DBSR_MCHF: A B-spline Dirac-Hartree-Fock program

Oleg Zatsarinny[a,*], Charlotte Froese Fischer[b]

[a] Department of Physics and Astronomy, Drake University, Des Moines, IA 50311, USA
[b] National Institute of Standards and Technology Gaithersburg, MD 20899-8420, USA
* Corresponding author.
E-mail address: oleg.zatsarinny@drake.edu

Abstract

A B-spline version of a general multiconfiguration Dirac-Hartree-Fock program is described.

Program summary

Program title: DBSR_MCHF version 1.00
Catalogue identifier: XXXX_v1_0
Program summary URL:http://cpc.cs.qub.ac.uk/summaries/XXXX_v1_0.html
Program obtainable from: CPC Program Library, Queen's University, Belfast, N. Ireland
Licensing provisions: Standard CPC license, http://cpc.cs.qub.ac.uk/licence/licence.html
No. of lines in distributed program, including test data, etc.: XXXXX
No. of bytes in distributed program, including test data, etc.: XXXXX
Distribution format: tar.gz
Programming language: Fortran 95
*Compute*r: Any system with a Fortran 95 compiler. Tested on Intel(R) Core(TM) i7-3632QM, 2.20 GHz
Operating system: Any system with a Fortran 95 compiler
Classification: XX
External routines: DBSR_BREIT, LAPACK (http://www.netlib.org/lapack/)
Nature of problem: Relativistic Dirac–Hartree-Fock wavefunctions are determined for atoms in a bound state. These wavefunctions may be used to predict a variety of atomic properties.
Solution method: The radial functions for large and small components of the one-electron spinor are expanded in B-spline bases. The variational principle applied to an energy functional that includes Lagrange multipliers for orthonormal constraints defines the Dirac–Hartree-Fock matrix for each orbital. Orthonormal transformations for a stationary solution were applied and Lagrange multipliers eliminated through projection operators.
Restrictions: There is no restriction on calculations for atomic states with shells whose angular momenta are less than or equal to 9/2.
Unusual features: The program allows the consideration of a few atomic states simultaneously. A simple interface through the command-line arguments allows the user to run the program with minimal initial preparations.
Running time: From a few seconds to a few hours depending on the size of the multiconfiguration expansion under consideration.
References:

1. Introduction

The multiconfiguration Hartree-Fock programs, both non-relativistic, atsp2K (Froese Fischer et al 2007), or relativistic versions, GRASP2K (Jönsson et al 2007, 2013), are all based on methods that implement and extend the numerical procedures described by Hartree (1947) far ago. Hartree viewed the problem as a system of second-order, homogeneous, integro-differential equations. The SCF process replaced these equations by a system of second-order differential equations where exchange, Lagrange multipliers and configuration interaction contributions defined the non-homogeneous part of the differential equation. Such equations have quite different mathematical properties from those of the original integro-differential, non-linear eigenvalue equations. This procedure deals with multiple off-diagonal Lagrange multipliers arising from orthogonality constraints, the fact that the self-consistent field (SCF) process might not converge, and that the equations have multiple solutions. Node counting was introduced as a means of control but node counting is an art – off-diagonal Lagrange multipliers might produce an extra node with a small amplitude in the tail, and similarly intermediate orbitals with high angular momenta might have an extra node near the origin for a few iterations and then disappear as the solutions converge.

With the development of faster computers with considerably more memory, an interesting question arises – what is the best method for solving the MCHF equations? A program implementing such a method should be flexible, easy to modify, and allow the user to control the accuracy of the desired solution.

One of alternative methods for solutions of HF and MCHF equations was presented by Froese Fisher (2007) in her articles on B-splines in variational atomic structure calculations. For a detailed introduction to the background theory of B-splines and their application to atomic physics, the reader is referred to the review article by Bachau *et al* (2001). B-spline were later employed in the non-relativistic and relativistic versions HF programs SPHF (Froese Fischer 2011) and DBSR_HF (Zatsarinny and Froese Fischer 2016).

The construction of a relativistic B-spline basis is somewhat different from the nonrelativistic case. Spurious states have been observed when the Dirac equation is expanded in a single finite basis set (Drake and Goldman 1981, Johnson et al 1988), whereas similar phenomena have not been encountered in nonrelativistic calculations. This problem has been widely discussed in the literature. One possible solution was suggested by Froese Fischer and Zatsarinny (2009) based on B-spline bases with different orders for the large and small components. In relativistic calculations, it is advantageous to use a finite nuclear potential instead of the Coulomb potential with its singularity at the origin, which is difficult to handle with B-splines. However, B-splines have the advantage of flexibility with regard to the distribution of the knot points near the nucleus.

B-splines also have received wide application in scattering problems, in particular, solving the close-coupling equations by the R-matrix method. Since the R-matrix approach is based on solving the

Schrödinger equation in a box, it could be expected that B-splines would be very useful as a basis for the R-matrix approach. The effective completeness of the B-spline basis means that no additional correction (known as the "Buttle" correction) is required. The general suite of codes, BSR, based on the use of B-splines in R-matrix approach was submitted to the CPC library in 2006 (Zatsarinny 2006) and was employed in a variety of scattering and atomic-structure calculations producing much more accurate results than with standard R-matrix method (see review Zatsarinny and Bartschat 2013).

