

CPC 50th anniversary article

# BSHF: A program to solve the Hartree–Fock equations for arbitrary central potentials using a B-spline basis

D.T. Waide, D.G. Green <sup>\*</sup>, G.F. Gribakin

Centre for Theoretical Atomic, Molecular and Optical Physics, School of Mathematics and Physics, Queen's University Belfast, Belfast, BT7 1NN, Northern Ireland, United Kingdom

## ARTICLE INFO

### Article history:

Received 12 August 2019  
 Received in revised form 29 November 2019  
 Accepted 2 December 2019  
 Available online 18 December 2019

### Keywords:

Hartree–Fock  
 B-spline basis  
 Atomic structure  
 Arbitrary central potentials

## ABSTRACT

BSHF solves the Hartree–Fock equations in a B-spline basis for atoms, negatively charged ions, and systems of  $N$  electrons in arbitrary central potentials. In the B-spline basis the Hartree–Fock integro-differential equations are reduced to a computationally simpler eigenvalue problem. As well as solving this for the ground-state electronic structure self-consistently, the program can calculate discrete and/or continuum excited states of an additional electron or positron in the field of the frozen-target  $N$ -electron ground state. It thus provides an effectively complete orthonormal basis that can be used for higher-order many-body theory calculations. Robust and efficient convergence in the self-consistent iterations is achieved by a number of strategies, including by gradually increasing the strength of the electron–electron interaction by scaling the electron charge from a reduced value to its true value. The functionality and operation of the program is described in a tutorial style example.

### Program summary

*Program Title:* BSHF

*Program Files doi:* <http://dx.doi.org/10.17632/fj3y6c58dy.1>

*Code Ocean Capsule:* <https://doi.org/10.24433/CO.1226817.v2>

*Licensing provisions:* GPLv3

*Programming language:* Fortran 90.

*External routines/libraries:* LAPACK.

*Nature of problem:* Self-consistent solution of electronic structure for atoms and electrons in arbitrary central potentials in the Hartree–Fock approximation.

*Solution method:* A B-spline basis is employed that transforms the Hartree–Fock integro-differential equations to a computationally simpler eigenvalue problem. The eigenvalue problem is solved iteratively until self-consistency is achieved.

*Unusual or notable features:*

1. Robust and efficient convergence in the self-consistent iterations is achieved by gradually increasing the value of the electron charge from a reduced value to its true value, i.e., increasing the strength of the electron–electron interaction. In this way all orbitals are calculated simultaneously at every iteration of the self-consistency loop, i.e., without the need to successively fill the occupied shells from the core to valence (as most other Hartree–Fock codes require for convergence).
2. In addition to atoms and negative ions, the program solves the Hartree–Fock equations for systems of  $N$  electrons confined in an arbitrary central potential specified by the user: thus the program can be used to e.g., calculate the structure of electrons confined in harmonic potentials, which are known to approximate the electron gas.
3. In addition to calculating the ground state of the  $N$ -electron system, the program calculates the discrete and/or continuum excited states for an additional electron or positron in the field of the ‘frozen’  $N$ -electron target. It thus provides an orthonormal basis that can be used as input for many-body theory calculations.

*Restrictions:* The program solves *non-relativistic* Hartree–Fock equations for spherically symmetric systems only (in modelling systems other than atoms, e.g., harmonically confined electron gas or jellium-type models of clusters, relativistic effects can often be negligible and a non-relativistic

<sup>\*</sup> Corresponding author.

E-mail addresses: [dwaide01@qub.ac.uk](mailto:dwaide01@qub.ac.uk) (D.T. Waide), [d.green@qub.ac.uk](mailto:d.green@qub.ac.uk) (D.G. Green), [g.gribakin@qub.ac.uk](mailto:g.gribakin@qub.ac.uk) (G.F. Gribakin).

treatment is perfectly suitable). For open-shell atoms the central-field approximation is used, i.e., the potential is angularly averaged.

© 2019 Elsevier B.V. All rights reserved.

## 1. Introduction

The ability to calculate accurate electronic structure of many-electron systems is essential to much of theoretical atomic, molecular and optical physics and related fields. A multitude of properties and processes involving many-electron atoms rely on the knowledge of the atomic structure, at least as a starting point. Examples include, but are not limited to, calculations of polarisabilities and dispersion coefficients for atoms and molecules [1], leptonic interactions with atoms [2–5], generating atomic data for astrophysics (see e.g., [6]), and atomic physics tests of fundamental symmetries, e.g., parity violating studies, searches of the variation of fundamental constants, and atomic interactions with cosmic fields [7]. The general  $N$ -electron problem is, however, intractable. A favoured starting point for approximations is the independent-particle approximation, and more specifically the Hartree–Fock (HF) approximation (described in detail below). The HF equations can be sufficient for simple problems, but they also provide an efficient basis from which higher-order calculations can be performed, e.g., many-body theory calculations.

Programs for solution of the HF equations have been in development since this problem became computationally feasible in the 1960s, driven by pioneering efforts by Froese–Fischer, Brage, Johnsson, Hibbert and others in HF and post-HF methods such as Multi-Configuration Hartree–Fock and Configuration Interaction [8–11]. Notable codes specifically designed for atomic structure HF include the radial-grid based solver of [12], but to our knowledge there is only one generally available code that solves the HF equations in a B-spline basis [13] (we note that programs that solve the Dirac–Fock equations have also been published [14]). Solution of the problem in a B-spline basis in particular is desired as the resulting states efficiently span the energy continuum and provide rapid convergence in sums over intermediate states involving energy denominators in many-body calculations.

This *Computer Programs in Physics* paper presents the program ‘BSHF: a program to solve the Hartree–Fock equations for arbitrary central potentials using a B-spline basis’. It solves the non-relativistic Hartree–Fock equations for arbitrary central potentials in a B-spline basis, not just Coulomb potentials. The additional ability to solve the HF equations for arbitrary potentials is useful e.g., as recent studies have shown that a finite number of electrons and positrons confined in harmonic potentials can approximate the electron gas [15], opening the way for finite-system many-body calculations. Moreover, our BSHF program employs a convergence strategy that relies on gradually increasing the strength of the electron–electron interaction from a weakened value to its true value, doing so by gradually scaling the value of elementary charge from below unity to unity (we work in atomic units throughout). For atoms, this approach enables all orbitals to be calculated at every iteration of the self-consistent procedure (rather than having to build up the orbitals), and in all cases results in robust and efficient even convergence, even for negative ions that otherwise struggle to converge. For electrons confined in harmonic potentials, we found that traditional convergence methods failed, and convergence could only be achieved through this scaling of the charge. Comparison of the results for neutral atoms using typical spline parameters we have found that BSHF gives ground-state energies with a relative

accuracy of  $\sim 10^{-8}$  with respect to tabulated HF benchmark values [16]. This paper presents the source code, focusing on a presentation of its functionality and operation via a tutorial-style example.

