with the transition operators $O_{\mu}^{[\lambda]}$ for electric and magnetic transitions. The dependence of the transition matrix elements on the total magnetic quantum numbers can be found by applying the Wigner-Eckart theorem,

$$< JM \mid O_{\mu}^{[\lambda]} \mid J'M' >= (-1)^{J-M} \begin{pmatrix} J & \lambda & J' \\ -M & \mu & M' \end{pmatrix} < J \parallel O^{[\lambda]} \parallel J' > . \qquad (9.1)$$

The problem is to evaluate the reduced matrix element (RME) of the electric and magnetic multipole operator of any order between arbitrary *jj*-coupled configurations. The calculations of the angular integrals follow the method based upon the representation of configuration wavefunctions through Slater determinants. This allows us to introduce a completely unrestricted version of a non-orthogonal scheme.

9.1.2. Calculation of RME between jj configuration wavefunctions

The present procedure is based upon the expansion of the *N*-electron configuration wave function in terms of Slater determinants

$$\Phi(\gamma JM) = \sum_{\{u\}} (\{u\} \mid \gamma JM) \mid u_1 ... u_N > \qquad (9.2)$$

where $(\{u\}|\gamma JM)$ denotes the expansion coefficient, and the individual determinant $\mid u_1 ... u_N\rangle$ has the form

$$|u_1 \ldots u_N> \;=\; \frac{1}{N!}\begin{vmatrix} <t_1|u_1> & <t_1|u_2> & \ldots & <t_1|u_N> \\ <t_2|u_1> & <t_2|u_2> & \ldots & <t_2|u_N> \\ \ldots & \ldots & \ldots & \ldots \\ <t_N|u_1> & <t_N|u_2> & \ldots & <t_N|u_N> \end{vmatrix} \tag{9.3}$$

Here t=($\mathbf{r}$,s) represents the spin and position of the individual electrons and u stands for the one-electron quantum numbers, $|u\rangle$ = $|\, nljm\rangle$ or $\left|u\right\rangle = \left|n\kappa m\right\rangle$, where m is the z-component of the total angular momentum j and κ is the relativistic quantum number defined as

$$\kappa = (l-j)(2j+1)\ . \tag{9.4}$$

Each of one-electron spin-orbitals in relativistic theory is a Dirac four-component spinor

$$\phi_{n\kappa m}(\mathbf{r}) = \frac{1}{r}\begin{pmatrix} P_{n\kappa}(r)\eta_{\kappa m}(\hat{\mathbf{r}},\sigma) \\ iQ_{n\kappa}(r)\eta_{-\kappa m}(\hat{\mathbf{r}},\sigma) \end{pmatrix}, \tag{9.5}$$

where the real and imaginary parts are referred as the large and small components, respectively, and $\eta_{\kappa m}$ is the spinor spherical harmonic

$$\eta_{\kappa m}(\hat{\mathbf{r}},\sigma) = \sum_{m_\ell m_s} (\ell m_\ell \tfrac{1}{2} m_s \mid jm) Y_{\ell m_\ell}(\theta,\phi)\chi_{\frac{1}{2}m_s}(\sigma), \tag{9.6}$$

where $(\ell m_\ell \tfrac{1}{2} m_s \mid jm)$ are Clebsch-Gordon coefficients, $Y_{\ell m_\ell}(\theta,\phi)$ are spherical harmonics and $\chi_{\frac{1}{2}m_s}(\sigma)$ are the usual two-component Pauli spin functions. The separation of variables in representation (4) allows the Hamiltonian matrix element to be split so that angular integrals are calculated using Racah algebra and radial integrals by numerical quadrature.

Thus, the determination of matrix elements between CSFs is reduced to that of matrix elements between separate Slater determinants, which, in turn, are reduced to one-electron integrals between spin-orbitals in the *nljm*-representation. In comparison to the existing codes based upon the Racah technique, the resulting formula explicitly include the magnetic quantum numbers and contain additional summation over Slater functions and over all electrons but not shells. This increases considerably the number of operations required, and the speed of the calculations is much reduced. On the other hand, the final formulae are much simpler and more convenient for programming. The principal advantage, however, is that the *nljm*-representation procedure facilitates the straightforward extension to the case of non-orthogonal orbitals to be made in the most general way. The main problem here is to obtain the determinantal representation of arbitrary atomic wavefunctions, including configurations with several open shells. An effective way to solve this problem has been proposed in case of the *LS*-coupling in the program ZAP_NO (Zatsarinny 1996). It is based upon the preliminary creation of tables of expansion coefficients for the individual shells. The determination of the coefficients for the CSFs is then reduced to the vector coupling of shell terms. The same procedure is used in the present code for the *jj*-coupling.

