BSR_HF: A B-spline Hartree-Fock program

This program is version of the SPHF program of Charlotte Froese Fischer, CPC, **182**, 1315 (2011) and designed to more close connection with BSR complex. I will not refer to all specific features, taken from SPHF. I refer the reader to above article and references there. I thankful Charlotte to strong cooperation in B-spline applications.

1. General remarks

The steady states of a many-electron system are defined by Schrödinger equation

$$H\Psi = E\Psi\ , \tag{1}$$

where *E* is the total energy of the system, Ψ is called the total wavefunction and describes the state of the system, and *H* is the *Hamiltonian* or *Hamiltonian operator*. For an *N*-electron atomic system the nonrelativistic Hamiltonian, in atomic units, is given by

$$H_{DC} = -\frac{1}{2}\sum_{i=1}^{N}\left(\nabla_i^2 + \frac{2Z}{r_i}\right) + \sum_{i<j}\frac{1}{r_{ij}}\ . \tag{2}$$

This expression assumes that the nucleus can be treated as a point charge with infinite mass. Z is the atomic number, r_i the distance of *i*th electron from the nucleus, and r_{ij} the distance between the *i*th and *j*th electron.

In the Hartree–Fock approach, the total wave function Ψ is defined as being an antisymmetrized product of one-electron spin-orbitals of the form

$$\phi(nlm_l m_s) = \frac{1}{r}P(nl;r)Y_{lm_l}(\theta,\varphi)\chi_{m_s}\ , \tag{3}$$

where *P*(*nl*; *r*) is a real-valued radial function, $Y_{lm_l}(\theta,\varphi)$ a complex-valued spherical harmonic function, and χ_{m_s} a spin function. In general, the total wavefunction can be through a single or combination of the Slater determinants. In atomic physics, we usually work with single configuration state function (CSF) Φ(*γLS*) for a configuration *γ* that is an eigenfunction of the total orbital-angular momentum and spin-angular operators $\mathbf{L}^2$, L_z, $\mathbf{S}^2$ and S_z.

The task is reduced to find the radial function *P*(*nl*; *r*), which can be determined by applying the variational principle to the energy, E_{HF} = < Φ(*γLS*) | *H* | Φ(*γLS*)>, subject to orthonormality constraints. The radial functions *P(nl*; *r)* depend only on *nl* quantum numbers, and angular integrations can be performed using Racah algebra producing an energy expression involving only real radial integrals:

$$E_{HF} = \sum_a q_a L(a,a) + \sum_{a<b} \sum_k \left(f_k(a,b) R_k(abab) + g_k(a,b) R_k(abba) \right) , \tag{4}$$

$$L(a,b) = \left\langle a \left| -\frac{1}{2} \frac{d^2}{dr^2} - \frac{Z}{r} + \frac{l(l+1)}{2r^2} \right| b \right\rangle \tag{5}$$

$$R^k(abcd) = \int_0^\infty \int_0^\infty P_a(r_1) P_b(r_2) \frac{r_<^k}{r_>^{k+1}} P_c(r_1) P_d(r_2) \, \mathrm{d}r_1 \mathrm{d}r_2 , \tag{6}$$

where the indices *a* and *b* refer to one-electron orbitals ($n_a l_a$) and ($n_a l_a$), respectively, the q_a is the occupation of the orbital *a*, and coefficients f_k and g_k depend on the angular symmetry of the atomic state under consideration. The Hartree-Fock equations now can be derived from the variation condition

$$\delta(E_{HF} - \sum_{ab} \delta(l_a, l_b) \lambda_{ab} < a | b >) = 0 \tag{7}$$

where Lagrange multipliers λ_{ab} ensure the orthogonality of orbitals with the same *l*-values. Eq. (7) applies to all allowed perturbations that satisfy orthogonality constraints and the boundary conditions. For the perturbation of a single orbital the resulting Hartree-Fock differential equations have the form