### 1.1. The radial Hartree–Fock equations

In the Hartree–Fock approximation [17,18], the total wavefunction of an  $N$ -electron system of energy  $E$  is approximated as a Slater determinant (sum of Slater determinants in general) that is an antisymmetrised product of  $N$  single-electron spin orbitals  $\phi_{\alpha_j}(x_j)$ , viz.  $\Psi_E(x_1, \dots, x_N) = \sqrt{N!} \hat{\mathcal{A}} \prod_j^N \phi_{\alpha_j}(x_j)$ , where  $\hat{\mathcal{A}}$  is the antisymmetrisation operator,  $\alpha_j$  represents a complete set of quantum numbers describing the  $j$ th orbital, and  $x_j = (\mathbf{r}, \sigma)$  represents coordinates of position and spin. Minimising the expectation value of the Hamiltonian through the variation of the  $\phi_{\alpha}$  [17,18], or in the diagrammatic approach summing a certain class of diagrams [19,20], yields the system of  $N$  integro-differential equations, the Hartree–Fock equations

$$\left( -\frac{1}{2} \nabla^2 + V(r) + \hat{V}^{\text{HF}} \right) \phi_j(x_j) = \varepsilon_j \phi_j(x_j). \quad (1)$$

Here the first term in the bracket is the kinetic energy operator. The second term is a local central potential  $V(r)$ , which for an atom with atomic number  $Z$  is  $V(r) = -Z/r$ , but in the BSHF program can also be chosen alternatively to be an arbitrary central potential (e.g., a harmonic confining potential, a system of electrons in which has been shown to approximate the electron gas [15]). The Hartree–Fock potential  $\hat{V}^{\text{HF}} = \sum_{i=1}^N (\hat{J}_i - \hat{K}_i)$  is a sum of the direct and (non-local) exchange terms  $\hat{J}_i \phi_j(x_j) = \int dx_i \phi_i^*(x_i) r_{ij}^{-1} \phi_i(x_i) \phi_j(x_j)$  and  $\hat{K}_i \phi_j(x_j) = \int dx_i \phi_i^*(x_i) r_{ij}^{-1} \phi_j(x_i) \phi_i(x_j)$ , where  $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$ . Eq. (1) demands a self-consistent solution due to the interdependency of the Coulomb mean field potential  $V^{\text{HF}}$  and the electron orbitals  $\phi$ .

For closed-shell central-field problems  $\phi_{nlm\sigma} = r^{-1} P_{nl}(r) Y_{lm}(\theta, \phi) \chi_{\sigma}$ , where  $P_{nl}$  is the radial wavefunction and  $Y_{lm}$  is a spherical harmonic with principal quantum number  $n$  and orbital angular momentum  $l$  and its projection  $m$ , and  $\chi_{\sigma}$  is the (two-component) spinor, Eq. (1) leads to the radial Hartree–Fock equations,<sup>1</sup>

$$\left( -\frac{1}{2} \frac{d^2}{dr^2} + V_i(r) + \hat{U}_i \right) P_i(r) = E_i P_i(r), \quad (2)$$

where  $i$  now enumerates the orbitals  $nl$ . Here  $V_i(r)$  is the local, direct (‘Hartree’) potential, representing the interaction between a single particle and the average field of the other particles

$$V_i(r) = V(r) + \frac{Y_i(r)}{r} + \frac{l_i(l_i + 1)}{2r^2}, \quad (3)$$

where

$$Y_i(r) = \sum_{j=1}^s (N_i - \delta_{ij}) Y_{ij}^0(r) - \frac{2}{N_i} \sum_{j=1}^s \sum_{\lambda>0} \gamma'_{\lambda}(j) Y_{ij}^{\lambda}(r), \quad (4)$$

<sup>1</sup> Eq. (2) may also be used for open-shell electronic configurations. In this case the solution is a further approximation where the wavefunctions have been spherically averaged. In the case of the ground state for open-shell neutral atoms this difference is relatively small since the variation from the ideal case arises due to only one incomplete subshell. BSHF handles open-shell atoms under this approximation.

$$Y_{ij}^\lambda(r) = r \int_0^\infty \frac{r_{<}^\lambda}{r_{>}^{\lambda+1}} P_i(r') P_j(r') dr' \quad (5)$$

Here  $r_{<}$  ( $r_{>}$ ) is the lesser (greater) of the radial variables  $r$ ,  $r'$ , and  $\lambda$  is the angular momentum transferred by the Coulomb interaction, in accordance with the triangle rule,  $|l_i - l_j| \leq \lambda \leq l_i + l_j$ , where  $l_i$  is the orbital angular momentum of the  $i$ th orbital, and  $\gamma_\lambda(j)$  are angular momentum coupling coefficients [21]. The potential

$$\hat{U}_i P_i(r) \equiv \int_0^\infty U_i(r, r') P_i(r') dr', \quad (6)$$

where  $U_i(r, r')$  is the non-local exchange kernel (Fock term), which is the interaction between two particles from the exchange of their coordinates:

$$U_i(r, r') = \frac{1}{N_i} \sum_{j=1, j \neq i}^s \sum_{\lambda} \gamma_\lambda(i, j) \frac{r_{<}^\lambda}{r_{>}^{\lambda+1}} P_j(r) P_j(r'). \quad (7)$$

Here,  $N_i$  is the number of electrons in the  $i$ th orbital and  $j$  sums over all pairs of distinct electron orbitals, where there are  $s$  ground-state orbitals in the system, and  $\gamma_\lambda(i, j)$  are angular momentum coupling coefficients [21].

