DBSR_HF: 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

User Guide – Extracted section of the published manuscript

4. Program structure

The block-scheme of the program is shown in the Figure 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 Dirac-Hartree-Fock equations by the main module **SOLVE_HF,** followed by the output of resulting wavefunctions in different formats. The main blocks will be described separately in the following subsections.
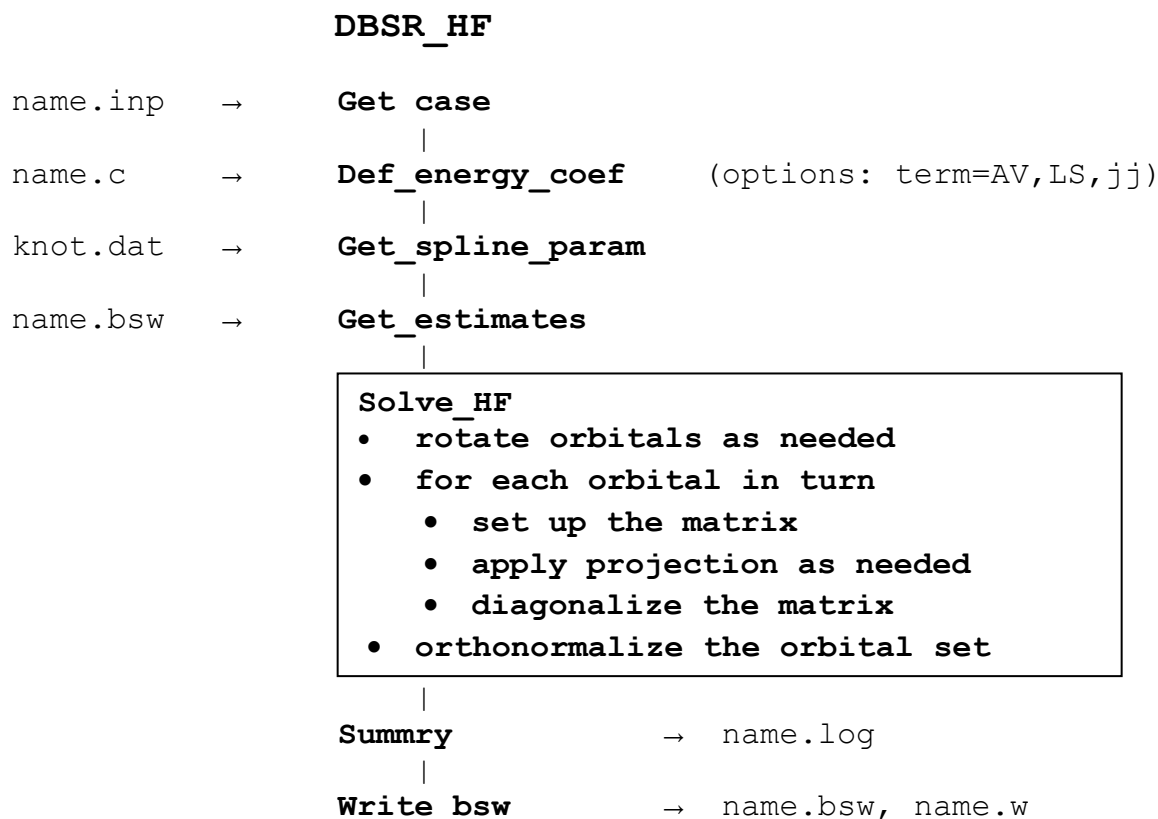
```
                    DBSR_HF

  name.inp    →     Get case
                        |
  name.c      →     Def_energy_coef      (options: term=AV,LS,jj)
                        |
  knot.dat    →     Get_spline_param
                        |
  name.bsw    →     Get_estimates
                        |
            ┌───────────────────────────────────────────┐
            │ Solve_HF                                   │
            │  •  rotate orbitals as needed              │
            │  •  for each orbital in turn               │
            │       •  set up the matrix                 │
            │       •  apply projection as needed        │
            │       •  diagonalize the matrix            │
            │  •  orthonormalize the orbital set         │
            └───────────────────────────────────────────┘
                        |
                  Summry              →   name.log
                        |
                  Write_bsw           →   name.bsw, name.w
```

4.1. Input parameters

Each calculation is supposed to have its own name, given as the first argument on the command line for which the variable name is not needed. Other input parameters can be given either on the command line or in the file **<name>.inp** if they differ from the default value. Here we have used the notation **<var>** to represent the value of a variable **var**.

In order to simplify calculations, many options are set when the name is that of an atomic symbol, also referred to as the value of the variable **atom**. In this case, the program itself prepares the input file **<atom>.inp** and runs the calculations automatically. The input file in this case contains all possible parameters with their default values, along with additional information and hints for running the program. If needed the user can change the parameters and rerun the case. A typical input file for the case where **<name>=Rb** contains the following atomic parameters:

```
atom   =  Rb
z      =  37.00
atw    =  85.47
term   =  AV
core   =  1s 2s 2p- 2p 3s 3p- 3p 3d- 3d 4s 4p- 4p   (or simply [Kr])
conf   =  5s(1)
varied =  all
```

The file contains the atomic symbol, nuclear charge, atomic weight, a core which is one of [He], [Be], [Ne], [Mg], [Ar], [Zn], [Kr], [Cd], [Xe], [Hg], or [Rn], the configuration of electrons outside the core, and an indication of which orbitals should be varied. All input parameters are given with their key words as **parameter=value,** and any parameter from the input file can be redefined on the command line with the same construction. The key-word format for input parameters simplifies their preparation: there is no restrictions on their position in the input file. 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.

