Nuclear Science and Techniques (2025) 36:234
https://doi.org/10.1007/s41365-025-01810-4

Numerical solution of the Dirac equation with scalar, vector, and tensor potentials

Alexander Wallén Kiessling[1] · Daniel Karlsson[1] · Yuxin Zhao[2] · Mário Bettencourt Amaro[1] · Chong Qi[1]

Received: 18 November 2024 / Revised: 11 March 2025 / Accepted: 17 March 2025 / Published online: 9 October 2025

Abstract
Solving the Dirac equation has played an important role in many areas of fundamental physics. In this work, we present the Dirac equation solver DiracSVT, which solves the Dirac equation with scalar, vector, and tensor nuclear potentials in spherical coordinate space. The shooting method was used with a Runge–Kutta 4 integration scheme. The potentials are parameterized in a Woods–Saxon form, which reproduce well the known single-particle states around all doubly magic nuclei and can be applied to study the shell evolution of exotic nuclei. The code can be easily extended to the study of other systems, including atomic, hadron, and molecular physics.

Keywords Dirac equation · Nuclear physics · Scalar · Vector and tensor potentials · Shell evolution

1 Introduction

The Dirac equation is a fundamental relativistic wave equation in quantum mechanics that describes the behavior of spin-1/2 massive particles. It is a generalization of the Schrödinger equation to account for relativistic effects. Solving Dirac equations with various potentials has played an important role in many areas of fundamental physics, including atomic, nuclear, hadron, and molecular systems. The Dirac equation has been studied in the context of nuclear physics, especially within the framework of relativistic mean-field theories [1, 2] and in explaining various single-particle effects [3–5]. One of the most important aspects of contemporary nuclear physics is the study of the shell evolution of exotic nuclei, which refers to the potentially dramatic changes in the shell structure as one approaches driplines with excessive protons or neutrons [6]. The evolution of the shell structure, which is crucial for our understanding of nuclear stability as well as the origin of heavy elements, can be induced by the isospin dependence of the spin and tensor potentials in various mean-field model approaches. The Dirac equation is unique for such studies because it provides not only insight into the origin of spin–orbital potentials as the competition between scalar and vector potentials [1] but also the possibility of adding a tensor potential at the mean-field level. This study focuses on numerically solving the Dirac equation with scalar, vector, and tensor potentials. We hope that it can be a useful tool for studying the structure of exotic nuclei, as well as other quantum systems where spin–orbit and tensor effects can be important.

Dirac equation solvers are not available in the nuclear physics community, except for the Fortran subroutine embedded in relativistic mean-field codes [7, 8]. Previous works in other areas on solving the Dirac equation focused particularly on the vector potential and utilized various schemes. In Ref. [9] the finite element method is applied to solve the Dirac equation, which is validated to converge to within floating point solutions. The problems inherent in the numerical solutions of the Dirac equation owing to spurious solutions are discussed in Ref. [10], where a stabilized version of the finite element method is also applied. The mapped Fourier method was used to numerically address the Dirac equation in. [11]. In Ref. [12] evolutionary algorithms are used and demonstrated in a case study of a muon orbiting the ^{208}Pb nucleus. The work of Ref. [13] also focuses on studying muonic atoms and applies the shooting method to integrate the Dirac equation. Moreover, a power series expansion method was employed in Ref. [14]. The numerical aspects of solving the Dirac equation are discussed in Ref. [15]. Other important methods include the

✉ Chong Qi
chongq@kth.se

[1] Department of Physics, KTH Royal Institute of Technology, Albanova University Center, 10691 Stockholm, Sweden

[2] Department of Physics, University of Florida, Gainesville, FL 32611, USA

Woods–Saxon basis [16], complex momentum representation [17], analytic continuation in the coupling constant (ACCC) [18], and Green's function methods [19, 20].

Our work solves the Dirac equation in spherical coordinate space by utilizing the shooting method with a Runge–Kutta 4 integration scheme. We have prepared the code, named DiracSVT, in three different popular programming languages: Python, MATLAB, and C++. These are gradually becoming the staple in physics, replacing Fortran as a more modern, versatile, and modular alternative. We include scalar, vector, and tensor potentials of an isospin-dependent Wood–Saxon form in the Dirac equation [21]. The code is used to solve the available single-particle states around doubly magic nuclei, including ^{16}O, ^{40}Ca, ^{48}Ca, ^{56}Ni, ^{100}Sn, ^{132}Sn, and ^{208}Pb. The same potential can also reproduce the shell structures of many open nuclei well over the entire nuclear chart.

The remainder of this paper is organized as follows. Section 2 describes the formulation of the Dirac equation with corresponding potentials and boundary conditions. Section 3 describes our method and the algorithm for solving the equation. Section 4 presents our results, and Sect. 5 discusses these results and presents an outlook on the possibilities for future extensions to this work.