The non-relativistic BSR code was recently extended to the fully-relativistic version, DBSR, based on the Dirac-Coulomb Hamiltonian (Zatsarinny and Bartschat 2008). The fully-relativistic calculations considerably improve the accuracy for electron scattering on heavy atoms (Zatsarinny and Bartschat 2010). The important ingredient for the scattering calculations is the accurate description of the target states. As a main option, the DBSR complex of codes uses the target wavefunctions generated with the GRASP code. GRASP is an elaborate code for many-configuration DHF calculations, however, its application in DBSR calculations requires conversion of radial orbitals to a B-spline basis. For this reason, we developed the DBSR_HF program, which runs the DHF calculations, directly in the B-spline basis. At the moment, the DBSR complex is in the process of preparation for submission to the CPC library. As a preliminary step towards this submission, the present DBSR_MCHF program is submitted separately as a code that can be used outside the DBSR complex for a variety of atomic applications.

2. Dirac-Hartree-Fock equations

We use the Dirac-Coulomb (DC) Hamiltonian to describe the arbitrary N-electron atom or ion. In atomic units, the DC Hamiltonian for N electrons in a central field for a nucleus of charge Z is given by

$$H_{DC} = \sum_{i}^{N} h_D(\mathbf{r}_i) + \sum_{i<j} \frac{1}{r_{ij}}, \tag{1}$$

$$h_D(\mathbf{r}) = c\boldsymbol{\alpha} \cdot \boldsymbol{p} + \beta c^2 + V_{\text{nuc}}(r), \tag{2}$$

where $\boldsymbol{\alpha}$ and β are Dirac matrices, $\boldsymbol{p}$ is the momentum operator of the electron, $c \approx 137$ is the speed of light, and V_{nuc} is the nuclear potential. For each total symmetry J^π, with J denoting the total electronic angular momentum in a jj-coupling scheme and π indicating the parity, the total wave function is constructed using anti-symmetrized products of Dirac four-component spinors

$$\phi_{n\kappa m}(\mathbf{r}) = \frac{1}{r} \begin{pmatrix} P_{n\kappa}(r)\chi_{\kappa m}(\hat{r}) \\ iQ_{n\kappa}(r)\chi_{-\kappa m}(\hat{r}) \end{pmatrix}, \tag{3}$$

where the real and imaginary radial Pauli spinors are the large and small components, respectively, $\chi_{\kappa m}$ is the spinor spherical harmonic, and κ is the relativistic angular momentum quantum number.

In the multiconfiguration Dirac–Fock model, an atomic state is approximated by a superposition of configuration state functions (CSF)

$$\left|\psi_\alpha(PJM)\right\rangle = \sum_r c_r(\alpha)\left|\gamma_r PJM\right\rangle. \tag{4}$$

From the symmetry of free atoms with respect to a rotation or inversion of coordinates it is clear that the Hamiltonian matrix is always block-diagonal in the total angular momentum and parity quantum numbers (*PJM*). For large-scale computations, therefore, ansatz (4) can be restricted to include only those basis functions which have the same overall symmetry.

A representation of an atomic state ψ_α in the given CSF basis, i.e. the set of mixing coefficients $\{c_i\}$, is obtained by solving the secular equation

$$\det(\mathbf{H} - E_\alpha \mathbf{I}) = 0, \tag{5}$$

where E_α denotes the eigenvalue and

$$\mathbf{H} = (H_{rs}) = (<\gamma_r PJM \mid H \mid \gamma_s \overline{P}\overline{J}\overline{M}> \delta_{P\overline{P}}\delta_{J\overline{J}}\delta_{M\overline{M}}) \tag{6}$$

the Hamiltonian matrix. Note that the block symmetry of this matrix (6) is independent of the particular choice of the electron–electron interaction in the Hamiltonian as discussed earlier. In addition, since this matrix is real and symmetric, all atomic states are orthogonal for $E_\alpha \ne E_\beta$ or can be chosen in this way for $E_\alpha = E_\beta$.

Of course, the set-up and decomposition of the Hamiltonian matrix elements eventually depend on how the many-particle basis is constructed from the one-particle functions. We assume *jj*-coupled CSF for which the angular integration can be carried out analytically (see write-up of the DBSR_BREIT program). The integration over the angular variables of all electrons gives rise to a decomposition of the Hamiltonian matrix elements in the form

$$H_{rs} = \sum_{ab} t_{rs}^0(ab) < a \| h_D \| b > + \sum_k \sum_{abcd} v_{rs}^k(abcd) R^k(abcd). \tag{7}$$

where $t^o{}_{rs}(ab)$ and $v^L{}_{rs}(abcd)$ are one- and two-particle (scalar) angular coefficients and the $R^k(abcd)$ are the Slater integrals:

$$R^k(abcd) = \int_0^\infty \int_0^\infty [P_a(r_1)P_c(r_1) + Q_a(r_1)Q_c(r_1)]\frac{r_<^k}{r_>^{k+1}}[P_b(r_2)P_d(r_2) + Q_b(r_2)Q_d(r_2)]\mathrm{d}r_1\mathrm{d}r_2, \tag{8}$$

where the indices a and b refer to one-electron orbitals $(n_a\kappa_a)$ and $(n_b\kappa_b)$, respectively. The Dirac-Hartree-Fock equations now can be derived from the variation condition