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. The knot points t_i are defined by the following relations:

$$
\begin{aligned}
&t_i = 0 && \text{for } i = 1,...,k \\
&t_{i+1} = t_i + h_i && \text{for } i = k,...,k + m_i \\
&t_{i+1} = t_i(1 + h_e) && \text{while } t_{i+1} - t_i < h_{max} \\
&t_{i+1} = t_i + h_{max} && \text{for the rest of } t_i \text{ while } t_i < r_{max} \\
&t_i = t_{max} && \text{for } i = n_s + 1, n_s + k.
\end{aligned}
\tag{23}
$$

We see that the knots are defined on the finite interval $(0, r_{max})$. Density of knots near the origin is defined by the initial step h_i, the exponential growth in the middle-radii region is defined by the factor h_e, and the maximum step at high radii is defined by h_{max}. By varying these parameters, we may easily change this semi-logarithmic grid to the equally-spaced grid or to a completely logarithmic grid (equally spaced with respect to log(r)). The number of B-splines, n_s, is a derived value, depending on all the above parameters. For bound-state calculations we can use the logarithmic grid in the outer region, whereas in scattering calculations the maximum size of the subintervals should be restricted according to the maximum energy of the scattering electron. As a practical implementation, we can use a less dense grid for the bound-state calculation and then, if needed, convert to B-spline expansions in any other grid.

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 parameters specific 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 **knot.dat** file incorporates the parameters from the file **isodata** used in the GRASP complex. An example of a **knot.dat** file for the Rb calculation with Fermi nuclear potential is the following:

```
grid_type =  1

ks =    9               !  order  of splines
ns =  113               !  number of splines
```

```
    nv =   105                    !   number of intervals
    ml =     1                    !   number of initial equal-spaced intervals

    ksp=     8
    ksq=     9

    hi =          0.25000000  !   initial step for r*zg
    he =          0.25000000  !   exponetial step size
    hmax =        1.00000000  !   maximum step size for r, given
    rmax =       50.00000000  !   maximum radius, given
    tmax =       50.53782203  !   maximum radius, actual

    atomic_number =   37.0000
    atomic_weight =   87.0000
    rrms          =    4.1989

    nuclear =  Fermi

    c_fermi =      0.9077800618470297D-04 !   in a.u.
    a_fermi =      0.9890591209340532D-05 !   in a.u.

    grid points:
    ............
```

This grid is indicated as **grid_type=1** and calculated with the routine **mkgrid_01,** based on the relations (23). Users can easily introduce their own grid type, adding the corresponding routine **mkgrid_xx.** The resulting grid points are also recorded to the **knot.dat** file. In case of extensive BSR or DBSR calculations, the user is strongly advised to first try different sets of parameters to produce the desired knot sequence with a minimum number of B-splines, because this parameter strongly affects the computation time for all subsequent calculations.

In contrast to the non-relativistic case, where solutions near the origin are polynomial in nature, there is a problem with using B-splines to approximate the $r^{\gamma-1}$ behavior of relativistic orbitals at small r. Here $\gamma = \sqrt{1-(Z\alpha)^2}$ and approximating this behavior in a B-spline basis with sufficient accuracy requires a very fine grid at small values of r. Thus extra care must be taken in this case, particularly for s or p_- orbitals. To some extent, this problem is eliminated when the potential is modified to account for the finite size of the nucleus. This requires, however, that the first few grid points be placed at very small distances and well inside the nucleus.

The present program supports three nuclear models, defined by the parameter **nuclear** with possible values of **point, Fermi** or **uniform**. 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. For example, in the **uniform** mode, the nucleus is described by a uniform ball of charge of radius R (parameter **r_uniform** in the **knot.dat** file). Under this assumption, the nuclear potential can be written as

$$V_{\text{nuc}}(r) = \begin{cases} -(Z/R)(3/2 - r^2/2R^2) & r < R, \\ -Z/r & r \geq R. \end{cases} \tag{24}$$

The root-mean-square radius of a uniform charge distribution R_{rms} is related to its radius R through

$$R = \sqrt{5/3}R_{\text{rms}}. \tag{25}$$

The radius R_{rms} of a nucleus with weight A can be provided by the user in file **knot.dat** or as input parameter, otherwise the radius will be taken form the available values in compilation [17] or, if absent, will be estimated using the empirical formulas

$$R_{\text{rms}} = 0.836A^{1/3} + 0.57 \text{ fm, } Z \leq 90, \tag{26a}$$

$$R_{\text{rms}} = 0.770A^{1/3} + 0.98 \text{ fm, } Z \geq 91, \tag{26b}$$

taken from references [18] and [19], respectively.

In atomic structure calculations, the above assumption of a uniform nuclear charge density is often replaced by the more realistic assumption of a nuclear charge density given by a Fermi distribution function:

$$\rho_{\text{nuc}}(r) = \frac{\rho_0}{1 + \exp[(r - c)/a]} \tag{27}$$

In this formula, c is the 50 % fall-off radius of the density, and a is related to the 90 % to 10 % fall-off distance $t = 4 \ln(3) a$. Again, these parameters can be provided by users, or they will be estimated based on R_{rms} in the same way as in the GRASP program.

Calculations show that the results will be independent of the B-spline grid only when the initial point is placed deep inside the nucleus. For this reason, the initial point for the uniform and Fermi distribution is chosen by the program as Rh_i, where R is the radius of nucleus. In this sense, the B-spline knot sequence depends on the nuclear model, and all related parameters are recorded in the same file, **knot.dat**.

As an alternative to the direct calculation of the grid points based on the given parameters in the **knot.dat** file, it is also possible to read them from an external text file with the input parameters **knot=<anyname>** and **grid_type=-1**, or from an unformatted bsw-file **<name>.bsw**) containing orbital wavefunctions from the previous calculations with the input parameter **grid_type=-2**.

