

Programming a Linear Algebra Solution to the Problem of Two Electrons Trapped in a Spherical Harmonic Oscillator Well

Elizabeth Drueke

March 8, 2016

Abstract

When solving complicated quantum systems, analytical solutions are rarely available. Thus, it is important to develop numerical methods which can solve complex differential equations arising from these systems. Here, we look particularly at the case of one or two electrons confined to a spherical quantum mechanical well, and analyze the effect and efficiency of both the Jacobi Rotation Algorithm and Householder's Algorithm in solving these systems. We conclude that Householder's Algorithm is the more efficient choice of the two.

1 Introduction

The quantum harmonic oscillator is hugely important both in quantum mechanics and in other areas of physics because a good many physical systems can be approximated as harmonic oscillators. In this report, we look at three possible cases of the quantum harmonic oscillator set up: one electron confined to a spherical harmonic oscillator, 2 non-interacting electrons confined, and 2 interacting electrons confined to the oscillator. We are particularly interested in developing numerical solutions to the differential equations posed by these systems. This is done using the Jacobi Rotation Algorithm and the Householder Algorithm.

Section 2 provides some motivation for the problem at hand and goes into some of the derivation of the Schrodinger equations for these systems. It sets up the math which is necessary to solve the problem. Section 3 discusses the algorithm derived for this project in particular, and discusses the code written and how everything was implemented. The basic framework for the code is C++ with ROOT [1] plotting utilities. This section also discusses the time dependence of the various algorithms used within the code. Section 4 discusses the calculations performed by the code, particularly in regards to the solutions plotted. In it, the results and patterns noticed in those results are mentioned. Section 5 discusses the conclusions of the project, including aspects of both the physics and the code.

2 Theory

Arguably the most important equation in quantum mechanics is Schrodinger's Equation, briefly stated as

$$H\Psi = E\Psi, \quad (2.1)$$

where H is the Hamiltonian for the system, E is the energy of the system, and Ψ is the state of the system. Already, this equation is reminiscent of an eigenvalue problem, in which a matrix (the representation of H) acts upon a vector (the ket representation of the state) and returns that vector multiplied by some constant (an eigenvalue, the energy of the system). Indeed, quantum mechanics is frequently formatted in this way, with state kets being represented by column vectors and the Hamiltonian by a Hermitian matrix [5].

However, the Schrodinger equation is also a differential equation, and it is also quite reasonable to attempt to construct an analytical solution. It is often sufficient to begin with the classical Hamiltonian, and then substitute in the quantum mechanical counterparts to the classical quantities. These usually involve derivatives, and so the system is solvable by a differential equation [5]. For example, quantum mechanical momentum is given by

$$p \rightarrow -\frac{1}{i\hbar} \frac{\partial}{\partial x}$$

In the case of a particle confined to a well with a harmonic oscillator potential, we have a potential

$$V(r) = \begin{cases} \frac{1}{2}kr^2 & |r| < a \\ \infty & |r| > a \end{cases}$$

for some $a > 0$, where m is the mass of the particle confined and $k = m\omega^2$ where ω is the harmonic oscillator frequency. Using separation of variables, and letting $\Psi(r, \theta, \phi) = R(r)\Theta(\theta, \phi)$, we are able to reduce an otherwise complicated coupled differential equation to an angular equation and a radial equation

$$-\frac{\hbar^2}{2m} \left(\frac{1}{r^2} \frac{d}{dr} \left(r^2 \frac{d}{dr} \right) - \frac{l(l+1)}{r^2} \right) R(r) + V(r) R(r) = ER(r), \quad (2.3)$$

the numerical solution to which will be the subject of this report. In three dimensions, we have the analytic solution

$$E_{nl} = \hbar\omega \left(2n + l + \frac{3}{2} \right) \quad (2.4)$$

for some $n, l \in \mathbb{N} \cup \{0\}$. We can further simplify this by letting $R(r) = \frac{1}{r}u(r)$ to get

$$-\frac{\hbar}{2m} \frac{d^2}{dr^2} u(r) + \left(V(r) + \frac{l(l+1)}{r^2} \frac{\hbar^2}{2m} \right) u(r) = Eu(r).$$

We further restrict our solutions to be physically significant. In particular, we require $u(0) = 0$ and $u(\infty) = 0$. We can then make our equation dimensionless by introducing $\rho = \frac{1}{\alpha}r$ for some constant α of dimension length. From there, we find

$$-\frac{\hbar^2}{2m\alpha^2} \frac{d}{d\rho^2} u(\rho) + \left(V(\rho) + \frac{l(l+1)}{\rho^2} \frac{\hbar^2}{2m} \right) = Eu(\rho)$$

In this project, we let $l = 0$, and so, substituting our value for $V(\rho)$ from Eq. 2.2, we see the Schrodinger equation becomes

$$-\frac{\hbar^2}{2m\alpha^2} \frac{d^2}{d\rho^2} u(\rho) + \frac{k}{2} \alpha^2 \rho^2 u(\rho) = Eu(\rho).$$

Up until this point, we have left α arbitrary, but now we choose it in such a way as to make the calculations simpler. In particular, we let

$$\alpha = \left(\frac{\hbar^2}{mk} \right)^{1/4} \quad (2.5)$$

We can see that α still has dimensions of length, as required. In particular, we will notice that α defines the length scale of the problem (eg. α will be the Bohr radius for problems on the nuclear scale). Then, letting

$$\lambda = \frac{2m\alpha^2}{\hbar^2} E,$$

we can rewrite Schrodinger's Equation as [4]

$$-\frac{d^2}{d\rho^2} u(\rho) + \rho^2 u(\rho) = \lambda u(\rho). \quad (2.6)$$

It was shown in Project 1 [2] that we can set up a tridiagonal matrix in order to compute a numerical approximation to a second derivative. In particular, we can define a $\rho_{min} = 0$ and a ρ_{max} and then, for some number of steps n_{step} , calculate a step size

$$h = \frac{\rho_{max} - \rho_{min}}{n_{step}}.$$

We then have $\rho_i = \rho_{min} + ih$ for $i = 0, 1, \dots, n_{step}$, and we can write the Schrodinger equation as

$$-\frac{u(\rho_i + h) - 2u(\rho_i) + u(\rho_i - h)}{h^2} + V_i = \lambda u(\rho_i) \quad (2.7)$$

for $V_i = \rho_i^2$. Thus, we have a tridiagonal matrix with diagonal elements

$$d_i = \frac{2}{h^2} + V_i$$

and off-diagonal elements

$$e_i = -\frac{1}{h^2}.$$

However, unlike in project 1, we are not solving a system of linear equations in this case, but rather an eigenvalue problem. Solving this eigenvalue problem will be accomplished with the help of Jacobi's rotation algorithm [4], which is outlined in Section 2.1. The reason for this is that we can rewrite the Schrodinger equation now in terms of the d_i and e_i as

$$d_i u_i + e_{i-1} u_{i-1} + e_{i+1} u_{i+1} = \lambda u_i.$$

Finding the eigenvalues to this problem will give us the various energy levels. From there, we can find the eigenvectors, which translate to the wave functions.

In particular, we are interested in two cases of electrons in a harmonic oscillator well. The first is the case of a single electron, where the radial Schrodinger Equation is

$$-\frac{\hbar^2}{2m} \frac{d^2}{dr^2} u(r) + \frac{1}{2} k r^2 u(r) = E^{(1)} u(r) \quad (2.8)$$

as before and the second is the case of two interacting electrons. To solve this case, we look first at the non-interacting situation, in which case the Schrodinger Equation reads

$$\left(-\frac{\hbar^2}{2m} \frac{d^2}{dr_1^2} - \frac{\hbar^2}{2m} \frac{d^2}{dr_2^2} + \frac{1}{2} k r_1^2 + \frac{1}{2} k r_2^2 \right) u(r_1, r_2) = E^{(2)} u(r_1, r_2). \quad (2.9)$$

To solve this case, we introduce the relative coordinate $r = r_1 - r_2$ and the center of mass coordinate $R = 1/2(r_1 + r_2)$. In terms of these coordinates, the Schrodinger equation reads

$$\left(-\frac{\hbar^2}{m} \frac{d^2}{dr^2} - \frac{\hbar^2}{4m} \frac{d^2}{dR^2} + \frac{1}{4} k r^2 + k R^2 \right) u(r, R) = E^{(2)} u(r, R). \quad (2.10)$$

We solve this again by separation of variables, letting $u(r, R) = \psi(r) \phi(R)$.

Next, we can introduce the repulsive Coulomb interaction between the two electrons,

$$V(r_1, r_2) = \frac{\beta e^2}{|r_1 - r_2|} = \frac{\beta e^2}{r} \quad (2.11)$$

where $\beta e^2 = 1.44 \text{ eVnm}$. Now the r -dependent Schrodinger equation becomes

$$\left(-\frac{\hbar^2}{m} \frac{d^2}{dr^2} + \frac{1}{4} k r^2 + \frac{\beta e^2}{r} \right) \psi(r) = E_r \psi(r).$$

As before, we introduce a dimensionless variable $\rho = r/\alpha$ to get

$$-\frac{d^2}{d\rho^2} \phi(\rho) + \frac{1}{4} \frac{mk}{\hbar^2} \alpha^4 \rho^2 \psi(\rho) + \frac{m\alpha\beta e^2}{\rho\hbar^2} \phi(\rho) = \frac{m\alpha^2}{\hbar^2} E_r \psi(\rho).$$

Next, we can define a frequency ω_r which defines the strength of the oscillator potential.

$$\omega_r^2 = \frac{1}{4} \frac{mk}{\hbar^2} \alpha^2$$

and fix

$$\alpha = \frac{\hbar^2}{m\beta e^2}.$$

Thus, the Schrodinger equation becomes [4]

$$-\frac{d^2}{d\rho^2}\psi(\rho) + \omega_r^2 \rho^2 \psi(\rho) + \frac{1}{\rho} = \lambda \psi(\rho).$$

Throughout this project, we will be studying this potential at the ground level and the first two excited states, plotting the wave functions as a function of r and for various values of ω_r . The results are discussed in Section 4.