$$\delta(E_\alpha(PJ) - \sum_{ab}\delta(\kappa_a,\kappa_b)\lambda_{ab} < a \mid b >) = 0 \tag{9}$$

where we introduce Lagrange multipliers λ_{ab} to ensure the orthogonality of orbitals with the same κ values. Equation (9) applies to all allowed perturbations that satisfy orthogonality constraints and the boundary conditions. For the perturbation of a single orbital the resulting MCDF differential equations have the form

$$
\begin{cases}
c\left(\dfrac{d}{dr} - \dfrac{\kappa}{r}\right)Q_a + \left(V_{MCDF} + V_{nuc} + c^2\right)P_a = \varepsilon_a P_a + \sum_{b \neq a} \varepsilon_{ab} P + X_P \\
-c\left(\dfrac{d}{dr} + \dfrac{\kappa}{r}\right)P_a + \left(V_{MCDF} + V_{nuc} - c^2\right)Q_a = \varepsilon_a Q_a + \sum_{b \neq a} \varepsilon_{ab} Q + X_Q
\end{cases}
\tag{10}
$$

where $\varepsilon_a = \lambda_{aa}/q_a$, $\varepsilon_{ab} = \lambda_{ab}/q_a$ are referred to as energy parameters, and q_a denotes the generalized orbital occupation number

$$
q_a = \sum_{r}^{n_c} c_r^2 q_a^r
\tag{11}
$$

The MCDF potentials in Eq. (10) are defined by the corresponding one- and two-electron integrals in the expression (7) and will be discussed in the next section.

3. B-spline basis

B-splines are functions designed to generate piece-wise polynomial functions for approximating arbitrary functions in some finite interval. Consider the interval $[a,b]$ divided into subintervals. The endpoints of these subintervals are given by the knot sequence $[t_i]$, $i = 1,2,...,n + k$. The B-splines of order k, $B_{i,k}(r)$, on this knot sequence are defined recursively by the relations

$$
B_{i,1}(r) = \begin{cases} 1, & t_i \leq r \leq t_{i+1} \\ 0, & \text{otherwise} \end{cases}
\tag{12.a}
$$

and

$$
B_{i,k}(r) = \frac{r - t_i}{t_{i+k-1} - t_i} B_{i,k-1}(r) + \frac{t_{i+k} - r}{t_{i+k} - t_{i+1}} B_{i+1,k-1}(r).
\tag{12.b}
$$

Each $B_{i,k}$ is defined over the interval $[t_i,t_{i+k}]$, which contains $k + 1$ consecutive knots, and is indexed by the knot where it starts. In order to confine all B-splines in the given interval $[a,b]$, the multiplicity of knots in the endpoints is usually chosen to be the maximum possible value, which is equal to the B-spline order k. The most common choice for the multiplicity at inner knots is unity, corresponding to the maximum continuity of B-spline functions inside the interval. With this choice, employed in the present program, the number of subintervals is related to the number of B-splines as $n_{int} = n + 1 - k$.

For a detailed introduction to the background theory of B-splines, the reader is referred to the review of Bachau et al (2001) and the description of the SPHF and BSR codes as examples of the numerical implementation of splines in atomic calculations.

In our implementation, we expand the radial functions for the large and small components $P(r)$ and $Q(r)$ in separate B-spline bases as

$$P(r) = \sum_{i=1}^{n_p} p_i B_i^{k_p}(r), \qquad Q(r) = \sum_{i=1}^{n_q} q_i B_i^{k_q}(r). \tag{13}$$

Both B-spline bases are defined on the same grid, with the same number of intervals, however, with different orders, k_p and k_q,. Only in this way can the calculations of various matrix elements and integrals of interest be performed with the same routines and the same computational resources as in the case of a single B-spline basis. At the same time, using the different bases avoids the appearance of pseudo-states that are a common problem for solutions of the Dirac equations in any finite-element basis. More details on this issue are discussed in our previous paper (Froese Fischer and Zatsarinny 2009). By default, $k_q = k_p + 1$.

With basis (12), the Dirac-Hartree-Fock equations (7) are reduced to matrix equations of the form

$$\begin{bmatrix} \mathbf{H}_{pp} & \mathbf{H}_{pq} \\ \mathbf{H}_{qp} & \mathbf{H}_{qq} \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix} = \varepsilon \begin{bmatrix} \mathbf{B}_{pp} & 0 \\ 0 & \mathbf{B}_{qq} \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix} + \begin{bmatrix} \mathbf{X}_p \\ \mathbf{X}_q \end{bmatrix}, \tag{14}$$

where $\mathbf{p}$ and $\mathbf{q}$ are vectors of the expansion coefficients for the given orbital, and $\mathbf{B}$ is the B-spline overlap matrix $\langle B_i | B_i \rangle$. For more detailed description of the Hamiltonian matrix $\mathbf{H}$, let us introduce the matrices for the one-electron terms

$$\mathbf{D}_\kappa^\pm \rightarrow \left\langle B_i \left| \pm \frac{d}{dr} + \frac{\kappa}{r} \right| B_j \right\rangle, \tag{15.a}$$

$$\mathbf{V}_{nucl} \rightarrow \left\langle B_i \left| V_{nucl}(r) \right| B_j \right\rangle, \tag{15.b}$$

and the four-dimensional arrays for Slater integrals between individual B-splines

$$R^k(ij;i'j') = \int_0^\infty \int_0^\infty B_i(r_1) B_j(r_2) \frac{r_<^k}{r_>^{k+1}} B_{i'}(r_1) B_{j'}(r_2) dr_1 dr_2. \tag{16}$$