2 Theory

We limit ourselves to spherical symmetry. The Dirac Hamiltonian H with a scalar potential S, a vector potential V, and a tensor potential U can be given by (see, e.g., [22, 23])

$$H = \vec{\alpha} \cdot \vec{p} + \beta(m + S) + V - i\beta\vec{\alpha} \cdot \hat{r}U. \tag{1}$$

The corresponding Dirac equation for the radial wave function can be expressed as

$$\left(\frac{\mathrm{d}}{\mathrm{d}r} + \frac{\kappa}{r} - U(r)\right)g_\kappa(r) = (E + m - \Delta(r))f_\kappa(r), \tag{2}$$

$$\left(\frac{\mathrm{d}}{\mathrm{d}r} - \frac{\kappa}{r} + U(r)\right)f_\kappa(r) = -(E - m - \Sigma(r))g_\kappa(r), \tag{3}$$

where f and g are the two components of the radial wave function and $\kappa = -(l + 1)$ for $j = l + \frac{1}{2}$ and $\kappa = l$ for $j = l - \frac{1}{2}$. For convenience, we define our potentials as $\Sigma = V + S$ and $\Delta = V - S$ as the sum of the vector and scalar potentials. The above equations can also be rewritten as

$$g'_\kappa = (2m + B - \Delta)f_\kappa + \left(U - \frac{\kappa}{r}\right)g_\kappa, \tag{4}$$

$$f'_\kappa = (-B + \Sigma)g_\kappa + \left(\frac{\kappa}{r} - U\right)f_\kappa, \tag{5}$$

where the binding energy B was obtained from the total energy $E = B + m$.

The potentials can be parameterized in a standard Woods–Saxon shape (see, e.g., [21, 24])

$$\Sigma = \frac{\Sigma_{0,\mathrm{n(p)}}}{1 + e^{\frac{r - R_\sigma}{a_\sigma}}} + V_{\mathrm{Coulomb}}, \tag{6}$$

$$\Delta = \frac{\Delta_{0,\mathrm{n(p)}}}{1 + e^{\frac{r - R_\delta}{a_\delta}}} + V_{\mathrm{Coulomb}}, \tag{7}$$

where $R_{\sigma,\delta} = r_{\sigma,\delta}A^{1/3}$ and a are radius and diffuseness parameters, respectively. Naturally, the Coulomb potential term exists only for protons.

Quantities $\Sigma_{0,\mathrm{n(p)}}$, and $\Delta_{0,\mathrm{n(p)}}$ are defined for the three possible scenarios. In Scenario 1, Δ_0 is assumed to be directly proportional to Σ_0 with a scaling factor. Scenario 2 does not explicitly consider the dependence of the potential on the neutron excess. Scenario 3, on the other hand, describes dependence via a spin–orbit-dependent term δ_{so}. For further details, please refer to Ref. [21]). For each of the three different scenarios, where n and p stand for neutron and proton states, respectively,

$$\Sigma_{0,\mathrm{p}} = V_0\left(1 + \delta\frac{N - Z}{A}\right), \tag{8}$$

$$\Sigma_{0,\mathrm{n}} = V_0\left(1 - \delta\frac{N - Z}{A}\right), \tag{9}$$

and $\Delta_{0,\mathrm{n(p)}}$ is defined for Scenario 1 as

$$\Delta_0 = -\lambda\Sigma_0, \tag{10}$$

for Scenario 2 as

$$\Delta_0 = -\lambda V_0, \tag{11}$$

which assumes no isospin dependence, and for Scenario 3 as

$$\Delta_{0,\mathrm{p}} = -\lambda V_0\left(1 - \delta_{\mathrm{so}}\frac{N - Z}{A}\right), \tag{12}$$

$$\Delta_{0,\mathrm{n}} = -\lambda V_0\left(1 + \delta_{\mathrm{so}}\frac{N - Z}{A}\right). \tag{13}$$

The parameters $V_0, \delta, \delta_{\mathrm{so}}, \lambda, a_\sigma = a_\delta, r_0$, and $r_{0,ls}$ are fitted to data for the three scenarios separately. In the simplest case, the diffuseness parameters a_σ and a_δ are assumed identical.

The Coulomb potential is defined as

$$V_{\mathrm{Coulomb}} = \begin{cases} c\frac{Z}{r} & r > r_\sigma \\ c\frac{Z(3r_\sigma^2 - r^2)}{2r_\sigma^3} & r \leq r_\sigma, \end{cases} \tag{14}$$

where

$$c = \frac{e^2}{4\pi\epsilon_0\hbar c} \equiv \alpha \approx 0.007297. \tag{15}$$

In addition, we have a tensor potential defined similarly in a Woods–Saxon shape:

$$U(r) = \frac{U_0}{1 + e^{\frac{\Delta^{\sigma}(r)}{a_\sigma}}} \tag{16}$$

The boundary condition for the bound states is defined separately for limiting cases of small ($r \to 0$) and large ($r \to \infty$) radii. For small r, we have

$$f = \begin{cases} -a_0\left(\frac{-B+\Sigma}{k}\right)\epsilon^{l+2} & k < 0 \\ a_0\epsilon^{l} & k > 0 \end{cases}, \tag{17}$$

$$g = \begin{cases} a_0\epsilon^{l+1} & k < 0 \\ a_0\left(\frac{2m+B-\Delta}{k}\right)\epsilon^{l+1} & k > 0 \end{cases}, \tag{18}$$

where ϵ denotes the radial variable in the small radius regime. For the boundary conditions in the case where $r \to \infty$, it is assumed that both wave functions approach zero in this limit.

