Monte Carlo Method for the nuclear Few-Body Problem

Pedro Castro Perez* Alejandro Godínez Sandí†

March 30, 2020

Abstract

In this work we will use a Monte Carlo method to find numerical solutions for the nuclear few-body problem. The algorithm we will follow is the Green's function Monte Carlo method, in which by means of a suitable Green's function we will transform the Schrödinger equation in an integral form, and solve this via a random walks procedure. The main goal is to replicate the results obtained in [1]. We were able to successfully implement this algorithm to calculate the potential depth required to generate the binding energy of the ground state, and the wave function for three particular systems, the Deuteron, Triton and Alpha particle using several potential shapes.

1 Introduction

The problem of solving the Schrödinger equation for the few-body nuclear problem is, in most cases, non-trivial. Solutions for systems such as hydrogenic atoms are well studied. However, an exact solution for the ground state of the N-body wave equation is generally not possible to be obtained analytically [2], due to the fact that knowledge about the many-body wave function is needed, which typically scales greatly in size with the number of particles. It is the attempt of this research, to work out a Monte Carlo algorithm for the solution of the nuclear few-body problem.

Many methods can be applied to solve numerically the Schrödinger equation for an specific system in general, such as Density functional theory or Hartree–Fock methods, however, for many-body systems the Quantum Monte Carlo methods are the ones that have been widely used [3, 4, 5].

Quantum Monte Carlo methods provide accurate, and in some times, exact solutions for the many-body problem. These methods generally aim to find and compute the ground state wave function via handling the N-dimensional integrals that commonly arise in the many-body problems, as it is often done in Monte Carlo methods.

The approach of this work is to look for a formulation of the Schrödinger equation in an integral way and attempt the solution via the Green's Function Monte Carlo method. This technique, in general uses a Green's function to solve the many-body integral Schrödinger equation performed by Monte Carlo integration. Green's Function Monte Carlo method has been successfully used to study the ground states properties of many-body systems [6].

The Green function in this method is generally not known, but with Monte Carlo methods of random walks the required Green function can be realized, this is manly used in diffusion problems [7, 8]. In our case, we will see that the operator, which mainly occurs when the binding energy of the system is known, has a Green's function which is positive definite.

*s6pecast@uni-bonn.de

†alegod02@gmail.com

In this work, we will analyze and solve the few-body nuclear problem by means of the Green's Function Monte Carlo method for the Triton and Alpha particles for several potential shapes, in this case the Squared, Gaussian and Exponential wells. With this, we will obtain the potential depth required to generate the ground state binding energy and the ground state wave function.

In the following sections, we will present the theoretical basis of our problem, how we approached the problem, what algorithm we used and how we analyzed the errors. Finally, we will conclude by showing and discussing our results, by comparing them with some analytical solutions and results obtained in [1], and discuss about the implemented method.

2 Theoretical Basis

The description of the nuclear few-body problem is given as follows by the N-body Schrödinger equation

$$\left(-\sum_i \frac{\hbar^2}{2m_i}\nabla_i^2 + V(\mathbf{x_1},...\mathbf{x_N})\right)\psi(\mathbf{x_1},...\mathbf{x_N}) = E\psi(\mathbf{x_1},...\mathbf{x_N}). \tag{1}$$

For our problem, we assume that the energy of the system is known, in this case we fixed the energy to be $E = -B$. A suitable change of variables for the position vector will allow us to rearrange equation 1 in a more compact form. Let $\mathbf{r_i}$ be the new position variable defined as:

$$\mathbf{r_i} = \sqrt{\frac{2m_iB}{\hbar^2}}\mathbf{x_i}. \tag{2}$$

We proceed to introduce a parameter λ, which is the eigenvalue for the binding energy, in this case, the strength of the potential, which is what we are interested in. With this new parameter a second change of variables is done for the potential:

$$V(\mathbf{x_1},...\mathbf{x_N}) = -\lambda BW(\mathbf{r_1},...\mathbf{r_N}). \tag{3}$$

With these changes, equation (1) now has the following form

$$\left(-\nabla^2 + 1\right)\psi(\mathbf{R}) = \lambda W(\mathbf{R})\psi(\mathbf{R}), \tag{4}$$

where $\mathbf{R}$ is just the set $(\mathbf{r_1},...\mathbf{r_N})$ of three dimensions vectors. The operator $\left(-\nabla^2 + 1\right)$ has a Green function [1] , which will transform this last equation into an integral equation, of the form:

$$\left(-\nabla^2 + 1\right)G(\mathbf{R_0},\mathbf{R}) = \delta\left(\mathbf{R_0} - \mathbf{R}\right). \tag{5}$$

In this case, to find the appropriate Green function, we have to take into account the boundary conditions of the problem. This means that the wave function has to be square integrable in all 3N coordinates. Following the procedure of [1] to find the suitable Green function, equation (1) is written in integral form as:

$$\psi(\mathbf{R}) = \lambda\int d\mathbf{R'}G(\mathbf{R'},\mathbf{R})W(\mathbf{R'})\psi(\mathbf{R'}). \tag{6}$$

Equation (6) is the one that it is going to be solved with our Monte Carlo algorithm, which will be more detailed in Section 3. In general, this algorithm is going to be based on random walks iterations, as most Monte Carlo methods.

To start solving by iteration, we can write from equation (6)

$$\psi_{n+1}(\mathbf{R}) = \lambda_0\int d\mathbf{R'}G(\mathbf{R'},\mathbf{R})W(\mathbf{R'})\psi_n(\mathbf{R'}), \tag{7}$$

which can also be written as

$$\psi_{n+1}(\mathbf{R}) = \lambda_0 \xi \psi_n(\mathbf{R}). \tag{8}$$

Expanding ψ_0 as a linear combinations of the eigenfunctions $\psi^{(k)}$ of the operator ξ, and expressing $\psi^{(k)}$ with its eigenvalue $\lambda^{(k)}$, we get

$$\psi_0 = \sum C_k \psi^{(k)}, \tag{9}$$

and

$$\psi^{(\lambda)} = \lambda^{(k)} \xi \psi^{(k)}. \tag{10}$$

From here, we can obtain in a general form:

$$\psi_n = \sum C_k \psi^{(k)} (\lambda_0 / \lambda^{(k)})^n. \tag{11}$$

From this expression, we can see that, as we proceed with the iteration, the state $\psi^{(k)}$ for which $|\lambda|$ is the smallest, it is going to represent the state bound with the weakest potential, the ground state. The value of λ will be the variable that we will try to estimate with our algorithm. Since, it will give us information about the depth of potential, V_0. Due to the fact that, λ and V_0 are deeply related through Equation 3, the relation explicitly turns out to be:

$$\lambda = \frac{V_0}{B} \tag{12}$$

Where B represents the Binding energy of the N-nuclei system, which can be obtained experimentally. This means that by estimating λ, we have a direct estimation of the potential depth with the use of a factor that can be calculated experimentally. This is why, the main focus of our method will fall on the estimation of λ.