$$\left(-\frac{1}{2} \frac{d^2}{dr^2} - \frac{1}{r} [Z - Y(nl,r) + \frac{l(l+1)}{2r^2} - \varepsilon_{nl,nl} \right) P(nl;r) = \frac{1}{r} X(nl;r) + \sum_{n' \neq n} \varepsilon_{nl,n'l} P(n'l;r) \tag{8}$$

where we introduce the direct and exchange HF potentials

$$Y(nl;r) = \sum_{n'l';k} \frac{(1+\delta_{nl,n'l'})}{q_{nl}} f_k(nl,n'l') Y^k(nl,n'l';r) \tag{9}$$

$$X(nl;r) = \sum_{n'l' \neq nl;k} \frac{1}{q_{nl}} g_k(nl,n'l') Y^k(nl,n'l') P(n'l';r) \tag{10}$$

$$Y^k(nl,n'l';r) = \int_0^r \frac{s_<^k}{r_>^k} P(nl;s) P(n'l';s) ds + \int_0^r \frac{r_<^{k+1}}{s_>^{k+1}} P(nl;s) P(n'l';s) ds \tag{11}$$

and $\varepsilon_{nl,n'l} = \lambda_{nl,n'l} / q_{nl}$ are referred to as energy parameters. This differential equation (8) can be solved numerically by finite difference methods in which the radial wave function is represented by a vector of numerical values $P(nl,r_i)$, $i = 1, \ldots, N$. The boundary conditions, $P(nl; 0) = 0$ and $P(nl;r) \to 0$ as $r \to \infty$, make this a two-point boundary value problem. The HF equations are solved self-consistently, iteration by iterations, beginning with some initial estimations.

3. B-spline basis

In our implementation, the HF solution is represented in terms of B-spline basis functions $B_i\,(r)$, namely

$$P(nl;r)=\sum_{i=1}^{n_s} a_i B_i\,(r) \tag{12}$$

and the expansion coefficients are determined by the Galerkin condition for differential equations. The solution of Eq. (8) then becomes a solution of the symmetric, generalized matrix-eigenvalue problem

$$(\mathbf{H}-\varepsilon\mathbf{S})a=0 \tag{13}$$

where **H=**(H_{ij}) are Hamiltonian matrix in the B-spline basis and **S** is the B-spline overlap matrix $\langle B_i | B_i \rangle$. For more detailed description of the Hamiltonian matrix **H**, let us introduce the matrices for the one-electron terms

$$L_{ij}=\left\langle B_i(r)\left|-\frac{1}{2}\frac{d^2}{dr^2}-\frac{Z}{r}+\frac{l(l+1)}{2r^2}\right|B_j(r)\right\rangle, \tag{14}$$

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

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

The direct and exchange potentials X(nl,r) and Y(nl,r) can then be expressed as matrices obtained by convolution of these arrays over two variables as

$$\mathbf{V}_b^{dir}\rightarrow R^k(\,\bullet\, b\,\bullet\, b)=\sum_{jj'} b_j b_{j'} R^k(ij;i'\,j')\,, \tag{17}$$

$$\mathbf{V}_b^{exc}\rightarrow R^k(\,\bullet\, b\, b\,\bullet\,)=\sum_{i'j} b_{i'} b_j R^k(ij;i'\,j')\,. \tag{18}$$

Note that the direct integrals are defined only by diagonal banded matrixes, but exchange requires full matrixes and numerically takes most of the computation time. Finally, the Hamiltonian matrix, $\mathbf{H}^{\boldsymbol{a}}$ for orbital a, has the form

$$\mathbf{H}^{a}=\mathbf{L}\;+\;\mathbf{V}_a^{dir}\;+\;\mathbf{V}_a^{exc}\quad, \tag{19}$$

where the direct and exchange potentials are represented by matrices obtained by summation and convolution of the corresponding Slater integrals (16).

$$\mathbf{V}_a^{dir}=\sum_b\sum_k\frac{(1+\delta_{a,b})}{q_a}\,f_k(a,b)R^k(.\,P_b\;.\,P_b)\,, \tag{20}$$

$$\mathbf{V}_a^{exc}=\sum_b\sum_k\frac{1}{q_a}\,g_k(a,b)R^k(.\,P_b\;P_b\;.)\,. \tag{21}$$

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 [16], 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

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

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

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 and summary of results.