In the case of an additional electron or positron in the frozen-core of an  $s$  orbital system the wavefunction of the additional particle, labelled  $s+1$ , is

$$\left( -\frac{1}{2} \frac{d^2}{dr^2} + V_{s+1}(r) + \hat{U}_{s+1} \right) P_{s+1}(r) = E_{s+1} P_{s+1}(r). \quad (8)$$

## 1.2. Radial Hartree–Fock equation in a B-spline basis

BSHF solves the Hartree–Fock equations in a B-spline basis [22]. The broad utility of B-splines in atomic physics has been reviewed in [23,24]. B-splines of order  $k_s$  are piecewise polynomials of degree  $k_s - 1$  defined over a restricted domain ('box') that is divided into  $n_s - k_s + 1$  segments by a knot sequence of  $n_s$  points:  $r_i \in [0, R]$ , where  $n_s$  is the number of non-zero splines in the basis [22]. (See Fig. 1 panel (a) for an example set of splines.) In BSHF two choices for the knot sequence are coded. The first is the exponential knot sequence

$$r_j = \begin{cases} 0, & \text{for } 1 \leq j \leq k_s, \\ \rho_0(e^{\sigma j} - 1), & \text{for } k_s \leq j \leq n_s + 1, \\ R, & \text{for } n_s + 1 \leq j \leq n_s + k_s, \end{cases} \quad (9)$$

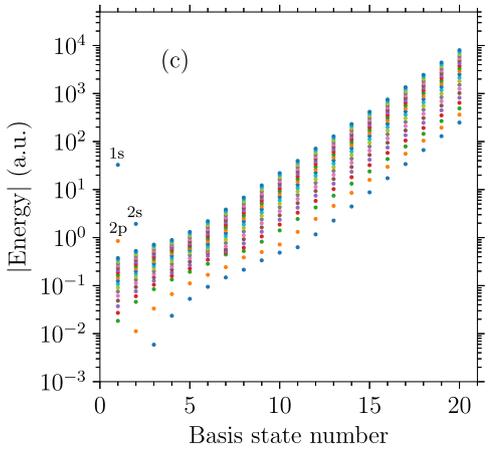
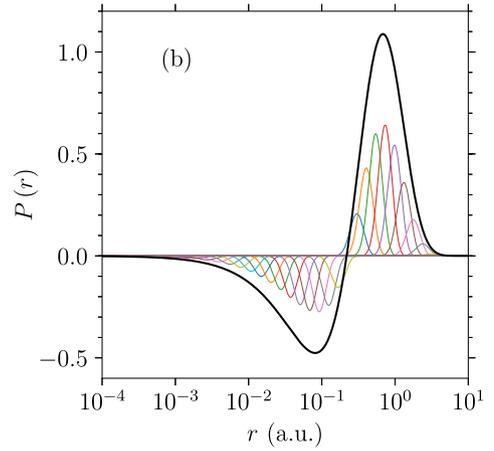
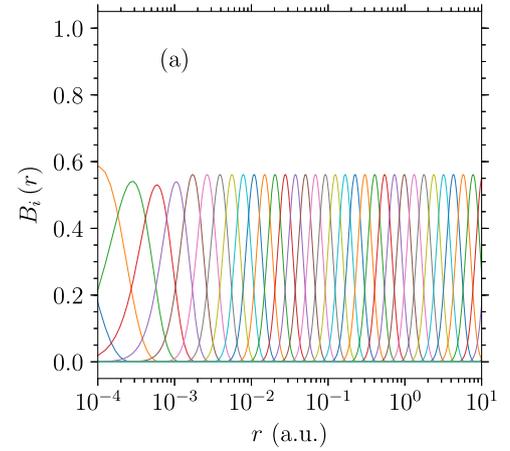
where typically  $\rho_0 = 10^{-3}$  and  $\sigma$  is determined by the condition that  $r_{n_s - k_s + 1} = R$  (see [3] for more details). The second is the quadratic-linear knot sequence

$$r_j = \begin{cases} 0, & \text{for } 1 \leq j \leq k_s, \\ A(j - k_s)^{m+1} R / (B^m + (j - k_s)^m), & \text{for } k_s \leq j \leq n_s + 1, \\ R, & \text{for } n_s + 1 \leq j \leq n_s + k_s, \end{cases} \quad (10)$$

where  $A = [B^m + (n_s + 1 - k_s)^m] / (n_s + 1 - k_s)^{m+1}$  and  $B$  is a parameter (typically  $B = 100$ ) (see [25] for more details). Both simultaneously provide an accurate representation of the atomic ground-state orbitals and provide positive-energy electron and positron states that efficiently span the energy continuum (see [3,25] for details of their relative merits and for the circumstances under which one knot sequence may be preferred).

Expanding the single-particle HF orbital of angular momentum  $l$  in the B-spline basis as

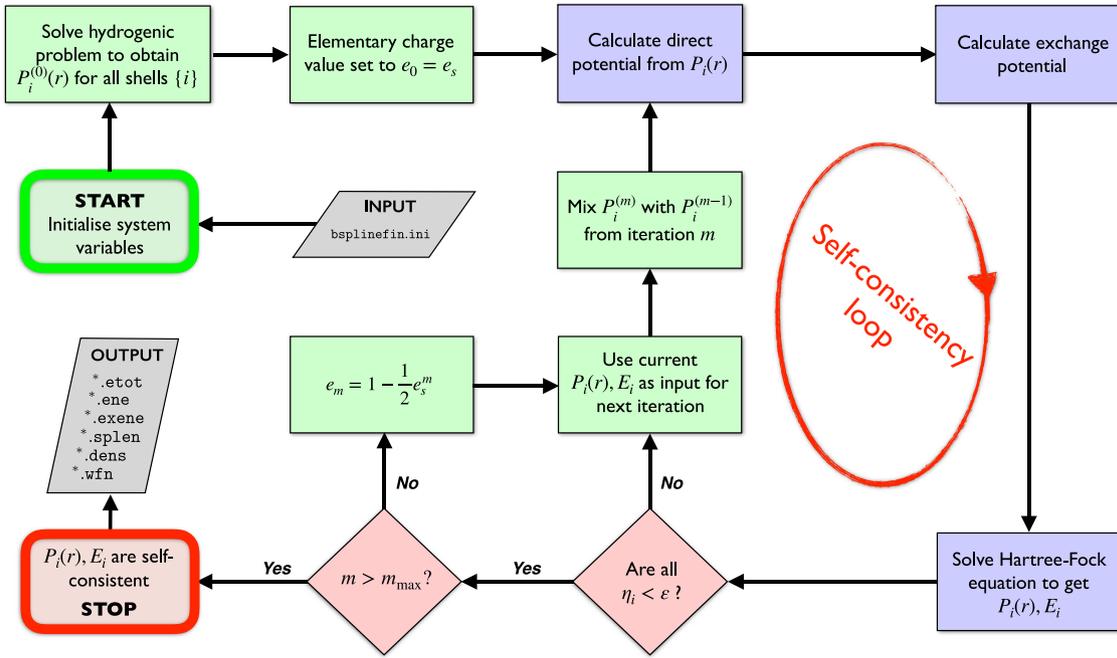
$$P_{nl}(r) = \sum_i c_i^{(nl)} B_{i, k_s}(r), \quad (11)$$