3 Methods

3.1 Green's Function Monte Carlo Method

This is the main algorithm that has been used in this project. It has been proposed in [1] for the resolution of the Schrödinger equation problem, already explained in the previous section. For the case of a number of nuclei N equal or greater than two and for any given shape for the potential $W(\mathbf{r_1}, ...\mathbf{r_N})$.

In this algorithm, a continuous random walk is implemented, in which any step can be interpreted as an iteration of Equation (7). So, for any step n of the algorithm, we have a population of points $\mathbf{R_1}, ...\mathbf{R_M}$, which must fulfill that the density number in the configuration space gives us the wave function of our N-nuclei system $\psi_n(\mathbf{R})$.

Once we have this initial condition for every iteration, we can begin describing the algorithm for any given step n. First, we calculate the value $\lambda_0 W(\mathbf{R}_m)$, $\forall m \in (1, M)$. Then, we use ϵ_m a random integer set with mean value $\lambda_0 W(\mathbf{R}_m)$. This is achieved by computing the integer and fractional part of $\lambda_0 W(\mathbf{R}_m)$ and make ϵ_m equal to the integer part plus one, with probability equal to the fractional part.

Now, for every point $\mathbf{R}_m$, we pick ϵ_m new points $(\mathbf{R}_1^{new}..., \mathbf{R}_{\epsilon_m}^{new})$ given by the Green function as a probability distribution function: $G(\mathbf{R}_m, \mathbf{R}_\ell^{new})$, $\forall \ell \in (1, \epsilon_m)$. This can be done by calculating a norm R and a unity vector $\mathbf{\Omega}$. The first one can be obtained with:

$$u = -ln(\xi_1 \xi_2 ... \xi_{3N}) \quad \text{and} \quad v = \sqrt{1 - \xi_0^{2/(3N-1)}} \tag{13}$$

$$R = uv \tag{14}$$

Where $\xi_0, \xi_1, \xi_2, ..., \xi_{3N}$ is a set of random numbers distributed by the uniform distribution between 0 and 1.

The unit vector $\mathbf{\Omega}$ must be homogeneously distributed in 3N-dimensional unit sphere. This can be done easily by setting the 3N components of $\mathbf{\Omega}$ to be normally distributed and then re-normalize them to unity.

Once we have obtained R and $\mathbf{\Omega}$, the creation of a new point $\mathbf{R}_\ell^{new}$, which uses the Green function as a distribution function, can be obtained in the following way:

$$\mathbf{R}_\ell^{new} = \mathbf{R}_m + R\mathbf{\Omega} \tag{15}$$

A proof to why this calculation of $\mathbf{R}_\ell^{new}$ follows the Green function as a distribution can be seen in the Appendix of [1]. Having followed these steps, we end up with M sets of points $(\mathbf{R}_1^{new}..., \mathbf{R}_{\epsilon_m}^{new})$, $\forall m \in (1, M)$. Finally, combining these M sets of points as a whole one we obtain $(\mathbf{R'}_1..., \mathbf{R'}_{M'})$. This set has the same property as the initial set, where the density of points in the configuration space is the wave function $\psi_{n+1}(\mathbf{R'})$ in the following $(n+1)$ iteration step.

This procedure is the one followed in every iteration of the random walk. Here, the parameter λ_0 is going to represent our estimation for the parameter described in the previous section as λ, which represents the ratio between the potential depth V_0 and the binding energy B. In our algorithm, when $\lambda_0 > \lambda$, the population of points $\mathbf{R}$ increases and vice-versa. This means, that at the beginning of the random walk, λ_0 is going to be assign a value much greater compared to the eigenvalue λ that we want to estimate. So the population of points rises to a number considered big enough to make Monte Carlo estimations. It is also convenient that for the initial step of $n = 0$, we chose just one point $\mathbf{R}_0$ with a value $W(\mathbf{R}_0) > 0$.

Once the population is considered to be big enough, we change the value for λ_0, for any given step n, as our estimate for λ:

$$\bar{\lambda}_n = \frac{M}{\sum_{i=1}^{M} W(\mathbf{R}_i)} \tag{16}$$

These changes in the value λ_0, will help to stabilize the population number, as well as, adapting the density of points to approach the wave function. After the number of steps L defined by us, the process would come to an end and we will be left with a set of samples $(\bar{\lambda}_1, ..., \bar{\lambda}_L)$. From this set, our estimation for $\bar{\lambda}$ can easily be obtained by simply calculating the mean. Although, there are a few aspects which will be discussed when we explain the error estimations for this method.

3.2 Well shapes

So far, we have not defined in a precise way how are defined the different potential functions $V(\mathbf{x_1}, ...\mathbf{x_N})$. First, we are going to consider Equation (3), which shows us the relation between the functions $V(\mathbf{R})$ and $W(\mathbf{R})$. As we can see, the relation between both function is the same but for a factor $-\lambda B = -V_0$ which represents the negative potential depth. This means that $W(\mathbf{R})$ could be considered as a "normalized" positive potential of $V(\mathbf{R})$. So, to define a given potential, we just need to define the function $W(\mathbf{R})$, which will be different for every case.

Now, we are going to explicitly show the expressions for the different wells that have been used in this project: Squared, Gaussian and Exponential Wells. The definition of these potentials has been done with a range parameter called b defined in [9].

Squared Well:

$$W(r) = \begin{cases} 1 & \text{if } r < b \\ 0 & \text{if } r \geq b \end{cases} \tag{17}$$

Gaussian Well:

$$W(r) = \exp\left(-(1.4354 \cdot r/b)^2\right) \tag{18}$$

Exponential Well:

$$W(r) = \exp\left(-3.5412 \cdot r/b\right) \tag{19}$$

It is important to know that the way these Wells are defined is as pair forces potentials, so the variable r represents the distance between one of the particles and another. Moreover, the potential must be additive, which means that:

$$W(\mathbf{R}) = W(\mathbf{r_1}, ...\mathbf{r_N}) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} W\left(\sqrt{(\mathbf{r}_i - \mathbf{r}_j)^2}\right). \tag{20}$$

Lastly, it must be noted that the range parameter b is defined originally with units of distance. However, because of the fact that the fraction r/b has no units, in our algorithm instead of b we will use the parameter without units b/R. Where R is called unit of length parameter and has the expression:

$$R = \frac{\hbar}{\sqrt{2MB}}, \tag{21}$$

with M being the mass of the proton and B the binding energy of the N-nuclei system.