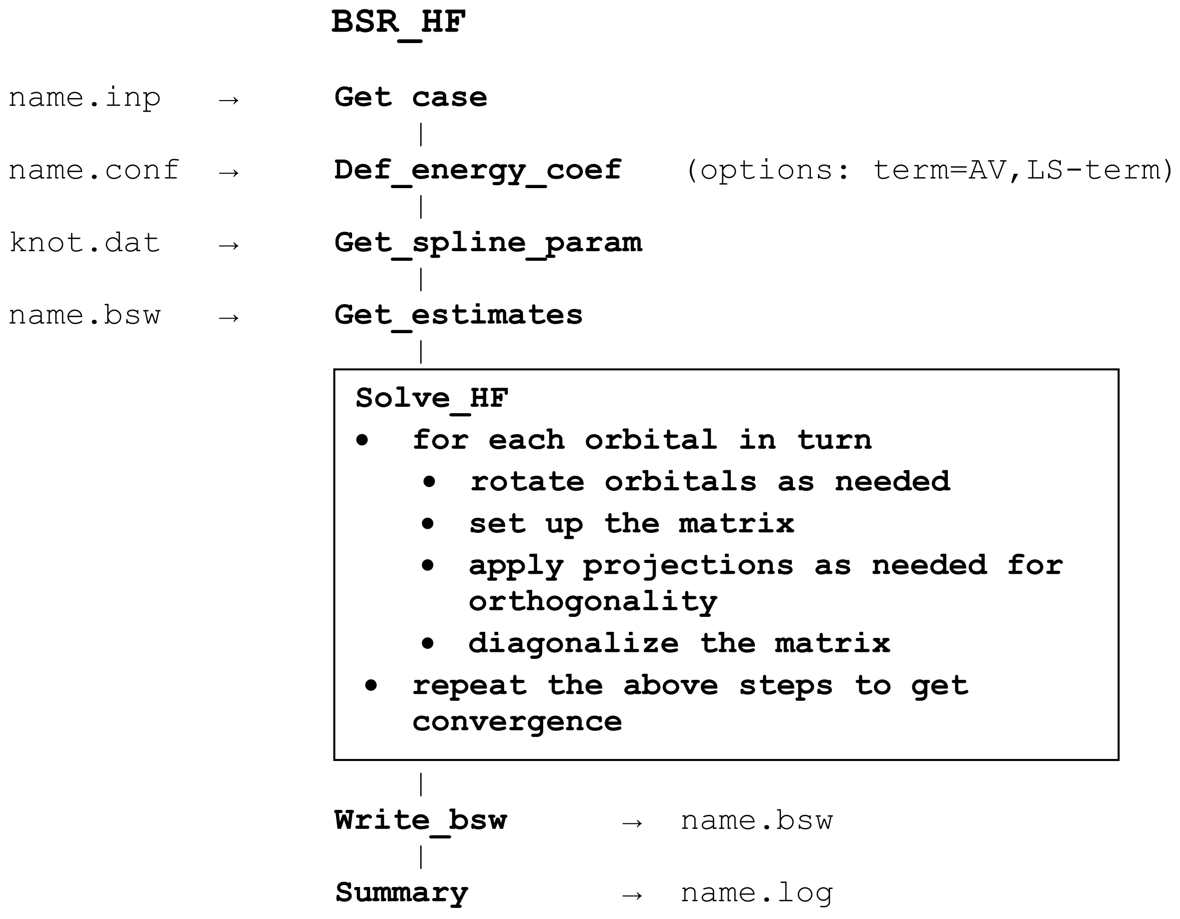
BSR_HF
name.inp → Get case
name.conf → Def_energy_coef (options: term=AV,LS-term)
knot.dat → Get_spline_param
name.bsw → Get_estimates
Solve_HF
• for each orbital in turn
• rotate orbitals as needed
• set up the matrix
• apply projections as needed for orthogonality
• diagonalize the matrix
• repeat the above steps to get convergence
Write_bsw → name.bsw
Summary → name.log

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

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.

