EN5004

ELECTRONIC DEVICES

# SIMULATION OF THE TIME INDEPENDENT SCHRÖDINGER'S EQUATION

208643N

K. G. RANASINGHE

PG Dip. / M.Sc. IN ELECTRONICS & AUTOMATION

UNIVERSITY OF MORATUWA

SRI LANKA

DEPARTMENT OF ELECTRONIC & TELECOMMUNICATION ENGINEERING

May 30, 2021

# Contents

# Abstract

The analytical solution of the Schrödinger’s Equation to obtain closed form solutions can be quite challenging due to complex nature of the potential fields a particle being investigated interacts with. This problem is circumvented by using methods of numerical computation that closely approximate this complex behavior. This study attempts to simulate the one-dimensional Schrödinger’s Equation and obtain solutions for a variety of potential functions using the finite difference method (FDM). The obtained solutions are observed for any divergence from expected behavior.

**Keywords – Time Independent Schrödinger’s Equation (TISE), Time Dependent Schrödinger’s Equation (TDSE), Finite Difference Method (FDM)**

# 1. Introduction

## 1.1 Dependance and Independence of Time

The one-dimensional Schrödinger’s wave equation in its full form can be written as,

$$-\frac{\hbar^2}{2m}\frac{\partial^2\psi(x,t)}{\partial x^2}+V(x)\,\psi(x,t)=j\hbar\,\frac{\partial\psi(x,t)}{\partial t} \tag{1}$$

Where $\psi(x,t)$, $V(x)$, represents the wave function and potential function respectively and $\hbar, m$, are the modified Planck constant and mass of the particle.

The Schrödinger’s wave equation can be broken down into two forms: the time dependent portion, and the time independent or position dependent portion, using the separation of variables technique.

$$\psi(x,t)=\psi(x)\phi(t) \tag{2}$$

Hence, $\psi(x)$ becomes a function of position alone with no time dependance. Rewriting the Schrödinger’s wave equation with the separated variables gives,

$$-\frac{\hbar^2}{2m}\,\phi(t)\frac{\partial^2\psi(x)}{\partial x^2}+V(x)\,\psi(x)\phi(t)=j\hbar\,\psi(x)\frac{\partial\phi(t)}{\partial t} \tag{3}$$

Dividing by the total wave function, $\psi(x)\phi(t)$ gives,

$$-\frac{\hbar^2}{2m}\frac{1}{\psi(x)}\frac{\partial^2\psi(x)}{\partial x^2}+V(x)=j\hbar\,\frac{1}{\phi(t)}\frac{\partial\phi(t)}{\partial t} \tag{4}$$

This form allows us to clearly observe the time dependent and independent portions of the Schrödinger’s wave equation, since either side of the equation is a function of a variable that is totally independent of the other. Hence, we can rewrite the dependent and independent portions with respect to some constant denoted by,

$$\frac{E}{\hbar} = \omega = \frac{\eta}{\hbar} \tag{5}$$

Time dependent portion written as,

$$E = j\hbar \, \frac{1}{\phi(t)} \frac{\partial \phi(t)}{\partial t} \tag{6}$$

Time independent portion written as,

$$-\frac{\hbar^2}{2m} \frac{\partial^2 \psi(x)}{\partial x^2} + V(x)\psi(x) = E\psi(x) \tag{7}$$

The Time Independent Schrödinger's Equation given above will be referred to as the TISE hereon, and is the concern of this study.

## 1.2 Quantum and Classical Relations

From de Broglie's hypothesis of the wave-particle duality principle, and the typical mathematical form of a wave function solution given by,

$$\lambda = \frac{h}{\rho} \;\; , \;\; \psi \approx \; exp\left(\frac{2\pi i z}{\lambda}\right) \tag{8}$$

Where $\lambda$, $h$, $\rho$, give the de Broglie's wavelength, Planck's constant and the momentum of the particle respectively.

$$\frac{\partial^2 \psi}{\partial z^2} = -k^2 \psi \;\; , \;\; k = \frac{2\pi}{\lambda} \tag{9}$$