3.3 Error estimates

Once a complete definition of the algorithm and the different potentials, which are going to be used, has been described. In this part of the report, we are going to discuss a way of estimating possible uncertainties taking place when estimating λ in our method.

One of the main sources of error that come from the Monte Carlo estimate of $\bar{\lambda}$ comes from the fact that the initial part of the samples is not going to be very "reliable" because the number of points is not yet stabilized and the density do not resemble the wave function of the system. For this reason, a good way of avoiding these initial contributions to the estimate of $\bar{\lambda}$ can be to not consider the initial percentage of the samples. In our case, we have always neglected the initial 10% of our samples to estimate our $\bar{\lambda}$ parameters.

The second source of uncertainties comes from the intrinsic nature of estimating parameters with the use of Monte Carlo methods. Any Monte Carlo estimate, obtained during a finite time, has always some degree of uncertainty. Due to the fact that there will always be random fluctuations during the estimation process. For this reason, it becomes important to find a way to estimate the uncertainties coming from a given Monte Carlo estimate. In our case, the method that we have followed has the name: **Simple Block Bootstrap**. This method is deeply explained in [10].

This method has two main processes. The first one consist in performing **Data Blocking** to the samples of $\bar{\lambda}$, in order to avoid the effect that Autocorrelation has in the estimate of the uncertainties. To do this, it is necessary to obtain the normalized Autocorrelation function of our sample $\Gamma(t)$, and estimate the blocking length as:

$$\ell = 2 \cdot \tau_{int} = \sum_{t=-W}^{W} \Gamma(t), \tag{22}$$

where W is the first value for t where the $\Gamma(t)$ becomes 0 within errors.

Once we have estimated the blocking length ℓ, is time to perform Data Blocking to our set of samples $(\bar{\lambda}_1, ..., \bar{\lambda}_L)$ to a blocked set of samples $(\bar{\lambda}_1^B, ..., \bar{\lambda}_{L'}^B)$:

$$\bar{\lambda}_i^B = \frac{1}{\ell} \sum_{k=1}^{\ell} \bar{\lambda}_{(i-1)\cdot\ell+k}. \tag{23}$$

With the blocked set of samples we can perform the second process to obtain the uncertainty to our estimate: **Bootstrap Method**. Which won't be explained here because of a matter of space and relevance to the topic of this paper. However, a description to this method can be seen in pages 64-66 in [10].

The use of Data Blocking and Bootstrapping methods to our samples, will give us a proper estimate of the intrinsic uncertainties coming from the Monte Carlo integration. However, we must also consider a way of checking for the correctness of our results. Two ways of doing this, are going to be shown in the following sections. The first one being a comparison with an analytical solution of the problem. This solution has been obtained in pages 263-268 of [11] for the case of the Squared Well, and the comparison with our results will be shown in the following section. Another option would be to compare with the results obtained in [1], which will also be done in the Discussion section.

Finally, it must be mentioned that the code created for this project can be consulted in the following public Github repository: `https://github.com/pedro-95/few-body`.

4 Results

First of all, the 2-Nuclei Deuteron particle has been studied. The reason for this has already been mentioned and is that this 2-Nuclei system has an analytical solution for the case of a Squared Well [11]. This would allow us to check the correctness of our implementation of the algorithm. In Figure 1, these results are shown together with the analytical solution. Even though, the analysis of the results obtained will be held in the following Section. It must be mentioned, that the integration process for all of the 15 points shown in the Figure 1, has been carried out with an average population of 8000 per step during 2000 steps each. As we said, a 10% of our 2000 samples has been discarded for every point. Lastly, the uncertainties to our estimations of λ are also being shown in Figure 1. These uncertainties do not appear visible in the Figure because they are too small, being the biggest one of the order of 0.1%.

Before continuing exposing our results, we must remark that in our algorithm, both the range parameter b of the potential, that we are setting, and λ, the parameter the we are estimating, are "normalized". The range parameter is normalized with the unit of length parameter R and λ is a normalized parameter of V_0 with the Binding Energy B. R depends directly on B as shown in Equation (21), and both parameters change with the number N of nuclei in our system. This means that for different N-nuclei systems, we have different normalization variables. With the data obtained by [12], Table 1 shows the different values of B and R for our 3 systems.

	Binding Energy	Unit of Length
Deuteron (N=2)	$B_d = 2.224$ MeV	$R_d = 3.053 \cdot 10^{-13}$ cm
Triton (N=3)	$B_t = 8.481$ MeV	$R_t = 1.563 \cdot 10^{-13}$ cm
Alpha particle (N=4)	$B_\alpha = 28.296$ MeV	$R_\alpha = 8.563 \cdot 10^{-14}$ cm

Table 1: Binding Energy B and Unit of Length R according to the N-nuclei system. The values for the Binding energy were obtained from [12].

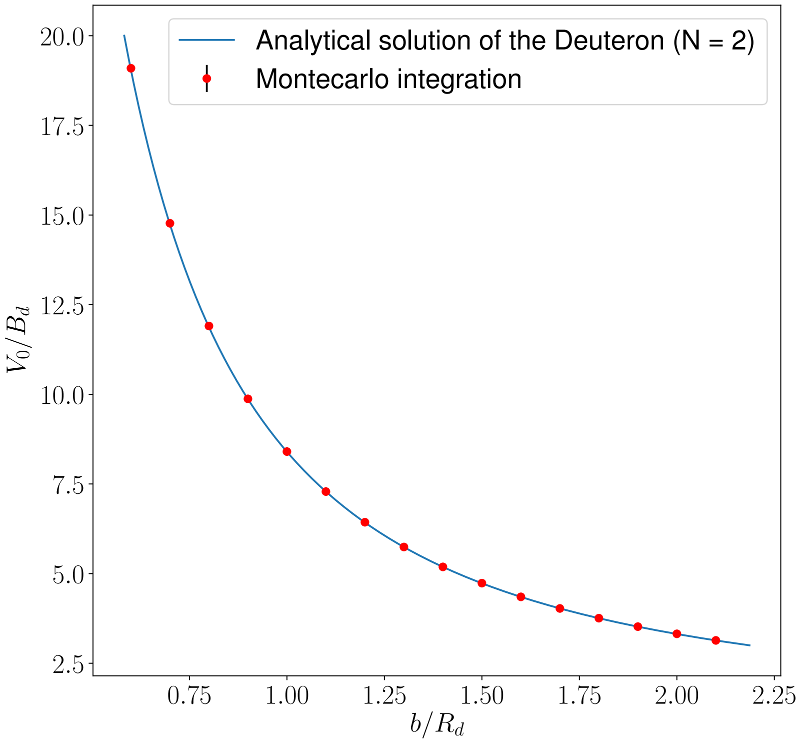


Figure 1: Comparison between the implemented Green's function Monte Carlo Method and the Analytical Solution obtained in [11] for the case of the 2- nuclei Deuteron particle in a Squared Well. The data plotted is shown in Table 2 in Appendix A.