Consider a matrix element between two Slater determinants, $|u_{1..}\ u_N>$ and $|v_{1...}\ v_N>$, for a transition operator $\boldsymbol{O}^k = \sum_{i=1}^{N} \boldsymbol{o}^{\lambda}(i)$, where $\boldsymbol{o}^{\lambda}(i)$ is the one-electron operator with multipole index λ, acting on electron i. Define the determinant D_{uv}=det{$<u_i|v_j>$} of the matrix formed from all overlap integrals between the one-electron orbitals u_i and v_j, (i,j = 1,...,N), and let the $D_{uv}(\rho,\rho')$ denote the first cofactors of the determinant D_{uv}, corresponding to the matrix of overlap integrals with orbitals u_ρ and $v_{\rho'}$ deleted from the initial and final sets, respectively. Then, it can be shown (Löwdin 1955) that

$$< u_1 u_2 ... u_N \mid O_\mu^{[\lambda]} \mid v_1 v_2 ... v_N >= \sum_{\rho,\rho'} (-1)^{\rho+\rho'} < u_\rho \mid o_\mu^{[\lambda]} \mid v_{\rho'} > D_{uv}(\rho,\rho') . \tag{9.7}$$

This formula, in connection with irreducible tensor algebra, allows us to reduce the configuration matrix elements to one-electron matrix elements between spin-orbitals *nljm*, multiplied by overlap factors in the case of non-orthogonal radial orbitals. Applying the Wigner-Eckart theorem, the one-electron matrix elements in (9.5) can be reduced to one-electron RME.

$$< nljm \mid o_\mu^{[\lambda]} \mid n'l'j'm' >= (-1)^{j-m} \begin{pmatrix} j & \lambda & j' \\ -m & \mu & m' \end{pmatrix} < nlj \parallel o^{[\lambda]} \parallel nl'j' > . \tag{9.8}$$

The present algorithm for the calculation of matrix elements consists of two independent steps: the determination of the determinantal representation for two input configuration wavefunctions with certain total magnetic quantum numbers M, and the calculation of matrix elements between all Slater states obtained by the above formulae. Our final task is to obtain the reduced matrix elements for configuration wavefunctions, $< \gamma J \parallel O^{[\lambda]} \parallel \gamma' J' >$. They are obtained on the basis of the calculated matrix elements $< \gamma JM \parallel O_\mu^{[\lambda]} \parallel \gamma' J' M' >$ by a reciprocal application of the Wigner-Eckart theorem (9.1). Note that the amount of calculations in the present approach crucially depends on the number of Slater states involved, which, in turn, depends on the chosen quantum numbers M and M'. In the present implementation, we use $M = J$, which guarantees a minimum number of relevant Slater states.

The one-particle reduced matrix element for transition amplitude can be written as

$$\left\langle \alpha \parallel \mathbf{o}^{\lambda} \parallel \beta \right\rangle^{e,m} = \left\langle j_\alpha \parallel \mathbf{C}^{\lambda} \parallel j_\beta \right\rangle \cdot \Pi^{e,m}(l_\alpha + k + l_\beta) \cdot \mathbf{M}_{\alpha\beta}^{e,m}(\omega; G_\lambda) \tag{9.9}$$

where

$$< j_\alpha \parallel \mathbf{C}^{(\lambda)} \parallel j_\beta >= (-1)^{j_\alpha + 1/2} \sqrt{[j_\alpha][j_\beta]} \begin{pmatrix} j_\alpha & \lambda & j_\beta \\ 1/2 & 0 & -1/2 \end{pmatrix} \tag{9.10}$$