From the one-dimensional representation of the wave equation in mathematics and the classical wave relation stated above, substitution and simplification lead to,

$$k = 2\pi \frac{1}{\lambda} = \frac{\rho}{\hbar} \tag{10}$$

$$\frac{\partial^2\psi}{\partial z^2} = -\psi\frac{\rho^2}{\hbar^2} \tag{11}$$

Multiplying by $\frac{1}{2m}$, and rearranging the terms gives,

$$-\frac{\hbar^2}{2m}\frac{\partial^2\psi}{\partial z^2} = \psi\frac{\rho^2}{2m} \tag{12}$$

$$\rho = mv \ \ , \ -\frac{\hbar^2}{2m}\frac{1}{\psi}\frac{\partial^2\psi}{\partial z^2} = \frac{mv^2}{2} \tag{13}$$

Hence the left and right side of the equation above relates the kinetic energies of the quantum mechanical picture with that of the classical.

Each segment of the TISE can be related to the classical energy picture as follows,

$$\underbrace{-\frac{\hbar^2}{2m}\frac{\partial^2\psi(\mathrm{z})}{\partial \mathrm{z}^2}}_{\text{Kinetic Energy}} + \underbrace{V(\mathrm{z})\psi(\mathrm{z})}_{\text{Potential Energy}} = \underbrace{E\psi(\mathrm{z})}_{\text{Total Energy}} \tag{14}$$

Representation in the Hamiltonian operator, *H*, form,

$$\underbrace{\left[-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial \mathrm{z}^2} + V(\mathrm{z})\right]}_{\mathrm{H}}\psi(\mathrm{z}) = E\psi(\mathrm{z}) \tag{15}$$

The energy eigenvalues of *H*, determine the set of energies corresponding to $E_n$, and the eigenfunctions of *H*, correspond to the set of wavefunctions $\psi_n$, where *n*, denotes the principal quantum number which states the excitation state from ground state onwards.

The probability density of the wave function has a more meaningful sense, as it tells the probability for finding a particle in a certain region of space, and is given by,

$$P(r): P(\vec{r}) \ = \ |\psi(\vec{r})\}|^2 \tag{16}$$

In an infinite space, the probability of finding a particle would be perfectly certain. Hence the normalized wave function probability density is given as,

$$\int_{-\infty}^{+\infty} |\psi^2| \ d\psi = 1 \tag{17}$$

# 2. Numerical Methods

The TISE in its complete three-dimensional representation is a complex second order differential. The analytical solution for even the one-dimensional TISE can get quite tedious when faced with bound potentials that are complex. Only a few potential models for the TISE and TDSE (Time Dependent Schrödinger's Equation) have complete analytical solutions. One such example is the analytical solution for the one-electron atom potential, whose analytical solution is realized through the symmetry of the problem. But the same analytical techniques cannot be extended to solve potentials with more than one electron, due to relativistic effects and quantum nature of electromagnetic interaction, which in turn effect the separability of the partial differential equation. Numerical computation methods provide a more computationally efficient approximate of the solutions to these complex differentials' equations.

## 2.1 Types of Available Methods

Many different numerical methods are available for solving the TISE. Each method offers some variances in computational efficiency and accuracy in terms of simulating the TISE for various potentials functions and multidimensional space.

A handful of the more mainstream methods are discussed in brief detail in this section.

### 2.1.1 Finite Difference Method (FDM)

The finite difference method is possibly the most commonly used method for numerically solving complex differential equations of a certain complexity. It also offers a very intuitive understanding of how the differential equations are handled numerically.

The fundamental principle in FDM is based on representing the derivatives in the TISE, by approximation of differences between successive points in space.