Once the comparison with the analytical solution for the Deuteron was made, we proceeded with obtaining the λ estimations for the cases without analytical solution: the Triton and Alpha particle. These results can be seen in Figure 2. As we can see, these results have been obtained with the use of the 3 potential shapes exposed in the previous Section: Squared, Gaussian and Exponential Wells. As in the previous figure, uncertainties are also shown in Figure 2. However, they are also not perceptible because of it's small value, which for the case of the biggest one does not reach 0.4%. The average population of points has been around 9000 per step during 2000 steps each estimation point. And like in the previous case, the initial 10% of the samples have been discarded for each point. The conclusions to these results will be expressed in the Discussion section.

Finally, the last results that will be exposed in this paper are the ones related with obtaining the wave function of a given N-nuclei system. As we said before in Section 3, for any given step n the population of points $(\mathbf{R_1}, ...\mathbf{R_M})$ fulfills the property that the density number in the configuration space gives us the wave function of the N-nuclei system $\psi_n(\mathbf{R})$. This fact has been used to determine an estimation for the wave function in the case of a given step. But before this, a good replacement for the wave function that can be estimated and used to work in an easy manner, is what we will call the "Medium distance between particles". This observable can be estimated as:

$$MD_n = \binom{N}{2}^{-1} \cdot \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} |\mathbf{r_i} - \mathbf{r_j}| \qquad (24)$$

The reason for the normalization to have the expression $\binom{N}{2}$ is just the result for the different achievable combinations between r_i and r_j without repeating and i and j not being the same. After this, a histogram can be drawn to study different situations and how are the different variables related with each other. This can be seen in Figures 3 and 4, where different Potential shapes, different N values and different range parameters b/R have been studied.

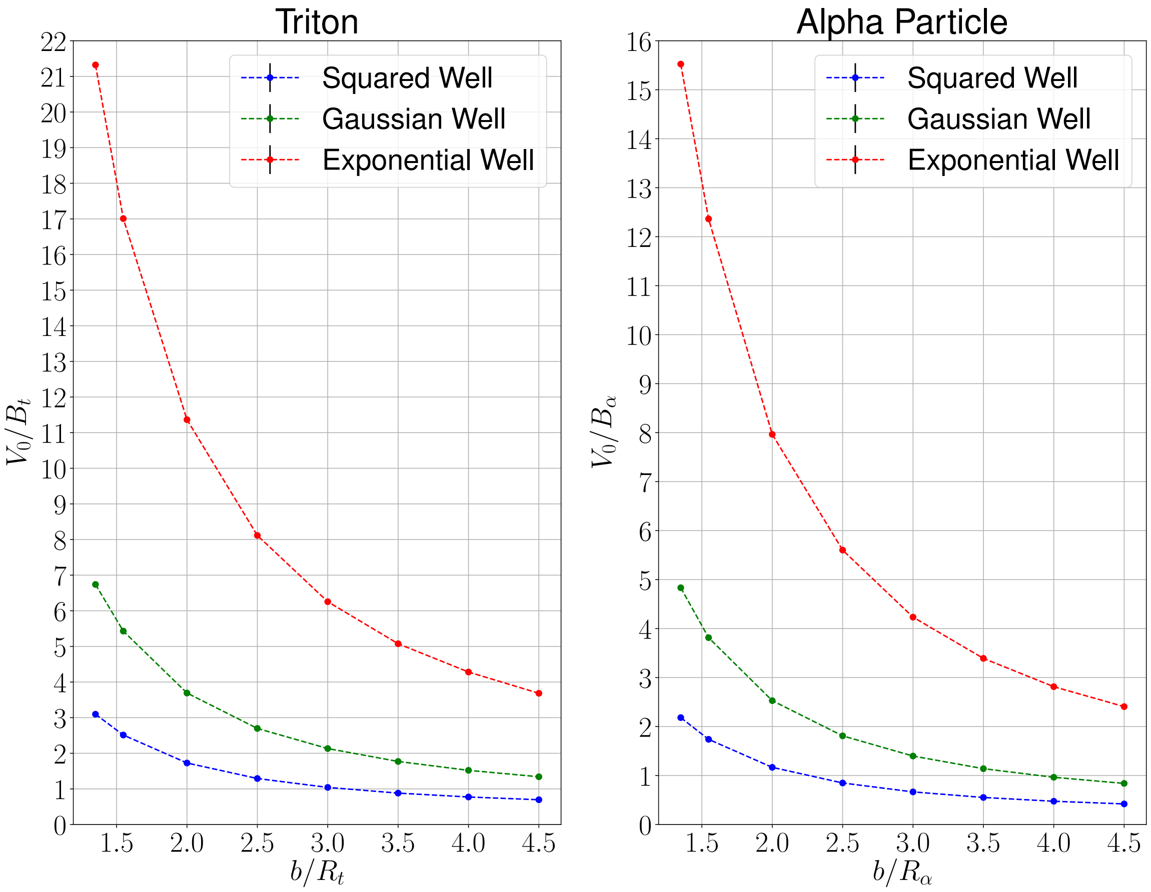


Figure 2: Results obtained by the Green's function Monte Carlo Method for different values of b/R and different Well shapes. The 3-nuclei Triton particle and the 4-nuclei Alpha particle were studied. The data plotted is shown in Tables 3 and 4 in Appendix A.

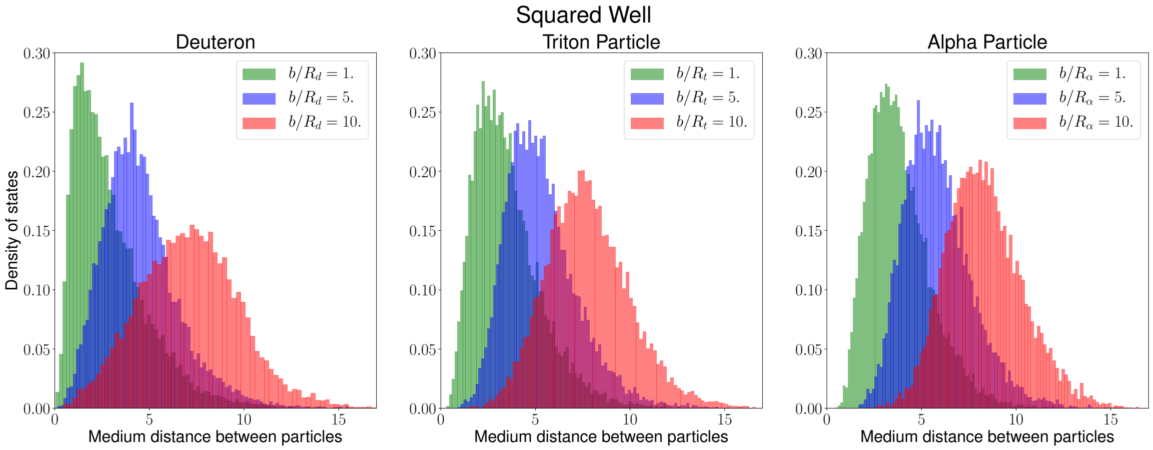


Figure 3: Medium distance between particles obtained for a Squared Well in the three different studied systems and for different range values b/R of the Potential.