$\mathbf{M}_{\alpha\beta}^{e,m}(\omega; G_\lambda)$ is a radial integral, ω stands for the angular frequency of the transition, and G_λ is a gauge factor. $\Pi^{e,m}(l_\alpha + k + l_\beta)$ is parity factor, which for electric transitions equal 0, when $(l_\alpha + k + l_\beta)$ is odd,

and for magnetic transitions equal 0, when $(l_\alpha + k + l_\beta)$ is even. The detailed expressions for the electric, $\mathbf{M}^e$, and magnetic, $\mathbf{M}^m$, integrals are given by Grant (2008). Let's introduce following one-electron integrals

$$J_\lambda = \int_0^\infty (P_\alpha(r)P_\beta(r) + Q_\alpha(r)Q_\beta(r))j_\lambda(\omega r/c)dr \tag{9.11a}$$

$$I_\lambda^\pm = \int_0^\infty (P_\alpha(r)Q_\beta(r) \pm Q_\alpha(r)P_\beta(r))j_\lambda(\omega r/c)dr \tag{9.11b}$$

The magnetic multipole amplitude (Grant (2008), Ch.8.2.1)

$$\mathbf{M}^m_{\alpha\beta}(\omega; G_\lambda) = -i^{\lambda+1}\frac{2\lambda+1}{\sqrt{\lambda(\lambda+1)}}(\kappa_\alpha + \kappa_\beta)I_\lambda^+ \tag{9.12}$$

is independent of G_λ. For electric multipoles,

$$\mathbf{M}^e_{\alpha\beta}(\omega; G_\lambda) = \mathbf{M}^e_{\alpha\beta}(\omega; 0) + G_\lambda \mathbf{M}^l_{\alpha\beta}(\omega) \tag{9.13}$$

where the Coulomb (velocity) gauge integral is

$$\begin{aligned}\mathbf{M}^e_{\alpha\beta}(\omega; 0) &= -i^\lambda\left\{\left(\frac{\lambda}{\lambda+1}\right)^{1/2}\left[(\kappa_\alpha - \kappa_\beta)I^+_{\lambda+1} + (\lambda+1)I^-_{\lambda+1}\right] - \left(\frac{\lambda+1}{\lambda}\right)^{1/2}\left[(\kappa_\alpha - \kappa_\beta)I^+_{\lambda-1} - \lambda I^-_{\lambda-1}\right]\right\} \\ &= -i^\lambda\sqrt{\frac{(\lambda+1)}{\lambda}}\left\{\left(\frac{\lambda}{\lambda+1}\right)\left[(\kappa_\alpha - \kappa_\beta)I^+_{\lambda+1} + \lambda I^-_{\lambda+1}\right] - (\kappa_\alpha - \kappa_\beta)I^+_{\lambda-1} + \lambda I^-_{\lambda-1}\right\}\end{aligned} \tag{9.14}$$

and the longitudinal part is

$$\mathbf{M}^l_{\alpha\beta}(\omega) = -i^\lambda\left\{\left[(\kappa_\alpha - \kappa_\beta)I^+_{\lambda+1} + (\lambda+1)I^-_{\lambda+1}\right] + \left[(\kappa_\alpha - \kappa_\beta)I^+_{\lambda-1} - \lambda I^-_{\lambda-1}\right] + (2\lambda+1)J_\lambda\right\} \tag{9.15}$$

Length gauge we obtain with $G_\lambda = [(\lambda+1)/\lambda]^{1/2}$):

$$\mathbf{M}^e_{\alpha\beta}(\omega; G_\lambda) = -i^\lambda(2\lambda+1)\sqrt{\frac{(\lambda+1)}{\lambda}}\left\{J_\lambda + \frac{(\kappa_\alpha - \kappa_\beta)}{(\lambda+1)}I^+_{\lambda+1} + I^-_{\lambda+1}\right\} \tag{9.16}$$

If we can assume that the transition wavelength is large compared with atomic dimensions, so that we need retain only the leading terms of the power series expansion of the Bessel function,

$$j_\lambda(x) = \frac{x^\lambda}{(2\lambda+1)!!} - \frac{x^{\lambda+2}}{2(2\lambda+3)!!} + \ldots \tag{9.17}$$

Then, keeping the first term, we get

$$I_\lambda^\pm = \frac{\omega^\lambda}{c^\lambda}\frac{1}{(2\lambda+1)!!}\int_0^\infty (P_\alpha(r)Q_\beta(r) \pm Q_\alpha(r)P_\beta(r))r^\lambda dr \tag{9.18}$$

$$J_\lambda = \frac{\omega^\lambda}{c^\lambda}\frac{1}{(2\lambda+1)!!}\int_0^\infty (P_\alpha(r)P_\beta(r) + Q_\alpha(r)Q_\beta(r))r^\lambda dr \tag{9.19}$$