2.1 The Jacobi Rotation Algorithm

For a real, symmetric matrix A , with eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_n$, we can create a diagonal matrix D such that

$$D = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \ddots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix}$$

by applying a particular real, orthogonal matrix S . That is, there exists an S such that $S^T A S = D$. The trick to the Jacobi Rotation Algorithm is to find this S in order to reduce A to its diagonal form. In particular, the Jacobi Algorithm looks for an S of the form

$$S = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \ddots & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & 1 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & 0 & \cdots & 0 & \cos(\theta) & 0 & \cdots & 0 & \sin(\theta) \\ \vdots & 0 & \cdots & 0 & 0 & 1 & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & -\sin(\theta) & 0 & \cdots & 0 & \cos(\theta) \end{pmatrix}$$

for some θ . And so of course the algorithm really boils down to finding the θ which will diagonalize A , or at least make the off-diagonal elements of A as small as possible [4]

But how do we define "as small as possible"? In particular, we compute the Frobenius norm of the off-diagonal elements and require it be less than some threshold ϵ . The Frobenius norm is defined to be

$$\|A\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^n |a_{ij}|^2}. \quad (2.12)$$

And so we require

$$\text{off}(\|A\|_F) = \sqrt{\sum_{i=1}^n \sum_{j=1, i \neq j}^n |a_{ij}|^2} < \epsilon. \quad (2.13)$$

If this norm is greater than ϵ , then we find the maximum off-diagonal element, a_{lk} and compute

$$\tau = \frac{a_{kk} - a_{ll}}{2a_{lk}}$$

and

$$\tan(\theta) = -\tau \pm \sqrt{1 + \tau^2}.$$

This will clearly yield two possible values for $\tan(\theta)$, and we choose the smallest one to continue. From there, we can calculate

$$\cos(\theta) = \frac{1}{\sqrt{1 + \tan^2(\theta)}}$$

and

$$\sin(\theta) = \tan(\theta) \cos(\theta).$$

With these values of $\sin(\theta)$ and $\cos(\theta)$, we can calculate the rotation matrix S and effectively set the largest off-diagonal element we started with, a_{lk} , to 0. We continue this process until $\|A\|_F < \epsilon$ as desired [4].

3 The Algorithm

The file for this project is `project2.C`. It contains three main functions: `partb()`, `partc()`, and `solve1e()`. We begin with a discussion of `partb()`. As shown in Section 2, for the single-electron case, we are essentially trying to solve the equation

$$-\frac{d^2u}{d\rho^2} + \rho^2u = \lambda u,$$

which is an eigenvalue problem with eigenvalue λ . Discretizing this, we see that we can let

$$\begin{aligned} u &\rightarrow u_i \rho_i \\ \rho &\rightarrow \rho_i = \rho_0 + ih \\ h &\rightarrow \frac{b-a}{n+1} \\ -\frac{d^2u}{d\rho^2} &\rightarrow \frac{u_{i+1} + u_{i-1} - 2u_i}{h^2} \end{aligned}$$

to get

$$-(u_{i+1} + u_{i-1} - 2u_i) + \rho_i^2 u_i h^2 = h^2 \lambda u_i. \quad (3.1)$$

We do this by using the `themat` and `thevec` classes developed in Project 1 [2]. We begin by defining the matrix

$$\begin{pmatrix} \frac{2}{h^2} + V_1 & -\frac{1}{h^2} & 0 & 0 & \cdots & 0 & 0 \\ -\frac{1}{h^2} & \frac{2}{h^2} + V_2 & -\frac{1}{h^2} & 0 & \cdots & 0 & 0 \\ 0 & -\frac{1}{h^2} & \frac{2}{h^2} + V_3 & -\frac{1}{h^2} & 0 & \cdots & 0 \\ \cdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \frac{2}{h^2} + V_{n_{step}-2} & -\frac{1}{h^2} \\ 0 & \cdots & \cdots & \cdots & \cdots & -\frac{1}{h^2} & \frac{2}{h^2} + V_{n_{step}-1} \end{pmatrix}$$

where

$$V_i = \rho_i^2.$$

We then send the matrix through the Jacobi Algorithm, `Jacobi_Method` (which is overloaded in order to count and graph the number of rotations required to diagonalize). `Jacobi_Method` runs another function, `Jacobi_Method_step`, over the matrix several times until it is diagonalized. The `Jacobi_Method_step` begins by determining the maximum element in the matrix and, if this is less than some threshold `eps`, calculates `tau`, `tan`, `cos`, and `sin` as follows:

```
double tau = (mat[col][col]-mat[row][row])/(2*mat[row][col]);
double tan = 0;
if(tau >= 0)
    tan = 1.0/(tau+sqrt(1 + tau*tau));
else
    tan = -1/(-tau+sqrt(1+tau*tau));
double cos = 1/sqrt(1+tan*tan);
double sin = tan*cos;
```

With these variables, it is possible to define the similarity transformation matrix `S` which can zero out (or essentially zero out) the maximal off-diagonal element. `S` is calculated as follows:

```
themat S = themat(n);
for(int i=0;i<n;i++){
    for(int j=i;j<n;j++){
        if((i==row && j==row) || (i==col && j==col))
            S[i][j]=cos;
        else if((i==row && j==col) || (i==col && j==row))
            S[i][j]=-sin;
        else if(i==j)
            S[i][j]=1;
        else
            S[i][j]=0;
    }
}
```

```
for(int i=0;i<n;i++){
    for(int j=0;j<i;j++){
```

```

        if(S[i][j]==0)
            S[i][j]=S[j][i];
        else
            S[i][j]=-S[j][i];
    }
}

```

From there, using the `transpose()` function developed for this project, we calculate $B = S \cdot \text{mat} \cdot S.\text{transpose}()$, zeroing out the maximal off-diagonal elements of `mat`, and feed it back into the function until the B returned is fully diagonal. Notice here that, in our calculation of `tan`, we chose the smallest value according to the quadratic formula. We do this in order to ensure that the θ corresponding to this `tan` satisfies $|\theta| < \frac{\pi}{4}$.

Once the Jacobi Algorithm has been run, the matrix has been diagonalized and its eigenvalues can be read off from its elements. In particular, we expect that these eigenvalues will converge, with increase numbers of steps, to the discretized energy values of the system. This was investigated, and, indeed, it was shown that the eigenvalues converge more quickly for larger values of ρ_{max} . This is reasonable because, in fact, we would expect that these would be the energy values as $\rho \rightarrow \infty$.

As discussed in Section 3.1, the Householder Algorithm from `lib.cpp` [3] is far more efficient than the Jacobi Algorithm. For this reason, the 2-electron problem is solved using `tqli` from that file. In particular, the 2-electron problem is solved by function `partc()` in `project2.C`. This function solves and plots the interacting and non-interacting 2-electron problem for various ρ_{max} , number of steps, and ω_r . The code is much the same as for the `partb()`, except the potential for the non-interacting case is

$$V_i = (\omega_r \rho_i)^2$$

and for the interacting case is

$$V_i = (\omega_r \rho_i)^2 + \frac{1}{\rho_i}.$$

Vectors are defined for the interacting and non-interacting cases for the diagonal elements of the matrices (`ds`) and for the off-diagonal elements (`es`), and identity matrices (`z`) are created as well for each case to be fed into the `tqli()` function, which gives the `ds` vectors the eigenvalues and the `z` matrices the corresponding eigenvectors. These eigenvectors are then plotted for the lowest eigenvalue (ie. we are looking at the lowest/ground state energy) as well as for the first two excited states. The plots are discussed in Section 4, along with the plots created by `solve1e()`, which creates plots in a similar way as `partc()`, but for the single electron case.

3.1 Time Dependence

There are several steps involved with the algorithm described in Section 3, each of which has its own time-dependence. We start with a discussion of the Jacobi Rotation Algorithm. Each step of this algorithm is designed to zero-out one off-diagonal element of the matrix A , and so one might expect that there would be $n^2 - n$ rotations required to diagonalize

A. However, what we find is that any given rotation might force a previously-zeroed non-diagonal element to become non-zero. And so in fact there are typically $3n^2 - 5n^2$ rotations required to fully diagonalize A , and each rotation requires $4n$ operations. This results in a total of $12n^3 - 20n^3$ operations to diagonalize A [4].

Because the matrix required to solve this problem is tridiagonal, it is appropriate to compare the computing time of the Jacobi Algorithm to that of the Householder Algorithm, provided in `lib.cpp` [3]. A short study was made into the time required for each algorithm, and it was found that the Jacobi Algorithm took a good deal more time to run than Householder's Algorithm. In particular, it is clear from Figure 1 that Jacobi's Algorithm does indeed go as the number of steps cubed.

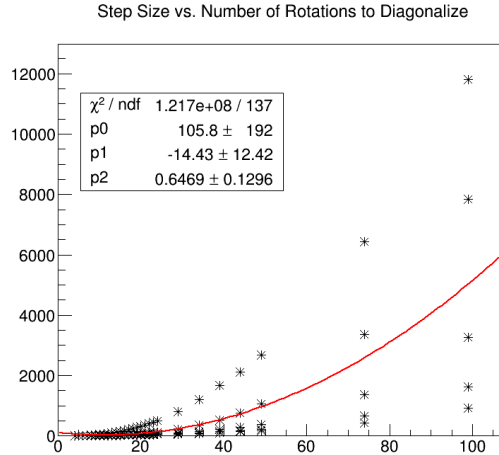


Figure 1: The number of rotations required by the Jacobi algorithm to diagonalize matrices of a given dimension.

4 Results and Benchmarks

We begin by discussing the solution to the single-electron problem. The ground state solutions are shown in Fig. 2, the 1st excited state in Fig 3, and the 2nd excited state in Fig 4. We see in all cases of that the eigenvalues (energy) do indeed converge to about what we would expect (the first three ground state energies should be 3, 7, and 11) [4], even over very few numbers of steps. We also see that, as expected, the distribution falls off quickly for larger radii. In addition, we see nodes forming in the excited state distributions, as we expect.

The analysis of the 2-electron problem is more involved, as we have multiple values of ω_r , interacting and non-interacting cases, and multiple values of ρ_{max} . These plots are shown in Figures 5-19. As with the single electron, we see that, in all variations, the eigenvalues converge to some finite quantity with increasing numbers of steps. We also see nodes appearing for excited states, and that the distribution drops off quickly for larger values of ρ_{max} . It appears that larger ω_r cause the distribution to fluctuate with greater frequency and become

