

DBSR_CI

The DBSR_CI program performs set-up and diagonalization of relativistic Hamiltonian matrices within an appropriate many-particle basis. This procedure is the base for the configuration interaction (CI) and the multiconfiguration Dirac–Fock (MCDF) methods widely used in the atomic structure calculations. Over the years, therefore, the implementation of the CI and MCDF methods has led to several powerful codes including, for example, the well-known GRASP suit of codes (Jönsson et al 2007, 2013) or RELCI program (Fritzsche et al 2002) as part of the RATIP package ((Fritzsche 2012) for the calculation of relativistic atomic transition and ionization properties.

The present DBSR_CI program differs from other similar codes by additional option of using the term-dependent non-orthogonal orbitals for representation of different atomic states. The DBSR_CI is the part of the DBSR complex, uses the DBSR libraries and the B-spline representation of the one-electron radial functions. The angular coefficients for Hamiltonian matrix elements are supposed to generated by the DBSR_BREIT program, as preliminary step for the DBSR_CI runs. The DBSR_CI employs the Dirac-Coulomb Hamiltonian with additional option to include Breit interaction in the long-wave approximations. All related formulae and notations are presented in the DBSR_BREIT descriptions. Below we only provide some details specific for the DBSR_CI performance.

1. Theoretical background

The CI calculations have two main steps, the set up of the overlap and Hamiltonian matrixes and the diagonalization procedure, which in case of the non-orthogonal one-electron orbitals lead to the generalized eigenvalue problem

$$\mathbf{H}\mathbf{c} = E\mathbf{O}\mathbf{c} \quad (1)$$

where \mathbf{c} is the vector of expansion coefficients, \mathbf{H} and \mathbf{O} are Hamiltonian and overlap matrix in the given basis. Each matrix element H_{ij} can be represented through the one- and two-electron integral as

$$\begin{aligned} \langle \Phi(\gamma J) | H | \Phi(\gamma' J) \rangle = & \sum_{a,b} t(ab) I(a,b) \times D^{ab}(\{n\kappa\}; \{n'\kappa'\}) \\ & + \sum_{\lambda} \sum_{abcd} f^{\lambda}(ab, cd) X^{\lambda}(ab, cd) \times D^{abcd}(\{n\kappa\}; \{n'\kappa'\}) \end{aligned} \quad (2)$$

where we introduced the generalized two-body Slater integral

$$\begin{aligned} X^{\lambda}(abcd) = & \Pi^e(l_a, l_c, \lambda) \Pi^e(l_b, l_d, \lambda) R^{\lambda}(abcd) \\ & + \sum_{\nu=k-1}^{\nu=\lambda+1} \Pi^o(l_a, l_c, \nu) \Pi^o(l_b, l_d, \nu) \sum_{\mu=1}^8 s_{\mu}^{\nu\lambda}(abcd) S_{\mu}^{\nu}(abcd) \end{aligned} \quad (3)$$

We use an abbreviation of the quantum numbers $a = (n_a, \kappa_a)$, $b = (n_b, \kappa_b)$, ... to describe the different subshells of equivalent electrons. $D(\{nl\}, \{n'l'\})$ stands for the corresponding overlap factors, see description of DBSR_BREIT. The relativistic Slater integrals R^λ can be expanded as sum of Slater integrals containing the large and small components of the Dirac orbital

$$R^\lambda(abcd) = \int_0^\infty \int_0^\infty [P_a(r_1)P_c(r_1) + Q_a(r_1)Q_c(r_1)] \frac{r_<^\lambda}{r_>^{\lambda+1}} [P_b(r_2)P_d(r_2) + Q_b(r_2)Q_d(r_2)] dr_1 dr_2 \quad (4)$$

$$= R^\lambda(P_a, P_b; P_c, P_d) + R^\lambda(Q_a, Q_b; Q_c, Q_d) + R^\lambda(P_a, Q_b; P_c, Q_d) + R^\lambda(Q_a, P_b; Q_c, P_d)$$