Let's introduce the radial integrals

$$R^{\lambda}(f,g)=\int_0^{\infty} f(r)g(r)r^{\lambda}dr \tag{9.20}$$

Then

$$\mathbf{M}^{m}_{\alpha\beta}(\omega;G_{\lambda})=-i^{\lambda+1}\frac{2\lambda+1}{\sqrt{\lambda(\lambda+1)}}\frac{\omega^{\lambda}}{c^{\lambda}}\frac{1}{(2\lambda+1)!!}(\kappa_{\alpha}+\kappa_{\beta})[R^{\lambda}(P_{\alpha},Q_{\beta})+R^{\lambda}(Q_{\alpha},P_{\beta})] \tag{9.21}$$

$$\mathbf{M}^{e}_{\alpha\beta}(\omega;0)=i^{\lambda}\left(\frac{\lambda+1}{\lambda}\right)^{1/2}\frac{\omega^{\lambda-1}}{c^{\lambda-1}}\frac{1}{(2\lambda-1)!!}\Big[(\kappa_{\alpha}-\kappa_{\beta}-\lambda)[R^{\lambda-1}(P_{\alpha},Q_{\beta})+(\kappa_{\alpha}-\kappa_{\beta}+\lambda)R^{\lambda-1}(Q_{\alpha},P_{\beta})]\Big] \tag{9.22}$$

$$\mathbf{M}^{l}_{\alpha\beta}(\omega)=$$

$$-i^{\lambda}\left\{\begin{array}{l}\dfrac{\omega^{\lambda+1}}{c^{\lambda+1}}\dfrac{1}{(2\lambda+3)!!}\Big[(\kappa_{\alpha}-\kappa_{\beta})[R^{\lambda+1}(P_{\alpha},Q_{\beta})+R^{\lambda+1}(Q_{\alpha},P_{\beta})]+(\lambda+1)[R^{\lambda+1}(P_{\alpha},Q_{\beta})-R^{\lambda+1}(Q_{\alpha},P_{\beta})]\Big]\\ +\dfrac{\omega^{\lambda-1}}{c^{\lambda-1}}\dfrac{1}{(2\lambda-1)!!}\Big[(\kappa_{\alpha}-\kappa_{\beta})[R^{\lambda-1}(P_{\alpha},Q_{\beta})+R^{\lambda-1}(Q_{\alpha},P_{\beta})]-\lambda[R^{\lambda-1}(P_{\alpha},Q_{\beta})-R^{\lambda-1}(Q_{\alpha},P_{\beta})]\Big]\\ \dfrac{\omega^{\lambda+1}}{c^{\lambda+1}}\dfrac{1}{2(2\lambda+1)!!}\Big[(\kappa_{\alpha}-\kappa_{\beta})[R^{\lambda+1}(P_{\alpha},Q_{\beta})+R^{\lambda+1}(Q_{\alpha},P_{\beta})]-\lambda[R^{\lambda+1}(P_{\alpha},Q_{\beta})-R^{\lambda+1}(Q_{\alpha},P_{\beta})]\Big]\\ \pm(2\lambda+1)\dfrac{\omega^{\lambda}}{c^{\lambda}}\dfrac{1}{(2\lambda+1)!!}[R^{\lambda}(P_{\alpha},P_{\beta})+R^{\lambda}(Q_{\alpha},Q_{\beta})]\end{array}\right\} \tag{9.23}$$

In the Babushkin (length) gauge (with $G_{\lambda}=[(\lambda+1)/\lambda]^{1/2}$) we find

$$\mathbf{M}^{e}_{\alpha\beta}(\omega;G_{\lambda})=\pm i^{\lambda}\left(\frac{\lambda+1}{\lambda}\right)^{1/2}(2\lambda+1)\frac{\omega^{\lambda}}{c^{\lambda}}\frac{1}{(2\lambda+1)!!}[R^{\lambda}(P_{\alpha},P_{\beta})+R^{\lambda}(Q_{\alpha},Q_{\beta})] \tag{9.24}$$

In order to use general formula for oscillator strengths and decay probabilities, we define the one-electron transition integrals as:

1. Electric multipole, Babushkin (length) gauge:

$$d^{\lambda}(\alpha\beta)=R^{\lambda}(P_{\alpha},P_{\beta})+R^{\lambda}(Q_{\alpha},Q_{\beta}) \tag{9.25}$$