**Fig. 1.** (a) the set of 40 B-splines of order 6 defined on an exponential knot sequence with box radius 30 a.u. (b) An example: the calculated Ne 2s radial wavefunction (thick black line) and its decomposition in the spline basis (thin lines) as in Eq. (11). (c) Eigenvalues  $\varepsilon^{(l)}$  of the HF equation in the B-spline basis Eq. (12) for angular momenta  $l = 0$  (lowest energies) to 15 (highest energies) for Ne. The energies efficiently span the positive-energy continuum.

where  $c_i^{(nl)}$  are the expansion coefficients and  $B_i$  is the  $i$ th spline of the basis, the set of  $s$  integrodifferential Hartree–Fock equations is reduced to a computationally simpler symmetric generalised eigenvalue equation for a specific angular momentum  $l$ ,

$$\mathcal{H}^{(l)} \mathbf{c}^{(l)} = \varepsilon^{(l)} \mathcal{B} \mathbf{c}^{(l)}, \quad (12)$$



**Fig. 2.** Schematic of the BSHF algorithm. The hydrogenic case is used as an ansatz and the solution is iteratively refined until the error in the energies and wavefunctions is sufficiently small.

where

$$\mathcal{H}_{ij}^{(l)} = \int_0^R dr \left\{ \frac{1}{2} \frac{dB_i}{dr} \frac{dB_j}{dr} + B_i(r) \left( V(r) + \frac{l(l+1)}{2r^2} + V^{\text{HF}}(r) \right) B_j(r) \right\}, \quad (13)$$

$\mathcal{H}_{ij}^{(l)} \equiv \langle B_i | H_0^{(l)} | B_j \rangle$  is the matrix element of the radial part of the HF Hamiltonian for each angular momentum  $l$  [23], and  $B_{ij} \equiv \langle B_i | B_j \rangle$ . Note that to implement the boundary conditions  $P(0) = P(R) = 0$  the first and last splines are discarded in the expansions. Thus, the solutions of the equation (e.g. see Fig. 1(b)) for a given angular momentum are a set of  $n_s - 2$  orthogonal eigenfunctions and the matrices  $\mathcal{H}^{(l)}$  and  $\mathcal{B}$  consist of  $(n_s - 2)(n_s - 2)$  elements. For the electron, the lowest energy states from this set correspond directly to the ground-state wave functions, and the remaining  $(n_s - 2) - s$  states of the set describe an additional electron propagating in the HF field of the ‘frozen’ ground-state  $N$ -electron target system (consider solution of Eq. (8), where the  $N$ -electron target is frozen). For an additional positron all states are positive-energy continuum states that describe the positron propagating in the HF field of the ground-state  $N$ -electron target system. Thus BSHF has the capability to provide an orthonormal<sup>2</sup> basis for higher-order calculations that include electron and positron correlations, e.g., through many-body theory, where the fact that B-splines efficiently span the positive energy continuum (see Fig. 1(c)) makes them particularly useful due to the rapid convergence of sums over intermediate states that contain energy denominators [2–5,20,26].

## 2. BSHF algorithm

### 2.1. Overview of the BSHF algorithm

Fig. 2 shows a schematic of the BSHF algorithm. The program initiates the self-consistent solution of Eq. (12) by first solving

<sup>2</sup> The diagonalisation routine called in the program returns orthogonal states for a given  $l$ . The ground state wavefunctions are normalised as  $\int_0^\infty |P_i(r)|^2 dr = 1$ , whereas the excited state wavefunctions are not normalised.

for the non-interacting electron case, i.e.,  $V_{\text{HF}} = 0$  in Eq. (13). The resultant orbitals are used to build the Hartree–Fock Hamiltonian matrix Eq. (13), and Eq. (12) is solved repeatedly, initially only for values of orbital angular momentum represented by the occupied orbitals, until the difference in successive approximations of the wavefunction and energy for all orbitals  $\eta_i < \epsilon$ , where  $\epsilon$  is a user-specified threshold (typically  $10^{-6}$ ), and

$$\eta_i = \max \left( \left| P_i^{(m)}(r) - P_i^{(m+1)}(r) \right|, \left| E_i^{(m)} - E_i^{(m+1)} \right| \right). \quad (14)$$

Typically, in the atomic case the orbital energies will converge much faster than the wavefunction. However, for an arbitrary central potential, which BSHF includes the capability of considering, the scales of the energies and wavefunctions may be different. This criterion ensures that both have converged, regardless of the nature of their interdependence.

Robust and efficient convergence in the self-consistent iterations is achieved by two main strategies. First, gradually increasing the value of the electron charge from a reduced value to its true value. The value of elementary charge is initially chosen to be smaller than unity so that the effect of the electron–electron interaction on the solution to the HF equations is weakened. Once the solutions are self-consistent, then the strength of the elementary charge is increased towards unity according to

$$e_m = 1 - \frac{1}{2} e_s^m, \quad m = 1, \dots, m_{\text{max}} \quad (15)$$

where  $m_{\text{max}}$  then is the number of increments to take before setting the charge to unity (typically  $m_{\text{max}} = 10$ ), and iterating further until self-consistency is reached. In this way *all* the occupied orbitals are calculated simultaneously at every iteration of the self-consistency loop, i.e., without the need to successively fill the occupied shells from the core to valence as most other Hartree–Fock codes do. We found that for calculations of electrons in harmonic potentials (and in some cases negative ions) ‘traditional’ convergence approaches failed, and this procedure was essential. Secondly, convergence can be accelerated by determining the new estimates of the wavefunctions and energies

at step  $m$  as a linear combination of their values at the current iteration and the previous one, viz.

$$P_i^{(m)}(r) = (1 - \alpha)P_i^{(m)}(r) + \alpha P_i^{(m-1)}(r), \quad (16)$$

where, following [21],

$$\alpha = \frac{E_i^{(m)} - E_i^{(m-1)}}{E_i^{(m)} - 2E_i^{(m-1)} + E_i^{(m-2)}}. \quad (17)$$

Finally, once the solutions of the occupied orbitals are self-consistent to tolerance  $\varepsilon$  then one further iteration is performed, finding the states of the additional electron or positron in the field of the frozen-core (i.e., ground state orbitals are invariant) for all angular momenta required as specified by the user.