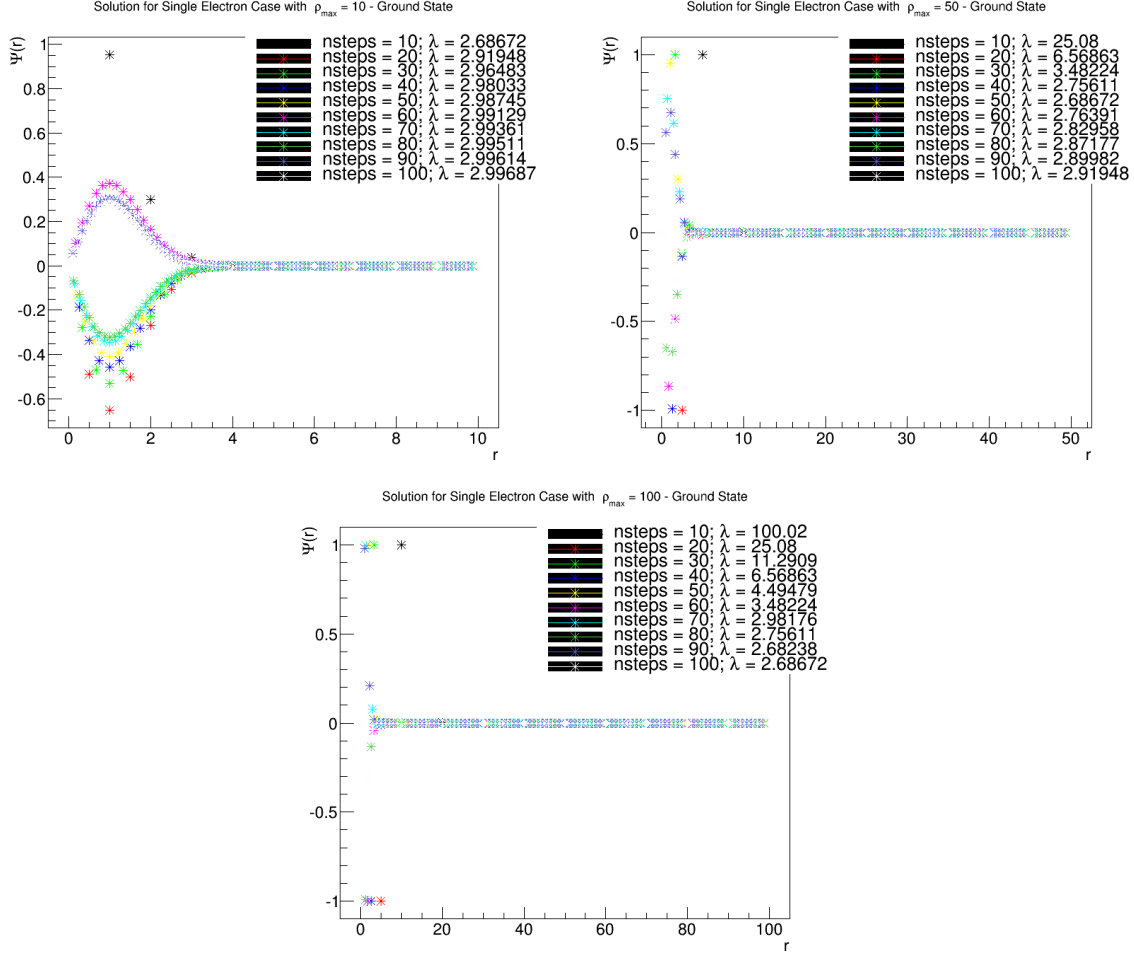


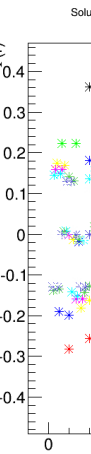
Figure 2: Solutions for the single electron case for various values of ρ_{max} in the ground state.

less stable near the origin. We also notice that the interacting potential peaks farther from the origin than the non-interacting potential. This makes sense because the electrons would try to be as far from each other within the well as possible, which will likely imply that they are farther from the origin than the non-interacting electrons. The plots for $\omega_r = 1.0$ in the non-interacting regime should be very similar to those from the single electron system.

5 Conclusions

In the end, a code was developed which was able to successfully reproduce anticipated physics results for the situations of one and two electrons confined to a spherical harmonic oscillator quantum well. Plots were made which demonstrated the state solutions, and it was shown that the energy eigenvalues converged to the anticipated discretized energy levels. We were able to use both Householder's Algorithm and Jacobi's Rotation Algorithm in this solution, and determined that the Householder Algorithm was a more efficient tridiagonal solver.

However, there were several issues with the code in the end, particularly with the analysis of the time dependence of the two algorithms. Several plots did not come out in the anticipated way, indicating that there is a fairly significant bug in `partb()`. This does not impact the physics shown or the solutions calculated, but nonetheless it hindered a good description of the coding and the motivation for choosing Householder's Algorithm over Jacobi's.



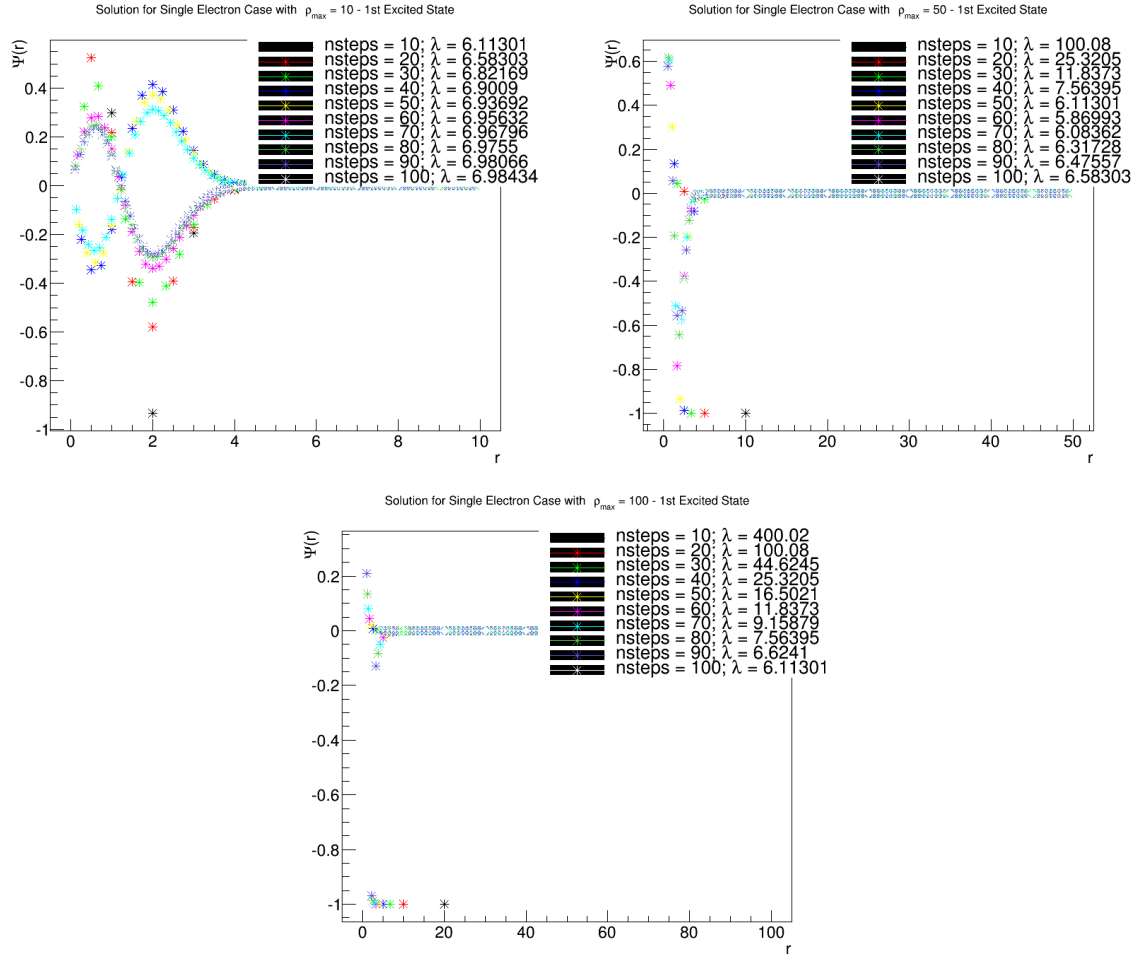


Figure 3: Solutions for the single electron case for various values of ρ_{\max} in the 1st excited state.

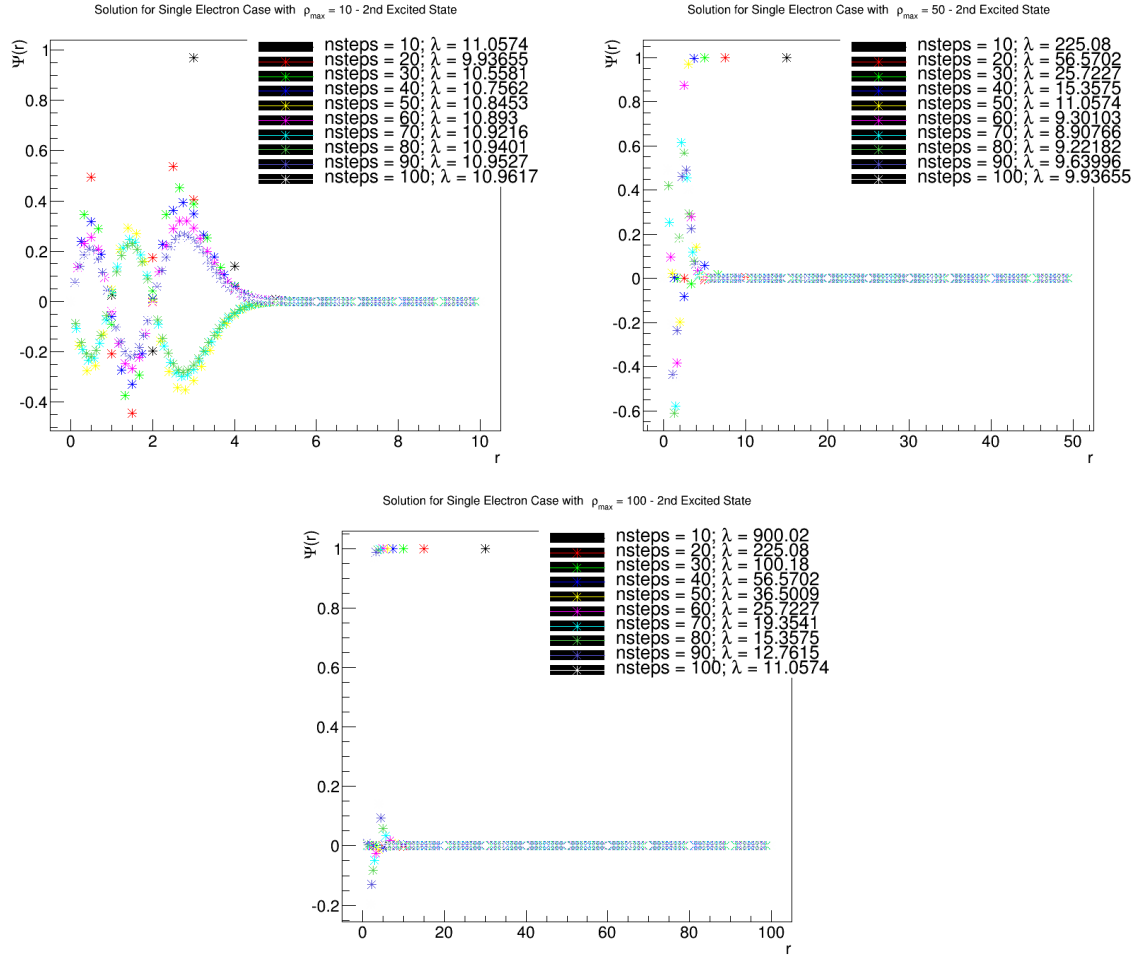


Figure 4: Solutions for the single electron case for various values of ρ_{max} in the 2nd excited state.