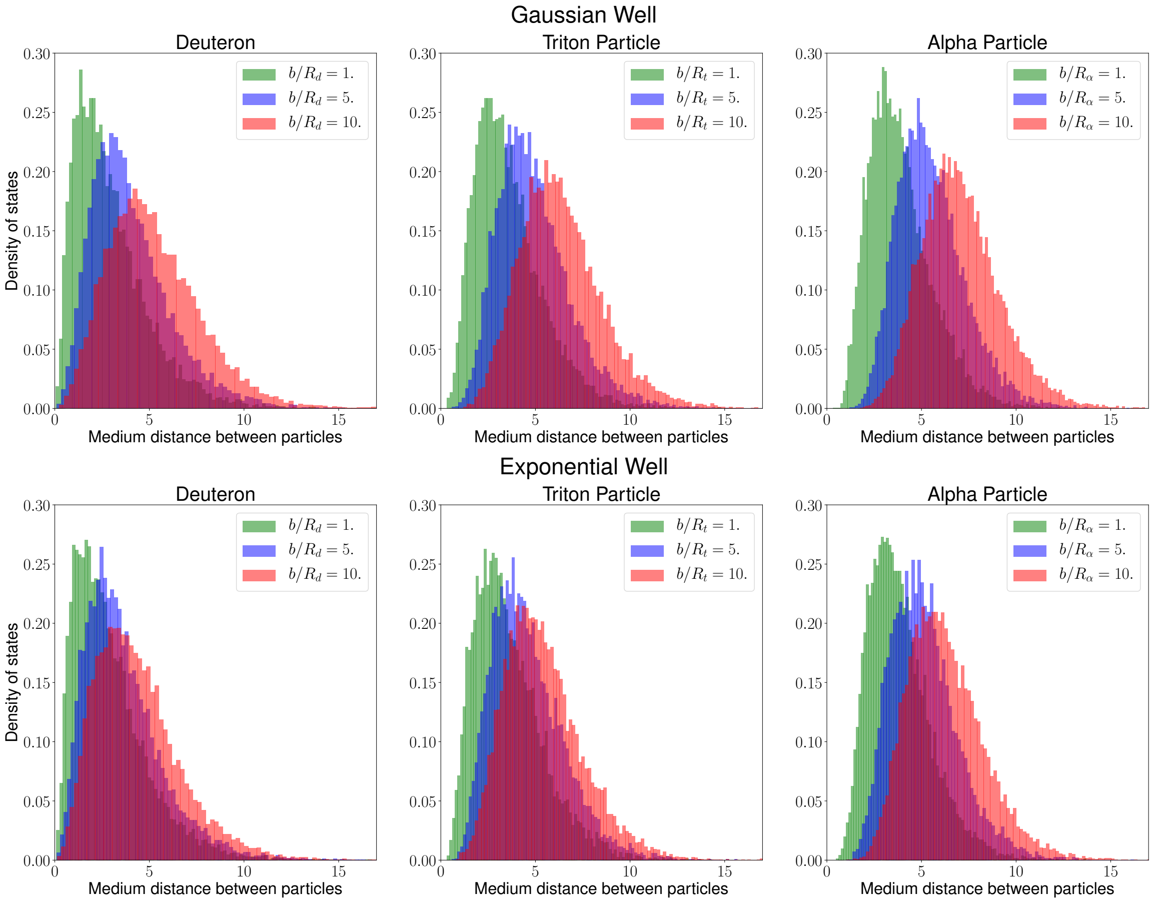


Figure 4: Medium distance between particles obtained for the Gaussian and Exponential Wells in the three different studied systems and for different range values b/R of the Potential.

The wave function can be obtained with a process similar to the Medium Distance. But instead of one parameter we must estimate three of them, which will represent each of the three dimensional axis of the wave function.

$$x_n - x_m = N \cdot \binom{N}{2}^{-1} \cdot \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \mathbf{r_i}(1) - \mathbf{r_j}(1) \tag{25}$$

$$y_n - y_m = N \cdot \binom{N}{2}^{-1} \cdot \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \mathbf{r_i}(2) - \mathbf{r_j}(2) \tag{26}$$

$$z_n - z_m = N \cdot \binom{N}{2}^{-1} \cdot \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \mathbf{r_i}(3) - \mathbf{r_j}(3) \tag{27}$$

Once these three observables are estimated, we proceed with the plot of two dimensional histograms for a better visualization of the wave function. These results can be seen, in Appendix B.

5 Discussion

First, we must begin with the comparison between the analytical solution and the implemented Method for the case of the Deuteron, showed in Figure 1. As we can see, our method is very capable of simulating the behaviour obtained by the Solution. This is a good sign that indicates that our implementation of the algorithm has been adequate. In addition to obtaining correct results, we can also see that the relative error is very small. This is due to the fact that we have used a population of points big enough during every step of the integration, which increased in a big amount the time that took us the integration of every point.

Apart from this initial comparison, we have already mentioned that another way of checking the correctness of our implementation was comparing the results obtained in [1]. The normalized range parameter values used in that paper are also included in ours, as can be seen in Figure 2 or in Tables 3 and 4 in Appendix A. This allows a direct comparison between results, and what we can see is a partial match between the results. The tendencies and order of the numerical results seem to match but when we analyze them in deepness, we can perceive certain differences statistically significant, both in the main value and in the uncertainty. These differences could be due to a couple of factors which have to be considered.

One of those factors could be the fact that the paper [1], where this method has been proposed, was published in the year 1962. The calculations were performed with the use of a computer called: IBM 7090, which had a computation speed of around 100 KFlop/s. To put this number in perspective, a smartphone nowadays has a computation speed around 400 millions times faster. Moreover, we have no information of the population size used in every step of the calculation and the quality of the random number generator. All these facts could lead us to think that the calculations obtained in [1] might have relevant deviations from the real values.

Another possible factor could be that our implementation is not completely correct, even though the comparison with the analytical solution was successful. Maybe, there is another factor that we have not considering that makes our algorithm underestimate the uncertainties to our results.