It is important in the present implementation because we use the different B-spline bases for the large and small components

$$P(r) = \sum_{i=1}^{n_p} p_i B_i^{k_p}(r), \quad Q(r) = \sum_{i=1}^{n_q} q_i B_i^{k_q}(r). \quad (5)$$

As results we need to use different procedures for calculation of different integrals in (4). First, we need to generate the following four-dimensional arrays

$$R^{\lambda, pppp}(ij; i' j') = \int_0^\infty \int_0^\infty B_i^{k_p}(r_1) B_j^{k_p}(r_2) \frac{r_<^\lambda}{r_>^{\lambda+1}} B_{i'}^{k_p}(r_1) B_{j'}^{k_p}(r_2) dr_1 dr_2, \quad (6a)$$

$$R^{\lambda, qqqq}(ij; i' j') = \int_0^\infty \int_0^\infty B_i^{k_q}(r_1) B_j^{k_q}(r_2) \frac{r_<^\lambda}{r_>^{\lambda+1}} B_{i'}^{k_q}(r_1) B_{j'}^{k_q}(r_2) dr_1 dr_2, \quad (6b)$$

$$R^{\lambda, pqpq}(ij; i' j') = \int_0^\infty \int_0^\infty B_i^{k_p}(r_1) B_j^{k_q}(r_2) \frac{r_<^\lambda}{r_>^{\lambda+1}} B_{i'}^{k_p}(r_1) B_{j'}^{k_q}(r_2) dr_1 dr_2, \quad (6c)$$

$$R^{\lambda, qpqp}(ij; i' j') = \int_0^\infty \int_0^\infty B_i^{k_q}(r_1) B_j^{k_p}(r_2) \frac{r_<^\lambda}{r_>^{\lambda+1}} B_{i'}^{k_q}(r_1) B_{j'}^{k_p}(r_2) dr_1 dr_2. \quad (6d)$$

The four-dimensional arrays, $R^\lambda(ij; i' j')$, effectively represents the Slater integral in the B-spline basis. For specific orbitals, the Slater integral can then be obtained as sum

$$R^\lambda(a, b; c, d) = \sum_i \sum_j \sum_{i'} \sum_{j'} a_i b_i c_i d_j R^\lambda(ij; i' j') \quad (7)$$

where the a_i, b_i, c_i, d_i are the B-spline expansion coefficients for corresponding large or small components of orbitals a, b, c, d , depending on the type of integral under consideration.

The similar procedure is used for Breit integrals $S^\lambda(abcd)$. All Breit integrals S_k in (3) can be reduced to the two type of integrals:

$$S^\lambda(P_a P_b Q_c Q_d) = \int_0^\infty dr_1 \int_{r_1}^\infty dr_2 \left[P_a(r_1) P_b(r_2) \frac{r_1^\lambda}{r_2^{\lambda+1}} Q_c(r_1) Q_d(r_2) \right], \quad (8)$$

and

$$S^\lambda(P_a Q_b Q_c P_d) = \int_0^\infty dr_1 \int_{r_1}^\infty dr_2 \left[P_a(r_1) Q_b(r_2) \frac{r_1^\lambda}{r_2^{\lambda+1}} Q_c(r_1) P_d(r_2) \right]. \quad (7)$$

In B-spline basis, we need then the following four-dimensional arrays

$$S^{\lambda, ppqq}(ij, i' j') = \int_0^\infty dr_1 \int_{r_1}^\infty dr_2 \left[B_i^{k_p}(r_1) B_j^{k_p}(r_2) \frac{r_1^\lambda}{r_2^{\lambda+1}} B_{i'}^{k_q}(r_1) B_{j'}^{k_q}(r_2) \right], \quad (8a)$$

$$S^{\lambda, pqqp}(ij, i' j') = \int_0^\infty dr_1 \int_{r_1}^\infty dr_2 \left[B_i^{k_p}(r_1) B_j^{k_q}(r_2) \frac{r_1^\lambda}{r_2^{\lambda+1}} B_{i'}^{k_q}(r_1) B_{j'}^{k_p}(r_2) \right]. \quad (8b)$$