To handle unbound neutron resonance states with positive energy, we implement for large r similar boundary conditions to Eq. (26), as in [25]:

$$\psi = \begin{pmatrix} [Cj_l(r) + Dy_l(r)]Y_{j,l,j_z} \\ \frac{\sqrt{B^2+2Bm}}{B+2\,m}[Cj_{l'}(r) + Dy_{l'}(r)]Y_{j,l',j_z} \end{pmatrix},$$

where j and y indicate the spherical Bessel functions $l' = j \pm \frac{1}{2}$ and $j_z = -j, -j+1, \ldots j$. Here, $Y_{j,l,jz}$ are the spinor spherical harmonics, as detailed in [26]. In the 1D case, one of the two components of the spinor spherical harmonics vanishes, and the solution is reduced to two components f and g. As such the boundary conditions for $r \to \infty$ become

$$f = Cj_l(r) + Dy_l(r), \tag{19}$$

$$g = \frac{\sqrt{B^2 + 2Bm}}{B + 2m}[Cj_{l'}(r) + Dy_{l'}(r)]. \tag{20}$$

The low-lying proton resonance states can be well approximated as bound states, as most of these states are pretty long-lived with negligibly small imaginary parts. The boundary condition should be refined as a Coulomb Hankel function for higher-lying proton resonance states.

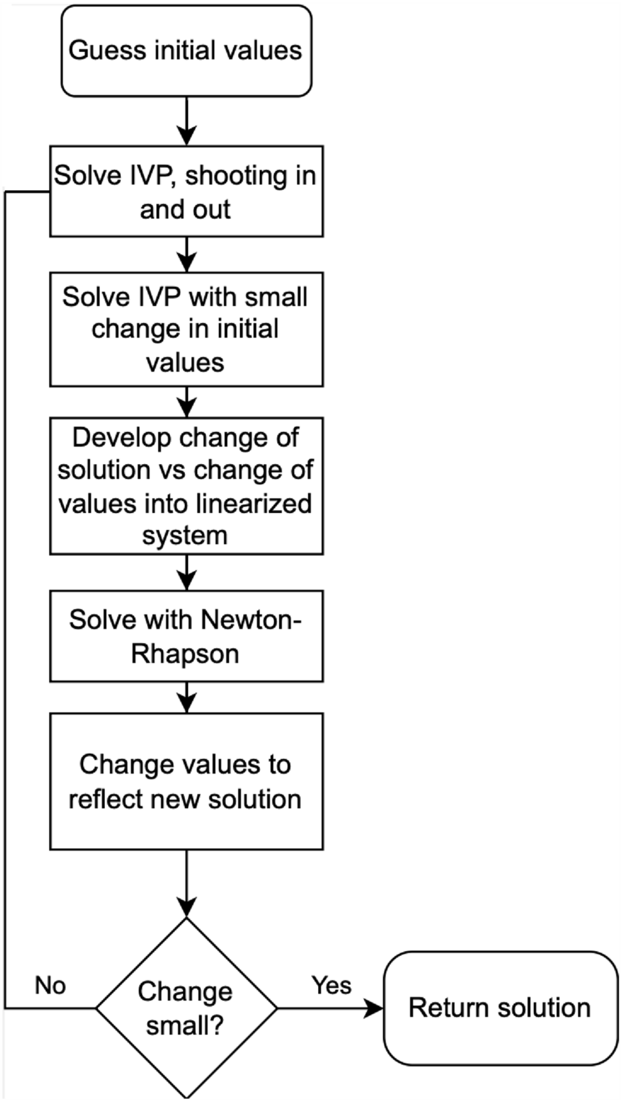
Guess initial values
Solve IVP, shooting in and out
Solve IVP with small change in initial values
Develop change of solution vs change of values into linearized system
Solve with Newton-Rhapson
Change values to reflect new solution
No
Change small?
Yes
Return solution

Fig. 1 Flowchart of the DiracSVT solver scheme. IVP stands for initial value problem

3 Method

We attempt to combine an inner solution and an outer solution in what is known as the shooting method. The shooting method is a numerical scheme suitable for boundary-value problems. More specifically, the shooting method is a technique for solving a boundary-value problem by transforming it into an initial value problem (IVP). The method guesses the values of the boundary conditions at one end of the interval and then integrates the differential equation from that end using an ordinary differential equation solver. The estimated boundary values were adjusted until the solution satisfied the boundary conditions at the other end of the interval. No verification was performed for the number of nodes.