4.3 Angular coefficients

In DBSR_HF, the orbitals are optimized so that an energy functional defined as a linear combination of energies in the form of Eq. (4) has a stationary energy value. This is an extension of the usual HF

definition. The program has different options for defining the energy functional governed by the input parameter **term**. When **term=AV** , the energy functional is a linear combination of the average energies of a list of configurations specified in relativistic notation, either in the command line (single configuration) or in a <**name**>.**conf** file (multiple configurations). When **term=LS**, the process is similar, but these configuration may be specified in the shorter non-relativistic notation which the program will expand into all possible configurations in a relativistic notation. In this case, all these configurations are included in the optimization process and none can be omitted. When **term=jj,** the energy functional is a weighted linear combination of matrix elements of the Dirac Coulomb Hamiltonian, $\left\langle \Phi \,|\, H^{DC} \,|\, \Phi \right\rangle$, of the coupled configuration state functions (CSFs) Φ given in the <**name**>.**c** file in the format used by GRASP.

To construct the Dirac-Hartree-Fock equations, we need the angular coefficients $f_k(a,b)$ and $g_k(a,b)$ in the energy expression (8). In the average-energy approximation, **term=AV**, these coefficients, for a single configuration, are defined by the following expression [21]:

$$E_{Av} = \sum_a q_a < a \,|\, h_D \,|\, a >$$

$$+ \sum_a \frac{1}{2} q_a (q_a - 1) R^0(aa,aa) - \sum_a \frac{1}{2} q_a (q_a - 1) \frac{[j_a]}{[j_a]-1} \sum_{k=2}^{2l_a} < j_a \,\|\, \mathbf{C}^{(k)} \,\|\, j_a >^2 R^k(aa,aa) , \quad (28)$$

$$+ \sum_{a,b>a} q_a q_b R^0(ab,ab) - \sum_{a,b>a} q_a q_b \sum_{k=|l_a-l_b|}^{k=l_a+l_b} < j_a \,\|\, \mathbf{C}^{(k)} \,\|\, j_b >^2 R^k(ab,ba)$$

where the reduced matrix element $< j_a \,\|\, \mathbf{C}^{(k)} \,\|\, j_b >$ is not zero when $l_a + l_b + k$ is even and has the form

$$< j_a \,\|\, \mathbf{C}^{(k)} \,\|\, j_b >= (-1)^{j_a+1/2} \sqrt{[j_a][j_b]} \begin{pmatrix} j_a & k & j_b \\ 1/2 & 0 & -1/2 \end{pmatrix} \quad (28)$$

In the case of the specific configuration states in the *jj*-coupling the coefficients $f_k(a,b)$ and $g_k(a,b)$ are obtained using the general module **jj_energy_coef,** which is part of the DBSR complex and will not be discussed here. For the given list of *jj*-coupled configurations, the Dirac-Hartree-Fock equations are designed to minimize the weighted sum of all the average energies of the configurations in the list. When **term=AV** or **term=LS,** the default weights for the average energy expressions are their statistical weights, whereas when **term=jj**, the weights of all coupled CSFs in the file <**name**>.**c** are equal. The weight coefficients may also be defined by the input parameter **eal**, in the same way as in the GRASP program (see also sections 5 and 6) except that configuration interactions are not included in the energy functional.

4.4 Initial estimations

Initial estimations for the radial functions are obtained either from screened hydrogen functions or from reading the file specified by the **inp=<filename>** parameter. This file should contain the B-spline expansions for the large and small components, P and Q, obtained from a previous calculation. These files may have the default extension **.bsw** and contain, along with the radial functions, also the B-spline grid used in their calculation. Should this B-spline grid 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 will be converted to the B-spline representation.

4.5 SCF procedure

The central routine of the program is the **Solve_HF** subroutine, which runs the SCF procedure for solving the DHF equations (7).

When several radial functions with the same orbital symmetry are present, the solutions of the DHF equations not only are stationary with respect to single excitations to unoccupied radial functions, they also are stationary with respect to orthonormal transformations. Such transformations represent the rotation of orbitals, and for two given orbitals a and b, the transformation can be described in terms of a single parameter ε as

$$\begin{pmatrix} \tilde{P}(a,r) \\ \tilde{P}(b,r) \end{pmatrix} = \begin{bmatrix} 1 & -\varepsilon \\ \varepsilon & 1 \end{bmatrix} \begin{pmatrix} P(a,r) \\ P(b,r) \end{pmatrix} / \sqrt{1+\varepsilon^2} . \tag{29}$$

The effect of a rotation on the energy can be expanded in powers of ε

$$E(\varepsilon) = E(0) + g\varepsilon + g'\varepsilon^2 + \text{higher - order terms} . \tag{30}$$

Then the stationary condition,

$$\partial E / \partial \varepsilon \cong g + 2g'\varepsilon = 0, \tag{31}$$

leads to $\varepsilon = -g/(2g')$. The coefficients g and g' can be expressed through the Slater integrals in the energy expression as given in [4] for the non-relativistic case. The default input parameter is **rotate=0,** in which case rotations are omitted, a choice that may slow convergence to a stationary solution. Otherwise, a rotation is performed as a preliminary step before applying the orthogonal projection operator to the Hamiltonian matrix (22). When orbital a is constrained to be orthogonal to a fixed orbital b, the projection operator can be applied directly. The rotation analysis can be omitted when the orthogonality is between orbitals of filled shells of the same symmetry. As discussed in [3], the energy in this case is invariant under rotation and g and g' are both zero.

The SCF procedure in the B-spline basis is reduced to setting up each of the interaction matrices (19), applying the projection operators (if needed for orbital orthogonalization), and diagonalization of the generalized eigenvalue problem (13). In principal, we can generate the total Hamiltonian matrix for all orbitals and obtain their wavefunctions in one diagonalization, however, due to different convergence rates of orbitals, such a procedure was found not to be efficient. Diagonalizaton is performed by a call to the LAPACK routine DSYGV [20] that returns the entire spectrum of Dirac-Hartree-Fock Hamiltonian energies, 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.

