

Numerical solution of the Schrödinger equation using Neural Networks in Python

A. Gkrepis^{1,2}, O. Kosmas⁴, D. Vlachos³ and T. Kosmas¹

¹ Department of Physics, University of Ioannina, Ioannina, Greece

² Department of Informatics, University of Western Macedonia, Kastoria, Greece

³ Department of Economics, University of Peloponnese, Tripoli, Greece

⁴ Conigital LTD, Birmingham-Coventry, Manchester, UK

E-mail: a.gkrepis@uowm.gr

Abstract. The motion of quantum mechanical systems in physical sciences is described by partial differential equations, usually of second order with respect to spatial coordinates. The required solutions of the time-dependent type of equations are, in general, functions of the temporal variable t and the state vectors determining the positions of the system's particles at the time t . Only a small number, however of those differential equations can however be solved analytically, while the majority of them must be solved numerically by applying specific very advanced integration techniques. Among these equations, the fundamental Schrödinger equation offers great insight towards numerical solving. In this work we present an effective method that solves numerically the time-independent Schrodinger equation on the basis of neural networks techniques. Analytical and numerical results for the radial part of this equation with the corresponding energies are then compared so as to estimate the performance of our method.

1. Introduction

It is well known that, many phenomena in classical and quantum physics can be described through the use of partial differential equations (PDEs). Towards this aim, about a century ago, Schrödinger derived a second order differential equation to account for the wave nature of a particle assuming that the particle behaves as wave. This equation is known as a wave-equation and serves as a mathematical model to describe the motion of a particle by means of a wave [1, 2].

The Schrödinger's equation yields solutions in terms of functions, called wave functions, while at the same time the corresponding energy E of the particle in question is obtained. In general, the wave-functions can describe the motion of microscopic systems in molecular, atomic and subatomic (nuclear and hadronic) physics but also of macroscopic systems, for example, the entire Universe. Hence, Schrödinger's equation is considered fundamental to all applications of quantum mechanics including the quantum field theory.

In mathematics, Schrödinger's equation in its two variants: the time-dependent and the time-independent differential form, is one of the basic equations that are studied in the sub-field of partial differential equations (PDEs), and has many applications in geometry, physics, scattering theory, integrable systems, etc.

In this work, after summarizing briefly (in Sect. 2) some basic details of the Schrödinger equation, we focus on methods providing analytical solutions of this equation. Then (in Sect. 3), we present a numerical method based on artificial neural networks for solving the time-independent Schrödinger equation. Next (in Sect. 4), we apply our method in order to obtain numerical solutions for the wave functions describing the bound states of some exotic leptonic atoms and compare them with the corresponding analytic wave functions. Finally (in Sect. 5), we summarize the main conclusions extracted from the present work and discuss future steps of our study.

2. Analytic solutions of the Schrödinger equation

For obtaining possible analytical solutions of the Schrödinger equation, the well known method of the separation of variables is utilized. For a system of two particles interacting via a potential $V(\vec{r})$, the time-dependent Schrödinger equation is formulated as [7, 8],

$$\left(-\frac{\hbar^2}{2\mu}\nabla^2 + V(\vec{r})\right)\Psi(\vec{r}, t) = i\hbar\frac{\partial\Psi(\vec{r}, t)}{\partial t} \quad (1)$$

where $\hbar = h/2\pi$ represents the well known reduced Plank constant. Assuming that the masses of the two particles are m_1 and m_2 (with $m_1 \geq m_2$), then the reduced mass of the system, μ , entering Eq. (1) is defined as

$$\mu = \frac{m_1 m_2}{m_1 + m_2} = \frac{m_2}{1 + m_2/m_1}. \quad (2)$$

Notice that, for $m_1 \gg m_2$ the resulting reduced mass is $\mu \approx m_2$.