If we place all these differences aside, we can see that the general behaviour of λ in the 3 systems is very similar, the major difference is just a change in the scale. This means that with the integration of just a few points of any desired system with any Well shape, one could do a regression with the functional behaviour of the analytical solution for the Deuteron. With this procedure, we can obtain a particular solution for any given N-nuclei system and any Well shape.

The global behaviour of the λ variable increases as the potential range parameter b/R decreases. This behaviour for the Deuteron is the one described by the analytical solution, which is similar to the behaviour of an inverse function ($\sim 1/b$) for b/R large enough. And as we said before, this behaviour is the same for the case of the Triton and Alpha particles. The reason why the λ values are greater in the case of the Exponential Well and smaller in the case of the Squared Well, being the Gaussian Well between both, can be seen in an easy way with our estimation of $\bar{\lambda}$ shown in Equation (16). The Squared Well has a maximal value for the Potential in a region of space quite big in comparison with the Gaussian or Exponential Wells. The second one has an exponential decrease in the value of the Potential when the distance between particles gets slightly greater than 0. This fact makes λ obtain greater values for the case of an Exponential Well and smaller ones for a Squared Well.

In a similar way, one can analyze the results obtained for the Medium Distance between particles shown in Figures 3 and 4. When comparing different Well shapes, for the case of the Squared Well the Medium Distance is statistically smaller when compared to the Gaussian or Exponential Wells. This is because the Well shape influences the distance between particles. In the Squared Well, the potential value does not change in a region quite large of the space between the nuclei, meaning that they can have a greater separation. But in the opposite case of the Exponential Well, the potential changes exponentially in the space between the particles, decreasing the separation between them. The Gaussian Well is just an intermediate case.

Finally, the last analysis will be a comparison between the different studied N-nuclei systems for the case of the Medium Distance between particles. As we can see by comparing Figures 3 and 4, with the increase in the number N of nuclei the Medium Distance also increases. This result is somehow expected, because of the fact the “size” of the system should increase with the number of particles and this is what we are able to perceive with these results. In the Wave function of the different cases shown in Appendix B, this trend can also be seen.

6 Summary

As a summary, in this report we have implemented a Monte Carlo Method called Green’s Function Monte Carlo, which has been used for the resolution of the nuclear Few-Body Problem. This problem in a lot of cases has no analytical solutions, where the relation between the depth of the potential for the Ground State V_0, and the shape and range of the potential can be found. With this method proposed in [1], we were able to compare our results with the available analytical solution of the Deuteron (N=2), which agree in an accurate manner. This first result told us how well our algorithm was. We also obtained the depth of the potential for the Triton (N=3) and Alpha particle (N=4) for different Potential shapes and ranges. We compared these results with [1] and as discussed in the last section, they reasonably agree with the expected results but some differences were found. Our error analysis also shows how well our algorithm was able to estimate this quantities. As a secondary task, we have also successfully estimated the Wave function of those systems and the average distance between the nuclei. In general, our implemented Monte Carlo algorithm gives effective and expected results for the calculation of the potential depths and wave functions for a given N-body system.

References

[1] MH Kalos. Monte carlo calculations of the ground state of three-and four-body nuclei. *Physical Review*, 128(4):1791, 1962.

[2] Brian Harold Bransden, Charles Jean Joachain, and Theodor J Plivier. *Physics of atoms and molecules*. Pearson education, 2003.

[3] VG Rousseau. Stochastic green function algorithm. *Physical Review E*, 77(5):056705, 2008.

[4] WMC Foulkes, Lubos Mitas, RJ Needs, and G Rajagopal. Quantum monte carlo simulations of solids. *Reviews of Modern Physics*, 73(1):33, 2001.

[5] Raimundo R dos Santos. Introduction to quantum monte carlo simulations for fermionic systems. *Brazilian Journal of Physics*, 33(1):36–54, 2003.

[6] James B Anderson. *Quantum Monte Carlo: origins, development, applications*. Oxford University Press, 2007.

[7] Malvin H Kalos and Paula A Whitlock. *Monte carlo methods*. John Wiley & Sons, 2009.

[8] Tao Pang. Diffusion monte carlo: A powerful tool for studying quantum many-body systems. *American Journal of Physics*, 82(10):980–988, 2014.

[9] John M Blatt and J David Jackson. On the interpretation of neutron-proton scattering data by the schwinger variational method. *Physical Review*, 76(1):18, 1949.

[10] Marcus Petschlies and Carsten Urbach. Computational physics. Computational Physics class script, 14th October 2019.

[11] Anwar Kamal. *Nuclear Physics*. Springer-Verlag Berlin Heidelberg, 2014.

[12] Aaldert Hendrik Wapstra and Kors Bos. The 1977 atomic mass evaluation: in four parts part i. atomic mass table. *Atomic Data and Nuclear Data Tables*, 19(3):177–214, 1977.

A Tables

b/R_d	Deuteron
	Squared Well
0.6	19.09 ± 0.02
0.7	14.770 ± 0.014
0.8	11.91 ± 0.01
0.9	9.87 ± 0.01
1.0	8.403 ± 0.007
1.1	7.287 ± 0.005
1.2	6.433 ± 0.005
1.3	5.742 ± 0.004
1.4	5.1872 ± 0.0024
1.5	4.7339 ± 0.0021
1.6	4.352 ± 0.002
1.7	4.0286 ± 0.0017
1.8	3.7573 ± 0.0019
1.9	3.521 ± 0.002
2.0	3.321 ± 0.002
2.1	3.1376 ± 0.0013

Table 2: Values obtained for λ with the use of the Green's Function Monte Carlo Method in the case of the Deuteron. These values were plotted in Figure 1.