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
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, 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

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

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

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 BSR_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.

4.3 Angular coefficients

In BSR_HF, the orbitals are optimized so that an energy functional defined as a linear combination of energies of input configuration or ASF's 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`** (default option) , 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 (in case of single configuration) or in a <**name>.conf** file (multiple configurations). When **term≠AV,** the energy functional is a weighted linear combination of matrix elements of the nonrelativistic Hamiltonian, $\left\langle \Phi^{LS} | H | \Phi^{LS} \right\rangle$, of the coupled configuration state functions (CSFs) Φ^{LS} given in the **name.cfg** file in the format used by MCHF program of Charlotte Froese Fischer. If this file is absent, it will be created by the program based on the configurations given in **name.conf** and the term given as **term=LS**.

To construct the Hartree-Fock equations, we need the angular coefficients $f_k(a,b)$ and $g_k(a,b)$ in the energy expression (4). In the average-energy approximation, **term=AV**, these coefficients, for a single configuration, are defined by following expression:

$$E_{Av} = \sum_a q_a < a | L | a > + \sum_a \frac{1}{2} q_a (q_a - 1) \left(R^0(aa,aa) - \frac{1}{(2l_a+1)(4l_a+1)} \sum_{k=2}^{2l_a} < l_a \| \mathbf{C}^{(k)} \| l_a >^2 R^k(aa,aa) \right)$$
$$+ \sum_{a,b>a} q_a q_b \left(R^0(ab,ab) - \frac{1}{2(2l_a+1)(2l_b+1)} \sum_{k=|l_a-l_b|}^{k=l_a+l_b} < l_a \| \mathbf{C}^{(k)} \| l_b >^2 R^k(ab,ba) \right) . \qquad (24)$$

In the case of the specific configuration states in the LS-coupling the coefficients $f_k(a,b)$ and $g_k(a,b)$ are obtained using the subroutine **Coef_ee_1conf,** which is simplification of general BSR_BREIT program in case of a single configuration and orthogonal orbitals, and has no restrictions on the number of open shells. The Hartree-Fock equations are then designed to minimize the weighted sum of all given configurations or CSFs. The default weights for the average energy expressions are the equal weights. The weight coefficients may also be defined by the input parameter **eal.**

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 radial functions, 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.

4.5 SCF procedure

The central routine of the program is the **`Solve_HF`** subroutine, which runs the SCF procedure for diagonalization of the B-spline HF equations (13).

When several radial functions with the same orbital symmetry are present, the solutions of the HF 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

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

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

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

Then the stationary condition,

$$\partial E / \partial \varepsilon \cong g + 2g'\varepsilon = 0 , \qquad (31)$$

leads to $\varepsilon = -g/(2g')$. The coefficients *g* and *g*' can be expressed through the Slater integrals in the energy expression.

$$
\begin{aligned}
g = {} & 2(q_b - q_a)L(a,b) \\
& + \sum_{k=0,2,\ldots}^{2\min(l_a,l_b)} 2(f_k(a,b) + g_k(a,b) - 2f_k(a,a))R^k(a,a,a,b) \\
& - \sum_{k=0,2,\ldots}^{2\min(l_a,l_b)} 2(f_k(a,b) + g_k(a,b) - 2f_k(b,b))R^k(a,b,b,b) \\
& + \sum_{c=1,c\neq a,b}^{n-orbitals} \sum_{k=0,2,\ldots}^{2\min(l_a,l_b)} 2(f_k(b,c) - f_k(a,c))R^k(a,c,b,c) \\
& + \sum_{c=1,c\neq a,b}^{n-orbitals} \sum_{k=|l_a-l_c|,+2,\ldots}^{|l_a+l_c|} 2(g_k(b,c) - g_k(a,c))R^k(a,c,c,b)
\end{aligned}
\tag{32}
$$