2. Electric multipole, Coulomb (velocity) gauge:

$$d^{\lambda}(\alpha\beta)=\frac{c}{\omega}\Big[(\kappa_{\alpha}-\kappa_{\beta}-\lambda)R^{\lambda-1}(P_{\alpha},Q_{\beta})+(\kappa_{\alpha}-\kappa_{\beta}+\lambda)R^{\lambda-1}(Q_{\alpha},P_{\beta})\Big] \tag{9.26}$$

3. Magnetic multipole:

$$d^{\lambda}(\alpha\beta)=\frac{1}{(\lambda+1)}(\kappa_{\alpha}+\kappa_{\beta})[R^{\lambda}(P_{\alpha},Q_{\beta})+R^{\lambda}(Q_{\alpha},P_{\beta})] \tag{9.27}$$

Then all $\mathbf{M}_{\alpha\beta}$ radial factors has the similar form

$$\mathbf{M}^{em}_{\alpha\beta}(\omega;G_\lambda) = \pm i^\lambda \left(\frac{\lambda+1}{\lambda}\right)^{1/2} (2\lambda+1)\frac{\omega^\lambda}{c^\lambda}\frac{1}{(2\lambda+1)!!}d^\lambda(\alpha,\beta)\,. \tag{9.28}$$

We see that all expressions for matrix elements have the same factor which is convenient to include on the final stage of calculations of oscillator strengths and transition probabilities (see below)..

10.2. Oscillator strengths and decay probabilities

For completeness of the description, we provide below the main formulae used in the present code for the calculation of radiative properties. The line strength for a given transition is defined as

$$S_{ij} = S_{ji} = \sum_{m's} |< \gamma_i J_i M_i \,|\, O^{[\lambda]}_\mu \,|\, \gamma_j J_j M_j >|^2 = |< \gamma_i J_i \,\|\, O^{[\lambda]} \,\|\, \gamma_j J_j >|^2 \tag{9.29}$$

It is given in atomic units, which are $e^2 a_0^{2\lambda}$ and $\beta^2 a_0^{2(\lambda-1)}$ for electric and magnetic transitions, respectively. Here $\beta = e\hbar/2mc$ is the Bohr magneton. The transition probability can be written as:

$$A_{ji} = \frac{2(2\lambda+1)(\lambda+1)}{\lambda[(2\lambda+1)!!]^2}\frac{1}{g_i}\frac{1}{\hbar}\left(\frac{E_j - E_i}{\hbar c}\right)^{2\lambda+1} S_{ij}\,. \tag{9.30}$$

The oscillator strength (unitless) is related to the transition probability by

$$f^\lambda_{ij} = \frac{mc^3}{2e^2}\frac{A_{ij}}{\omega^2} = \frac{2(2\lambda+1)(\lambda+1)}{\lambda[(2\lambda+1)!!]^2}\frac{1}{g_i}\frac{mc}{2e^2\hbar}\left(\frac{E_j - E_i}{\hbar c}\right)^{2\lambda-1} S^\lambda_{ij}\,, \tag{9.31}$$

independent of the nature of the radiative transition.

9.1.3. Output results

The DBSR_MULT program provides, as a final result, the expressions for the reduced matrix elements of the electric and magnetic operators (9.1) and (9.2) in the form

$$< \Phi(\{nlj\}\gamma J) \,|\, \|\mathbf{O}\|\, \Phi'(\{n'l'j'\}\gamma' J') >= \sum a_k \times d^k(i,i') \times D(\{nlj\},\{n'l'j'\})\,, \tag{9.32}$$

where a_k are the numeric coefficients which depend only on the angular symmetry of the configuration state functions involved, d^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 constructing the wavefunctions Φ and Φ', respectively. It should be emphasized that the coefficients a_k do not depend directly on the set of principal quantum numbers $\{n\}$ and $\{n'\}$. They only

depends on the orthogonality conditions imposed on the one-electron radial functions. For example, in the limiting case of orthogonal radial orbitals, the overlap factors are reduced trivially to factors 1 or 0, and we obtain the weighted sum of radial integrals d^k, which is ordinarily used in atomic structure calculations. In the case of intermediate orthogonality conditions, some *D*-factors disappear while others are simplified. Thus, once we have obtained 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 angular symmetry and for all other orthogonal conditions, only by analyzing the overlap factors $D(\{nl\},\{n'l'\})$ and replacing the set $\{n,n'\}$ by the specific one. Such a scheme is realized in the present code. In large-scale calculations, this can lead to a considerable reduction of 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 for the given set of CSFs, we can save only the expansions for the matrix elements with different angular symmetries, supposing that all radial functions in the initial and final sets are non-orthogonal. These data may be considered as a *data bank* for angular coefficients. They allow us to obtain the matrix element expansions for any set of CSFs constructed with the angular symmetries included in the bank. For new symmetries, we only have to perform the additional calculations and the databank can be extended.