Table 1 Parameters used in the code for Scenario 1 (yielding the values in Tables 2 and 3)

Scenario	V_0 (MeV)	δ, δ_δ	λ	r_σ (MeV^{-1})	a, a_δ (MeV^{-1})	r_δ (MeV^{-1})
1	−61.44	0.73	13.36	0.006515	0.003638	0.004867

In our case, we employed shooting in two directions: outward and inward. We used the Runge–Kutta 4 integration scheme as the ordinary differential equation solver. This is akin to the methods employed in earlier studies, such as [27].

The outward shooting solution originated from the origin. However, setting $r = 0$ causes a singularity in the model. To circumvent this, we set the inner point at a small distance $r = \epsilon$ from the origin. As for initial values, the boundary conditions from Eqs. (17–20) are used. It should be noted that an initial guess for the binding energy is required. For simplicity, we used the experimental values with some variations for the example cases studied below.

Inward shooting originates from an asymptotic distance from the origin $r = \infty$. As an approximation, we consider a point that we set to a relatively large distance from the origin. We used the boundary conditions for a large r in this shooting case.

Both the outward and inward solutions can be used independently, as is. However, there is a major tendency for the calculated solutions of $f(r)$ and $g(r)$ to blow up and become numerically unstable. Thus, the matching concept of the shooting method is presented. We only solve from each direction up to a certain "middle" point which we denote as r_m (note, this point does not have to be halfway in distance) where the two solutions are then matched.

To define the difference $\Delta(B_0, a_0)$ between the two solutions at the matching point, there are a few options [15]. These include simple direct differences

$$\Delta f(B_0, a_0) = f_\mathrm{out}(r_\mathrm{m}) - f_\mathrm{in}(r_\mathrm{m}), \tag{21}$$

$$\Delta g(B_0, a_0) = g_\mathrm{out}(r_\mathrm{m}) - g_\mathrm{in}(r_\mathrm{m}), \tag{22}$$

as well as relative average value-based differences

$$\Delta f(B_0, a_0) = \frac{2(f_\mathrm{out}(r_\mathrm{m}) - f_\mathrm{in}(r_\mathrm{m}))}{f_\mathrm{out}(r_\mathrm{m}) + f_\mathrm{in}(r_\mathrm{m})}, \tag{23}$$

$$\Delta g(B_0, a_0) = \frac{2(g_\mathrm{out}(r_\mathrm{m}) - g_\mathrm{in}(r_\mathrm{m}))}{g_\mathrm{out}(r_\mathrm{m}) + g_\mathrm{in}(r_\mathrm{m})}. \tag{24}$$

Here, B_0 and a_0 are the shooting parameters that must be adjusted such that the inside and outside Dirac spinor solutions match at the chosen matching radius r_m, that is, the direct differences in the equations above yield zero. After the trial, simple direct differences were found to converge most consistently and were used in the solver.

To drive the differences to zero, we must define the change in the difference in terms of the parameters. We do this by linearizing through a first order Taylor expansion

$$\Delta f + \frac{\partial \Delta f}{\partial B}(B_\mathrm{new} - B_\mathrm{old}) + \frac{\partial \Delta f}{\partial a_0}(a_{0,\mathrm{new}} - a_{0,\mathrm{old}}) = 0, \tag{25}$$

$$\Delta g + \frac{\partial \Delta g}{\partial B}(B_\mathrm{new} - B_\mathrm{old}) + \frac{\partial \Delta g}{\partial a_0}(a_{0,\mathrm{new}} - a_{0,\mathrm{old}}) = 0. \tag{26}$$

This defines a linear system, which we solve iteratively using Newton's root-finding method. The resultant solution is considered to be the correct solution after satisfactory convergence, that is, the change in the solution becomes insignificant with further iterations. The overall solution method is shown in Fig. 1.

The implementation of the numerical solver is written in three different languages: Python, MATLAB, and C++. It was developed and tested mainly for Linux, and the experiments were run on both Intel i7-2600 @3.40 GHz and AMD Ryzen 5 3600 @3.60 GHz. It is worth noting that the sensitivity of the solution to the precision of the system solution is very high in some cases. Changing the float precision can induce convergence to very different solutions. Similarly, merely varying the underlying linear algebra routine version (e.g., BLAS/LAPACK) also resulted in similar problems. As such, while the solver has been tested on different architectures, discrepancies remain regarding convergence that the reader should be aware of, resulting from different back-end libraries being utilized.