To demonstrate, Fig. 3(a) shows the convergence of the energy levels of both Ne and the negatively charged ion  $F^-$  using  $n_s = 40$ ,  $k_s = 6$ , and  $R = 40$  a.u., (b) for a system of  $N$ -electrons in a harmonic potential  $V(r) = \omega r^2/2$  with  $\omega = 1.0$ , and (c) the number of iterations required to obtain converged calculations for the noble gas atoms at a tolerance value of  $\varepsilon = 10^{-6}$  and  $10^{-12}$  [see Eq. (14)]. As the aim of this paper is to introduce and explain the operation of the code, the reader is referred to [27] for comprehensive benchmarks for both the noble-gas atoms for which comparisons with experiment are made, and for systems of  $N$ -electrons ( $N$  ranging from 2 to 106) in harmonically confined potentials (of interest as an approximation to the electron gas [15]) for which comparisons with the Thomas–Fermi method are made. Briefly, comparison of the results for neutral atoms using 40 non-zero splines of order 6 shows that the ground-state energies have a relative accuracy of  $\sim 10^{-8}$  with respect to tabulated HF benchmark values [16].

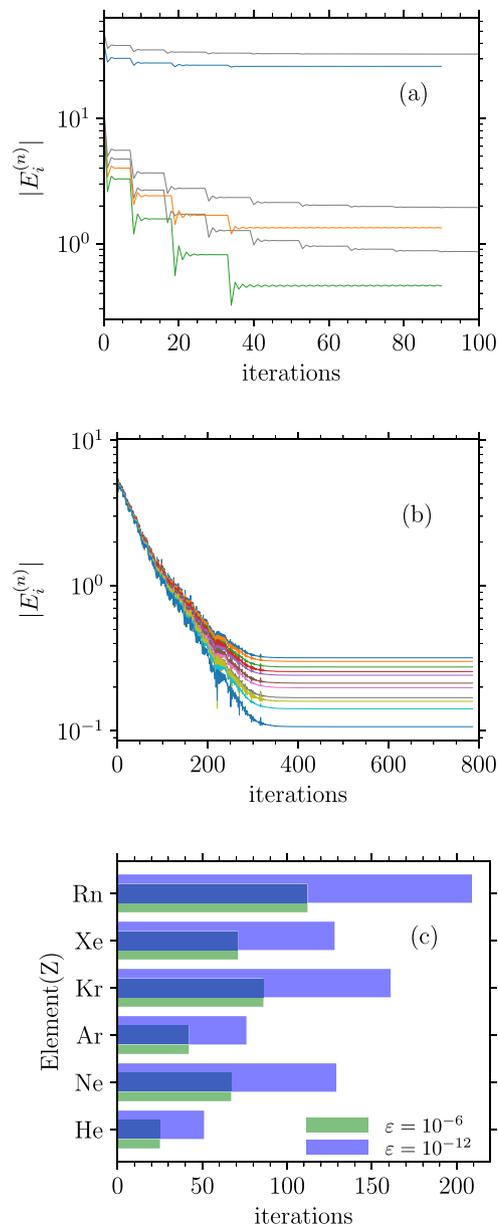
## 2.2. Calculation of integrals

The wavefunctions are principally stored as an array of the  $n_s$  B-spline coefficients  $c_i^{(l)}$  for each distinct orbital angular momentum  $l$  Eq. (11). This is sufficient for solving the exchange integrals, whose calculation is described in Section 2.2.2. For the calculation of the direct potential and also for post-processing purposes, it is instructive to also determine the wavefunctions on radial grids as discussed below.

### 2.2.1. Direct potential integrals

The contribution of the direct part of the two-body potential is calculated by recasting it as a system of two ordinary differential equations that divide the domain of the potential into two regions separated by a cusp at  $r = r'$ . These are solved via the three-point Adams–Moulton method, following [12,21] (where full details of the method are given). This requires an equally spaced grid but due to a singularity in the potential at  $r = 0$  we introduce a coordinate grid  $\rho = \alpha r + \beta \ln r$ , to allow for more points close to the singularity (where  $\alpha$  and  $\beta$  are constants evaluated by the program). The resulting solution is then quadratically interpolated onto the exponential grid on which the B-splines are defined, before being projected onto the spline basis to create the elements of the direct potential matrix.

The main advantage of this technique is that the computational resources required scale linearly in the total number of radial points, rather than quadratically as would be the case for a direct calculation of the two-body Coulomb integral. Moreover, this procedure accurately handles the cusp at  $r = r'$  which is especially prominent when the angular momenta transferred is large.



**Fig. 3.** (a) Energy levels of Ne (grey) and  $F^-$  (coloured) at each iteration of the self-consistent procedure. Note the sharp drop in the energy when the elementary charge is increased. Between these drops the charge is constant and the energy reaches a self consistency specified by the parameter `step_eps`. The parameters used were  $n_s = 40$ ,  $k_s = 6$ ,  $r_0 = 0.001$ , and  $R = 30$ . (b) Energy levels for a system of  $N$ -electrons in a harmonic potential  $\omega = 1.0$  at each iteration of the self-consistent procedure. (c) Number of iterations required to obtain converged energies for all orbitals for the noble gases He to Rn to the user specified thresholds  $\varepsilon = 10^{-6}$  and  $10^{-12}$  [see Eq. (14)] with  $m_{\max} = 10$ . (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

### 2.2.2. Exchange potential integrals

The majority of the computational time is taken by the calculation of the non-local potential Eq. (7). In the B-spline basis its calculation can be divided into a sum of basis coefficients and an integral over groups of B-splines. Given a polynomial of order  $2k_s - 1$  then a grid of  $k_s$  Gauss–Legendre points over the domain of the function allows exact integration [28]. Thus the integral over products of B-splines (each a polynomial of order  $k_s$ ) is independent of any particular wavefunction and may be pre-calculated exactly and stored in a file for reuse.