9.2. Structure and data flow

The block diagram of the program DBSR_MULT, along with the data flow, is given in Fig. 9.1. The only input data files are *c*-files with a list of configuration state functions for the initial and final states involved. The DBSR_MULT program considers only one type of transition operator in a single run. The type of the operator is given as E1,E2,…, M1,M2, …, where the first letter indicate if it is an electric or magnetic transition while the second letter defines the multipole index. In this convention, the parameter E0 indicates the calculations of overlap matrix elements. Since DBSR_MULT only requires a small number of input parameters, all of them must be defined in the command line as arguments.

The **read_conf_jj** routine reads the input configuration lists in spectroscopic notation, decodes them to the internal 'integer' representation, and tests the CSFs as to the correctness of angular coupling, number of electrons, parity and AFTER conditions. **read_conf_jj** also allocates (or reallocates) all main arrays with dimensions specific for the given partial wave. Then the CSFs are sorted according to their angular and configuration symmetries. At this stage the program also checks if this is a continuing calculation, i.e., whether there exists a relevant databank from some previous calculations. The program analyses the information contained in the databank and defines the subset of configurations that need

additional consideration, or it decides that the information in the data bank is sufficient for the given input configuration lists.

In the next block, **pre_det_exp**, the determinant expansions for all configurations are generated and recorded in the scratch file. It allows us avoids the large amount of recalculations of repeating determinant coefficients. The determinant expansion for a given angular symmetry is generated in the subroutine **det_expn_jj**, on the basis of angular coupling coefficients and the coefficients for determinant expansions of individual shells, stored in the module **det_jq**.

The calculations of the full set of integral coefficients and overlap factors for the input angular symmetries are run in the block **conf_loop**. First, the routine **det_mult** computes the radial expansions for matrix elements between all Slater determinants involved. This information is accumulated in the module **zoef_list**. The cases of overlaps, zero-orbital or one-orbital difference in the Slater determinants under consideration are treated separately by the routines **zno_overlap**, **zno_000**, and **zno_001**, respectively. The routine **me_jj** contains expressions for one-electron matrix elements in the *nljm*-representation (see section 9.1.2). The function **idet_fact** defines the corresponding overlaps factors $D_{uv}(\rho,\rho')$ in the expression (9.12).

Then, in the routine **term_loop**, the angular coefficients between two given angular symmetries are calculated on the basis of angular coefficients between the individual determinants involved (stored in module **zoef_list)** and 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 configuration symmetries are exhausted, the final angular coefficients for different angular symmetries involved are calculated in the subroutine **term_loop** and accumulate in the module **coef_list**. Finally, the angular coefficient for given pair of configurations are recorded in the subroutine **add_res** in the scratch file **nui**, and all array are nullified. The above calculations are repeated in the block **conf_loop** for all combinations of input configuration symmetries.

The new data and data from previous calculations are stored in the file **mult_res**. This file then is renamed to the initial name **mult_bnk** (_E1, M1, E2, …). This procedure prevents loosing the data in the initial databank if an error occurs during running the DBSR_MULT program. The format of the output databank is given in Section 9.5.

All calculations in DBSR_MULT are carried out in full non-orthogonal mode, without specifying the 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. Hence, 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 a specific set of one-electron

orbitals. Instead, the matrix element for specific orbitals are calculated dynamically when the required transition data are needed.