4 Numerical results

For example, we calculated the single-particle energies of all known single-particle states and single-hole states around the doubly magic nuclei for all three scenarios and all three language frameworks in which the solver is written. The binding energies against which the parameters were fitted were extracted from the experimental nuclear data. The parameters were fitted to minimize the RMS error between the numerical solutions and the experimental single-particle energies, as specified in Tables 2 and 3, so that they replicate the energies of the spin–orbit partners. The tensor potential is not included in the fitting of the parameters given below because of the scarcity of experimental data that can fully constrain its strength. Its effect is expected to be more

Table 2 Experimental and calculated single-particle energies for neutrons around doubly magic nuclei. These energies were retrieved after a single run for the same input and scenario, using the three codes. Outliers and discrepancies are kept for illustrative purposes, and the details are discussed in the text

Nucleus	State	Experimental B	MATLAB B	Python B	C++ B
^{16}O	$1p_{1/2}$	−15.66	−15.8374	−15.8374	−15.8374
^{16}O	$1d_{5/2}$	−4.14	−6.3839	−6.3839	−6.38387
^{16}O	$2s_{1/2}$	−3.27	−3.8203	−3.8203	−3.82027
^{40}Ca	$1d_{5/2}$	−22.39	−21.6375	−21.6375	−21.6375
^{40}Ca	$2s_{1/2}$	−18.19	−15.0881	−15.0881	−15.0881
^{40}Ca	$1d_{3/2}$	−15.64	−15.8337	−15.8337	−15.8337
^{40}Ca	$1f_{7/2}$	−8.36	−9.4823	−9.4823	−9.48232
^{40}Ca	$2p_{3/2}$	−5.84	−4.4926	−4.4926	−4.49257
^{40}Ca	$2p_{1/2}$	−4.2	−2.7787	−2.7787	−2.77869
^{40}Ca	$1f_{5/2}$	−1.56	−3.9294	−3.9294	−3.92936
^{48}Ca	$1d_{5/2}$	−15.61	−19.6859	−19.6859	−19.6859
^{48}Ca	$2s_{1/2}$	−12.55	−14.1267	−14.1267	−14.1267
^{48}Ca	$1d_{3/2}$	−12.53	−15.1527	−15.1527	−15.1527
^{48}Ca	$1f_{7/2}$	−10	−8.7770	−8.7770	−8.77702
^{48}Ca	$2p_{3/2}$	−4.6	−4.2724	−4.2724	−4.27242
^{48}Ca	$2p_{1/2}$	−2.86	−2.9638	−2.9638	−2.96378
^{48}Ca	$1f_{5/2}$	−1.2	−4.2483	−4.2483	−4.2483
^{56}Ni	$1f_{7/2}$	−16.64	−16.0488	−16.0488	−16.0488
^{56}Ni	$2p_{3/2}$	−10.25	−9.1691	−9.1691	−9.16906
^{56}Ni	$1f_{5/2}$	−9.48	−10.0171	−10.0171	−10.0171
^{56}Ni	$2p_{1/2}$	−9.13	−7.2951	−7.2951	−7.29513
^{100}Sn	$2p_{1/2}$	−18.38	−16.9797	−16.9797	−16.9797
^{100}Sn	$1g_{9/2}$	−17.93	−17.4011	−17.4011	−17.4011
^{100}Sn	$2d_{5/2}$	−11.13	−9.2034	−9.2034	−9.20345
^{100}Sn	$1g_{7/2}$	−10.93	−11.1915	−11.1915	−11.1915
^{100}Sn	$3s_{1/2}$	−9.3	−6.3580	−6.3580	−6.3580
^{100}Sn	$1h_{11/2}$	−8.6	−7.6706	−7.6706	−7.6706
^{100}Sn	$2d_{3/2}$	−9.2	−6.7376	−6.7376	−6.7376
^{132}Sn	$1g_{7/2}$	−9.75	−10.9534	−10.9534	−10.9534
^{132}Sn	$2d_{5/2}$	−8.97	−8.6191	−8.6191	−8.6191
^{132}Sn	$3s_{1/2}$	−7.64	−6.4832	−6.4832	−6.4832
^{132}Sn	$1h_{11/2}$	−7.54	−7.1366	−7.1366	−7.136
^{132}Sn	$2d_{3/2}$	−7.31	−7.0763	−7.0763	−7.0763
^{132}Sn	$2f_{7/2}$	−2.47	–	−1.3848	–
^{132}Sn	$3p_{3/2}$	−1.57	−0.2211	−16.8335	−16.8335
^{132}Sn	$1h_{9/2}$	−0.86	−2.8903	−2.8903	−2.8903
^{132}Sn	$2f_{5/2}$	−0.42	0.2823	−19.1602	−19.1602
^{208}Pb	$1h_{9/2}$	−11.4	−12.4667	−12.4667	−12.4667
^{208}Pb	$2f_{7/2}$	−9.81	−9.3144	−9.3144	−9.3144
^{208}Pb	$1i_{13/2}$	−9.24	−9.7762	−9.7762	−9.7762
^{208}Pb	$3p_{3/2}$	−8.26	−6.3349	−6.3349	−6.3349
^{208}Pb	$2f_{5/2}$	−7.94	−7.3793	−7.3793	−7.3793
^{208}Pb	$3p_{1/2}$	−7.37	−8.0489	−23.3705	−6.0271
^{208}Pb	$2g_{9/2}$	−3.94	−2.5108	−2.5108	−2.5108
^{208}Pb	$1i_{11/2}$	−3.16	−5.2959	−5.2959	−5.2959
^{208}Pb	$1j_{15/2}$	−2.51	−2.2689	−2.2689	–
^{208}Pb	$3d_{5/2}$	−2.37	−0.2044	−16.6099	−16.6099
^{208}Pb	$4s_{1/2}$	−1.9	−0.1299	−14.2321	−14.2321
^{208}Pb	$2g_{7/2}$	−1.44	−0.2843	−19.6005	−19.6005

significant as it approaches the neutron-rich nuclei around the neutron dripline.