The direct and exchange potentials, V^{dir} and V^{exc} can then be expressed through the matrices obtained by convolution of these arrays over two variables as

$$\mathbf{V}_a^{dir}: \qquad R^k(abac) \rightarrow R^k(.\,b\,.\,c) = \sum_{jj'} b_j c_j R^k(ij;i'j'), \tag{17.a}$$

$$\mathbf{V}_a^{exc}: \qquad R^k(abca) \rightarrow R^k(.\,bc\,.\,) = \sum_{i'j} b_j c_i R^k(ij;i'j'). \tag{17.b}$$

The interaction potentials X_p, X_q arise from the one-electron integrals and from R^k integrals with one entrance of the orbital under consideration:

$$\mathbf{X}_a^{int}: \quad <a\,\|\,h_D\,\|\,b> \rightarrow h_D(.b\ ) = \sum_j b_j h_D(ij)\,. \tag{18.a}$$

$$\mathbf{X}_a^{int}: \quad R^k(abcd) \rightarrow R^k(.b\,c\,d\ ) = \sum_{i'jj'} b_j c_i d_{j'} R^k(ij;i'\,j')\,. \tag{18.b}$$

More details are given by Froese Fischer (2007) and Zatsarinny (2006) in the non-relativistic case. Note that the direct integrals are defined only by diagonal banded blocks with respect to small and large components, but exchange requires full matrixes and numerically takes most of the computation time. Because we have two components of different order, calculations of interaction matrices are approximately 4 times more time consuming than those in the nonrelativistic calculations. According to the definition of Slater integrals (5), we actually have four different integrals, which we denote as $R^k(PP;PP)$, $R^k(QQ;QQ)$, $R^k(PQ;PQ)$, and $R^k(QP;QP)$, and because we use different bases for P and Q components, they all are different. Finally, the Hamiltonian matrix, $\mathbf{H}^a$ for orbital a, has the form

$$\mathbf{H}^a = \begin{bmatrix} \mathbf{V}_{pp}^{nucl} & c\mathbf{D}_{pq}^{-} \\ c\mathbf{D}_{qp}^{+} & \mathbf{V}_{qq}^{nucl} - 2c^2 B_{qq} \end{bmatrix} + \begin{bmatrix} \mathbf{V}_{pp}^{dir} & 0 \\ 0 & \mathbf{V}_{qq}^{dir} \end{bmatrix} + \begin{bmatrix} \mathbf{V}_{pp}^{exc} & \mathbf{V}_{pq}^{exc} \\ \mathbf{V}_{qp}^{exc} & \mathbf{V}_{qq}^{exc} \end{bmatrix}, \tag{19}$$

where the direct and exchange potentials are represented by matrices obtained by summation and convolution of the corresponding Slater integrals (16). As an example, for the PP sub-block we have

$$\mathbf{V}_{pp}^{dir} = \sum_{rs} c_r c_s \sum_{b,c} \sum_k f_{rs}^k(a,b,a,c)\Big(R^k(.\,P_b\,.\,P_c) + R^k(.\,Q_b\,.\,Q_c)\Big), \tag{20}$$

$$\mathbf{V}_{pp}^{exc} = \sum_{rs} c_r c_s \sum_{b} \sum_k f_{rs}^k(a,b,c,a) R^k(.\,P_b\,P_c\,.)\,, \tag{21}$$

where coefficients f_{rs}^k are related with angular coefficients v_{rs}^k in the expansion (11).

When two orbitals, connected by an orthogonality condition, are varied simultaneously as in a unitary transformation, the energy must also be stationary. This results in an implicit condition on the solution of the differential equations. At such a solution, orthogonality of orbitals can be achieved through the use of the projection operator method (Bentley 1994), what leads to a modification of the Hamiltonian matrix (19), one modification for each orthogonality constraint. If, for example, we have the constraint $<a\,|\,b>=0$, the modification of the Hamiltonian matrix for orbital a is

$$\mathbf{H}^a \rightarrow (1 - \mathbf{B}\,\mathbf{b}\mathbf{b}^t)\mathbf{H}^a(1 - \mathbf{b}\mathbf{b}^t\mathbf{B})\,, \tag{22}$$

where $\mathbf{B}$ is the B-spline overlap matrix. This method allows us to work without the explicit presence of Lagrange multipliers in Eq. (13).