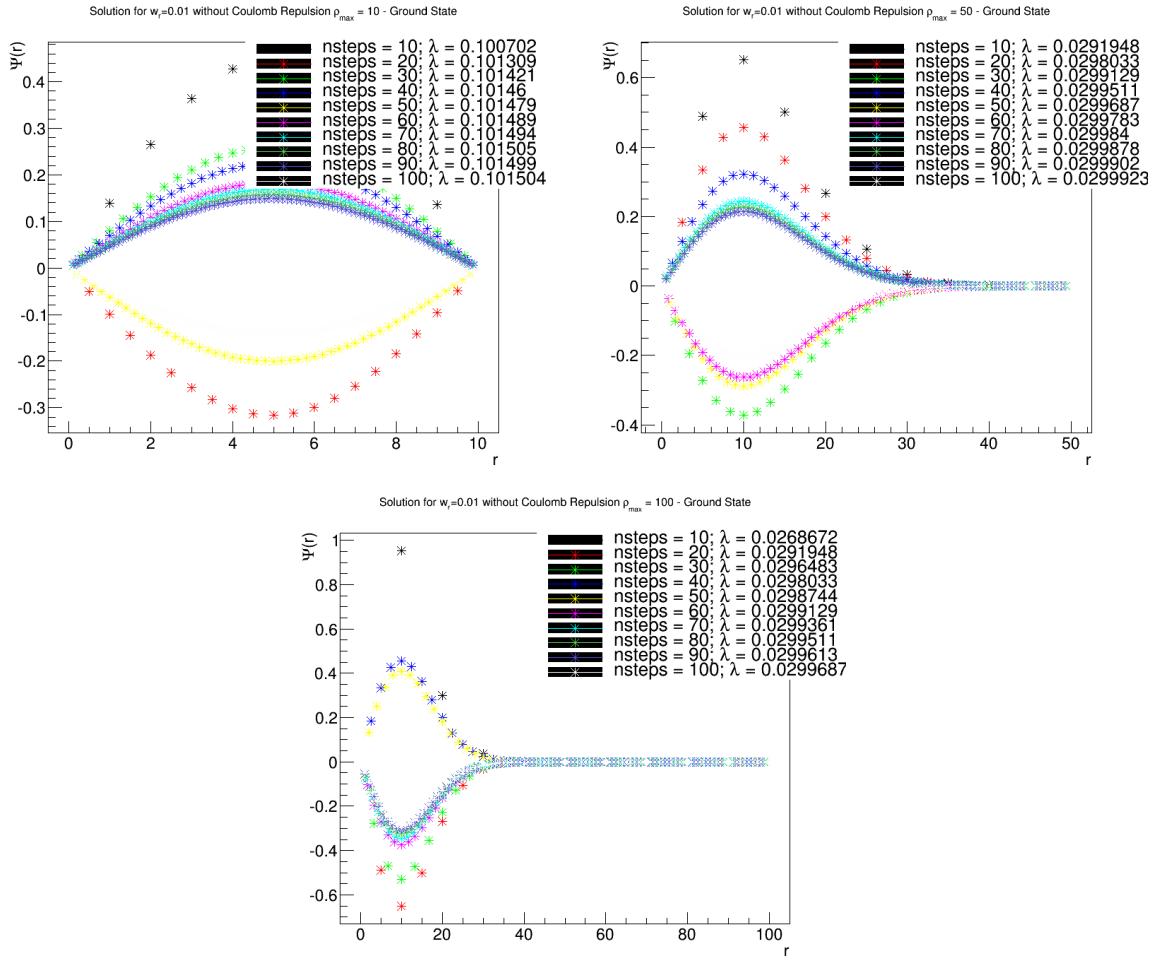


Figure 5: Solutions for the non-interacting two-electron system for various values of ρ_{\max} in the ground state for $\omega_r = 0.01$.

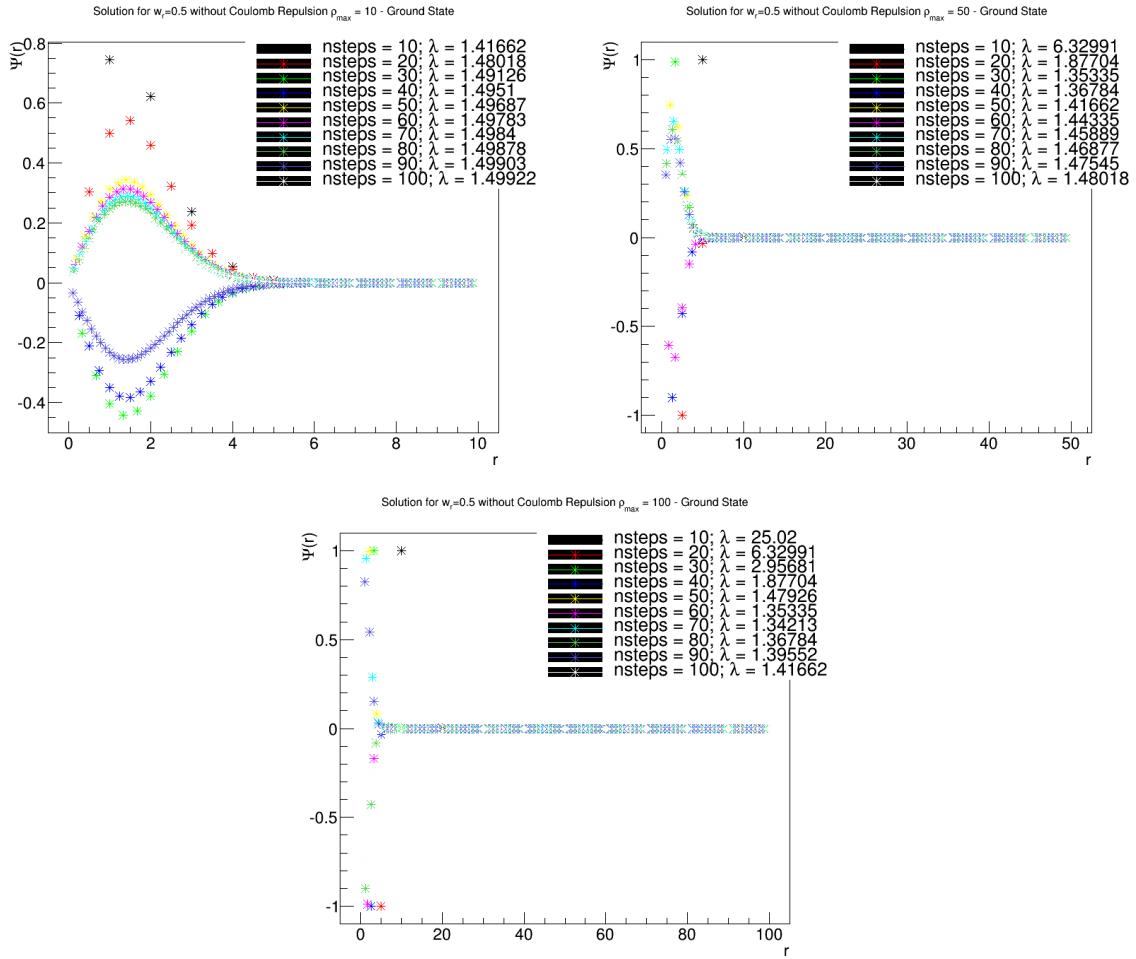


Figure 6: Solutions for the non-interacting two-electron system for various values of ρ_{\max} in the ground state for $\omega_r = 0.5$.

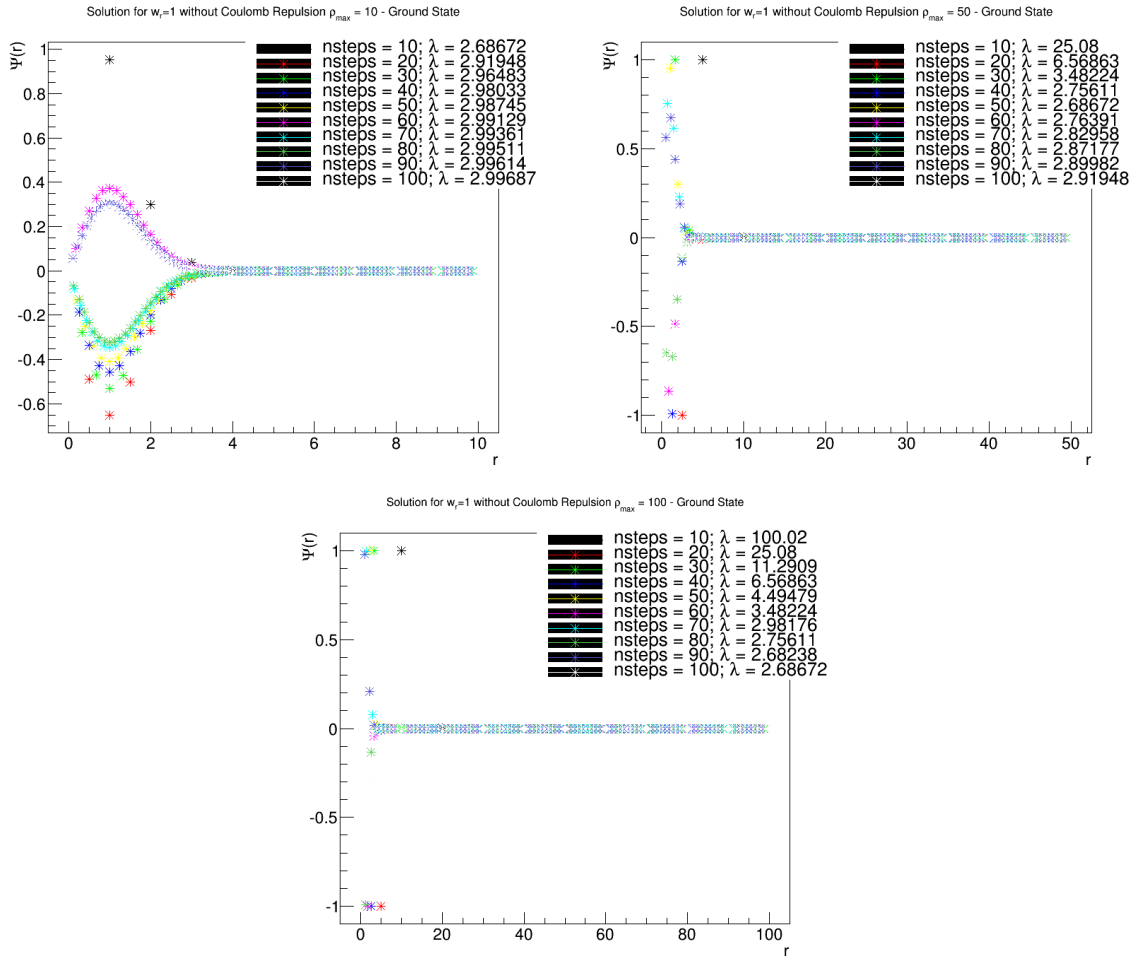


Figure 7: Solutions for the non-interacting two-electron system for various values of ρ_{\max} in the ground state for $\omega_r = 1.0$.