$$
\begin{aligned}
g' = {} & (q_a - q_b)(L(b,b) - L(a,a)) \\
& + \sum_{k=0,2,\ldots}^{2\min(l_a,l_b)} 2f_k(a,a)(R^k(a,b,a,b) - R(a,a,a,a) + 2R(a,b,b,a)) \\
& + \sum_{k=0,2,\ldots}^{2\min(l_a,l_b)} 2f_k(b,b)(R^k(a,b,a,b) - R^k(b,b,b,b) + 2R^k(a,b,b,a)) \\
& - \sum_{k=0,2,\ldots}^{2\min(l_a,l_b)} (f_k(a,b) + g_k(a,b))(R^k(a,a,a,a) + R^k(b,b,b,b) - 2R^k(a,b,a,b) - 4R^k(a,b,b,a)) \\
& + \sum_{c=1,c\neq a,b}^{n-orbitals} \sum_{k=0,2,\ldots}^{2\min(l_a,l_b)} (f_k(b,c) - f_k(a,c))(R^k(a,c,a,c) - R^k(b,c,b,c)) \\
& + \sum_{c=1,c\neq a,b}^{n-orbitals} \sum_{k=|l_a-l_c|,+2,\ldots}^{|l_a+l_c|} (g_k(b,c) - g_k(a,c))(R^k(a,c,c,a) - R^k(b,c,c,b))
\end{aligned}
\tag{33}
$$

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. 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, the energy in this case is invariant under rotation and *g* and *g*' are both zero.

After the projections have been applied, the SCF procedure in the B-spline basis is reduced to setting up each of the interaction matrices (19) followed by the 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 that returns the entire spectrum of Hartree-Fock Hamiltonian energies for given orbital. The present program selects the needed solution, as a rule the lowest eigenvalue of the spectrum, taking into account the orthogonality constraints.

The convergence of the SCF procedure cab be very slow or even fails. In tis case we may use the acceleration procedure

$$P_{nl}^{i} = \alpha P_{nl}^{i} + \beta P_{nl}^{i-1}, \tag{34}$$

where index *i* refers to the current iteration. The parameters α and β are defined by input parameters **aweight** and **bweight**. The procedure automatically applied when the current orbital difference max(|P^{i}-P^{i-1}|) is bigger the previous one, max(|P^{i-1}-P^{i-2}|). The user may force the procedure (34) for all iterations using the parameter **ac**=1.

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 Fe for term=AV calculations.

```
                        ATOM    Fe      TERM    AV


   nl        E(nl)         I(nl)         KE(nl)        AZ(nl)       S(nl)     ns
   1s  -261.400295   -337.854165   325.695737   260.199767      0.624     48
   2s   -31.964701    -82.692619    60.272545   -78.912923      3.650     55
   2p   -27.442601    -82.021131    58.955749   523.171085      4.825     58
   3s    -4.189226    -31.873266    12.631327    29.261907      9.505     60
   3p    -2.760507    -29.981101    11.064242  -189.485882     11.544     67
   3d    -0.607857    -24.737002     7.218550   150.447537     16.321     69
   4s    -0.260130     -9.405865     0.830199    -6.306391     18.596     82


   nl        1/R**3         1/R            R            R**2       Delta(r)  max_R
   1s     0.000000     25.521150     0.059112      0.004686   5387.707     4.277
   2s     0.000000      5.498660     0.268454      0.084979    495.549     9.670
   2p   462.820502      5.422188     0.236124      0.068370      0.000    12.670
   3s     0.000000      1.711715     0.818433      0.771723     68.139    14.670
   3p    54.757269      1.578667     0.864672      0.880293      0.000    21.670
   3d     4.930615      1.229060     1.084830      1.532102      0.000    23.670
   4s     0.000000      0.393695     3.241624     12.253685      3.165    36.670


     TOTAL ENERGY (a.u.)        -1262.29086341
     REL ENERGY                 -1270.90148574
                   Kinetic       1262.29086474
                   Potential    -2524.58172814
                   Ratio           -2.00000000
```

The output includes atom, term, closed shells, configuration, followed by orbital properties, where **E(nl)** is the diagonal energy parameter of the orbital, **ns** is the size of the spline expansion, **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 4*s*-orbital is the last point of the spline grid in this example. This is followed by the total energy including the energy with relativistic shift.