4.6 Breit interaction and QED corrections

The Breit interaction describes the relativistic corrections to the electronic motion due to magnetic and retardation effects. These are important corrections and they should be included into accurate atomic-structure calculations. An effective operator for the Breit interaction can be derived from quantum electro-dynamics (QED) in perturbation theory with respect to the number of virtually exchanged photons [21], and in Coulomb gauge, and in the long-wavelength approximation, the Breit operator reads as

$$H_B = -\frac{1}{2r_{12}}\left[\boldsymbol{\alpha}_1 \cdot \boldsymbol{\alpha}_2 + \frac{(\boldsymbol{\alpha}_1 \cdot \mathbf{r}_{12})(\boldsymbol{\alpha}_2 \cdot \mathbf{r}_{12})}{r_{12}^2}\right], \tag{32}$$

where $\boldsymbol{\alpha}$ are the Dirac matrices. Similar to the Coulomb interaction, the matrix elements of these operator between atomic wavefunctions can be expressed through the relativistic two-electron integrals

$$S^k(abcd) = \int_0^\infty \int_0^\infty P_a(r_1)Q_c(r_1)\frac{r_1^k}{r_2^{k+1}}\varepsilon(r_2 - r_1)P_b(r_2)Q_d(r_2)\mathrm{d}r_1\mathrm{d}r_2, \tag{33a}$$

where

$$\varepsilon(r_2 - r_1) = \begin{cases} 1 & \text{if} \quad r_2 > r_1 \\ 0 & \text{if} \quad r_1 > r_2 \end{cases}. \tag{33b}$$

The corresponding angular coefficients are closely related and can be obtained from the $f_k(a,b)$ and $g_k(a,b)$ coefficients (4) for Coulomb interaction [21], although different decompositions of the operators lead to different angular momenta and parity selection rules.

In the present program, there are two options to include the Breit interaction, based on the input parameter **mbreit:** the default value of 0 omits the Breit correction. For **mbreit = 1,** the Breit correction is added to the total energy in the final stage, whereas for **mbreit = 2,** the Breit

interaction is included in the setup of the Hamiltonian matrix (19) and thereby may influence the resulting one-electron orbitals. The latter procedure, however, considerably increases the computational time.

The QED corrections are calculated in the same approximations as in the GRASP [14] and RELCI [22] programs, directly using the corresponding modules from these programs. The dominant corrections arise from the self-energy of the electrons and the polarization of the vacuum due to the nuclear field. The effect of the vacuum polarization can be represented in terms of the Uehling potential [23,24] and has been included in relativistic structure calculations for many years. This correction is included according to the parameter `mode_VP` (see section 6). The computation of the self-energy requires a considerably larger effort. The self-energy contributions are usually estimated from tabulations that are available for the hydrogen-like ions [25,26]. Different methods have been proposed to determine an *effective* nuclear charge, in order to interpolate these data for individual subshells of a many-electron system. The GRASP package, e.g., approximates the screening coefficient by taking the overlap integral of the wavefunction and a hydrogenic wavefunction in the region $r < 0.0219\, a_0$. The MDFGME code [30] uses the Welton method without further details. Recently Lowe et al. [27] discussed Welton's concept of electron-self-energy [27] with respect to GRASP calculations. They suggest implementing a screening factor

$$E_{SE} = \frac{<\phi\,|\,\nabla^2 V(\mathbf{x})\,|\,\phi>}{<\phi^H\,|\,\nabla^2 V(\mathbf{x})\,|\,\phi^H>}\,E_{SE}^H \tag{34}$$

for electron moving in a potential $V(\mathbf{x})$. Here superscript H refers to the hydrogen self-energy E_{SE}^H and orbital ϕ^H. This implementation provides results similar to those of MDFGME. We include both methods for estimating E_{SE} and the user is able to select a screening approximation from the following options: `mode_SE = 0`, no correction; `mode_SE = 1`, no screening; `mode_SE = 2`, Welton concept; `mode_SE = 3`, GRASP mode, default.

In the present method, the self-energy contributions are not part of any *effective* Hamiltonian and, hence, cannot be incorporated into the optimization of radial orbitals. They are added to the total level energies according to the occupation of the individual subshells. We should mention that a new model-operator approach to the Lamb shift calculations [28,29] was suggested recently. This operator, h^{QED}, can be used to calculate Lamb shift in many-electron atomic systems either by evaluating the matrix element of h^{QED} with the many-electron wave function, or by adding h^{QED} to the Dirac–Coulomb–Breit Hamiltonian. This approach has not been implemented in the present version of the code. However, we plan to do so in the future.

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 final results for Rb calculations with Breit and QED corrections.

```
ATOM  Rb

core: 1s   2s   2p-  2p   3s   3p-  3p   3d-  3d   4s   4p-  4p

conf: 5s(1)


   nl          e(nl)           dmp          ns      <r>       max_r

   1s     -562.13287940      5.94D-08       59     0.040      4.538
   2s      -77.50141943      2.65D-08       63     0.177      8.538
   2p-     -70.01640669      8.90D-08       60     0.152      5.538
   2p      -67.76413186      5.81D-08       61     0.156      6.538
   3s      -12.55986969      7.35D-08       67     0.507     12.538
   3p-      -9.81823304      8.34D-08       70     0.509     15.538
   3p       -9.46482338      8.45D-08       70     0.519     15.538
   3d-      -4.67579913      6.62D-08       70     0.521     15.538
   3d       -4.61576786      7.07D-08       70     0.524     15.538
   4s       -1.56626656      9.82D-08       77     1.472     22.538
   4p-      -0.83227800      9.30D-08       82     1.703     27.538
   4p       -0.79624318      7.70D-08       83     1.739     28.538
   5s       -0.14004083      6.32D-08      102     5.545     47.538


Orbital convergence:      9.82D-08
Energy  convergence:      2.44D-15

Coulomb energy:      -2979.805546685149238 au       -81084.119911063040490 eV

Breit:                   1.568416632891987             42.678517218478419
Self-energy:             1.164124259846862             31.677231818635189
Vacuum polarization     -0.123992732212402             -3.373992500280763
Sum:                 -2977.196998524622813         -81013.138154526212020

time:     37.05  sec
```