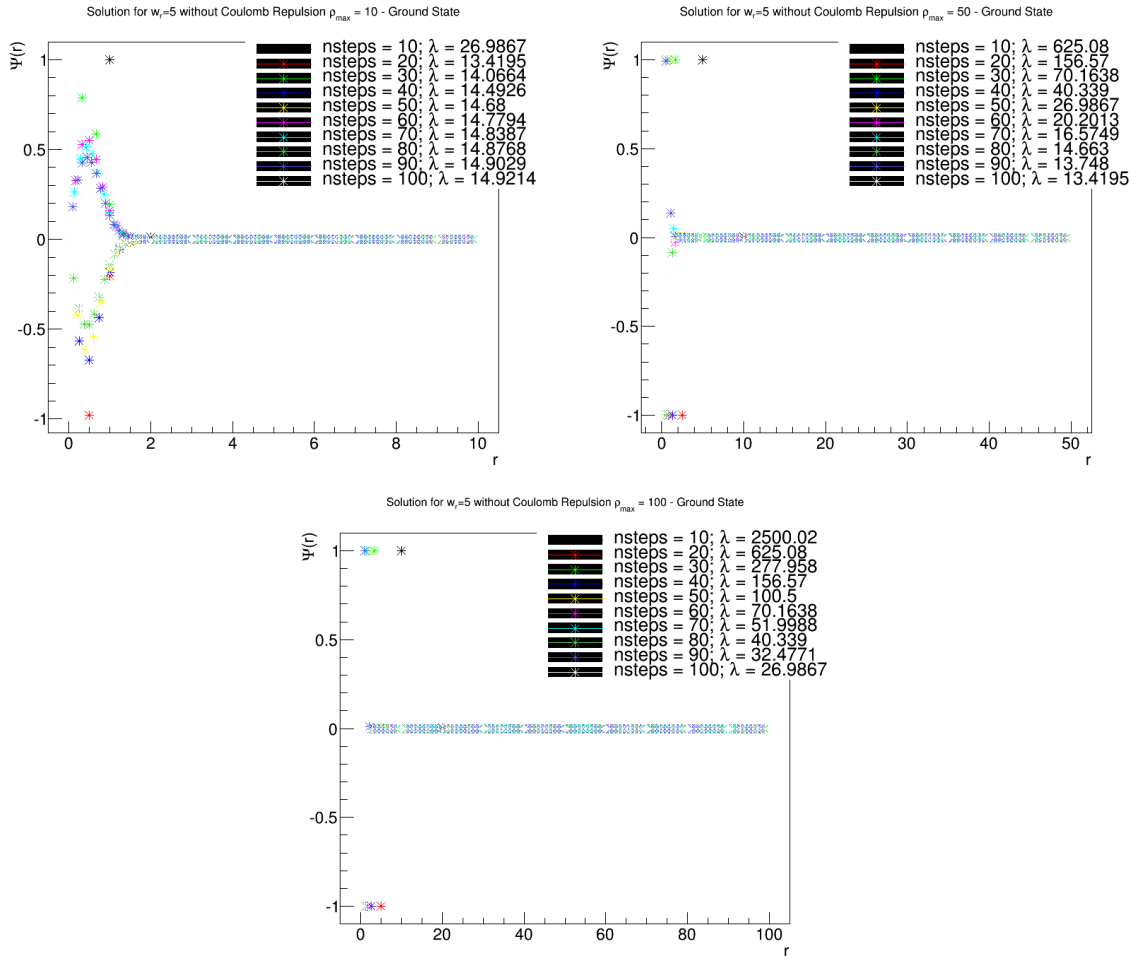


Figure 8: Solutions for the non-interacting two-electron system for various values of ρ_{\max} in the ground state for $\omega_r = 5.0$.

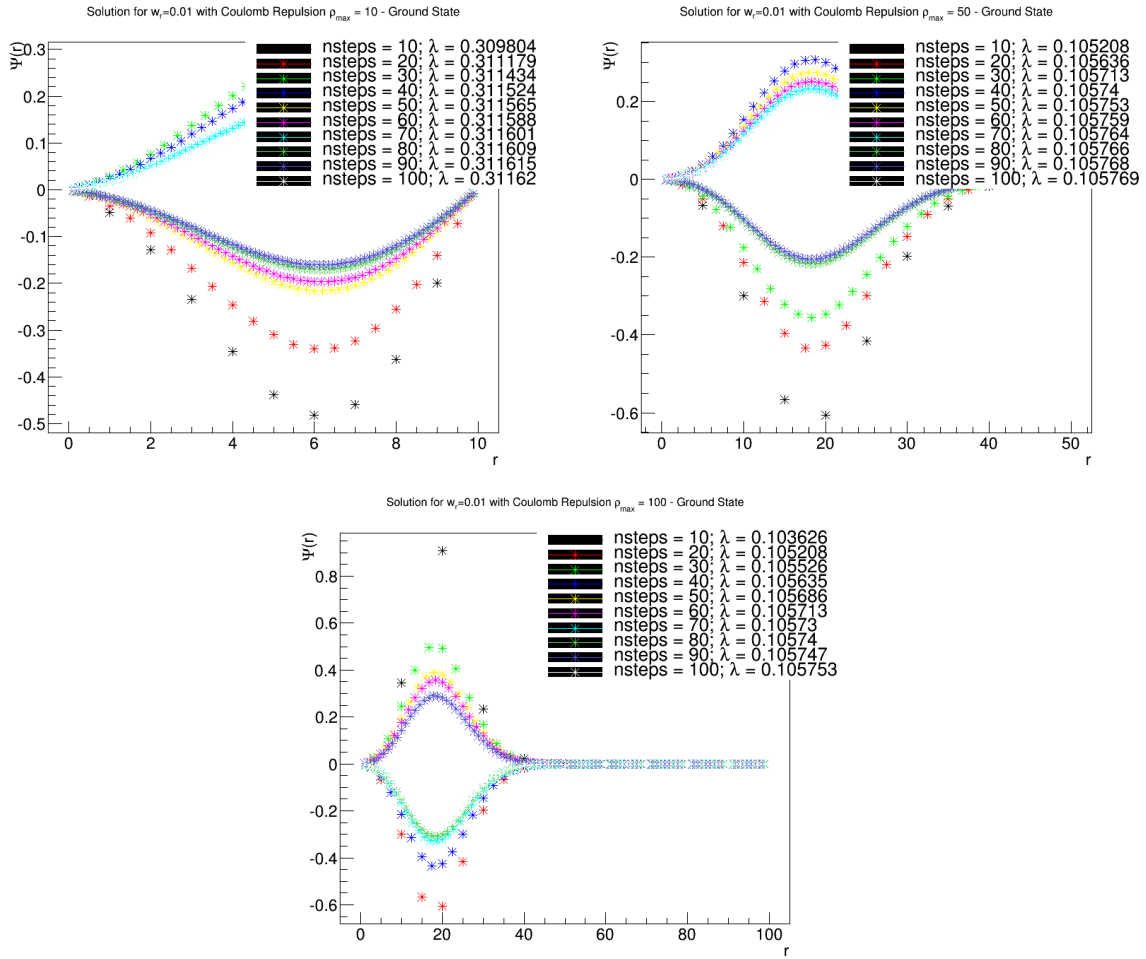


Figure 9: Solutions for the interacting two-electron system for various values of ρ_{\max} in the ground state for $\omega_r = 0.01$.

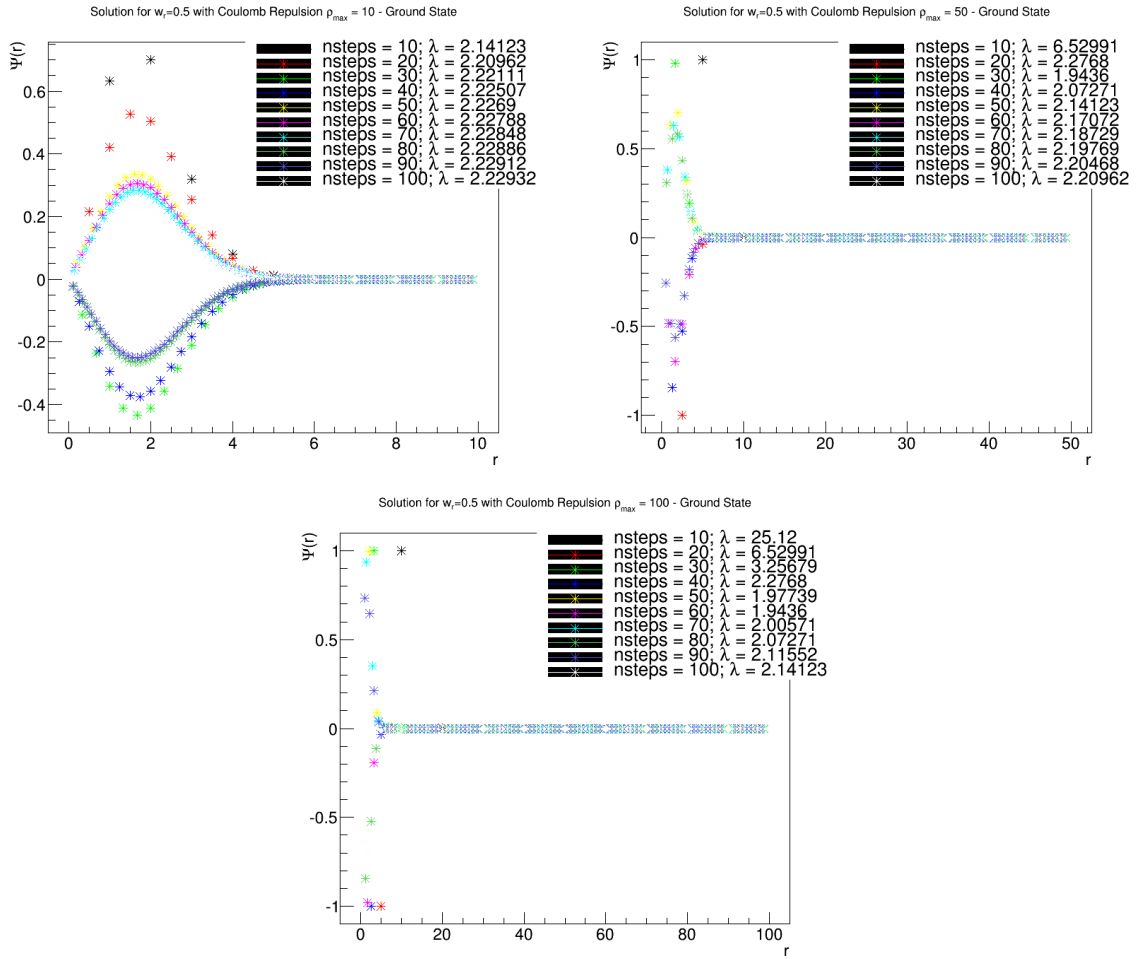


Figure 10: Solutions for the interacting two-electron system for various values of ρ_{max} in the ground state for $\omega_r = 0.5$.