In Table 1, the fitted parameters are given for Scenario 1, which were used to obtain the values in Tables 2 and 3. The difference between the scenarios relies on the value of δ_{so}: Scenario 2 is identical to Scenario 1 but takes $\delta_{so} = 0$ and Scenario 3 takes $\delta_{so} = -\delta$. The position was scaled by $1/\hbar c$, acquiring units of MeV^{-1}, yielding a Coulomb barrier in MeV as per Eq. 15.

Some of the results are listed in Tables 2 and 3 for comparison purposes. The solutions from the three different code frameworks were identical within the numerical errors and were in reasonable agreement with the experimentally determined energies from Ref. [21]. The results for the three codes with the same inputs are listed. In some rare cases, one of the solvers can display a solution, while another diverges (marked as in the tables). This seems likely to be the result of different languages using various routines to invert matrices. Moreover, the solver can sometimes converge to the energy levels of different states (e.g., the proton $2d_{3/2}$ and $2f_{7/2}$ states in ^{208}Pb). This artifact can be easily spotted and is sensitive to the initial conditions and area of integration. We did not correct for comparison purposes. The convergence can be improved by slightly modifying the initial value, matching radius, and maximal radius. The values in Tables 2 and 3 are given for initial guesses equal to the experimental value for standardization. However, it may not always be the most suitable choice. In the case of unsuccessful convergence, this can be tweaked until the converged value improves. For some outliers, the code converges to a state with different nodes, which is easy to identify because the calculated energy differs significantly from the desired value.

The solver also captured the waveform solution of the Dirac equation for the considered particle and state, as shown in Fig. 2. In the present work, we focus on bound states and very low-lying unbound states, as all quantities are assumed to be real. The bound and unbound states appear quite similar, except for the boundary conditions. This also implies that the code is not yet suitable for handling wide resonances.

Table 3 Experimental and calculated single-particle energies for protons

Nucleus	State	Experimental B	MATLAB B	Python B	C++ B
^{16}O	1p$_{1/2}$	−12.13	−11.9334	−11.9334	−11.9334
^{16}O	1d$_{5/2}$	−0.6	−2.9170	−2.9170	−2.9170
^{16}O	2s$_{1/2}$	−0.11	−0.8547	−0.8547	−0.8547
^{16}O	1d$_{3/2}$	4.688	4.7575	4.7575	4.7575
^{40}Ca	1d$_{5/2}$	−15.07	−14.1180	−14.1180	−14.1180
^{40}Ca	2s$_{1/2}$	−10.92	−7.7847	−7.7847	−7.7847
^{40}Ca	1d$_{3/2}$	−8.33	−8.5334	−8.5334	−8.5334
^{40}Ca	1f$_{7/2}$	−1.09	−2.5333	−2.5333	−2.5333
^{40}Ca	2p$_{3/2}$	0.69	1.5992	1.5992	1.5992
^{48}Ca	1d$_{5/2}$	−21.47	−22.4904	−22.4904	−22.4904
^{48}Ca	1d$_{3/2}$	−16.18	−15.4556	−15.4556	−15.4556
^{48}Ca	2s$_{1/2}$	−16.1	−14.2643	−14.2643	−14.2643
^{48}Ca	1f$_{7/2}$	−9.35	−10.5523	−10.5523	−10.5523
^{48}Ca	2p$_{3/2}$	−6.44	−3.2336	−3.2336	−3.2336
^{48}Ca	2p$_{1/2}$	−4.64	11.7435	−0.9598	11.7435
^{56}Ni	1f$_{7/2}$	−7.17	−6.9046	−6.9046	−6.9046
^{56}Ni	2p$_{3/2}$	−0.69	−0.5831	−0.5831	−0.5831
^{56}Ni	1f$_{5/2}$	0.33	8.5469	8.5469	8.5469
^{56}Ni	2p$_{1/2}$	0.41	1.0938	1.0938	1.0938
^{100}Sn	1f$_{5/2}$	−8.71	−7.0267	−7.0267	−7.0267
^{100}Sn	2p$_{3/2}$	−6.38	−4.5904	−4.5904	−4.5904
^{100}Sn	2p$_{1/2}$	−3.53	−3.0105	−3.0105	−3.0105
^{100}Sn	1g$_{9/2}$	−2.92	−3.7950	−3.7950	−3.7950
^{100}Sn	1g$_{7/2}$	3.9	1.9453	12.8860	12.8860
^{132}Sn	2p$_{3/2}$	−16.01	−14.1064	−14.1064	−14.1064
^{132}Sn	1g$_{9/2}$	−15.71	−18.0759	−18.0759	−18.0759
^{132}Sn	1g$_{7/2}$	−9.68	−10.0481	−10.0481	−10.0481
^{132}Sn	2d$_{5/2}$	−8.72	−6.9269	−6.9269	−6.9269
^{132}Sn	2d$_{3/2}$	−6.97	−3.5858	–	−28.5546
^{132}Sn	1h$_{11/2}$	−6.89	−8.9825	−8.9825	−8.9825
^{208}Pb	1g$_{7/2}$	−12	−13.3095	−13.3095	−13.3095
^{208}Pb	2d$_{5/2}$	−9.82	−9.3826	−9.3826	−9.3826
^{208}Pb	1h$_{11/2}$	−9.36	−13.2378	−13.2378	−13.2378
^{208}Pb	2d$_{3/2}$	−8.36	−6.7044	−6.7044	−28.4605
^{208}Pb	3s$_{1/2}$	−8.01	−5.3560	−25.9307	−25.9307
^{208}Pb	1h$_{9/2}$	−3.8	−5.7976	−5.7976	−5.7976
^{208}Pb	2f$_{7/2}$	−2.9	−1.6873	−1.6873	−27.3765
^{208}Pb	1i$_{13/2}$	−2.1	−5.5308	−5.5308	–
^{208}Pb	2f$_{5/2}$	−0.97	2.0965	−20.8512	−20.8512
^{208}Pb	3p$_{3/2}$	−0.68	3.5255	−17.4631	−17.4631