In spherical coordinates (r, θ, ϕ) , we use, first, the separation of variables method in Eq.(1) to separate the temporal dependence $\Phi(t)$ from the spatial dependence $\psi(\vec{r})$ of the wavefunction as $\Psi(\vec{r}, t) = \psi(\vec{r})\Phi(t)$.

Then, if the potential $V(\vec{r})$ is spherically symmetric, i.e. if it holds $V(\vec{r}) \equiv V(r)$, we apply again the separation of variables method to separate the radial wave function $R(r)$ from the angular part of $\psi(\vec{r})$ (here we take for granted that the angular dependent part is a spherical harmonic of the type $Y_{\ell, m}(\theta, \phi)$), as $\psi(\vec{r}) = R(r)Y_{\ell, m}(\theta, \phi)$. This procedure, results to a radial differential equation of the form

$$\frac{d}{dr}\left(r^2\frac{d}{dr}\right)R(r) - \frac{2\mu r^2}{\hbar^2}[V(r) - E]R(r) = \ell(\ell + 1)R(r). \quad (3)$$

We may, furthermore, simplify the latter equation by using the transformation $R(r) = u(r)/r$ and obtain the reduced radial differential equation,

$$\left[-\frac{\hbar^2}{2\mu}\frac{d^2}{dr^2} + V(r) + \frac{\ell(\ell + 1)\hbar^2}{2\mu r^2}\right]u(r) = Eu(r). \quad (4)$$

In this work, we are interested in the solutions of the radial equation (3), in the special case for which $V(r)$ is a Coulomb type potential given by

$$V(\vec{r}) = V(r) = -e^2/r, \quad r > 0 \quad (5)$$

(note here that the square of elementary charge is $e^2 = 1.4399764$ MeV fm). In this specific case, the differential equation satisfied by the reduced radial wave function $u(r)$ is written as

$$\left[-\frac{\hbar^2}{2\mu}\frac{d^2}{dr^2} - \frac{e^2}{r} + \frac{\ell(\ell + 1)\hbar^2}{2\mu r^2}\right]u(r) = Eu(r), \quad (6)$$

with boundary conditions $u(r=0) = 0$ and $u(r) \rightarrow 0$ as $r \rightarrow \infty$.

The radial equation (3) admits analytic solutions of the form [8],

$$R_{n,\ell}(r) = \sqrt{\left(\frac{2}{na_0}\right)^3 \frac{(n-\ell-1)!}{2n[(n+\ell)!]^3}} e^{-r/na_0} \left(\frac{2r}{na_0}\right)^\ell L_{n-\ell-1}^{2\ell+1}\left(\frac{2r}{na_0}\right) \quad (7)$$

where $L_{n-\ell-1}^{2\ell+1}(x)$ denote the associated Laguerre polynomials which are defined as,

$$L_n^a(x) = \sum_{m=0}^n \frac{(-1)^m}{m!} \binom{n+a}{n-m} x^m. \quad (8)$$

The square root coefficient in Eq. (7) comes out of the normalization condition written as

$$\int_0^\infty |R_{n,\ell}(r)|^2 r^2 dr = \int_0^\infty |u_{n,\ell}(r)|^2 dr = 1$$

which must hold for the Radial wavefunction $R(r)$ and the reduced radial wavefunction $u(r)$.

As mentioned in the Introduction, the energy eigenvalues are determined simultaneously with the corresponding wave functions and are given by

$$E_n = -\frac{\hbar^2}{2\mu a_0^2 n^2}. \quad (9)$$

We note that only the quantum number n (primary quantum number) enters the definition of E . In Eq. (9), a_0 denotes the Bohr radius of the particle of mass m_2 orbiting around the attractive particle of mass m_1 placed at the position $r = 0$ (in the relative coordinate system) and is defined by

$$a_0 = \frac{\hbar}{\mu c \alpha} \quad (10)$$

where α is the well known fine structure constant

$$\alpha = \frac{e^2}{4\pi\epsilon_0 \hbar c}. \quad (11)$$

In the latter definition ϵ_0 stands for the “vacuum electric permittivity” constant (c the speed of light).