> This reduces the differential equation to a difference equation. An analytical problem becomes an algebraic one. A problem with an infinite degree of freedom is replaced by

> one with a finite degree of freedom. A continuous problem goes over to a discrete one. (Kalnay, 2012)

A more detailed description of the mechanics of the finite difference method is explained in Section 2.2

### 2.1.2 The Shooting Method

The shooting method handles the derivative term differently yet again. It is primarily used in applications where "boundary conditions are specified at two different values of the independent variable" (Young, 2007)

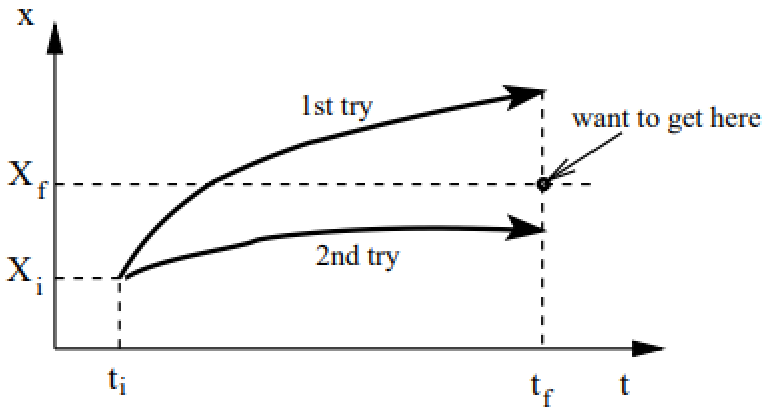


*Figure 1 Shooting Method (Young, 2007)*

Take the instance shown in Figure 1, where the displacement-time graph for two points is presented. In order to proceed from the initial point $X_i$ to final point $X_f$, two parameters have to be specified at the initial time instance: namely, the initial displacement, which is already provided, and the initial velocity, which is unknown. Hence the essence of the shooting method is that it tries to guess some initial velocity and proceed with computing the difference between the computed final value from the desired final value and converges on a solution for the initial velocity, where this error is eventually minimized. The velocity term here is analogous to the derivative term in the TISE.

$$x(t_f) - X_f = 0 \tag{18}$$

In the numerical simulation of the TISE, the shooting method is primarily used to obtain the solutions for the discrete energy eigen states and the corresponding eigen functions describing the wave function at that energy state.

### 2.1.3 Runge-Kutta Approach (RK)

The RK algorithm is more complex than the previous methods discussed. It is still a very widely used and effective BVP solver for complex differential equations. “It is stable and can be made quite accurate, although it is not always the most efficient algorithm” (Robertson, 2009).

The RK algorithm has two main orders, namely RK2 and RK4, for second and fourth order respectively. RK4 is more accurate and more complex of the two. The derivation of the RK algorithms are based on approximating the higher order derivative terms similar to that of the Euler method and FDM. Unlike the latter cases, various partial steps can be taken across a single step iteration to cancel out the error terms of higher orders. The derivation of the RK2 algorithm is presented in detail in the work of (Robertson, 2009).

RK2 algorithm.

$$k_1 = \epsilon f(x, y) \ \ , \ k_2 = \epsilon f(x + \epsilon/2, y(x) + k_1/2) \tag{19}$$

$$y(x + \epsilon) = y(x) + k_2 + \mathcal{O}(\epsilon^3) \tag{20}$$

### 2.1.4 Numerov’s Algorithm

The Numerov algorithm is a variation of the RK approach, that is computationally more efficient and provides a higher order of accuracy than RK4. The difference between the former and latter lie in the way they handle the TISE derivatives. RK4 represents the TISE as two first order equations, whereas Numerov’s algorithm uses the TISE in its native second order form.

(Young, 2009) presents a detailed derivation of Numerov’s algorithm for the TISE. To initiate this method, the TISE has be to rewritten to follow the form given below.

$$\frac{\partial^2 \psi}{\partial z^2} + k^2(z)\psi(z) = 0 \ \ , \ k^2(z) = \frac{2m}{\hbar^2}[E - V(z)] \tag{21}$$

The Numerov algorithm for a single time step is given as follows, (Young, 2009)

$$\psi_{n+1} = \frac{2\left(1 - \frac{5}{12}h^2 k_n^2\right)\psi_n - \left(1 + \frac{1}{12}h^2 k_{n-1}^2\right)\psi_{n-1}}{1 + \frac{1}{12}h^2 k_{n+1}^2} \tag{22}$$

## 2.2 Implementation

The preceding section provides a brief description of a few commonly used numerical differential solvers. There are many more different types of variants derived from these methods that may offer significant performance improvements to accuracy and computational efficiency.