In general, the MCHF equations (13) are the matrix equations with the right-hand side. Using the B-spline basis allows us to convert these equations to the pure eigenvalue problem. Indeed, we may multiply the interaction potentials X on the overlap integral $<a\,|\,a>$, which is close to the unity due to the normalization requirements. Then we have

$$\mathbf{X}_a^{int}: \quad R^k(abcd) \to R^k(\,\boldsymbol{.}\,b\,c\,d\,) = R^k(\,\boldsymbol{.}\,b\,c\,d\,) < a\,|\,a >= R^k(\,\boldsymbol{.}\,b\,c\,d\,)B(a\,\boldsymbol{.}\,)\,|\,a >, \qquad (23)$$

where $B(a\,\boldsymbol{.}\,)$ is the B-spline overlap matrix convoluted for one dimension. In results, the interaction potentials also can be represented in the matrix form (actually, this matrix is an external product of two vectors). In this case, the SCF procedure is reduced to diagonalization of Hamiltonian matrixes, first for orbitals and then for expansion coefficients:

$$\mathbf{H}^a \mathbf{a}_i = \varepsilon_a \mathbf{B} \mathbf{a}_i; \quad \mathbf{a}_i^T = (\mathbf{p}_i^T, \mathbf{q}_i^T), \qquad i = 1, 2, ..., n_{orb} \qquad (24)$$

$$\mathbf{H}^{DC} \mathbf{c} = E\mathbf{c}; \quad \mathbf{c}^T = (c_1, c_2, ..., c_{n_c}) \qquad (25)$$

DBSR_MCHF

get_case ← read arguments from **command line** and/or from **name.inp**

read_conf_jj ← **name.c**

def_energy_coef ← **int_bnk | name.bnk** ← DBSR_BREIT

get_spline_param ← **knot.dat | name.knot**

get_estimates ← **name.bsw** or any file defined by argument **inp**

SCF_MCHF - self-consistent field calculations

loop over varied orbitals

mchf-matrix

solve_eq - if **neuton** = 0

method_direct - rhs $\neq 0$ and method = 1

method_inverse - rhs $\neq 0$ and method = 2

method_eigenvalue - rhs=0

solve_nr - if **neuton** > 0

diag - diagonalization of the CI matrix

check the convergence

output_jfile ⟶ **name.j**

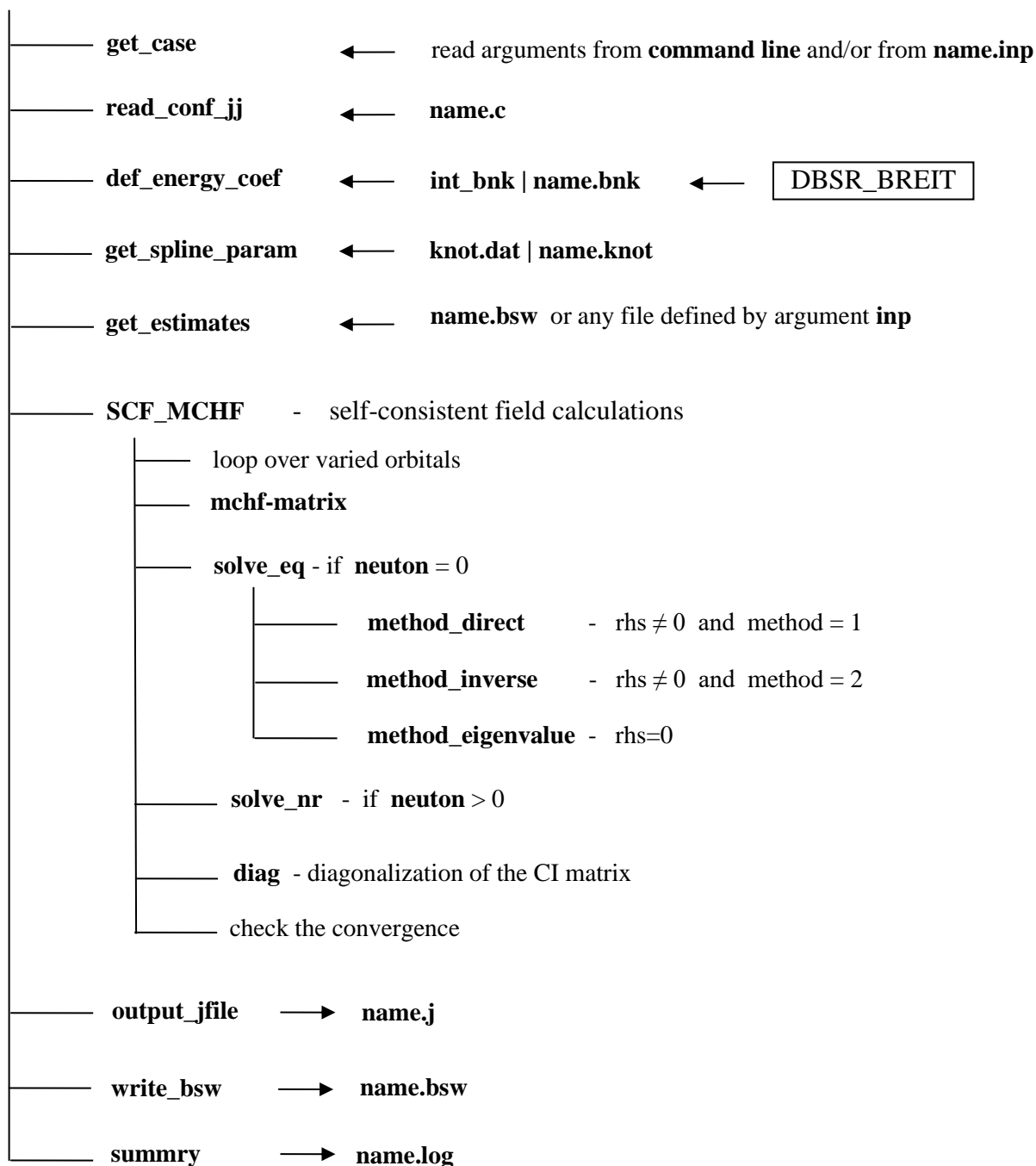**write_bsw** ⟶ **name.bsw**

summry ⟶ **name.log**

Fig. 1 Block diagram for the program DBSR_MCHF and data flow.