The output includes atom, term, nuclear charge, closed shells, configuration, followed by orbital properties, where **`e(nl)`** is the diagonal energy parameter of the orbital, **`dmp`** is the maximum change in the orbital in the last iteration, `ns` is the size of the spline expansion after the tail cut-off, **`<r>`** is the mean radius of the orbital (in atomic units), and **`max_r`** is the maximum extent in the radius of the orbital after the tail cut-off. Note that **`max_r`** for the 5s-orbital is the last point of the spline grid in this example. This is followed by convergence information and the total energy based on the Dirac-Coulomb

Hamiltonian. Also given are the Breit corrections in the long-wave approximation (low frequency limit), and two leading QED corrections, the self-energy and vacuum polarization, computed as expected values of the related operators.

5 Examples

To begin the calculations for a given atom, it is enough to run the program with one command-line argument, indicating the corresponding atomic symbol. For example, the command

```
>   dbsr_hf Rb mbreit=1
```

will run the calculations for the ground configuration of the rubidium atom with the inclusion of the Breit and QED corrections. All other parameters have their default values (see Section 6). The **<name>** for this calculation is **Rb** and so the orbital radial functions will be recorded in the file **Rb.bsw**, the B-spline parameters – in the file **Rb.knot**, the default parameters – in the file **Rb.inp**, and information about convergence of the self-consistent procedure and final energies – in the file **Rb.log**. Examples of these files were presented earlier. If needed, the user may change any parameter in the **Rb.inp** and **Rb.knot** files and rerun the case with the same name: **dbsr_hf Rb**. In this case, the above files will be treated as input files. The output also contains files **Rb.conf** and **Rb.c**, with description of configuration and possible atomic state functions, respectively. These files simplify further consideration of individual atomic states if need.

In the above example, the name of the case coincides with an atomic symbol, but it is not a mandatory option. The command

```
>   dbsr_hf  Rb_5s atom=Rb
```

would provide the results for the ground state configuration of Rb, with all input-output files beginning with Rb_5s. If we need configurations other than the ground configuration, the configuration should be indicated explicitly as an argument:

```
>   dbsr_hf  Rb_5p  atom=Rb  conf=5p(1) term=AV
>   dbsr_hf  Rb_5p-  atom=Rb  conf=5p-(1) term=AV
```

Note that all orbitals will be optimized separately for each excited configuration. To keep core orbitals the same, we may add parameters **varied=5p** and **inp=Rb_5s.bsw**, thereby optimizing on the $5p$ orbital using core orbitals from the previous calculations with name Rb_5s.

The above example is given with the **term=AV,** a mode where the wavefunction is determined for the given relativistic configuration in the average-energy approximation. If more configurations need to

be optimized simultaneously, the user should add them to the **<name>.conf** file. When the file **<name>.conf** is provided and **term=AV,** this file is considered as the input file for the given case and the program will optimize one-electron orbitals for the energy expression that is the weighted sum of the average energies of the configurations listed in this file. Weights can be given in the file or they will be defined by the parameter **eal:** if eal=1, each configuration provides an equal contribution, if eal=5, their contributions are proportional to their statistical weights. The latter are defined as a number of Slater determinants in the configuration expansion (at mode **term=AV** or **LS),** or as (2J+1) value for the atomic state with total momentum J (at mode **term=jj**).This approach is similar to the extended average level (EAL) calculation in GRASP. Note that an EAL calculation does not include any interaction between configurations.

Consider as an example the $2s^2$ and 2s2p configurations in Be, given in the file **Be.conf**:

```
Core subshells:
  1s
Peel subshells:
  2s   2p-   2p
CSF(s):
  2s ( 2)
  2s ( 1)   2p-( 1)
  2s ( 1)   2p ( 1)
*
```

In this case the command

> **> dbsr_hf   Be term=AV   eal=1**

will lead to simultaneous optimization of orbitals in the given configurations with the results:

```
ATOM  Be

core: 1s

conf: 2s(1.3)2p-(0.3)2p(0.3)


   nl         e(nl)          dmp          ns        <r>        max_r

   1s       -4.75336080     6.70D-08      71       0.415      16.538
   2s       -0.33625561     6.23D-08      84       2.599      29.538
   2p-      -0.20985356     7.22D-08      91       3.044      36.538
   2p       -0.20983110     7.23D-08      91       3.044      36.538

Orbital convergence:      7.23D-08
Energy  convergence:      7.82D-12

Coulomb energy:        -14.511413168965333 au         -394.851559901681412 eV

Configuration energies:

  2s ( 2)                -14.575547621579169 au         -396.596640712430542 eV
```

```
    2s ( 1)   2p-( 1)       -14.479357172086381 au         -393.979325045952294 eV
    2s ( 1)   2p ( 1)       -14.479334713230452 au         -393.978713946661458 eV

time:       4.06  sec
```

In cases with multiply occupied open shells, a non-relativistic configuration may correspond to several relativistic shells, and consequently several relativistic configurations. For nitrogen, e.g., we have one non-relativistic configuration **2p(3)** and three relativistic configurations: **2p-(2)2p(1)**, **2p-(1)2p(1)** and **2p(3).** In this case, it is convenient to use the default option **term=LS:**