This study focusses on the use of the finite difference method (FDM) to simulate the TISE for various potential functions. This section focuses on the FDM and its implementation to simulate the TISE. The derivation of the FDM is presented below.

From Taylor's expansion, we have,

$$f(x+\delta x) = f(x) + \frac{\delta x}{1!} f'^{(x)} + \frac{\delta x^2}{2!} f''^{(x)} + \ . \ . \ . \tag{23}$$

$$f(x-\delta x) = f(x) - \frac{\delta x}{1!} f'^{(x)} - \frac{\delta x^2}{2!} f''^{(x)} - \ . \ . \ . \tag{24}$$

Higher order terms are neglected as the $\delta x$, term gets smaller with increasing order. A marginal increase in accuracy may be observed for higher orders over the increased computation time. Re-writing derivative approximations as,

$$f'(x) = \frac{f(x+\delta x) - f(x)}{\delta x} = f'_F \tag{25}$$

$$f'(x) = \frac{f(x) - f(x-\delta x)}{\delta x} = f'_B \tag{26}$$

Equations (25,26) represent the forward and backward difference of the first derivative term. Subtracting the two equations yields the center difference term.

$$f'(x) = \frac{f(x+\delta x) - f\big(x-\delta(x)\big)}{2\delta x} = f'_C \tag{27}$$

The forward and backward difference method have similar approximation errors, whereas the center difference method has the least amount of truncation error, hence it gives better performance in the FDM algorithm. Figure 2, provides a visual depiction of the above.

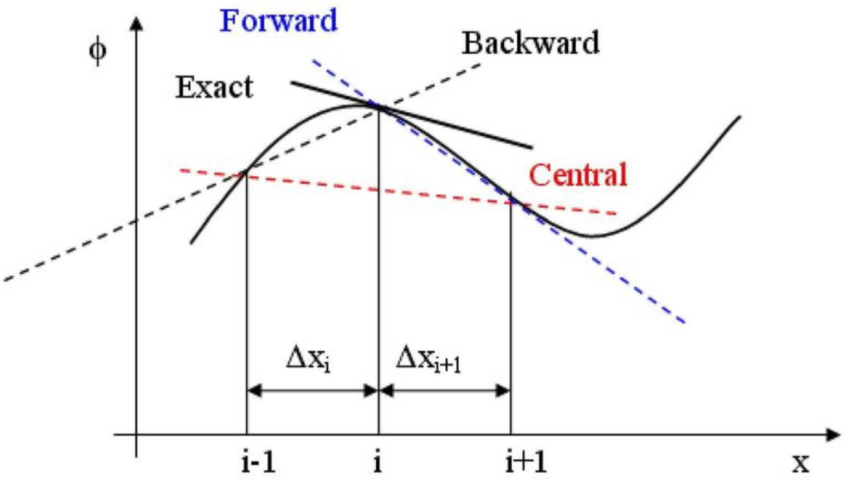


*Figure 2 Finite Difference Variations (Kesserwan, 2011)*

Adding equations (25,26) yield the second derivative expression given as,

$$f''(x) = \frac{f(x+\delta x) - 2f(x) + f(x-\delta x)}{\delta x^2} \tag{28}$$

Since the derivatives can now be represented in their equivalent finite difference form, it is now necessary to implement the vectorized form of the FDM terms to solve for the TISE.

From equation (15), the vectorized form can be written as,

$$\begin{pmatrix} \\ H \\ \\ [N \times N] \end{pmatrix} \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_N \end{pmatrix} = E \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_N \end{pmatrix} \tag{29}$$

Where $N$, is the number of discrete difference points in space chosen for the simulation space.