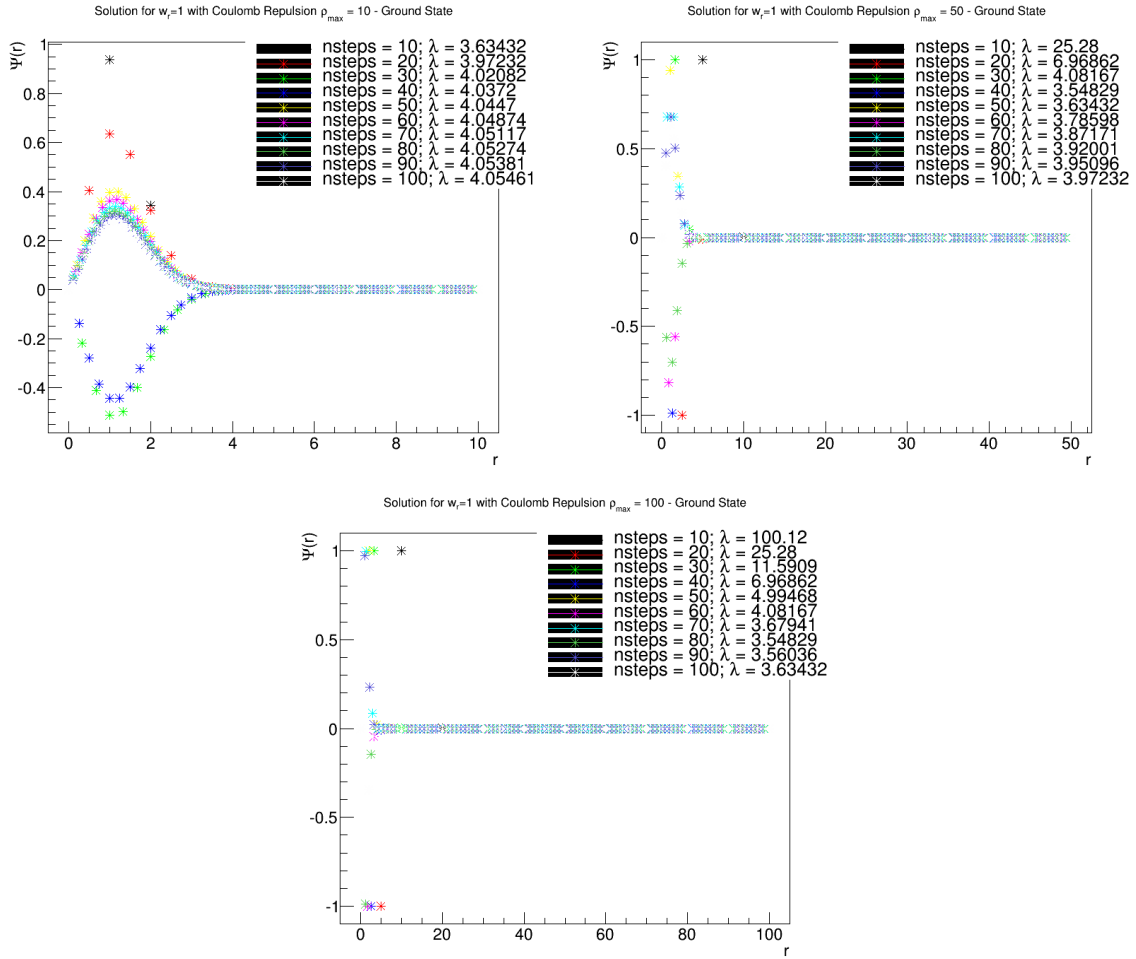


Figure 11: Solutions for the interacting two-electron system for various values of ρ_{\max} in the ground state for $\omega_r = 1.0$.

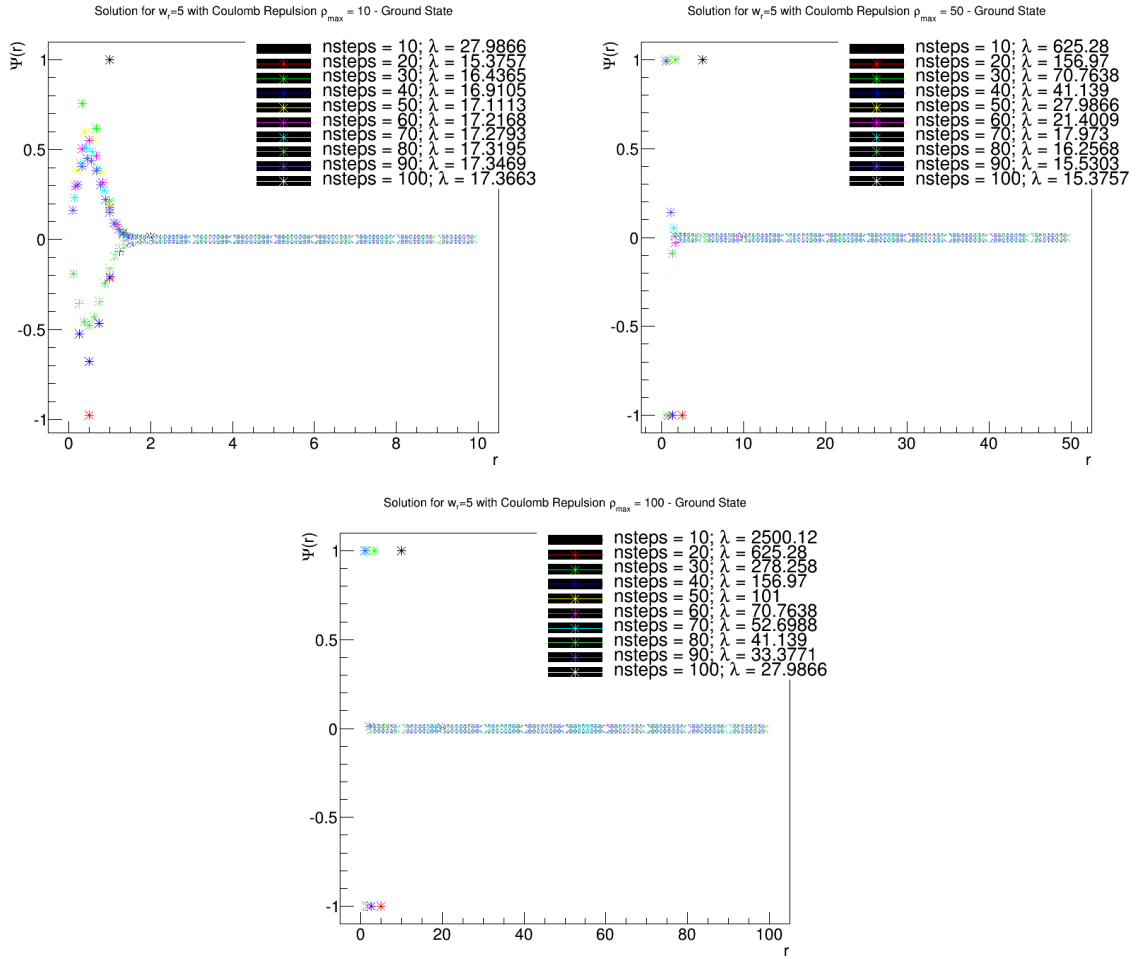


Figure 12: Solutions for the interacting two-electron system for various values of ρ_{\max} in the ground state for $\omega_r = 5.0$.

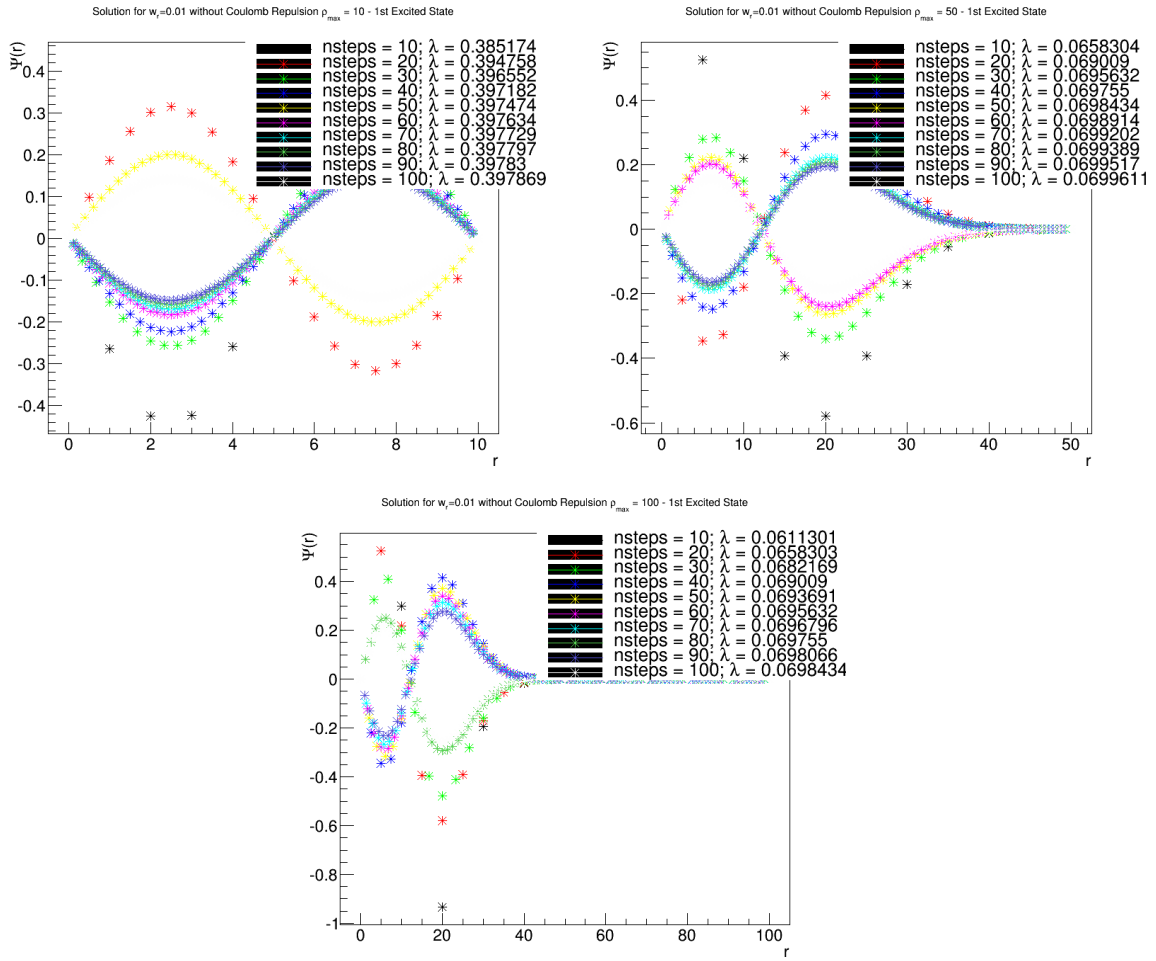


Figure 13: Solutions for the non-interacting two-electron system for various values of ρ_{\max} in the 1st excited state for $\omega_r = 0.01$.