4. Program structure

The block-scheme of the program is shown in Fig. 1, along with optional input/output data files. The primary steps of the program consist of the analysis of input parameters by the module **get_case**, the calculation of all needed angular coefficients by the module **def_energy_coef**, the definition of the grid and spline parameters by the module **def_spline_param**, the reading of the initial estimations by the module **get_estimates**, the solution of mchf equations by the main module **solve_mchf,** followed by the output of resulting wavefunctions and expansion coefficients.

4.1. Input parameters

All input parameters are optional except the name of case. Each calculation is supposed to have its own name which defines the associated data files: **name.c** – configuration expansion; **name.bsw** - one-electron radial functions; **name.bnk** - angular coefficients, needed to define matrix elements. Other input parameters can be given either on the command line or in the file `name.inp` if they differ from the default values. Any parameter from the input file can be redefined on the command line. All parameters have their default values, so any missing parameters will not halt the calculations (unless the **name** of the case is missing). Formal detailed description of all input parameters is given in the section 7. The `name.inp` file is optional; if this file is absent the program will create **name.inp** file to show the default parameters used in the calculations. For user convenient, this file is also added with information about different options for the input parameters and input/output data files.

4.2 Spline parameters and nuclear description

As we can see from the definition of B-splines in Eq. (10), B-splines are fully determined by the given knot sequence and the spline order k. Although there is complete freedom in choosing the mesh of knots, the optimal choice depends on the type of the result we are interested in. With splines, we can create any composite grid when needed, but too dense a grid can lead to a rapid saturation of the computational resources. Experience has shown that the most appropriate choice is a logarithmic grid that reflects the exponential behavior of atomic orbitals. On the other hand, in continuum calculations, the wavelength of the scattering particle cannot be smaller than the grid step; otherwise, the B-spline basis hardly describes the oscillating behavior of the wavefunction. In the present implementation, we use a mixed sequence which can easily be adjusted to the task under consideration. Detailed description of possible knot grids and their parameters is given in write-up of the DBSR_HF program.

Consistent with the BSR and DBSR suites of codes, the B-spline parameters are given in a separate file, **knot.dat**, by default. This file can be reassigned with input parameter **knot**. If this file is absent, the DBSR_HF program will create the **name.knot** file with suggested parameters for the given case. Separation of atomic parameters from the knot parameters allows us to keep the same grid for different calculations. However, we cannot completely separate the atomic parameters from the spline parameters due to different models for the nuclear potential. For this reason, the **knot.dat** file also includes the nuclear parameters so that the knot grid depends on the nuclear model. The present program supports three nuclear models, defined by the parameter **nuclear** with possible values of **point, Fermi** or **uniform** (see write-up of DBSR_HF). This allows us to include the effect of the finite nuclear size by replacing the nuclear Coulomb potential, $-Z/r$, by the potential with a finite charge distribution.

4.3 Angular coefficients

In DBSR_MCHF, the orbitals are optimized so that an energy functional defined as a linear combination of energies of selected atomic states has a stationary energy value. The weight coefficients are defined by the input parameter **eol**, in the same way as in the GRASP program (see also sections 5 and 6). The energy functional is expressed through the one- and two-electron radial integrals as in the Eq. (7). All needed angular coefficients $t_{rs}^0(ab)$ and $v_{rs}^k(abcd)$ are supposed to be prepared in the file **name.bnk**. These coefficient are calculated by the program DBSR_BREIT (..). If the data file **name.bnk** is absent, the program looks for the **int_bnk** file, which may be used for the several cases. If both of these files are absent, or the files do not contain all needed angular coefficients, the DBSR_MCHF program will stop and ask to run DBSR_BREIT program first.

4.4 Initial estimations

Initial estimations for the radial functions are obtained either from screened hydrogen functions or from the file specified by the **inp** parameter (default is the **name.bsw** file). This file contain the B-spline expansions for the large and small components, P and Q, obtained from previous calculations. These files are supposed have the default extension **.bsw** and contain, along with the radial functions, also the B-spline grid used in their calculation. If this B-spline grid do not coincide with the grid used in the current calculations, the corresponding B-spline expansions will be converted to the current grid. If the input file has the extension **.w**, it will mean that the radial functions are written in the GRASP format and they will be converted to the B-spline representation. (?)

4.5 SCF procedure

The central routine of the program is the **SCF_MCHF** subroutine, which runs the SCF procedure for solving the MCDF equations (14). The SCF procedure in the B-spline basis is reduced to setting up each of the orbital matrices (19), applying the projection operators (if needed for orbital orthogonalization), and solution the MCDF equation or diagonalization of the generalized eigenvalue problem (13). In the last case we obtain the entire spectrum of one-electron Hamiltonian, including both the positive and negative energies. The present program selects the needed solutions from the middle of the full spectrum, namely the lowest eigenvalue in the positive part of the spectrum, taking into account the orthogonality constraints. Then the program diagonalize the atomic Hamiltonian to obtain the expansion coefficients. Diagonalization is performed by a call to the LAPACK routine DSYGV (http://www.netlib.org/lapack/). …………….