5 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 name of the case; it is the first argument in the command line and is given without 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.

core list of closed shells; used only for a more compact description of the configuration; can be also 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 is given in the the **`name.conf`** file.

term [AV] **`term=AV`** - optimization of a single 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 set of non-relativistic CASs given in the file **`name.cfg`**. If this file is absent, the program will try to generate all relevant CASs base on the configurations given in the **name.conf** file.

eal [1] indicates the mode for the state 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).

ilzero [0] initial zero B-splines (0 means delete *l*+1 B-splines)

ibzero [2] zero B-splines in the end

aweigh [0.7] α acceleration parameter, Eq.(34)

bweight [0.7] β acceleration parameter, Eq.(34)

ac [0] if not equal 0, acceleration procedure apply everytime

out_plot [0] create **name.plot** file with radial functions ready for plotting (0/1 →no/yes).

nl all or list of orbitals with additional output of all spectrum.

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

6 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.

Table 1 Input/output files.

name.inp **dat=…** File type: formatted sequential input.
Written by the user or created by the program if absent.
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. Written by the user or created by the program if absent. Description: defines the B-spline grid.
name.conf	**conf=…**	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.cfg	**cfg=…**	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.
name1.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.
name2.bsw	**out=…**	File type: unformatted sequential input. Description: one-electron orbitals in B-spline representation. Created by program.
name_*orb*.nl	**nl=**	File type: unformatted sequential output. Description: one-electron orbitals in *B*-spline representation for all possible solutions for the given orbitals; 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.

8 Program installation and testing

The BSR_HF program is a part of the BSR complex and extensively uses common routines from the BSR libraries: ZCONFLS, BS, ZCOM, ZCONFJJ deals with the state description and calculation of the angular coefficients, BS 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. The BSR_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.

5 Examples

We provide a few examples N, Al and Fe atoms to illustrate the possible calls of BSR_HF. This examples are included in the distribution files as subdirectories named as atom itself.

N:

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

```
>  bsr_hf N
```

will run the calculations for the ground configuration of the nitrogen atom. All other parameters have their default values (see Section 6). The **`name`** for this calculation is **N** so all the output files will have the same name with different extensions, e.i. **N.log**, **N.bsw**, **N.inp**, **N.knot** and so on. For example, N.log will contain information about convergence and in the end – final energies:

```
                        ATOM   N      TERM   AV


   nl       E(nl)        I(nl)        KE(nl)       AZ(nl)       S(nl)    ns
   1s   -15.666391   -24.429801    22.147354    35.134012      0.429    48
   2s    -0.963670    -5.317273     2.264974    -7.780467      2.476    54
   2p    -0.508655    -4.770701     1.823837     9.781394      3.544    59


   nl       1/R**3        1/R           R          R**2      Delta(r)  max_R
   1s     0.000000     6.653879     0.228260     0.070236     98.230    13.815
   2s     0.000000     1.083178     1.326323     2.129412      4.817    19.815
   2p     3.020512     0.942077     1.446623     2.707158      0.000    24.815


     CORE ENERGY (a.u.)         -44.73398626


     TOTAL ENERGY (a.u.)         -54.29616935
     REL ENERGY                  -54.32329459
                   Kinetic        54.29616925
                   Potential    -108.59233860
                   Ratio          -2.00000000

time:      5.88  sec
```

If needed, the user may change any parameter in the created **`N.inp`** and **`N.knot`** files and rerun the case. In this case, the above files will be treated as input files and this information will replaced the default values. The output also contains files **`N.conf`**, with description of configuration(s), and if **term=LS** the program also create **N.c** with atomic state function(s) in format of the BSR(MCHF) complex. These files simplify further consideration 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

```
>  bsr_hf  N_ground atom=N
```

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

```
>  bsr_hf  2s_2p4  atom=N  conf=2s(1)2p(4)
```

The ground configuration of the N has three terms ^{4}S, ^{2}D, ^{2}P. We may run term-dependent calculations as

```
>  bsr_hf  N_4S  atom=N   term=4S  inp=N.bsw  varied=2s,2p
>  bsr_hf  N_2D  atom=N   term=2D  inp=N.bsw  varied=2s,2p
>  bsr_hf  N_2P  atom=N   term=2P  inp=N.bsw  varied=2s,2p
```

Here we also illustrate the fixed-core calculations, keeping 1s-orbitals from the ground-state calculations.