5 Summary

We presented the Dirac equation solver DiracSVT, which solves the Dirac equation in coordinate space with scalar, vector, and tensor nuclear potentials. The code is written in three different languages and uses the shooting method with the Runge–Kutta 4 integration scheme. The above potentials were parameterized in the Saxon form. We show that the potential can reproduce the known single-particle states very well around all the doubly magic nuclei. This code is freely available in Ref. [28].

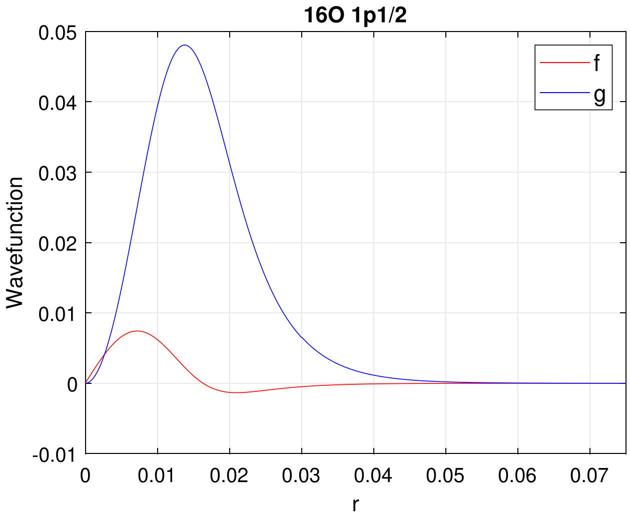

Fig. 2 (Color online) Radial wave function for ^{16}O, with neutron hole in state 1p$_{1/2}$. The results from the three codes were indistinguishable. The position was scaled by $1/\hbar c$, acquiring units of MeV^{-1}

Author Contributions All authors contributed to the study conception and design. Material preparation, data collection, and analysis were performed by Chong Qi. First MATLAB version of the code presented in this paper was written by Yuxin Zhao. The Python and C++ versions were done by Alexander Kiessling and Daniel Karlsson. The first draft of the manuscript was written by Chong Qi, Alexander Kiessling, and Daniel Karlsson, and all authors commented on previous versions of the manuscript. Mário Amaro tested the code and contributed to the revision of the manuscript. All authors read and approved the final manuscript.