4.6 Breit interaction and QED corrections

The Breit interaction and QED corrections are not included in the present version. We decided to not overload the program, considering the main goal of the program to generate the one-electron radial functions (physical and correlated). These functions then can be used to construct the large-scale configurations expansions for following CI calculations with program DBSR_CI. At this stage, we may included both Breit and QED corrections. The one possible advantage of such approach is that we may generate different sunset of correlated functions for different type of correction such as valence, core-valence, inner-core or others. These subsets may be not orthogonal that supposes more quick convergence. Note that DBSR_CI program may treat any level of orbital orthogonality.

4.7 Output of results

The essential information about the calculation is summarized in the **name.log** file, which contains the parameters of the computational model, B-spline parameters, the final results of the SCF iteration process, convergence estimates, and final energies in atomic unites (au) and (eV) . Below is a typical example of the final results for ….…………………………………….???

5 Examples

To begin the calculations for a given atom, the user should first prepare the file **name.c** with the configuration expansions. It can be done with the genjconf and genjterm utilities from the DBSR complex, or with corresponding routines from the GRASP complex. Then we need run the DBSR_BREIT program to obtain the needed angular coefficients, **file name.bnk**. It is the only files needed to begin the calculations. All other input file are optional.

As example, we prepared the file **Be.c** for the Be ground state:

………………………………….

6 List of input parameters

Input parameters can be provided in the command line or in file **name.c** in the format '*paramter=value*' (except **name** of the case, which is given as a simple one-word argument). The key-word format for input parameters simplifies their preparation: there is no restrictions on their position in the input file or in command line. All data except name of case are optional (their default values are indicated in the brackets).

name	name of the case
atom	symbolic name of the atom under consideration.
z	nuclear charge; can be real number (fraction) if needed for some special cases.
awt	atomic weight, by default is chosen for the most abundant or most stable isotope.
rrms	the root-mean-square radius of nuclear, defined from its atomic weight or taken from the compilation of Angeli and Marinova (2013).
varied [all]	list of one-electron orbitals included in the SCF optimization; accepted value: **all**, **none,** a **list** of orbitals separated by commas, or **number**, where **number** indicates the number of orbitals involved in the SCF procedure (from the end of the list) .
eol [5]	indicates the mode for the state weights: =1 - equally weighted; =5 - statistically weighed; =9 - defined by the user in the `name.inp` file through the parameters **levels** or **weights** (see below).
levels	j-block index and list of states to be optimized; repeat for each block.
weights	j-block index, level index, and corresponding weight; repeat for each level to be optimized.
scf_tol [1.0E-10]	energy convergence tolerance.
orb_tol [1.0E-7]	orbital convergence tolerance.
end_tol [1.0E-6]	orbital tail cut-off.

max_it [25]   maximum number of iterations .

metod [1]   method for solution of MCHF equations

irs [0]   method for treatment of right-han side,

rotate [0]   rotations during optimization (0/1 →no/yes).

newton [0]   using of the newton method (0/1 →no/yes).

debug [0]   additional debug output (0/1 →no/yes).

n_corr ?

ipzero[-1]   number of initial zeroes for P-component

iqzero[-1]   number of initial zeroes for Q-component

jpzero[1]   number of ending zeroes for P-component

jqzero[1]   number of ending zeroes for Q-component

7 Data files

The program employs the name-driven file convention. All filenames have the structure **name.extension**, where **name** identifies the specific calculation. There is also an option for user-defined filenames that is defined by the corresponding keywords in the command line and can be useful in some specific cases. For example, to keep the knot grid consistent with the previous calculations it is enough to indicate the corresponding knot file through the parameter **knot**, or rename it as the default generic name **knot.dat**. In this case, all associated calculations for the given atom or ion will be carried out with the same grid. The list of data files and their description is given in table 1. The first column present the default names, second column indicate the parameters which can be used to redefine these files, the last column contain shot description of the files.

Table 1 Input/output files.

name.inp		File type: formatted sequential input (optional). Written by the user or created by the program if absent. Description: input parameters for the given run.
name.log		File type: formatted sequential output. Description: Summary of the running information.
name.knot	**knot=…**	File type: formatted sequential input/output. Written by the user or created by the program if absent. Description: the B-spline grid and nuclear parameters.
name.c	**c=…**	File type: formatted sequential input. Created by the user (using **genjconf** and **genjterm** utilities). Description: input configuration states in the GRASP format.
name.bnk	**bnk=…**	File type: unformatted sequential input. Created by the DBSR_BREIT program. Description: angular coefficient for Dirac-Coulomb interaction.
name.j	**j=…**	File type: formatted sequential output. Created by the program. Description: CI expansion coefficients.
name.bsw	**inp=…**	File type: unformatted sequential input (optional). Provided by the user (from the prior runs). Description: one-electron orbitals in B-spline representation
name.bsw	**out=…**	File type: unformatted sequential output. Description: one-electron radial functions in B-spline representation reoptimized in the current run.