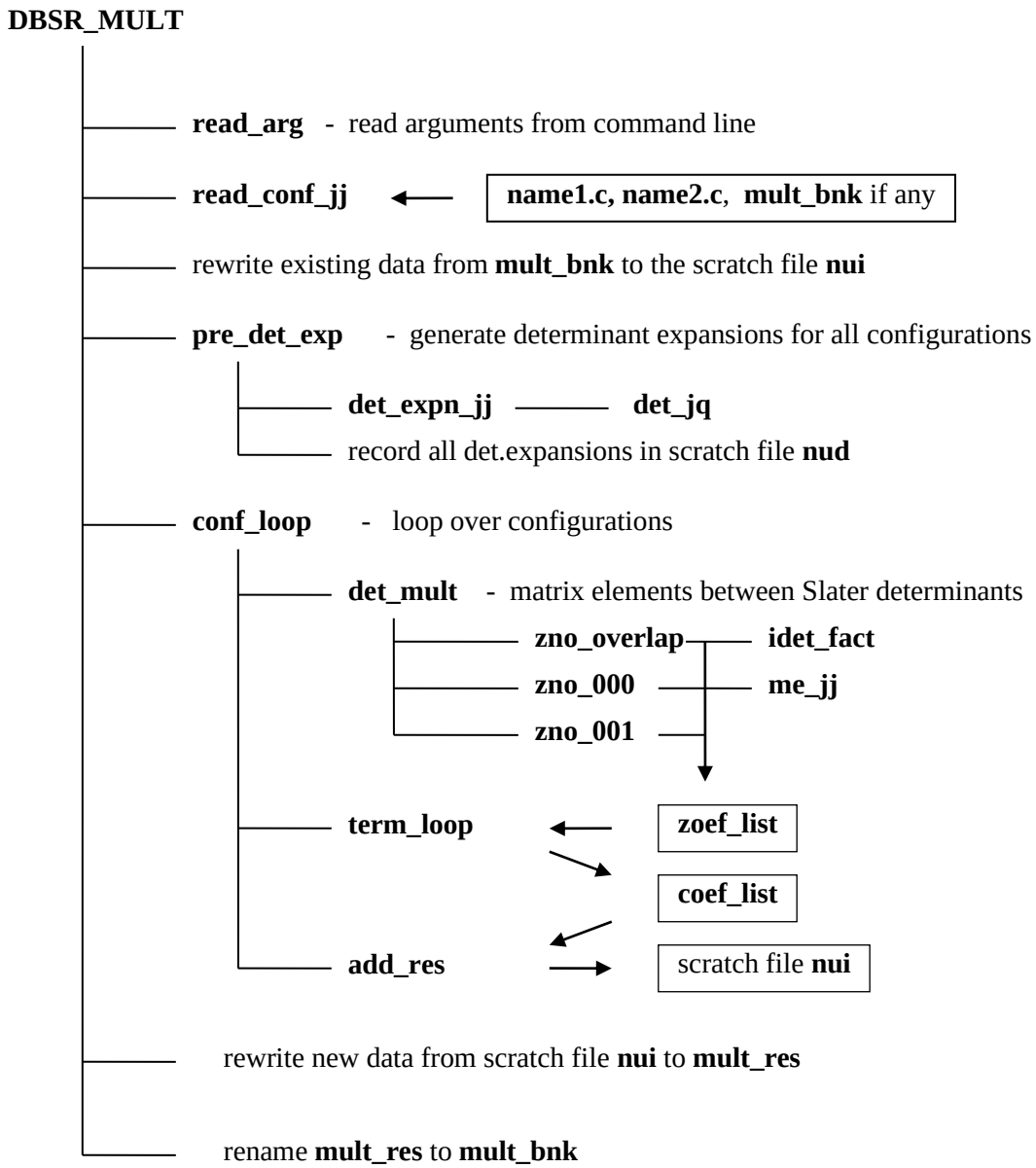

Fig. 9.1. Block diagram for the program DBSR_MULT and data flow (see text).

9.3. Data files

name1.c	File type: formatted sequential input. Created by user. Description: contains configuration expansion for the initial state**.**
name2.c	File type: formatted sequential input. Created by user. Description: contains configuration expansion for the final state**.**
mult_bnk_AA	File type: unformatted sequential output. Written by program DBSR_MULT**.** Read by DBRS_DMAT, DBSR_ZF. Description: databank for angular coefficients of the multidipole operator.
dbsr_mult.log	File type: formatted sequential output. Written by program DBSR_MULT**.** Read by user. Description: running information.

9.4.Input parameters

Input parameters must be provided in the command line.

name1.c	name for *c*-file of initial state
name2.c	name for *c*-file of finall state
AA	character*2, type of operator and multipole index: E1,E2,..,M1,M2,…

9.5. Structure of the dipole databank

Summary of the output records in the **mult_bnk** file. For more detailed description of different blocks see the **int_bnk** format in DBSR_BREITR write-up.