> **dbsr_hf   N term=LS**

In this mode, the input configuration is supposed to be given in the non-relativistic notation and the program generates all needed relativistic configurations. The one-electron orbitals with this option will be optimized for an energy expression that is a statistically weighted sum of the individual average energy expressions for each configuration. The printout includes the configuration energies, along with statistically weighted average energy:

```
ATOM  N

core: 1s  2s

conf: 2p-(1)2p(2)


  nl          e(nl)          dmp          ns       <r>        max_r

  1s      -15.67641987     5.67D-08       66       0.228      11.538
  2s       -0.96478796     4.71D-08       74       1.325      19.538
  2p       -0.50818410     7.75D-08       77       1.447      22.538
  2p-      -0.50888171     7.90D-08       77       1.446      22.538

Orbital convergence:      7.90D-08
Energy  convergence:      0.00D+00

Coulomb energy:        -54.327722044059890 au        -1478.274548666663804 eV

Configuration energies:

  2p ( 2)   2p-( 1)     -54.335364217343098 au        -1478.482494625669915 eV
  2p ( 1)   2p-( 2)     -54.316839950403363 au        -1477.978443446650999 eV
  2p ( 3)               -54.315677617866768 au        -1477.946816009657823 eV

time:       4.33  sec
```

In this mode, the program also outputs the file **N.conf:**

```
Core subshells:
  1s   2s
Peel subshells:
  2p   2p-
CSF(s):
  2p ( 3)                                                               0.2000
```

```
  2p ( 2)   2p-( 1)                                                           0.6000
  2p ( 1)   2p-( 2)                                                           0.2000
*
```

This file contains the list of all relativistic configurations included in the calculation and their weights (statistical weights in the given example). The **term=LS** option can be used for any nonrelativistic configuration. For example, command

> dbsr_hf B core=1s conf=2p(3) term=LS

will provide the results for all relativistic configurations connected with the excited $1s^2 2p^3$ configuration in boron, similar to the nitrogen example above. The **term=LS** option can also be applied to multiple nonrelativistic configurations provided by the user in the file **<name>.LS**.

Another option is **term=jj,** where the user has full control over the set of *coupled* configuration states (CSFs) and their weights. In this case, atomic states under consideration should be given in the input **<name>.c** file, in the format accepted in GRASP. This file should be created by the user, but the program itself may create this file whenever there exists a file **<name>.conf.** For example, after the run **dbsr_hf N term=LS,** the program also creates a file **N.c**:

```
Core subshells:
  1s   2s
Peel subshells:
  2p   2p-
CSF(s):
  2p ( 2)   2p-( 1)
        0       1/2
                    1/2-
 *
  2p ( 3)
      3/2
          3/2-
  2p ( 2)   2p-( 1)
        2       1/2
                    3/2-
  2p ( 1)   2p-( 2)
      3/2
                    3/2-
 *
  2p ( 2)   2p-( 1)
        2       1/2
                    5/2-
*
```

Now it is possible to run

> dbsr_hf N term=jj

The output file **N.log** in this case will contain energies for all atomic configuration states involved along with the orbital parameters:

```
ATOM   N

core: 1s    2s

conf: 2p(3)

   nl         e(nl)           dmp           ns        <r>        max_r

   1s      -15.67641086      4.37D-08       66       0.228      11.538
   2s       -0.96478794      7.95D-08       74       1.325      19.538
   2p       -0.50818409      7.75D-08       77       1.447      22.538
   2p-      -0.50888171      7.90D-08       77       1.446      22.538

Orbital convergence:       7.95D-08
Energy  convergence:       7.85D-16

Coulomb energy:         -54.327722044059932 au         -1478.274548666664941 eV

Atomic state energies:

2p(2)_2.2p-(1).3/2*     -54.430870327656955 au               0.00000 eV
2p(1).2p-(2).3/2*      -54.327728918206411 au               2.80651 eV
2p(2)_2.2p-(1).5/2*     -54.327719341815659 au               2.80677 eV
2p(2)_0.2p-(1).1/2*     -54.258971076471560 au               4.67743 eV
2p(3).3/2*             -54.258946423477113 au               4.67810 eV

time:      1.42   sec
```

Note that output now contains not the single-configuration energies but the atomic state energies obtained by diagonalization of the atomic Hamiltonian on the basis of all states included in the **<name>.c** file. These energies can be directly compared with experiment or other calculations. The configuration-mixing coefficients are recorded in the file **<name>.j.**

The option **term=jj** is the most general and the most difficult to program since it includes coupling information for multiple open shells. To perform such calculations, the program contains a module for the calculation of angular coefficients for any atomic configuration state, containing one-electron orbitals with angular momentum j up to 9/2. The present module is a simplified version for orthogonal orbitals of our general program **DBSR_BREIT,** which is a part of the DBSR complex and includes the more general case of non-orthogonal orbitals.

All the above examples concern the atomic state. In case of ions, we should separately define the atom (i.e. the nuclear properties) and the ionic configurations. That can be achieved for the ground configuration of an ion specified by the input parameter **ion** or through the files **<name>.conf** or **<name>.c** in more complicated cases. As an example, let us consider Li-like uranium:

> dbsr_hf Li_092 atom=U ion=Li mbreit=1

```
ATOM   U

core: 1s
```

```
conf: 2s(1)

   nl        e(nl)          dmp          ns        <r>       max_r

   1s   -4760.19460057     5.72D-08      47       0.014     0.312
   2s   -1209.71404307     7.66D-08      50       0.054     0.609

Orbital convergence:      7.66D-08
Energy  convergence:      3.35D-13

Coulomb energy:      -10846.660937604610808 au      -295151.969070185499731 eV

Breit:                  13.364538364755932              363.666739170965002
Self-energy:            28.684698321383472              780.548524605240800
Vacuum polarization     -7.503672554861111             -204.184840160982191
Sum:                -10846.115373473332511          -294211.938646570313722

time:       1.53  sec
```

In this case, <name> = **Li_092** and all output files will have this name with the corresponding extensions. Note that the program assigns the same ground configuration (taken from the neutral atom) for all ions in the isoelectronic series. Actually, the ground configuration may change with increasing the ionic charge and for many ions the ground state is still unknown.