Rewriting the TISE with the FDM terms gives,

$$-\left(\frac{\hbar^2}{2m}\right)\left(\frac{\psi_{n+1} - 2\psi_n + \psi_{n-1}}{d^2}\right) + V_n\psi_n = E\psi_n \tag{30}$$

Where $d$, is the difference between each successive point. Hence the constant terms are given as,

$$-\left(\frac{\hbar^2}{2md^2}\right)\underbrace{(\psi_{n+1}-2\psi_n+\psi_{n-1})}_{A}+V_n\psi_n=E\psi_n \tag{31}$$

Segment $A$ and $V_n$, of the above equation can then be rewritten in vectorized matrix form as,

$$A=\begin{pmatrix}-2 & 1 & 0 & \dots & 0\\ 1 & -2 & 1 & & \vdots\\ 0 & 1 & \ddots & \ddots & 0\\ \vdots & & \ddots & \ddots & 1\\ 0 & \dots & 0 & 1 & -2\end{pmatrix}\begin{pmatrix}\psi_1\\ \psi_2\\ \psi_3\\ \vdots\\ \psi_n\end{pmatrix}\ ,\ V_n=\begin{pmatrix}V_1 & 0 & \dots & \dots & 0\\ 0 & V_2 & & 0 & \vdots\\ \vdots & & \ddots & & \vdots\\ \vdots & 0 & & \ddots & 0\\ 0 & \dots & \dots & 0 & V_n\end{pmatrix} \tag{32}$$

Hence equation (30), in vectorized form is form is given as,

$$\underbrace{-\left(\frac{\hbar^2}{2md^2}\right)\begin{pmatrix}-2 & 1 & 0 & \dots & 0\\ 1 & -2 & 1 & & \vdots\\ 0 & 1 & \ddots & \ddots & 0\\ \vdots & & \ddots & \ddots & 1\\ 0 & \dots & 0 & 1 & -2\end{pmatrix}}_{K}\begin{pmatrix}\psi_1\\ \psi_2\\ \psi_3\\ \vdots\\ \psi_n\end{pmatrix}+\underbrace{\begin{pmatrix}V_1 & 0 & \dots & \dots & 0\\ 0 & V_2 & & 0 & \vdots\\ \vdots & & \ddots & & \vdots\\ \vdots & 0 & & \ddots & 0\\ 0 & \dots & \dots & 0 & V_n\end{pmatrix}}_{V}\begin{pmatrix}\psi_1\\ \psi_2\\ \psi_3\\ \vdots\\ \psi_n\end{pmatrix}=E\begin{pmatrix}\psi_1\\ \psi_2\\ \psi_3\\ \vdots\\ \psi_n\end{pmatrix}$$

$$K\psi_n+V\psi_n=(K+V)\psi_n=H\psi_n=E\psi_n \tag{33}$$

Hence equations (33) and (29) are of similar form.

The next step would be developing the algorithm and solving for the vectorized implementation of the TISE observed above. MATLAB is chosen in this context to build and simulate the performance of the FDM model on various potential functions.

### 2.2.1 MATLAB Implementation

The following discussion explains key code snippets from the complete code found in Appendix A of this document. The TISE for one dimension is explained here.

```
%% ------Initiating the constants------------------------------------------

hbar = 1.055e-34;
m = 9.11e-31; % Rest mass of electron
eV = 1.602e-19;
```

The standard constant terms in the TISE are initiated here. The particle to be simulated inside a potential field is an electron. The relative mass of the electron is not taken into account here and it is assumed the mass remains constant at all excitation states.

```
%% ------Initiating the stepped lattice points-----------------------------

L = 10e-10; % Potential well width
z_min = -L; % -ve Z range
z_max = L; % +ve Z range
n = 1000; % Number of lattice points
z = linspace(z_min,z_max,n); % Vector of 'z' lattice points
d = z(2) - z(1); % Successive lattice spacing
```

In this section the discretization of the continuous space domain is performed. The lattice window spanning $-L$ to $+L$ is divided into a vector of $n$, lattice points spaced at $d$ intervals.

```
%% ------Generating the potential energy matrix----------------------------

Wh = 10; % Well Height
Vo = Wh*eV; % Max finite potential
B1 = -L/2; % -ve well boundary
B2 = L/2; % +ve well boundary
```