1. ktype,kpol

ktype - 'E' or 'M', for electric and magnetic transitions, respectively.
kpol - multipole index.

2. Block describing the configuration symmetries involved:

```
nsymc,lsymc
JT_conf(1:nsymc)
no_conf(1:nsymc)
ip_conf(1:nsymc)
iq_conf(1:lsymc)
kn_conf(1:lsymc)
```

`nsymc` - number of different configuration symmetries
`lsymc` - length of the involved arrays
`JT_conf(i)` - total angular momentum of the given configuration i
`no_conf(i)` - number of relativistic subshells
`ip_conf(i)` - pointer to the list of subshells in the `iq_conf` and `kn_conf` arrays, `ip`
`iq_conf(ip+j)` - occupation number for the subshell `j` in the configuration `i`
`kn_conf(ip+j)` - quantum number κ for the subshell `j` in the configuration `i`

(see also routines **read_symc** and **write_symc** in the module **mod_symc.f90**)

3. Block describing the angular symmetries involved:

```
nsymt,lsymt
it_conf(1:nsymt)
ip_term(1:nsymt)
JS_term(1:lsymt)
VS_term(1:lsymt)
JI_term(1:lsymt)
```

`nsymt` - number of angular symmetries
`lsymt` - length of the involved arrays
`it_conf(i)` - pointer to the configuration for the given angular symmetry i
`ip_term(i)` - pointer to the list of subshell terms, `ip`
`JS_term(ip+j)` - total angular momentum for the subshell `j` in the angular symmetry `i`
`VS_term(ip+j)` - seniority number for the subshell `j` in the angular symmetry `i`
`JI_term(ip+j)` - intermediate term for the given subshell

(see also routines **read_symt** and **write_symt** in the module **mod_symt.f90**)

4. Pointer to the considered matrix element:

```
n
it_done(1:n)
```

`n` – number of recorded elements `( = nsymt*(nsymt+1) )`
`it_done(ij) > 0` means that the matrix element between angular symmetries `i` and `j` is already in the data bank; `ij = i*(i-1)/2 + j` with `i > j;` the inclusion or not of the Breit operator is not recorded – it is a responsibility of the user to remember this information.

(see also routines **read_done** and **write_done** in the module **mod_symt.f90**)

5. List of overlap determinants involved:

```
ndet,ldet,jdet
for i=1,ndet:
kpd(i),ipd(i),jpd(i),NPD(ipd(i)+1:ipd(i)+kpd(i))
```

`ndet` – total number of overlap determinants.
`ldet` – length of the `NPD` array, containing the description of all overlap determinants.
`jdet` – average size of one overlap determinant.
`kpd(i)` – dimension of the `i`-th overlap determinant, `kd`.
`ipd(i)` – pointer to the `i`-th overlap determinant in array `NPD`, `ip`.
`jpd(i)` – ordering pointer, allowing quick searching the given overlap.
`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 configurations.

(see also routines **read_det** and **write_det** in the module **mod_det.f90**)

6. List of overlap factors involved:

```
ndef,ldef,jdef
for i=1,ndef
kpf(i),ipf(i),jpf(i),NPF(ipf(i)+1:ipf(i)+kpf(i))
```

`ndef` – total number of different overlap factors.
`ldef` – length of the `NPF` array, containing the description of all overlap factors.
`jdef` – average size of one overlap factor.
`kpf(i)` – dimension of the `i`-th overlap factor, `kd`.
`ipf(i)` – pointer to the `i`-th overlap factor in array `NPF`, `ip`.
`jpf(i)` – ordering pointer, allowing quick searching the given overlap factor.
`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.

(see also routines **read_detf** and **write_detf** in the module **mod_det.f90**)

7. List of angular coefficients:

```
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 to the relevant radial integral in the packing form
`int = m×b^8+k×b^4+i2×b+i1`, where `m` – type of integral; `k` – multipole index; i_1, i_2 – pointer on the involved orbitals; `b = 10` – packing basis.
`idf` – pointer on the corresponding overlap factor.

REMARKS:

- Only one type of transition can be in the given **mult_bnk** as indicated by corresponding extension: **mult_bnk_E1 (M1,E2, …)**.
- Matrix element changes as $(-1)^{J_1-J_2}$ if we interchange initial and final state.