As the last example, consider the option for including the Breit interaction in the orbital optimization:

> dbsr_hf Li_092 atom=U ion=Li mbreit=2

```
ATOM  U

core: 1s

conf: 2s(1)

   nl        e(nl)          dmp          ns        <r>       max_r

   1s   -4754.18855539     5.73D-08      47       0.014     0.312
   2s   -1209.72030267     7.66D-08      50       0.054     0.609

Orbital convergence:      7.66D-08
Energy  convergence:      6.13D-12

Coulomb energy:      -10846.654037569465800 au      -295151.781311116472352 eV

Breit:                  13.342199536634887              363.058870155368822
Self-energy:            28.677277522486179              780.346594866751730
Vacuum polarization     -7.486131758448225             -203.707531924825048
Sum:                -10846.120692268792482          -294212.083378019160591

time:       1.88  sec
```

6 List of input parameters

Input data can be provided in the input file `<name>.inp` or in the command line as `keyword=value` (the data from the command line overrides the data from the input file `<name>.inp`). Below we describe input parameters with their default values indicated in brackets. The zero value for many parameters implies a negative answer. **name** is the only mandatory parameter. The command line (or input file) may contain any subset of input parameters, the rest will be replaced by their default values.

name	name of the case; it is the first argument in the command line and is given without a keyword.
atom	symbolic name of the atom under consideration; can be used as the name of the case.
an	atomic number, integer, can be used instead of the symbolic name of the atom.
ion	symbolic name of the ion under consideration; used to find the ground configuration and core.
ai	atomic number for the ion, integer, can be used instead of the symbolic name of the ion.
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 from the compilation [17].
rms	the root-mean-square radius of nuclear, defined from its atomic weight or taken from the compilation [17].
core	list of closed shells; used only for a more compact description of the configuration; can be specified as [He], [Be], [Ne], [Mg], [Ar], [Zn], [Kr], [Cd], [Xe], [Hg], or [Rn].
conf	electron configuration under consideration in the `temm=AV` or `term=LS` modes; not used if the description of configurations comes from the `<name>.conf` file. Under Unix environment, the configurations in the command line should be included into apostrophes (e.g., `conf='2p(3)'`).
term [LS]	`term=AV` - optimization of a single relativistic configuration (given by the input parameter `conf`) or a set of configurations (given in the file `<name>.conf`) in the average-energy approximation. `term=LS` - optimization of a single non-relativistic configuration (given by the input parameter `conf`) or a set of non-relativistic configurations (given in the file `<name>.LS`); otherwise, it is supposed the ground configuration of the atom. In this option, all relevant relativistic configurations will be generated by the program. `term=jj` - optimization of all atomic states given in the `<name>.c` file in the GRASP format.

mbreit [0]	inclusion of the Breit and QED corrections:
	=0 - no corrections;
	=1 - as the first-order corrections to the final energies;
	=2 - Breit interaction is included into orbital optimization and QED as the first order correction.
mode_SE [3]	self-energy correction mode:
	=0 - no correction;
	=1 – pure hydrogen-based results;
	=2 - Welton concept;
	=3 - GRASP mode.
mode_VP [1]	vacuum-polarization correction mode:
	=0 - no correction;
	=1 – the Uehling potential mode.
eal [5]	indicates the mode for the statistical weights:
	=1 - equally weighted;
	=5 - statistically weighed;
	=9 - defined by the user in the <name>.conf or <name>.c files (in format of expansion coefficients).
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) .
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 .
rotate [0]	rotations during optimization (0/1 →no/yes).
out_w [0]	create <**name>.w** file with radial functions in GRASP format (0/1 →no/yes).
out_plot [0]	create <**name>.plot** file with radial functions ready for plotting (0/1 →no/yes).
out_nl [0]	create <**name>.nl** file with all solutions for outer electron in the frozen-core approximation (0/1 →no/yes).
debug [0]	additional debug output (0/1 →no/yes).

Default values for atomic parameters are defined in the program file **dbsr_zcom.f90** in the module **atoms_par**. Included are the atomic weight, **an**, the 2-character symbol for the atom, the core, the ground configuration, **conf**, and the weight, **awt**, for the most abundant or most stable isotopes. The module includes values for atomic numbers 1-104. The values may easily be edited or the range extended by adding data for higher atomic numbers. This file also contains a module for physical constants used in the program.

7 Data files

The program employs the name-driven file convention. All filenames have the structure <**name**>.<**extension**>, where <**name**> (the value of **name**) identifies the specific calculation. The list of data files and their description is given in table 1. 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, the input wavefunctions from GRASP calculations can be introduced through the instruction **inp=<case>.w.** 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 **knot.dat** file. In this case, all associated calculations for the given atom or ion will be carried out with the same grid.

All input files are optional and the program will use the default input parameters if needed. Note also that the input files can be modified by the program by adding the absent information, additional comments, or replacing some parameters. The latter may happen if the parameters from the command line do not agree with the parameters in the files.