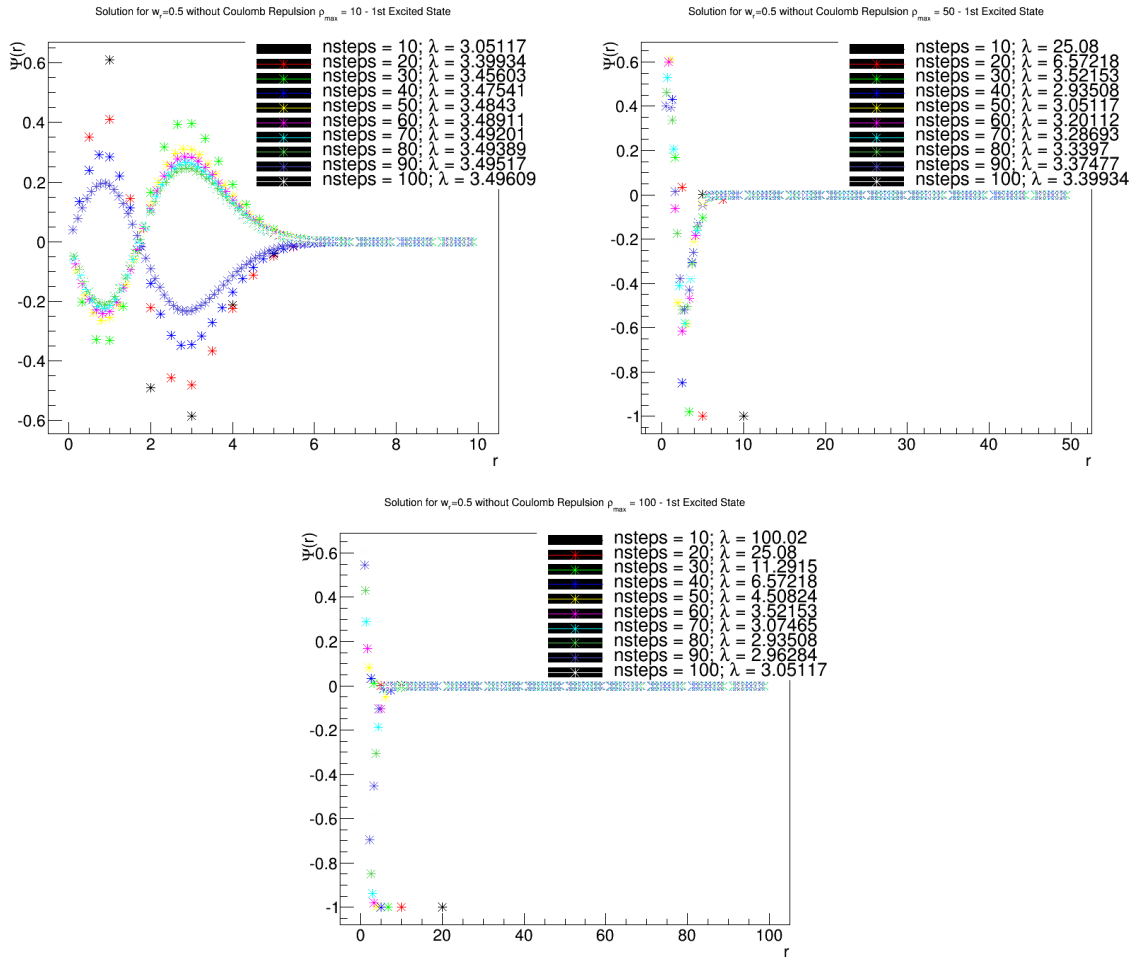


Figure 14: Solutions for the non-interacting two-electron system for various values of ρ_{\max} in the 1st excited state for $\omega_r = 0.5$.

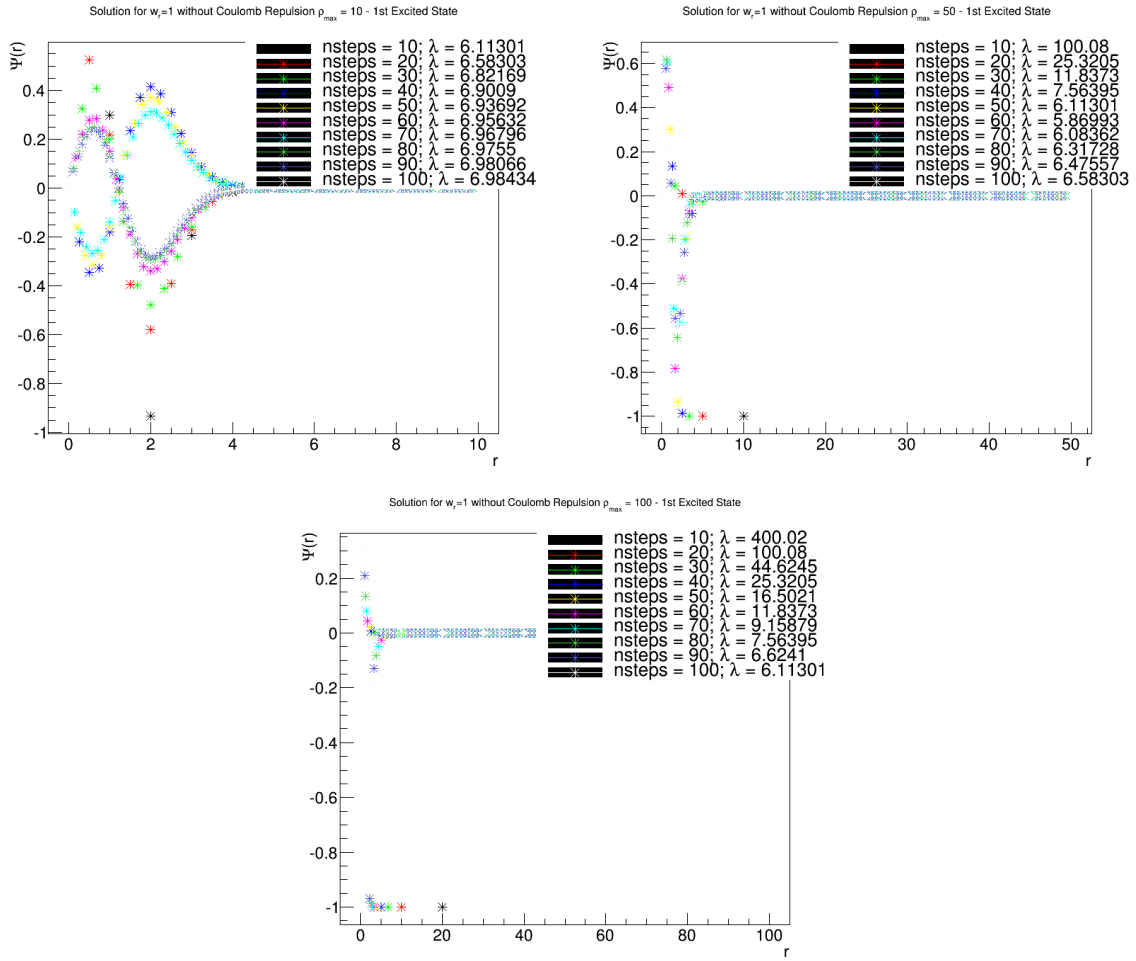


Figure 15: Solutions for the non-interacting two-electron system for various values of ρ_{\max} in the 1st excited state for $\omega_r = 1.0$.

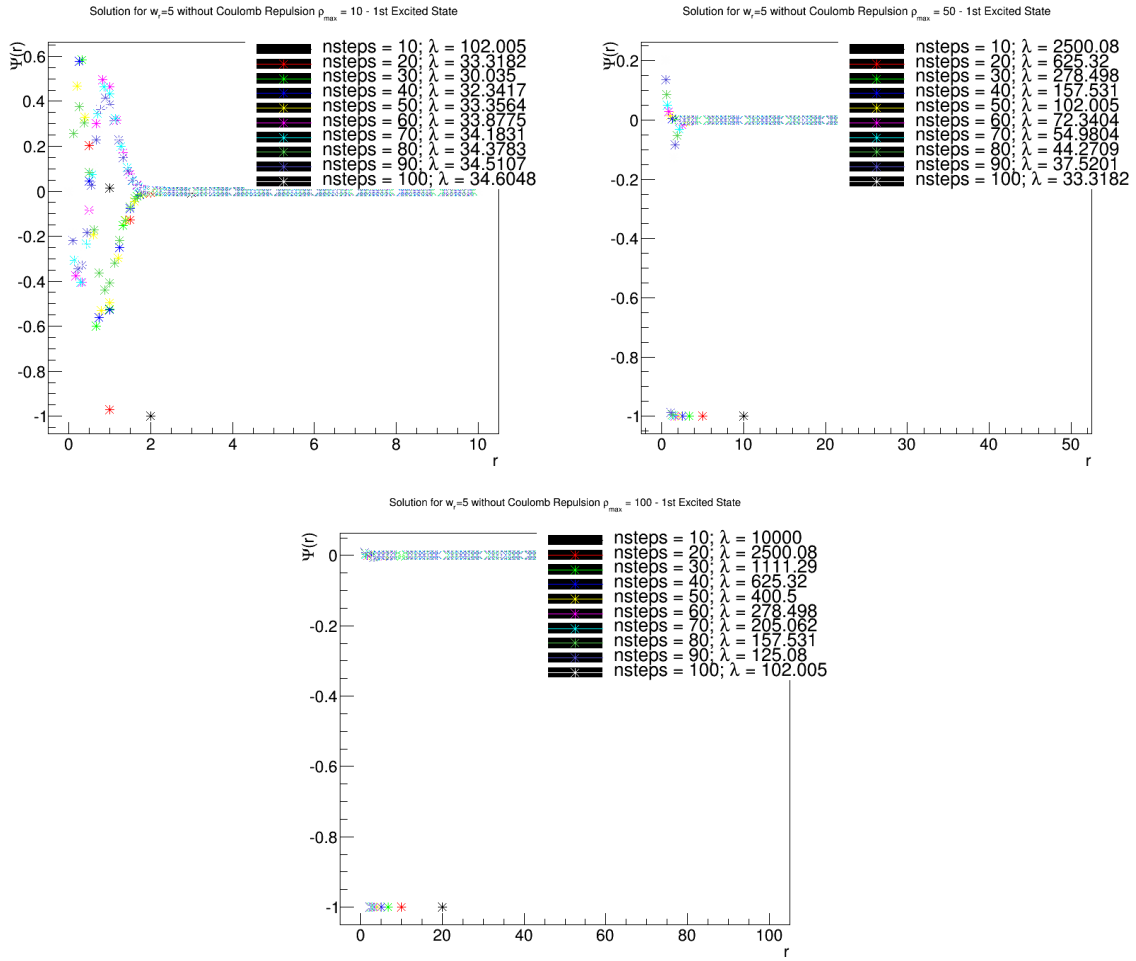


Figure 16: Solutions for the non-interacting two-electron system for various values of ρ_{\max} in the 1st excited state for $\omega_r = 5.0$.