7.1 Format of the **name.bsw** file

First record contains the B-spline grid parameters:

1. `grid_type, ns, ks, t(1:ns+ks), ksp, ksq`

`grid_type`	– type of grid
`ns`	– number of B-splines
`ks`	– order of B-splines
`t`	– knot sequence
`ksp`	– order of B-splines for the large component
`ksq`	– order of B-splines for the small component

Then for each one-electron orbital:

2. `ebs, mbs, e`

`ebs`	– orbital spectroscopic symbol
`mbs`	– number of splines in the orbital expansion
`e`	– one-electron energy parameter

3. `P(1:mbs)` – B-spline coefficients for the large component
4. `Q(1:mbs)` – B-spline coefficients for the small component

8 Program installation and testing

The DBSR_MCHF program is a part of the DBSR complex and extensively uses common routines from the DBSR libraries: ZCONFJJ, DBS, and ZCOM. ZCONFJJ deals with the CSF description and calculation of the angular coefficients, DBS contains subroutines for different operations with B-splines, including two-electron integrals, and ZCOM contains commonly used subroutines, such as reading arguments or arrays ordering. For convenience of independent use, the DBSR_MCHF program is submitted as a stand-alone program, which contains routines from the libraries, collected in separate files, dbsr_zconfjj.f90, dbsr_dbs.f90, and dbsr_zcom.f90 (?). The DBSR_MCHF program also uses the linear algebra LAPACK and BLAS libraries which are widely used in different applications and freely available at the site: http://www.netlib.org/lapack/ . It is recommended that users employ the LAPACK and BLAS libraries optimized for the given computer. Makefile contains all needed information for the compilation.

The program is modular in design and some features can be easily modified. In particular, the default input parameters and default file names are defined in the **module dbsr_mchf** in the file **MOD_dbsr_mchf.f90.** As mentioned earlier, the user may wish to include a new subroutine to generate a grid optimized for a particular application in the file **DBS_grid.f90** from ZCONFJJ library .

The program was thoroughly tested for different atoms and ions by comparing with GRASP [15] and MDFGME [30] results. Close agreement was found both for the total Colomb energy and the Breit and QED corrections. We should note however that total energies for heavy atoms noticeably depend on the nuclear model. All the above examples use the Fermi nuclear charge distribution, and the resulting total energies may strongly depends on the root-mean-square nuclear radius used.

No special tests are provided with the present write-up. To check the program, the user is suggested to repeat the examples given in the text. If the run for a specific case is failed, it is advised to delete all files with the given **<name>**, otherwise they may contain wrong information which will confuse the program for the runs with the same **<name>.**

9 Conclusions
………………………………..

References

Angeli I and Marinova K P 2013 *Atomic Data and Nuclear Data Tables* **99** 93
   Table of experimental nuclear ground state charge radii: An update
Bachau H, Cormier E, Decleva P, Hansen J E and Martin F 2001 *Rep. Prog. Phys.* **64** 1815
   Applications of B-splines in atomic and molecular physics
Bentley M 1994 *J. Phys. B* **27** 637
   Orthogonality constraints in finite basis set calculations
Drake G W F and Goldman S P 1981 *Phys. Rev. A* **23** 2093
   Appication of discrete-basis-set methods to the Dirac equation
Froese Fischer C, Brage T, and Jönsson P (1997) *Computational Atomic Structure. An MCHF Approach*
   (Institute of Physics Publishing: Bristol/Philadelphia)
Froese Fischer C (2007) *Advances in Atomic and Molecular Physics* **55** 539
   B-splines in variational atomic structure calculations
Froese Fischer C, Tachiev G, Gaigalas G and Godefroid M 2007 *Comput. Phys. Commun.* **176** 559
   An MCHF atomic-structure package for large-scale calculations
Froese Fischer C 2011 *Comput. Phys. Commun.* **182** 1315
   A B-spline Hartree–Fock program
Froese Fischer C and Zatsarinny O 2016 *Comput. Phys. Commun.* **202** 287
   A B-spline Galerkin method for the Dirac equation
Hartree D 1947 *Rep. Prog. Phys.* **11** 113
   The calculation of atomic structures
Johnson W R, Blundell S A and Sapirstein J 1988 Phys. Rev. A 37 307
   Finite basis sets for the Dirac equation constructed from B splines
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.* **182** 2197

New version: Grasp2K relativistic atomic structure package

Zatsarinny O 2006 *Comp. Phys. Commun.* **174** 273

BSR: B-spline atomic R-matrix codes

Zatsarinny O and Bartschat K 2008 *Phys. Rev. A* **77** 062701

Relativistic B-spline R-matrix method for electron collisions with atoms and ions: Application to low-energy electron scattering from Cs

Zatsarinny O and Froese Fischer C 2009 *Comput. Phys. Commun.* **180** 879

DBSR_HF: A B-spline Dirac–Hartree–Fock program

Zatsarinny O and Bartschat K 2013 *J. Phys. B* **46** 112001

The B-spline R-matrix method for atomic processes: application to atomic structure, electron collisions and photoionization

Zatsarinny O and Bartschat K 2010 *J. Phys. B* **43** 074031

Benchmark calculations for near-threshold electron-impact excitation of krypton and xenon atoms