3. Numerical solution of Schrödinger equation within N.N. techniques

Schrödinger’s equation doesn’t always possess analytic solutions, consequently a numerical approach must necessarily be employed. In this Section we present an advanced method derived within the context of neural networks (NN) which solves numerically the time independent Schrödinger equation (6) using the method developed by Kosmas and Lagaris [1].

Assuming that $u(r)$ is a trial solution of Eq. (6), we proceed with the following steps. At first, we define a grid from $r = 0$ up to a point $r = b$, where the wavefunction is practically vanishing (see below Sect. 4), and denote it by r_i , for $i = 1, 2, \dots, s$. Then, because equation (6) must hold at every point r_i of the grid, we write, equivalently, the expression

$$\sum_{i=1}^s \left[-\frac{\hbar^2}{2\mu} \frac{d^2 u(r_i)}{dr^2} + \left(-\frac{e^2}{r_i} + \frac{\ell(\ell+1)\hbar^2}{2\mu r_i^2} \right) u(r_i) - E u(r_i) \right]^2 = 0. \quad (12)$$

The next step towards solving the latter equation and obtain the $u(r)$, consists in (i) parametrizing $u(r)$, and (ii) minimizing the left-hand side of equation (12). In order to avoid the trivial solution $u(r) = 0$ everywhere in the domain of r , we divide the above expression by the normalization $\int |u(r)|^2 dr$. To facilitate the minimization process, we use a parametrization of the form

$$u(r) = r e^{-kr} N(r, \vec{u}, \vec{v}, \vec{w}) \quad (13)$$

where $k \in \mathbb{R}$ is an optimization constant (in our case $k \approx \sqrt{-2E\mu c^2/\hbar c}$), $u_j, v_j, w_j \in \mathbb{R}^d$ are the parameters which need to be determined through the minimization process.

The neural network $N(r, \vec{u}, \vec{v}, \vec{w})$ entering Eq. (13) is defined as

$$N(r, \vec{u}, \vec{v}, \vec{w}) = \sum_{i=1}^d u_i f(v_i r + w_i) \quad (14)$$

where $f(s)$ denotes the activation functions employed.

In our present work, we utilized the following three activation functions [9]:

(i) The sigmoid activation function defined as

$$f(s) = \frac{1}{1 + e^{-s}} \quad (15)$$

(ii) The ReLU activation function defined as

$$f(s) = \max(0, s) \quad (16)$$

(iii) The Softplus activation function defined as

$$f(s) = \ln(1 + e^s) \quad (17)$$

It is worth noting that, the function $N(r, \vec{u}, \vec{v}, \vec{w})$ of Eq. (14) constitutes a 2-layer neural network. The reason we parameterized u in this way is because it satisfies the conditions $u(0) = 0$ and $u(r) \rightarrow 0$ as $r \rightarrow \infty$. By inserting u into the left-hand side of Eq. (12), we derive the error (or loss) function as

$$E_f(u, v, w) = \sum_{i=1}^s \left[-\frac{\hbar^2}{2\mu} \frac{d^2 u(r_i)}{dr^2} + \left(-\frac{e^2}{r_i} + \frac{\ell(\ell+1)\hbar^2}{2\mu r_i^2} \right) u(r_i) - E u(r_i) \right]^2. \quad (18)$$

Our task, in the last step, is to minimize the error function E_f so as to approximate as close as possible the value zero. In practice, a value $E_f \approx 10^{-13} - 10^{-10}$ is considered zero. We mention that, the effectiveness of the minimization becomes obvious from the comparison of the numerical function $u(r)$ with the corresponding analytic solution (see below Sect. 4).