Funding Open access funding provided by Royal Institute of Technology.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. P. Ring, Relativistic mean field theory in finite nuclei. Prog. Part. Nucl. Phys. **37**, 193–263 (1996). https://doi.org/10.1016/0146-6410(96)00054-3
2. J. Meng (ed.), *Relativistic Density Functional for Nuclear Structure, International Review of Nuclear Physics* (World Scientific, Singapore, 2016)
3. J.N. Ginocchio, Pseudospin as a relativistic symmetry. Phys. Rev. Lett. **78**, 436–439 (1997). https://doi.org/10.1103/PhysRevLett.78.436
4. Z.X. Ren, S.Q. Zhang, J. Meng, Solving Dirac equations on a 3d lattice with inverse Hamiltonian and spectral methods. Phys. Rev. C **95**, 024313 (2017). https://doi.org/10.1103/PhysRevC.95.024313
5. T. Oishi, Time-dependent Dirac equation applied to one-proton radioactive emission. (2022). https://doi.org/10.48550/ARXIV.2212.03271
6. T. Otsuka, A. Gade, O. Sorlin et al., Evolution of shell structure in exotic nuclei. Rev. Mod. Phys. **92**, 015002 (2020). https://doi.org/10.1103/RevModPhys.92.015002
7. P. Ring, Y. Gambhir, G. Lalazissis, Computer program for the relativistic mean field description of the ground state properties of even-even axially deformed nuclei. Comput. Phys. Commun. **105**, 77–97 (1997). https://doi.org/10.1016/S0010-4655(97)00022-2
8. T. Nikšić, N. Paar, D. Vretenar et al., Dirhb-a relativistic self-consistent mean-field framework for atomic nuclei. Comput. Phys. Commun. **185**, 1808–1821 (2014). https://doi.org/10.1016/j.cpc.2014.02.027
9. M. Piibeleht, Numerical investigations of the Dirac equation and bound state quantum electrodynamics in atoms: a thesis presented in partial fulfilment of the requirements for the degree of doctor of philosophy in physics at Massey University, Albany, New Zealand. Ph.D. thesis, Massey University (2022)
10. H. Almanasreh, S. Salomonson, N. Svanstedt, Stabilized finite element method for the radial Dirac equation. J. Comput. Phys. **236**, 426–442 (2013). https://doi.org/10.1016/j.jcp.2012.11.020
11. E. Ackad, M. Horbatsch, Numerical solution of the Dirac equation by a mapped Fourier grid method. J. Phys. A Math. Gen. **38**, 3157 (2005). https://doi.org/10.1088/0305-4470/38/14/007
12. I.G. Tsoulos, O.T. Kosmas, V. Stavrou, DiracSolver: a tool for solving the Dirac equation. Comput. Phys. Commun. **236**, 237–243 (2019). https://doi.org/10.1016/j.cpc.2018.10.010
13. S. Sturniolo, A. Hillier, Mudirac: a Dirac equation solver for elemental analysis with muonic X-rays. X-Ray Spectrom. **50**, 180–196 (2021). https://doi.org/10.1002/xrs.3212
14. F. Salvat, J.M. Fernández-Varea, radial: a Fortran subroutine package for the solution of the radial Schrödinger and Dirac wave equations. Comput. Phys. Commun. **240**, 165–177 (2019). https://doi.org/10.1016/j.cpc.2019.02.011
15. R.R. Silbar, T. Goldman, Solving the radial Dirac equations: a numerical odyssey. Eur. J. Phys. **32**, 217 (2010). https://doi.org/10.1088/0143-0807/32/1/021
16. S.G. Zhou, J. Meng, P. Ring, Spherical relativistic Hartree theory in a woods-saxon basis. Phys. Rev. C **68**, 034323 (2003). https://doi.org/10.1103/physrevc.68.034323
17. S.Z. Xu, S.S. Zhang, X.Q. Jiang et al., The complex momentum representation approach and its application to low-lying resonances in ^{17}O and ^{29,31}F. Nucl. Sci. Tech. **34**, 5 (2023). https://doi.org/10.1007/s41365-022-01159-y
18. S.S. Zhang, J. Meng, S.G. Zhou et al., Analytic continuation of single-particle resonance energy and wave function in relativistic mean field theory. Phys. Rev. C **70**, 034308 (2004). https://doi.org/10.1103/physrevc.70.034308
19. T.T. Sun, S.Q. Zhang, Y. Zhang et al., Green's function method for single-particle resonant states in relativistic mean field theory. Phys. Rev. C **90**, 054321 (2014). https://doi.org/10.1103/physrevc.90.054321
20. T.T. Sun, L. Qian, C. Chen et al., Green's function method for the single-particle resonances in a deformed Dirac equation. Phys. Rev. C **101**, 014321(2020). https://doi.org/10.1103/physrevc.101.014321
21. Z.X. Xu, C. Qi, Shell evolution and its indication on the isospin dependence of the spin-orbit splitting. Phys. Lett. B **724**, 247–252 (2013). https://doi.org/10.1016/j.physletb.2013.06.018
22. H. Akcay, Dirac equation with scalar and vector quadratic potentials and Coulomb-like tensor potential. Phys. Lett. A **373**, 616–620 (2009). https://doi.org/10.1016/j.physleta.2008.12.029
23. H. Hassanabadi, E. Maghsoodi, S. Zarrinkamar et al., Dirac equation under scalar, vector, and tensor Cornell interactions. Adv. High Energy Phys. **2012**, 707041 (2012). https://doi.org/10.1155/2012/707041
24. P. Kennedy, The Woods-Saxon potential in the Dirac equation. J. Phys. A Math. Gen. **35**, 689 (2002). https://doi.org/10.1088/0305-4470/35/3/314
25. V. Alonso, S. De Vincenzo, L. Mondino, On the boundary conditions for the Dirac equation. Eur. J. Phys. **18**, 315 (1997). https://doi.org/10.1088/0143-0807/18/5/001
26. D.A. Varshalovich, A.N. Moskalev, V.K. Khersonskii, *Quantum Theory of Angular Momentum* (World Scientific, Singapore, 1988)
27. J. Meng, Relativistic continuum Hartree-Bogoliubov theory with both zero range and finite range Gogny force and their application. Nucl. Phys. A **635**, 3–42 (1998). https://doi.org/10.1016/s0375-9474(98)00178-x
28. https://codeocean.com/capsule/7384015/tree/v1