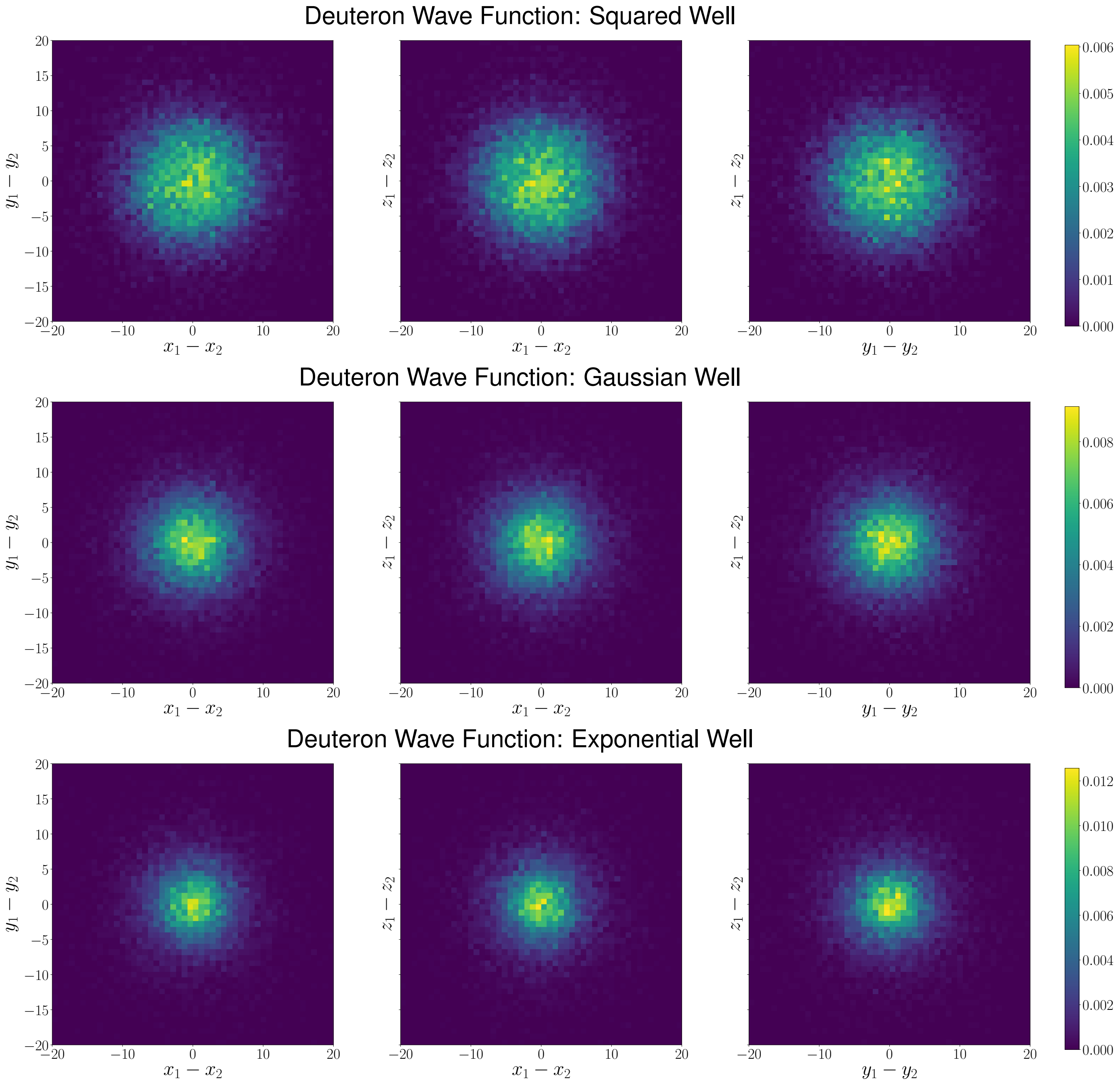
b/R_t	Triton		
	Squared	Gaussian	Exponential
1.35	3.0961 ± 0.0028	6.739 ± 0.007	21.32 ± 0.03
1.5477	2.5124 ± 0.0024	5.430 ± 0.007	17.008 ± 0.021
2.0	1.7275 ± 0.0012	3.692 ± 0.002	11.364 ± 0.014
2.5	1.2918 ± 0.0006	2.6973 ± 0.0017	8.114 ± 0.008
3.0	1.0421 ± 0.0004	2.1328 ± 0.0012	6.255 ± 0.007
3.5	0.8820 ± 0.0004	1.7704 ± 0.0012	5.075 ± 0.003
4.0	0.77321 ± 0.00024	1.5213 ± 0.0008	4.281 ± 0.005
4.5	0.69567 ± 0.00021	1.3412 ± 0.0006	3.6840 ± 0.0018

Table 3: Values obtained for λ with the use of the Green's Function Monte Carlo Method in the case of the Triton. These values were plotted in Figure 2.

b/R_α	Alpha Particle		
	Squared	Gaussian	Exponential
1.35	2.183 ± 0.003	4.83 ± 0.01	15.52 ± 0.05
1.5477	1.7390 ± 0.0018	3.815 ± 0.007	12.36 ± 0.04
2.0	1.1674 ± 0.0008	2.529 ± 0.003	7.965 ± 0.014
2.5	0.8482 ± 0.0006	1.8103 ± 0.0019	5.602 ± 0.009
3.0	0.6661 ± 0.0004	1.3982 ± 0.0009	4.234 ± 0.007
3.5	0.5523 ± 0.00024	1.1404 ± 0.0009	3.392 ± 0.006
4.0	0.4751 ± 0.0002	0.9644 ± 0.0005	2.814 ± 0.003
4.5	0.42068 ± 0.00017	0.8393 ± 0.0006	2.4072 ± 0.0023

Table 4: Values obtained for λ with the use of the Green's Function Monte Carlo Method in the case of the Alpha Particle. These values were plotted in Figure 2.

B Wave Function


Deuteron Wave Function: Squared Well

Deuteron Wave Function: Gaussian Well

Deuteron Wave Function: Exponential Well

Figure 5: Wave Function for the Deuteron system using Squared, Gaussian and Exponential Wells with a range of the potential of $b/R_d = 5$..