Computation of integrals is the most time-consuming part of the calculations. The using the non-orthogonal term-dependent orbital lead to big amount of two-electron integrals, with different multipole indexes and types. We cannot predict in advance all possible integrals involved into set up of Hamiltonian matrix, therefore in DBSR_CI program all Slater and Breit integrals calculated dynamically. To minimize the computations of relevant B-splines arrays (6) and (8), the program first creates the list of all integrals involved. Then this list is ordered according the multipole index and type of integrals, and only then we calculated the radial integrals and update the corresponding matrix elements. Such procedure allows to avoid the repeating calculations of B-spline arrays and considerably reduces the time of calculations.

For final stage of calculations, the diagonalization of the Hamiltonian matrix, we use standard LAPACK routines.

2. Structure and data flow

The block diagram of the DBSR_CI program, along with the data flow, is shown in Fig. 1. The program first read all input parameters if any from the command line, see section 4. All input parameters are optional except the name of case. This name defined the associated data files: **name.c** – configuration expansion; **name.bsw** - one-electron radial functions; **name.bnk** - angular coefficients, needed to define matrix elements, see Eq.(2). All input parameters are read from the command line by the routine **read_arg**.

All input data are then read and checked in subroutine **read_data**. The **read_conf_jj** routine reads the input configurations in spectroscopic notation form file **name.c**, decodes them to the internal 'integer' representation and allocates all the main arrays with dimensions specific for a given case. Then the CSFs are sorted according to their angular and configuration symmetries. At this stage the program also checks if the associated databank **name.bnk** contains all needed angular coefficients, otherwise the program stops. Then the program reads the B-spline grid from **knot.dat** file, routine **def_grid**, and reads all one-

electron radial functions from the **name.bsw** file, routine **read_bsw**. On this stage also is checked the correspondence between **name.c** and **name.bsw** files, and are defined all one-electron overlaps.

Then the **def_blocks** routine defines the J -blocks in the input c-file, the J is the total angular momentum. The **dbs_core_pq** routine defines the core energy, the **gen_dhl_core** routine sets up the one-electron integrals B-spline matrixes corrected to interaction with core (see the DBSR_BREIT description).

The following calculations are provided for each J -value separately. The routine **setup_matrix** reads all angular coefficients from databank **name.bnk**, calculate the associated overlaps factors and puts the associated integral together with corresponding angular coefficient to the module **c_data**, subroutine **add_integrals**. Note that this module saves integrals in different blocks according their multipole indexes and types. It provides preliminary ordering the overall integral list. On this stage, the overlap matrix is also generated because requires only the overlap factors. When all angular coefficients from the databank is processed, routine **add_matrix** begin the generation of the Hamiltonian matrix. In loop other the different types of integrals, it prepare the corresponding integral list ordering also over the one-electron indexes. After that the radial integrals are evaluated, routine **Rk_evaluate**, and the results are added to the corresponding matrix elements.

On the final stage, routine **DIAG**, the Hamiltonian matrix is diagonalize and the resulting solutions are placed in the output file **name.j**. It is the mane results of the calculations.