Inserting Eqs. (7) and (11) into Eq. (6), projecting onto the B-spline basis and using Gauss–Legendre quadrature with weights  $w(\cdot)$  where  $\cdot$  is an index into an array of radial abscissae, we find the exchange part of  $\mathcal{H}_{ij}^{(l)}$  [Eq. (13)] as

$$U_{ij}^{(l)} = \frac{1}{2} \sum_{b=1}^s \sum_{\lambda} N_b \begin{pmatrix} l & \lambda & l_b \\ 0 & 0 & 0 \end{pmatrix}^2 \sum_{i'j'} c_{i'}^{(b)} c_{j'}^{(b)} \mathcal{B}_{ij'i'j'}^{(\lambda)}, \quad (18)$$

where  $b$  denotes the set of quantum numbers of an orbital and  $\mathcal{B}$  are the precalculated terms in the exchange integrals, defined by

$$\mathcal{B}_{ij'i'j'}^{(\lambda)} = \sum_{x,y} \frac{r_{<}^{\lambda}}{r_{>}^{\lambda+1}} w(x)w(y)B_i(x)B_j(x)B_{i'}(y)B_{j'}(y), \quad (19)$$

where  $x$  and  $y$  enumerate the quadrature points,  $r_{<} = \min\{r(x), r(y)\}$ ,  $r_{>} = \max\{r(x), r(y)\}$ , and  $r(x)$  and  $r(y)$  are the corresponding values of the radius. Naively it would take  $\lambda_{\max} \cdot n_s^4$  matrix elements, where  $\lambda_{\max}$  is the greatest value of coupled angular momentum necessary, memory entries but by exploiting symmetry and properties of the splines only  $\lambda_{\max} \cdot n_s^2(2k_s - 1)^2$  are required in practice.

### 3. BSHF compilation, code structure, input, execution and output

#### 3.1. Compilation

BSHF is written in the programming language Fortran 90. It was developed and tested on an x86\_64 Linux Beowulf cluster using the Intel compiler ifort©v19. The matrix diagonalisation routines require compiler linkage to the Intel MKL library, or alternatively LAPACK if using gfortran. Compilation of the code is facilitated via a Makefile located in directory 'BSHF/'. The Makefile assumes either an Intel Fortran compiler or the gfortran compiler on a computer with either x86\_64 or i386 architecture: the user should modify the specified architecture or compiler Makefile variables if alternatives are to be used. To compile the program, simply navigate to the 'BSHF/' directory and type 'make clean; make BSHF'. A directory will be created named according to the machine architecture, e.g. 'BSHF/x86\_64/', in which all the object files and the compiled binary 'BSHF' are generated.

#### 3.2. Code structure

The program is driven by the main routine `./src/schf.f90` which reads in the input, generates the splines and radial grid, and manages the self-consistency loop via calls to subroutines. Whilst the user is free to modify and subroutine, three sections of the code are designed specifically to be easily modified by the user. The first is the ability to specify a customised central potential. The code provides a function `custom_potential()` in the file `./src/mod_potential` which may be straightforwardly modified to include any user defined background potential, other than the Coulomb potential provided. The namelist definition in this file also allows easy specification of related input parameters from the main configuration file. These new entries should then be listed in the main configuration file under the `potential_parameters` heading. The second is the ability to define a customised B-spline knot sequence, as the user may wish to explore problems for which tailored knot sequences may be more suited than either of the exponential and quadratic knot sequences that the code provides (these knot sequences have been described above). The knot sequence used is defined by the `getknots` subroutine in the file `./src/mod_splines.f90`. This subroutine returns a real-valued ordered list of the knot

points and can be replaced by any sequence desired by the user. Thirdly, all the data output to file is specified in subroutines in `./src/mod_bshf_output.f90`. These files are described in detail below and further in the README file supplied with the code. All the files are formatted, human-readable output defined in a separate subroutine for each file.

#### 3.3. Input configuration

The BSHF program reads a single input file 'bsplinefin.ini' by default, which contains the input data values as a Fortran namelist. Example input files have been provided in the 'BSHF/init' directory, including for atoms and for electrons in a harmonic potential. Here we consider the example of Ne, using a basis of 40 splines of order 6 defined on an exponential knot sequence with box radius 30 a.u. The various namelist variables (grouped by which properties they control) are thus set as follows (note that the initial numbers are for reference in this text only, and the order of the parameters in the namelist file is irrelevant):

##### A. parameters that specify the B-spline basis (see Section 1.2)

1. n\_s = 40
2. k\_s = 6
3. r = 30.d0
4. knotseq = 0
5. ro = 0.001
6. mpowr = 1
7. bcons = 100.d0

which specify respectively: 1. the number of splines  $n_s$  in the basis; 2. the order of splines  $k_s$ , 3. the radius (in a.u.) of the bounding 'box' in which the atom is placed at the centre; 4. the choice of knot sequence (set `knotseq = 0` for exponential knot sequence or `knotseq = 1` for quadratic-linear); 5. the value of the constant used for the exponential knot sequence  $\rho$  [see Eq. (9)] (used only if `knotseq = 0`); 6. the value of the power  $m$  for the quadratic linear knot sequence [see Eq. (10)] (used only if `knotseq=0`); 7. the value of  $B$  for the quadratic-linear knot sequence [see Eq. (10)] (used only if `knotseq = 0`).

##### B. parameters related to integration

8. np = 15
9. numax = 15
10. ltot\_max = 25
11. intfilename = "n40k6.dat"

which specify: 8. the number of points for Gauss–Legendre integration (see Section 2.2); 9. the number of points between knot-points on the exponential grid (see Section 2.2); 10. the maximum combined value of angular momentum for which to pre-calculate the integral part of each matrix element; 11. the name of the file to which the pre-calculated integral data is stored (see Section 2.2).