The well boundaries and the maximum finite potential is set in the snippet above. The potential function is generated in the section below. Different types of potential functions are simulated and their results are compared in the next section.

```
% -------Generation of potential function----------------------------------

%Select potential well type

V = ones(1,n)*Vo; V(z>= B1 & z<=B2) = 0; % Square Well
%V = Vo/(L^2)*(z.^2); % Harmonic Well
%V = ones(1,n)*Vo; V(z>= B1 & z<=B2) = Vo/(B2^2)*(z((z>= B1 & z<=B2)).^2);
% Truncated Parabolic Well
%V((z<=B1)) = Vo/(B2-L)^2*((z(z<=B1)) + L).^2; V(z>= B1 & z<=B2) =
0;V((z>=B2)) = Vo/(L-B2)^2*((z(z>=B2)) - L).^2; % Quadratic Well

%--------------------------------------------------------------------------

Vn = eye(n,n);
Vp = V'.*Vn;
```

Since the potential function should be of the form depicted in vectorized form (32), the potential function is transformed from a vector to a matrix with a diagonal of values.

```
%% ------Generating the kinetic energy matrix using FDM--------------------

K = eye(n,n); % Initiating the kinetic energy matrix
K = K*(-2); % Setting the main diagonal
for t = 1:n-1 % Setting the two flanking diagonals
   K(t,t+1) = 1;
   K(t+1,t) = 1;
end
```

```
%% ------Generating the Hamiltonian----------------------------------------

H = (-(hbar^2)/(2*m*d^2))*K+Vp ;
```

From (32), the kinetic energy matrix, *A*, is implemented in code in FDM vectorized form as shown in the previous snippet. First the main diagonal is initialized and then the two side diagonals flanking the main diagonal are set to create the second order difference matrix. The Hamiltonian matrix is then developed following equation (33).

```
%% ------Generating the energy eigen values/states-------------------------

[U,EV] = eig(H); % Obtaining the wavefunctions and the eigen energies

P = U.^2; % Probability density of wavefunctions

E = diag(EV); % Accumulating the energy states to a column vector
```

MATLAB eigenvector function is called on the Hamiltonian matrix from which matrices of all the potential wavefunctions from ground state excitation onwards and the possible energy eigen values satisfying the TISE are obtained. The energy eigenvalues are accumulated into a vector from the diagonal matrix. The results are then plotted for wavefunctions and their respective probability densities.

```
%---------Plotting the allowable energy states-----------------------------

Ea = Ec(Ec<Wh)
```

The allowable energy states are obtained by comparing all the discrete energy eigen values with the predefined maximum potential ceiling. All the energies lower than this maximum are collected and graphs are plotted to show the allowable energy states within each type of well.