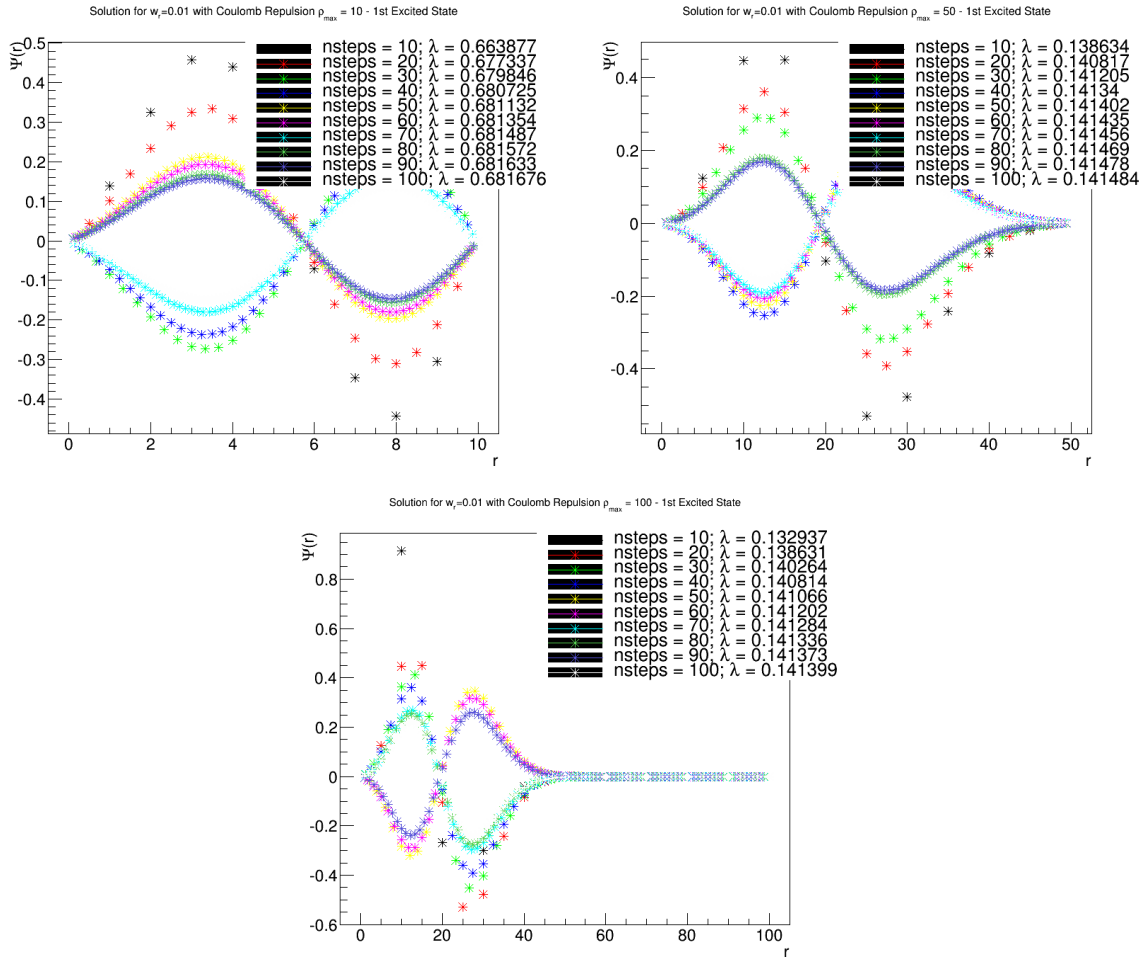


Figure 17: Solutions for the interacting two-electron system for various values of ρ_{\max} in the 1st excited state for $\omega_r = 0.01$.

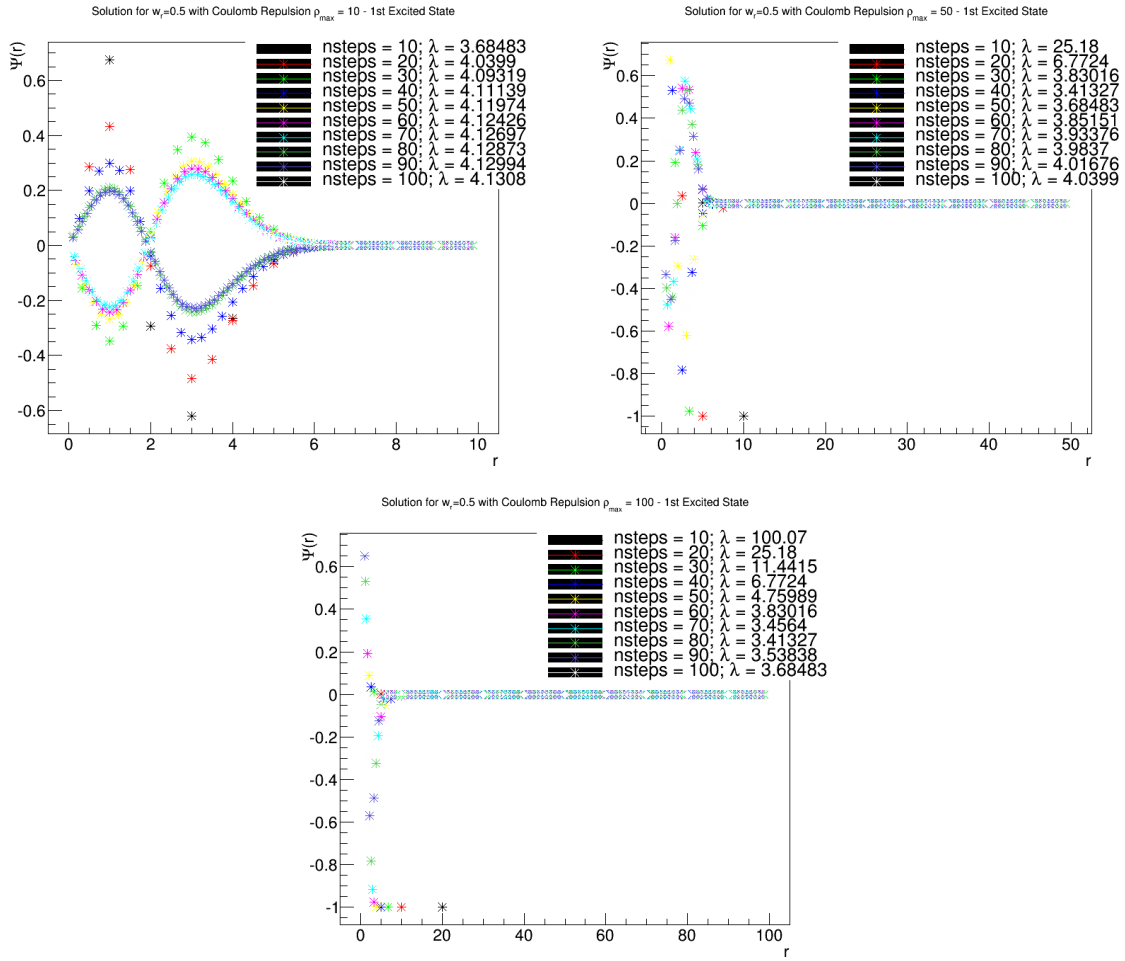


Figure 18: Solutions for the interacting two-electron system for various values of ρ_{\max} in the 1st excited state for $\omega_r = 0.5$.

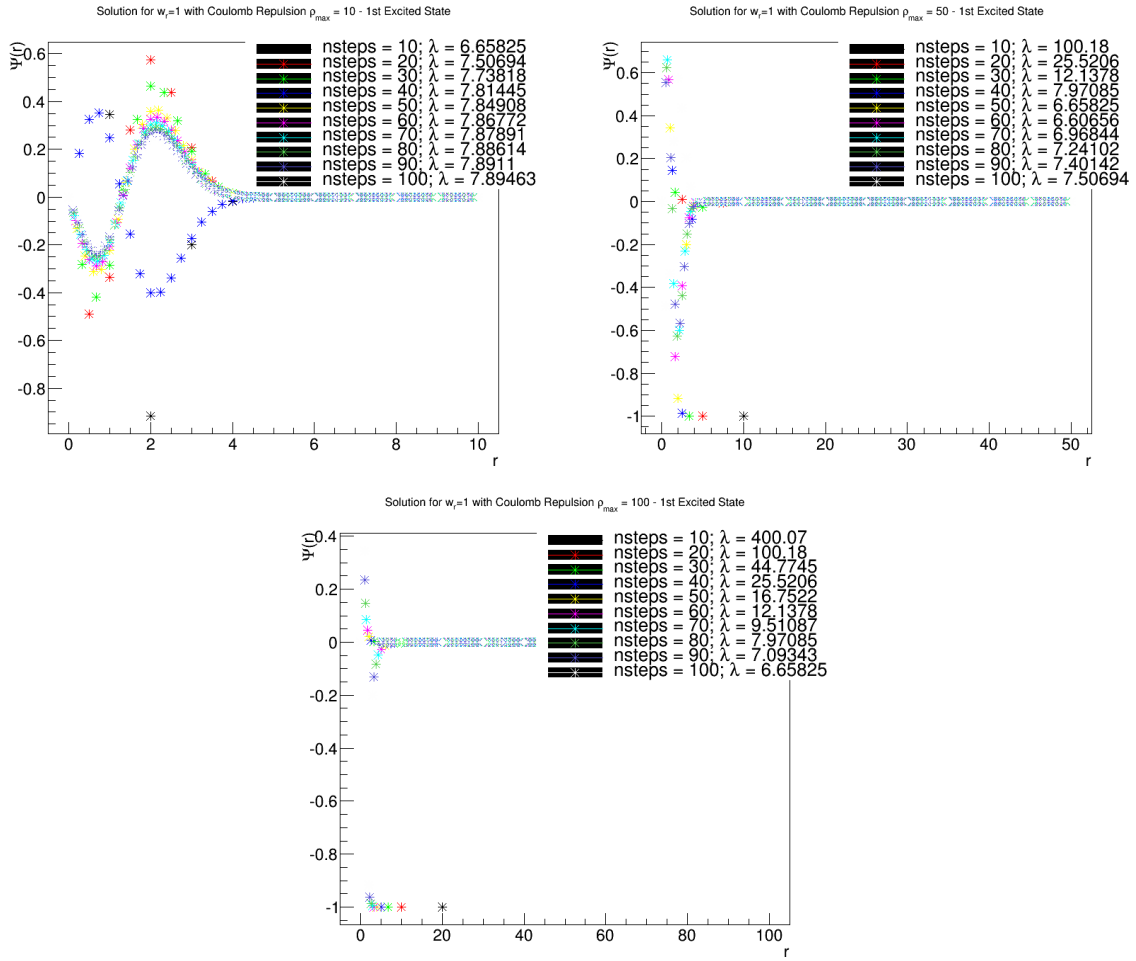


Figure 19: Solutions for the interacting two-electron system for various values of ρ_{max} in the 1st excited state for $\omega_r = 1.0$.