##### C. parameters that specify the system properties

12. z = 10
13. is = 3
14. npq = 1,2,2
15. lorb = 0,0,1
16. nel = 2,2,6
17. potential = "coulomb"
18. omega = 1.0

which specify: 12. the atomic number of the atom or ion (if the system is atomic-like); 13. the number of occupied orbitals; 14–16. the orbital principal quantum number (npq), angular momentum lorb and occupation nel of orbital  $P_{nl}$ ; 17. nature of

the central potential. Arbitrary functions can be used via a native Fortran function or using values on a grid read in from a file: set to "coulomb", "harmonic", "function" or "file"; 18. the angular frequency of a harmonic potential  $U = 1/2\omega r^2$ , if selected using "potential=harmonic".

#### D. parameters controlling self-consistency

19. scf\_eps = 1.0d-12
20. max\_iter = 1000
21. step\_eps = 1.0d0
22. e\_s = 0.5
23. m\_max = 10
24. do\_mixing = .false.

which specify, respectively: 19. the maximum discordance  $\varepsilon$  below which the self-consistent procedure will be deemed to have converged [see Eq. (14)]; 20. the maximum number of iterations in the self-consistent loop; 21. the value of  $\varepsilon$  used while  $e < 1$ ; 22. initial value of the elementary charge, calculated by  $e_m = 1 - e_s^m/2$  [see Eq. (15)]; 23. the number of values to take for  $e$  before setting it to unity; 24. logical variable switch to turn the orbital mixing on (.true.) or off (.false.).

#### E. parameters that control calculation for additional particle in field of frozen-core $N$ -electron ground state

25. ich = 0
26. lmax = 20

which specify respectively the charge of the excited particle (ich = 0 if electron or ich = 1 if positron), and the maximum angular momentum  $l$  for which to calculate the wavefunctions  $P_{nl}$  for the additional particle in the field of the frozen  $N$ -electron ground state.

#### F. parameters that control output

27. iprintlv = 0
28. ksav = 10
29. outfile = "Ne.wf"

which specify: 27. the verbosity level of the output, 28. the unit number of additional scratch output for diagnostic level information, 29. the name of the output file (described below).

#### 3.4. Execution of the program

The program can be executed with the command './x86\_64/BSHF' if using the default bsplinefin.ini input file or './x86\_64/BSHF -c filename.ini' if using a user-named input file.

As discussed in Section 2.2 integrals involving the spline parameters only are pre-calculated before the self-consistent solution of the HF equations for a given system. This integral data is written to the file 'intfilename' named by the user in the bsplinefin.ini input file. Computation of these integrals happens automatically if 'intfilename' does not exist. If it does exist, then the integrals are not re-computed, and instead 'intfilename' is read (the parameters used to generate it are checked against their current values in the namelist file, and an error is reported if these differ). One can also generate the pre-computed integral data file only, i.e., without continuing with the self-consistent calculation. This can be useful if one wishes to run BSHF for many systems with the same spline parameters without having to compute the integral data each time. To do this, one should first ensure that the parameters 'ro', 'r', 'n\_s', 'k\_s', 'lmax\_tot', and 'intfilename' are set correctly in the input file (bsplinefin.ini by default). Then, assuming the input file is in the 'BSHF/' directory, type './x86\_64/BSHF -p'

if using the default bsplinefin.ini file, or './x86\_64/BSHF -p -c filename.ini' if using a user-named input file. Once the integrals have been computed and saved to disk, the program can again be executed with the command './x86\_64/BSHF' if using the default bsplinefin.ini file or './x86\_64/BSHF -c filename.ini' if using a user-named input file. There are a number of additional available execution options, e.g., specifying: './BSHF -f' will write extra files to disk containing knots, splines, energies; and './BSHF -e' will make the code print excited state energies to screen etc. More details are given in the README.md file.

#### 3.5. BSHF output

##### 3.5.1. Output to screen

At each step of the self-consistent cycle the program prints the iteration number, the current value of  $e_n$ , the calculated value of  $\eta_i$  for orbital  $i$ , and the energy for each of the occupied orbitals  $i$ , e.g., for our example, at iteration step number 54 the following output is given:

$e_n = 1.0$			
iter	$\eta$	$nl$	$\varepsilon_{nl}$
54	9.41E-14	1s	-3.27724E+01
	3.06E-13	2s	-1.93039E+00
	7.64E-13	2p	-8.50414E-01

Once convergence has been achieved the contributions to the total energy of the  $N$ -electron system, calculated as

$$E_{\text{tot}} = \sum_{i=1}^s N_i \left[ \varepsilon_i - \frac{1}{2} \sum_{a,b}^{n_s} c_a^{(i)} c_b^{(i)} U_{ab} \right], \quad (20)$$

where  $U_{ab}$  is an element of the two-body potential matrix, are given

```
e_kin      = 1.285470940071851942E+02
e_pot1     = -3.111331610437827635E+02
e_pot2     = 5.40389499426360089E+01
e_tot_gen  = -1.285471170939615604E+02
e_tot_gen / e_kin = -1.0000001795978084E+00
```

where the lines are the kinetic energy, the potential energy due to the local potential  $U(r)$ , the potential energy due to, the total energy and the ratio of the total energy to the kinetic energy.

##### 3.5.2. Main output file

The program generates a single output file by default, whose name is specified in the input namelist ('Ne.wf' in our example). The first line of this file lists  $n_s$  (the number of splines),  $k_s$  (the spline order),  $Z$  (atomic number, or value of omega if harmonic potential is used),  $np$  (the number of Gauss-Legendre points used),  $l_{\text{max}}$  (the maximum angular momenta for which the eigenvalue problem was solved). The second line lists the  $n_s + k_s$  radial knot points, the  $n_{\text{pt}} = (n_s + k_s) * np$  radial grid points resulting from interlacing the spline grid with the Gauss-Legendre points, and the corresponding  $n_{\text{pt}}$  Gauss-Legendre weights. The third record is a list of the value of the  $n_s$  B-splines ( $B_j(r_i)$ ,  $j = 1, n_s$ ),  $i = 1, n_{\text{pt}}$  at the  $n_{\text{pt}}$  grid points  $r_i$ . The fourth record lists the occupation numbers and angular momenta of all occupied orbitals. The subsequent  $l_{\text{no}}+1$  lines, where  $l_{\text{no}}$  is the orbital angular momentum for which Eq. (12) is solved, list the  $n_s - 2$  eigenvalues and elements of the eigenvectors as  $(e_n(j, l_{\text{no}}+1), j=1, n_2), ((eiv(1, j, l_{\text{no}}+1), l=1, n_2), j=1, n_2)$ .