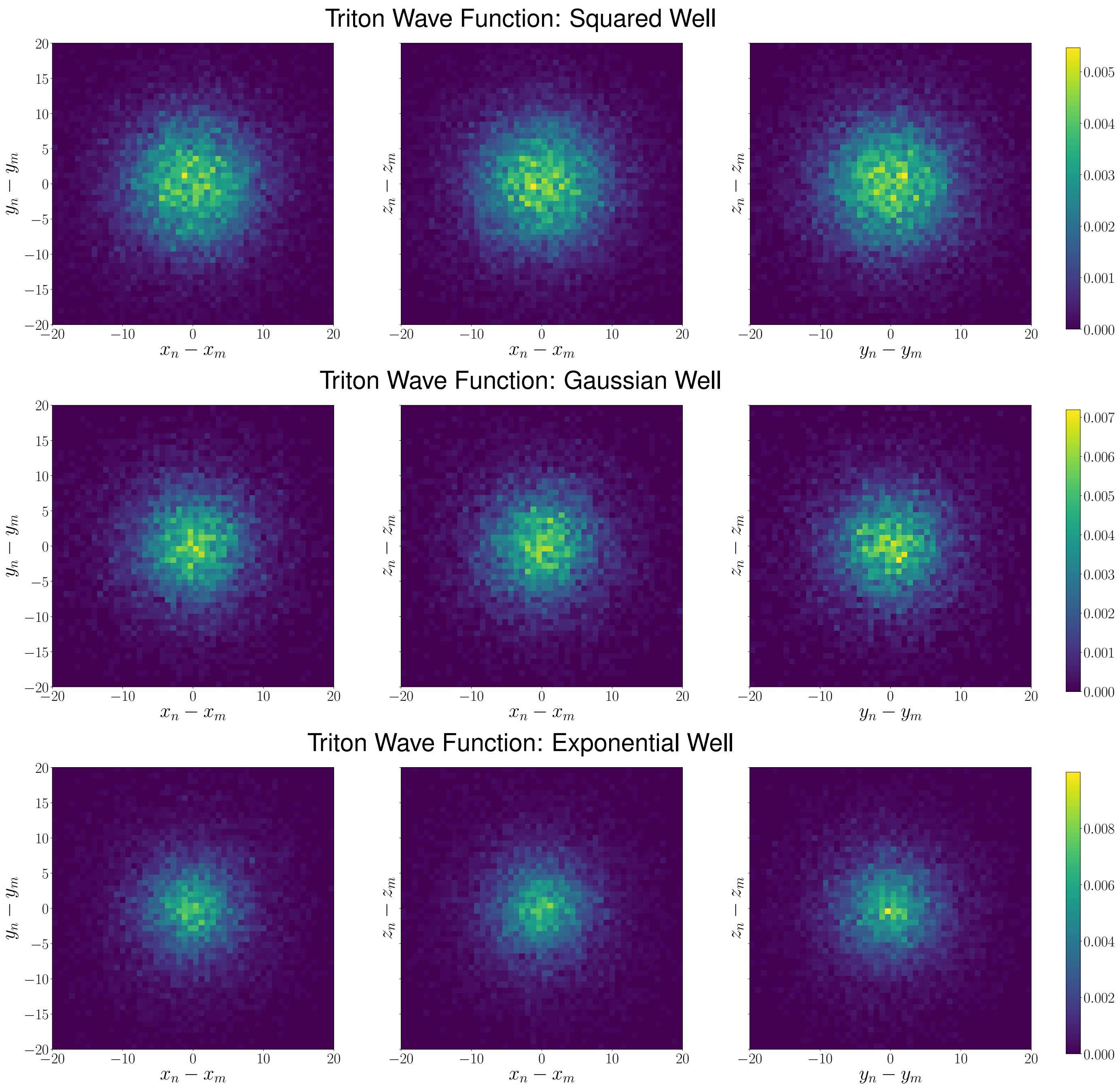
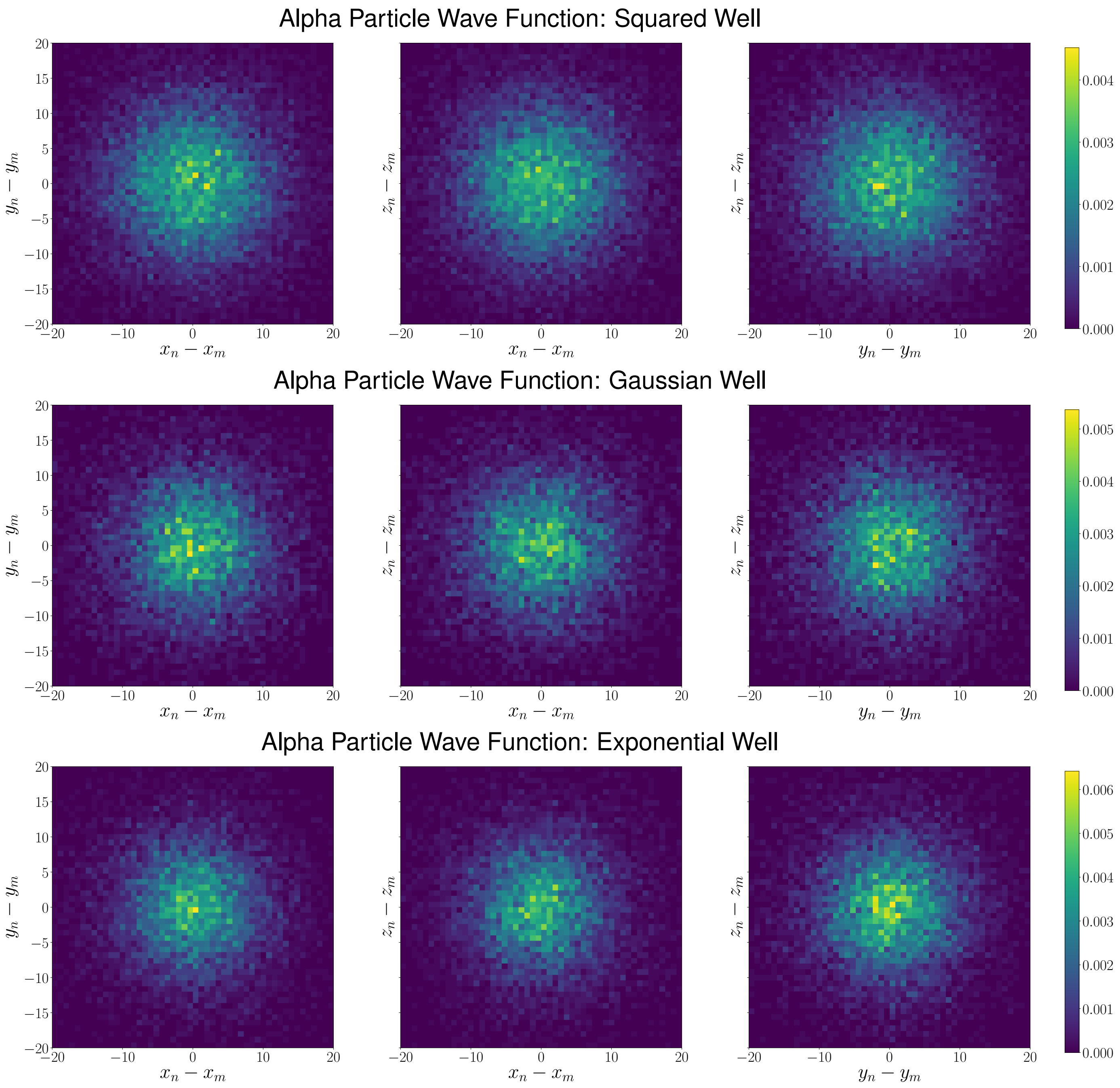


Figure 6: Wave Function for the Triton system using Squared, Gaussian and Exponential Wells with a range of the potential of $b/R_t = 5$..


Alpha Particle Wave Function: Squared Well

Alpha Particle Wave Function: Gaussian Well

Alpha Particle Wave Function: Exponential Well

Figure 7: Wave Function for the Alpha system using Squared, Gaussian and Exponential Wells with a range of the potential of $b/R_\alpha = 5$...