In this work, we used two methods for the minimization procedure (in Python language): (i) the BFGS, and (ii) the trust-constr, both provided by the Python submodule `scipy.optimize`[10]. Up to now, from these methods the performance of trust-constr proved to be better even though it requires longer training.

Regarding the determination of the energy E , this is calculated from the expression

$$E = \left(\int_0^\infty u^2(r) dr \right)^{-1} \int_0^\infty \left(\frac{\hbar^2}{2\mu} \left[\frac{du(r)}{dr} \right]^2 + \left[-\frac{e^2}{r} + \frac{\ell(\ell+1)\hbar^2}{2\mu r^2} \right] u^2(r) \right) dr. \quad (19)$$

as long as the wavefunction $u(r)$ is obtained.

4. Bound state wave functions for leptonic atoms

In this section, we use the numerical method derived in Sect. 3 in order to calculate the bound energy-spectra and the corresponding state wave functions in the purely leptonic (exotic) atom: Muonium (μ^+, e^-). The results obtained for the radial wave functions $R_{n,\ell}(r)$ for small quantum numbers n and ℓ are plotted in Fig. 1.

For the estimation of the level of performance we compare the numerical wave functions with the analytic ones given by Eq.(7).

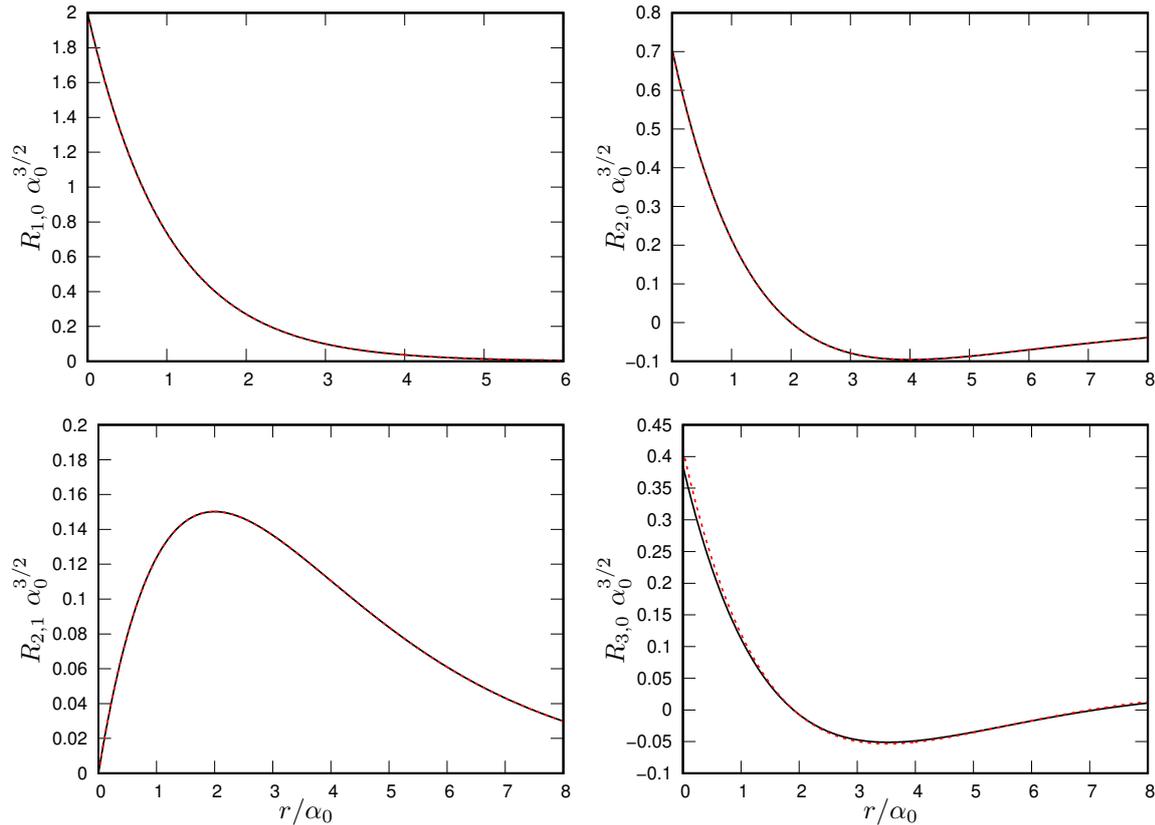


Figure 1. Comparison of the numerical Schrödinger solutions (red color) of the Muonium, (μ^+, e^-), radial wave functions: $R_{1,0}(r)$, $R_{2,0}(r)$, $R_{2,1}(r)$ and $R_{3,0}$, with the corresponding analytic ones (black color).

Table 1. Analytical and Numerical energies of the low-lying bound states of Muonium (μ^+, e^-), purely leptonic atom

(n, ℓ)	$E_{anal} [\times 10^{-6}]$	$E_{numer} [\times 10^{-6}]$
(1, 0)	-13.540200	-13.540370
(2, 0)	-3.385052	-3.385107
(2, 1)	-3.385052	-3.385107
(3, 0)	-1.504467	-1.585150

Numerical results show clearly that our method is a very powerful numerical tool for solving the time independent Schrödinger equation. As can be seen from Fig. 1, the performance of our numerical method is reduced as n gets larger. Regarding the numerical solution of these

equations, several programs have been developed [2, 3] utilizing different methods such as the Simulated Annealing [5], etc.