Al:

```
>  bsr_hf  Al
```

Always it is good to begin the calculations from the ground state, e.i.
to fixed the core functions.

```
>  bsr_hf  3s3p2  atom=Al  conf=3s(1)3p(2)   inp=Al.bsw   varied=3s,3p
```

This is example to get the excited state with fixed core.

```
>  bsr_hf  3s24p  atom=Al  conf=3s(2)4p(1)  varied=3s,4p  inp=Al.bsw
>  bsr_hf  3s25p  atom=Al  conf=3s(2)5p(1)  varied=5p  inp=Al.bsw
```

Another example to get excited states from Rydberg serious.

Fe:

Iron is a tough atom for calculations with slow convergence.

```
>  bsr_hf Fe  acc=1
```

We begin with ground state configuration with additional parameter **acc=1** which indicate to use "acceleration" procedure, Eq.(34), for all iterations. To get the lowest state of spectrum, $3d^64s^2$ ^{5}D, we run

```
>  bsr_hf Fe_5D atom=Fe term=5D inp=Fe.bsw acc=1 varied=3s,3p,3d,4s
```

The low part of iron spectrum is dominated by $3d^64s^2$ and $3d^7$4s configurations. The 3d74s configuration we may get as

```
>  bsr_hf Fe_3d7_4s atom=Fe conf=3d(7)4s(1) inp=Fe.bsw  orb_tol=1.d-5
   varied=3s,3p,3d,4s
```

Here we reduce the orbital convergence to 10^{-5}, what is enough for most of applications.

Then considering the whole spectrum, it is important to get correct difference between levels. To improve this difference, we may apply simultaneous optimization of different configurations or separate levels. Let's prepare the **Fe_AV_conf.conf** file (in the BSR format)

```
Fe
  1s  2s  2p  3s  3p
  3d( 6)  4s( 2)                                                        2.0000
  3d( 7)  4s( 1)                                                        1.0000
  3d( 8)                                                                3.0000
*
```

When the run

```
>  bsr_hf Fe_AV_conf atom=Fe inp=Fe.bsw  varied=3s,3p,3d,4s
```

will optimized the above configurations with indicated weights. Changing the weights, we in principal may change the difference between energies of configurations. To do such procedure for individual states we need prepare the list of these state, e.g. file **Fe_AV_term.cgf** in BSR format, where we put first three state from iron spectrum

```
Fe
  1s  2s  2p  3s  3p
  3d( 6)  4s( 2)                                                        1.0000
 5D4 1S0 5D
  3d( 7)  4s( 1)                                                        1.0000
 4F3 2S1 5F
  3d( 7)  4s( 1)                                                        1.0000
 4F3 2S1 3F
*
```

When the run

```
>  bsr_hf Fe_AV_conf atom=Fe inp=Fe.bsw  varied=3s,3p,3d,4s
```

will optimize all three state on the same set of one-electron functions.

9 Conclusions

The single-configuration Hartree-Fock approximation may be considered a good starting point for the study of various atomic processes with atoms and ions. We tested the program for neutral atoms up to Z=104 and obtained good convergence in all cases, though HF wavefunctions are more appropriate for light atoms. The HF wavefunctions can be used as an initial approximation for more elaborate calculations in multiconfiguration Hartree-Fock approach. Due to the simple and flexible input-output interface, the program can be used in different small projects, e.g., in education. Because the results on one grid may be mapped onto a refined grid, it is possible to quickly obtain rough estimates of solutions and, as needed, refine the calculations for greater accuracy. Though larger B-spline expansions are needed, the present program can also easily obtain solutions for high *n* Rydberg states, or provide solutions at positive energies, which simulate the continuum spectrum. Such continuum pseudostates have found wide application in dealing with physical processes explicitly involving the atomic continuum.