### 3.5.3. Additional output files

Using the option '-f' also outputs a series of diagnostic files with the same base name as 'outfile' but different extensions. These files are all formatted, i.e. plain text records, and contain the following:

1. 'outfile.etot' stores the following on separate lines:

- number of iterations before convergence
- total energy of the atom by the virial theorem, or by statistical summation of the mean energies of individual electrons where the virial theorem does not apply.
- ground state energies of occupied orbitals in increasing order

2. 'outfile.ene' [used to produce Fig. 1(a) and (b)] stores the following, one line per iteration:

- current iteration
- ground state energies of occupied orbitals in respective columns

3. 'outfile.exene' stores the following per line:

- $l$ , orbital angular momentum quantum number.
- $n$  principal quantum number.
- energy of the occupied or excited state ' $(n, l)$ '.

4. 'outfile.splen' [used to produce Fig. 1(a) and (b)] stores the following per line:

- $l$ , orbital angular momentum quantum number.
- $i$ , B-spline basis state index  $i = 1, n_s - 2$ .
- energy of the basis state  $i$ .

5. 'outfile.dens' stores in columns, the radial grid and the total density of the ground state electrons at each point on that grid.

6. Finally, 'outfile.wfn' stores the exponential grid with  $(n_s - k_s + 1) * \text{numax}$  points and the value of each occupied and excited wavefunction at each point on the grid in separate columns. The occupied states are prefixed by 'gnd\_wfn $s$ ' and the excited states are prefixed by 'lno= $x$ ' where ' $x$ ' is the angular momentum of the excited state. Full details of the files, including the format specifiers for each record are available in the README file supplied with the code.

## 4. Summary

We presented the BSHF program that solves the Hartree-Fock equations in a B-spline basis for atoms, negatively-charged ions and electrons in arbitrary central potentials, detailing its operation and functionality. BSHF achieves convergence in the self-consistent iterations by gradually increasing the value of the electron charge from a reduced value to its true value. In this way all orbitals can be calculated simultaneously at every iteration of the self-consistency cycle. In addition to calculating the ground-state of the  $N$ -electron system of interest, BSHF provides the states that describe an additional electron or positron propagating in the Hartree-Fock  $N$ -electron frozen target in the B-spline basis.

It thus provides orthonormal bases and can be used as a starting program for higher-order calculations that involve electron and positron correlations, e.g., many-body theory calculations.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## CRediT authorship contribution statement

**D.T. Waide:** Conceptualization, Methodology, Software. **D.G. Green:** Conceptualization, Methodology, Software, Supervision. **G.F. Gribakin:** Conceptualization, Methodology, Software, Supervision.

## Acknowledgements

DTW was supported by a Ph.D. studentship from Department for Education, Northern Ireland, United Kingdom. DGG gratefully acknowledges funding from by the European Research Council, Starting Grant project 'ANTI-ATOM' (804383).

## References

- [1] J. Mitroy, M.S. Safronova, C.W. Clark, *J. Phys. B* 43 (2010) 202001.
- [2] Y. Cheng, L.Y. Tang, J. Mitroy, M.S. Safronova, *Phys. Rev. A* 89 (2014) 012701.
- [3] G.F. Gribakin, J. Ludlow, *Phys. Rev. A* 70 (2004) 032720.
- [4] D.G. Green, J.A. Ludlow, G.F. Gribakin, *Phys. Rev. A* 90 (2014) 032712.
- [5] D.G. Green, A.R. Swann, G.F. Gribakin, *Phys. Rev. Lett.* 120 (2018) 183402.
- [6] J.F. Babb, R. Côté, H.R. Sadeghpour, P.C. Stancil, *J. Phys. B* 50 (2017) 140202.
- [7] M.S. Safronova, D. Budker, D. DeMille, D.F.J. Kimball, A. Derevianko, C.W. Clark, *Rev. Modern Phys.* 90 (2018) 025008.
- [8] C. Froese Fischer, T. Brage, P. Johansson, *Computational Atomic Structure: An MCHF Approach*, CRC Press, 1997.
- [9] A. Hibbert, *Comput. Phys. Comm.* 9 (1975) 141.
- [10] C. Froese Fischer, *Comput. Phys. Comm.* 43 (1987) 355.
- [11] C.F. Fischer, *Comput. Phys. Comm.* 64 (1991) 369.
- [12] L. Chernysheva, N. Cherepkov, V. Radojević, *Comput. Phys. Comm.* 11 (1976) 57.
- [13] C. Fischer, *Comput. Phys. Comm.* 182 (2011) 1315.
- [14] O. Zatsarinny, C.F. Fischer, *Comput. Phys. Comm.* 202 (2016) 287.
- [15] A. Zubiaga, M.M. Ervasti, I. Makkonen, A. Harju, F. Tuomisto, M.J. Puska, *J. Phys. B* 49 (2016) 064005.
- [16] S.L. Saito, *At. Data Nucl. Data Tables* 95 (2009) 836.
- [17] D.R. Hartree, *Math. Proc. Camb.* 24 (1928) 89.
- [18] V. Fock, *Z. Phys.* 61 (1930) 126.
- [19] J. Goldstone, *Proc. R. Soc. A* 239 (1957) 237.
- [20] W.H. Dickhoff, D.V. Neck, *Propagator Description of Quantum Mechanics in Many-Body Systems*, second ed., World Scientific, Singapore, 2008.
- [21] M.Y. Amusia, L.V. Chernysheva, *A Handbook for the ATOM Programs*, Institute of Physics Publishing Ltd, 1997.
- [22] C. de Boor, *A Practical Guide to Splines*, Springer, New York, 1977.
- [23] J. Sapirstein, W.R. Johnson, *J. Phys. B* 29 (1996) 5213.
- [24] H. Bachau, E. Cormier, P. Decleva, J.E. Hansen, F. Martin, *Rep. Progr. Phys.* 64 (2001) 1815.
- [25] A.R. Swann, G.F. Gribakin, *Phys. Rev. A* 97 (2018) 012706.
- [26] D.G. Green, G.F. Gribakin, *Prog. Theor. Chem. Phys.* 31 (2018) 243.
- [27] D. Waide, D. Green, G. Gribakin, 2020, To be submitted.
- [28] J. Stoer, R. Bulirsch (Eds.), *Introduction to Numerical Analysis*, Springer-Verlag, 2002.