In Table 1, the corresponding values of the energies, obtained by using the eigenvalue formula (9) and the integral formula (19) are listed. We see that the agreement between the analytical and numerical results is very good.

5. Conclusions - Outlook

In this work, we derived a numerical method (in Python language) for solving the time independent Schrödinger equation. The algorithm is relying upon a two-layer artificial neural network. Then, we applied this method in order to solve this differential equation and determine the low-lying bound states (for $n = 1, \ell = 0$ and for $n = 2, \ell = 0$ and $\ell = 1$) in purely leptonic atoms focusing on the Muonium (μ^+, e^-) leptonic atom.

From the calculations performed in this work, we concluded that for larger values of the principal quantum number n and $\ell > 0$ a better training is required in order to achieve high confidence level of the agreement between the analytical and numerical wave functions.

In the next step, we are going to extend the Dirac equation since it is a relativistic description.

Acknowledgments

The authors A.G., O.K. and T.K. wish to acknowledge financial support from the Association for Advancement of Research on Open Problems (OPRA Association, Tel Aviv, Israel) in Nuclear Physics & Particle Physics. A.G. wishes to thank Theodora Papavasileiou for their discussions on the Schrödinger equation.

References

- [1] Kosmas T S and Lagaris I E 2002 *J. Phys. G* **28** 2907
- [2] Tsoulos I G, Kosmas O T and Stavrou V N 2019 *Comput. Phys. Commun.* **236** 237-43
- [3] Salvat F and Fernandez-Varea M 2019 *Comput. Phys. Commun.* **240** 165-77
- [4] Martinez-y-Romero R P 2000 *Am. J. Phys.* **68** 1050-5
- [5] Kosmas O T and Vlachos D S 2012 *Comput. Oper. Res.* **39** 576-81
- [6] Giannaka et al 2021 *J. Phys.: Conf. Ser.* **1730** 012140
- [7] Fleisch D 2020 *A Student's Guide to the Schrödinger Equation* (Cambridge: Cambridge University Press)
- [8] Bethe H A and Salpeter E E 1957 *Quantum Mechanics of One- and Two-Electron Systems* (Berlin: Springer)
- [9] Apicella A, Donnarumma F, Isgrò F and Prevete R 2021 *Neural Netw.* **138** 14-32
- [10] Nocedal J and Wright S J 2006 *Numerical Optimization* (New York: Springer)