DBSR_CI

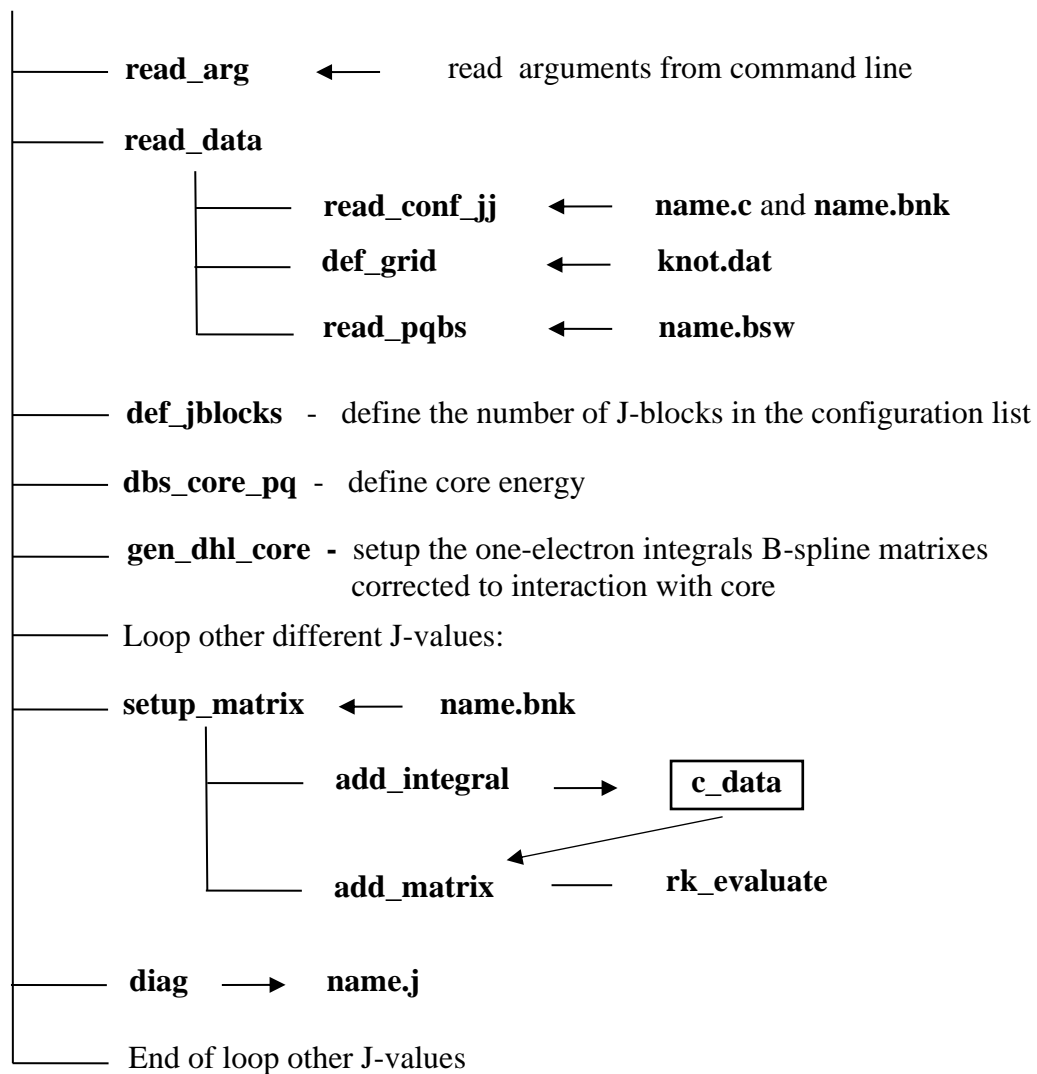


Fig. 1. Block diagram for the program DBSR_CI and data flow (see text).

3. Data files

name.c	File type: formatted sequential input. Created by genjterm or written by other user programs. Description: contains configuration list for the given case.
name.bnk	File type: unformatted sequential input. Created by DBSR_BREIT. Description: databank for angular coefficients.
name.bsw	File type: unformatted sequential input. Created by DBSR_HF, rw_bsw or other user programs. Description: contains one-electron radial functions in the B-spline representation.
knot.dat	File type: formatted sequential input. Created by DBSR_HF, DBSR_PREP or other user programs. Description: contains description of the B-spline knots grid and nuclear model.
name.j	File type: formatted sequential output (main result). Created by DBSR_CI. Description: contains configuration expansion for the obtained solutions.
name.ci	File type: formatted sequential output. Created by DBSR_CI. Description: : running information.

3.1 Format of the name.j file.

This file contains three key-words:

ncfg = ... - number of configuration states (first line)
nsol = ... - number of solutions (last line)
Solutions: - define the beginning of solution records.