Table 1 Input/output files.

name.inp	**dat=…**	File type: formatted sequential input.
		Usually created by the program and modified by user if needed.
		Description: input parameters for the given run.
name.log	**log=…**	File type: formatted sequential output.
		Description: Summary of the running information.
name.knot	**knot=…**	File type: formatted sequential input/output.
		Provided by the user or created by the program if absent.
		Description: defines the B-spline grid.
name.conf	**confs=…**	File type: formatted sequential input.
		Written by the user or created by the program for **term=LS**.
		Description: input configurations in the GRASP format (without coupling information) or a single non-relativistic configuration.
name.c	**c=…**	File type: formatted sequential input.
		Created by the user (one may use GRASP tools **csl** or **jjgen**) or created by the program for **term=AV** or **term=LS**.
		Description: input configuration states in the GRASP format.
name.j	**j=…**	File type: formatted sequential output.
		Created by the program for **term=jj**.
		Description: state mixing coefficients.
name.bsw	**inp=…**	File type: unformatted sequential input.

		Provided by the user or can be created by the program in prior runs. Description: one-electron orbitals in B-spline representation; if the specified filename has an extension **.w**, the file is supposed to be generated by the GRASP code.
name.bsw	**out=…**	File type: unformatted sequential output. Description: one-electron orbitals in B-spline representation. Created by program.
name.w	**w=**	File type: unformatted sequential output. Description: one-electron orbitals in the GRASP format. Created by program when parameter **out_w** > 0.
name.nl	**nl=**	File type: unformatted sequential output. Description: one-electron orbitals in *B*-spline representation for all possible solutions for the outer electron; can be used as a basis for the other calculations, e.g., in perturbation methods. Created by program when parameter **out_nl** > 0.
name.plot	**plot=**	File type: formatted sequential output. Description: one-electron orbitals in table form. Created by the program when parameter **out_plot** > 0.

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 *i*:

```
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_HF program is a part of the DBSR complex and extensively uses common routines from the DBSR libraries: ZCONFJJ, DBS, ZCOM. ZCONFJJ deals with the state 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 those that read arguments or order arrays. For convenience of independent use, the DBSR_HF program is submitted as a stand-alone program, which contains specific subroutines with names beginning with **hf_** and routines from the libraries, collected in separate files, dbsr_zconfjj.f90, dbsr_dbs.f90, and dbsr_zcom.f90. The DBSR_HF 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_hf** in the file **hf_MOD_dbsr_hf.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 **dbsr_lib_dbs.f90**.

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>.**

References

[1] C. Froese Fischer, T. Brage, P. Jönsson, Computational Atomic Structure. An MCHF Approach, Institute of Physics Publishing, Bristol/Philadelphia, 1997.
[2] H. Bachau, E. Cormier, P. Decleva, J. E. Hansen, and F. Martin, Rep. Prog. Phys. 64 (2001) 1815.
[3] C. Froese Fischer, Advances in Atomic and Molecular Physics 55 (2007) 539.
[4] C. Froese Fischer, Comput. Phys. Commun. 182 (2011) 1315.
[5] W.R. Johnson, J. Sapirstein, Phys. Rev. Lett. 57 (1986) 1126.
[6] G.W.F. Drake, S.P. Goldman, Phys. Rev. A 23 (1981) 2093.
[7] C. Froese Fischer and O. Zatsarinny, Comput. Phys. Commun. 180 (2009) 879.
[8] H.W. van der Hart, J. Phys. B 30 (1997) 453.

[9] O. Zatsarinny, C. Froese Fischer, J. Phys. B 33 (2000) 313.

[10] O. Zatsarinny, Comput. Phys. Commun. 174 (2006) 273.

[11] O. Zatsarinny, K. Bartschat, J. Phys. B 46 (2013) 112001.

[12] O. Zatsarinny, K. Bartschat, Phys. Rev. A 77 (2008) 062701.

[13] O. Zatsarinny, K. Bartschat, J. Phys. B 43 (2010) 074031.

[14] P. Jönsson, X. He, C. Froese Fischer, I.P. Grant, Comput. Phys. Commun. 177 (2007) 597.

[15] P. Jönsson, G. Gaigalas, J. Bieroń, C. Froese Fischer, I.P. Grant, Comput. Phys. Commun. 184 (2013) 2197.

[16] M. Bentley, J. Phys. B 27 (1994) 637.

[17] I. Angeli, K. P. Marinova, Atomic Data and Nuclear Data Tables 99 (2013) 93.

[18] W.R. Johnson, G.Soff, Atomic Data and Nudear Data Tables 33 (1985) 405.

[19] I. Goidenko, Private communication.

[20] http://www.netlib.org/lapack/

[21] I.P. Grant, Relativistic Quantum Theory of Atom and Molecules, Springer, 2007.

[22] S. Fritzsche, C. Froese Fischer, G. Gaigalas, Comput. Phys. Commun. 148 (2002) 103.

[23] E.A. Uehling, Phys. Rev. 48 (1935) 55.

[24] L. W. Fullerton, G. A. Rinker, Jr,, Phys Rev A13 (1976) 1283.

[25] P.J. Mohr, Ann. Phys. (N.Y.) 88 (1974) 52; At. Data Nucl. Data Tables 29 (1983) 453.

[26] P.J. Mohr, Y.-K. Kim, Phys. Rev. A 45 (1992) 2723.

[27] J.A. Lowe, C.T. Chantler, I.P.Grant, Radiation Physics and Chemistry 85 (2013) 118.

[28] V.M. Shabaev, I.I. Tupitsyn, V.A. Yerokhin, Phys. Rev. A 88 (2013) 012513.

[29] V.M. Shabaev, I.I. Tupitsyn, V.A. Yerokhin, Comp. Phys. Comm. 189 (2015) 175.

[30] P. Indelicato and J.-P. Desclaux, *MDFGME, a multiconfiguration Dirac-Fock general matrix element program* (release 2005), http://dirac.spectra.jussieu.fr.