Each solution is defined by three records:

1. **i, Label** - index and label of solution, format (i9,2x,a64)
2. **E, 2J, parity** - energy (in a.u.), 2J-value, parity (\pm), format (f16.8,2i8)
3. **C(1:ncgf)** - expansion coefficients, format (6f20.15)

Note that working with file, it is convenient to define first the key-words parameters using the routines **read_ipar** or **Ifind_position** from the library ZCOM.

4. Input parameters

Input parameters can be provided only in the command line in the format ‘*parameter=value*’ (except **name** of the case, which is given as simple one-word argument) . All data except name of case are optional (their default values are indicated in the brackets).

name	name of of case; it is supposed that name.c , name.bsw and name.bnk exist
msol [0]	if $\neq 0$, maximum number of solutions for output
nzero [0]	zero-order dimension ($0 \rightarrow$ all configurations)
E_{max} [0]	if $\neq 0$, restricts the output solutions if energy $> E_{\text{max}}$
mbreit [0]	if $= 1$, the Breit interaction will be included
debug [0]	if $\neq 0$, additional running information will be outputted
eps_c [1.d-10]	tolerance for angular coefficients
eps_{det} [1.d-10]	tolerance for determinant factors
eps_{ovl} [1.d-10]	tolerance for one-electron overlaps
eps_o [0.1]	tolerance for configurations overpals
c_data module parameters:	
mk [9]	maximum multipole index
nb [9]	number of blocks
mb [9]	size of block
kb [9]	number of blocks for one type of integrals

References

Jönsson P, He X, Froese Fischer C and Grant I P 2007 *Comput. Phys. Commun.* **177** 597

The grasp2K relativistic atomic structure package

Jönsson P, Gaigalas G, Bieroń J, Froese Fischer C and Grant I P 2013 *Comput. Phys. Commun.* **184** 2197

New version: Grasp2K relativistic atomic structure package

Fritzsche S, Froese Fischer C and Gaigalas G 2002 *Comput. Phys. Commun.* **148** 103

RELIC: A program for relativistic configuration interaction calculations

Fritzsche S 2012 *Comput. Phys. Commun.* **183** 1525

The Ratip program for relativistic calculations of atomic transition, ionization and recombination properties

Additional notes:

CI and perturbation theory

Let x be the CSF expansion vector for a variational calculation where $H^{00}x = E^0x$ defines the zero-order approximation with $x^t S x = 1$. The CSFs in this expansion are the zero-order set of CSFs. Let y be the CFS expansion vector for the first_order approximation determined to first order. Then the CI problem for the combined set is

$$\begin{bmatrix} H^{00} & H^{01} \\ H^{10} & H^{11} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = (E^0 + E^1) \begin{bmatrix} S^{00} & S^{01} \\ S^{10} & S^{11} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \quad (\text{xxx})$$

where H^{01} is the matrix of interaction matrix elements between the zero-order and first-order sets, and H^{11} the interaction matrix elements within the first-order set. In a first-order calculation the latter may be replaced by the diagonal matrix D for which $D_{ii} = H_{ii}^{11}$. The same for the overlap matrix S with diagonal elements $C_{ii} = S_{ii}^{11}$. From the first line of equation (xxx) we get

$$\begin{aligned} H^{00}x + H^{01}y &= (E^0 + E^1)(S^{00}x + S^{01}y) \\ H^{01}y &= E^1(S^{00}x + S^{01}y) + E^0 S^{01}y \\ E^1 &= \frac{x^t H^{01}y - E^0 x^t S^{01}y}{(1 + x^t S^{01}y)} \end{aligned} \quad (\text{xxx})$$

and from second equation (xxx) we get

$$\begin{aligned} H^{10}x + Dy &= (E^0 + E^1)(S^{10}x + Cy) \\ ((E^0 + E^1)C - D)y &= H^{10}x + (E^0 + E^1)S^{10}x \end{aligned} \quad ()$$

Suppose $E^1 \ll E^0$, then

$$y = (E^0 - D)^{-1}(H^{10}x - E^0 S^{10}x) \quad ()$$

Once the first-order corrections have been determined, the CFS expansions can be **normalized** and used in the calculation of other atomic properties.