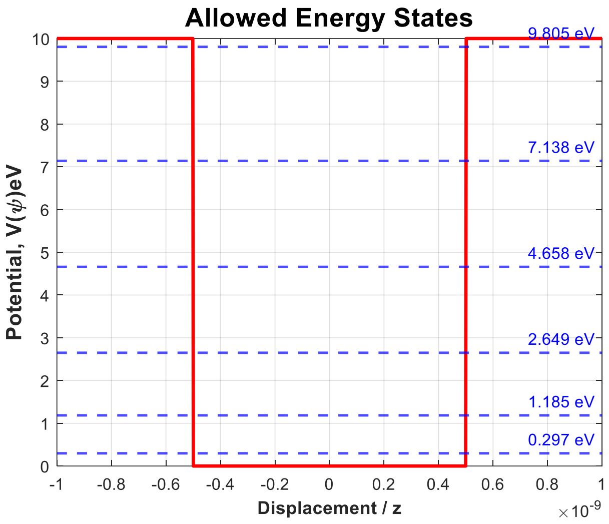
# 3. Simulation Outcomes

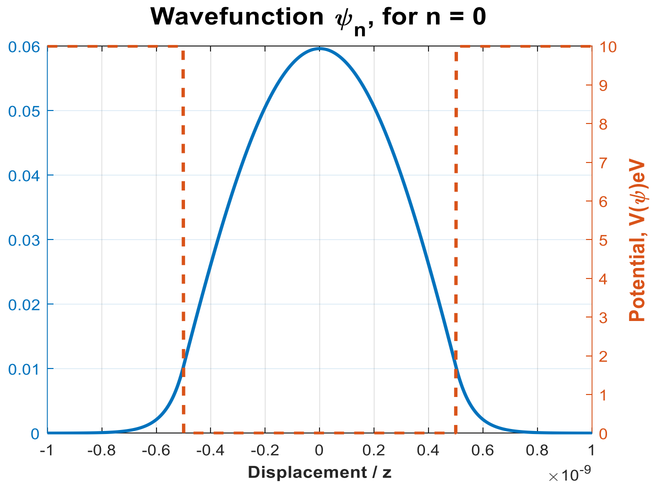
## 3.1 Symmetric Potentials

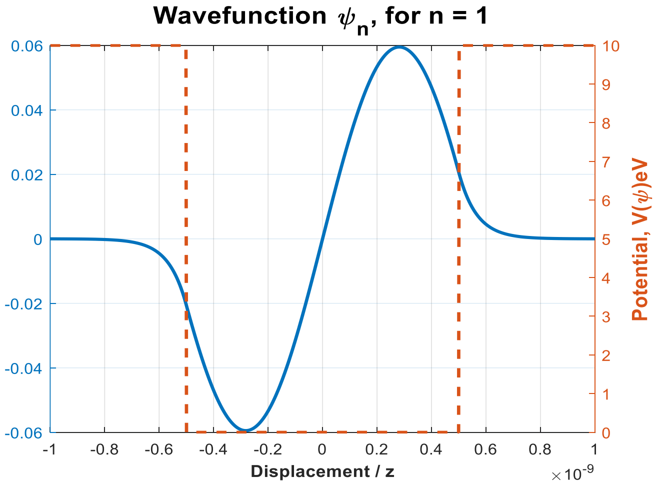
Some simple symmetric potential functions are simulated and the results are presented.

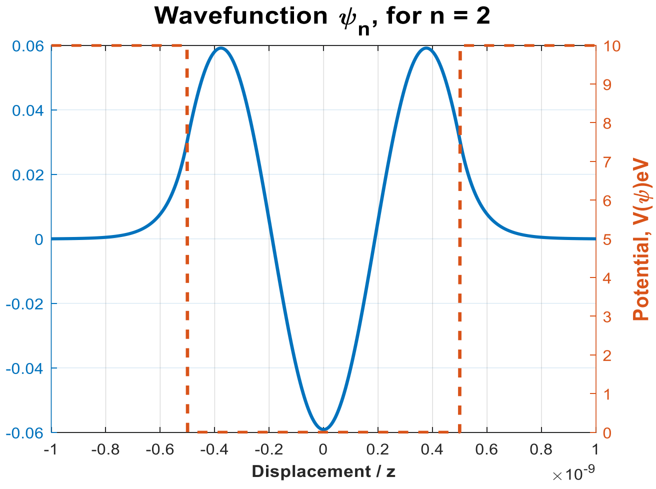
Most orders of the wavefunction and density function are presented below where possible.

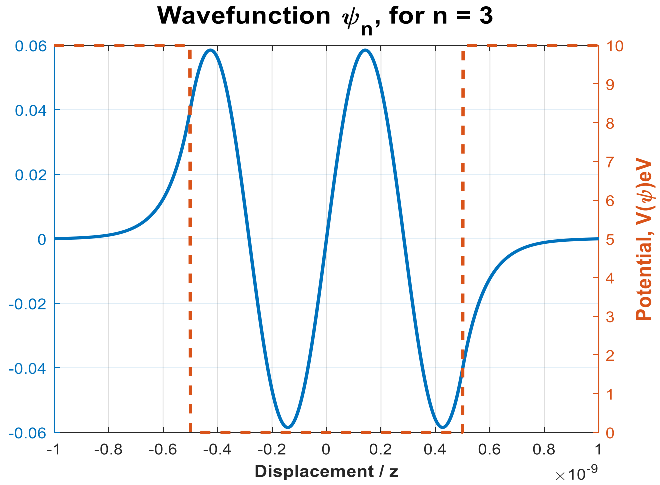
### 3.1.1 Square Potential Well



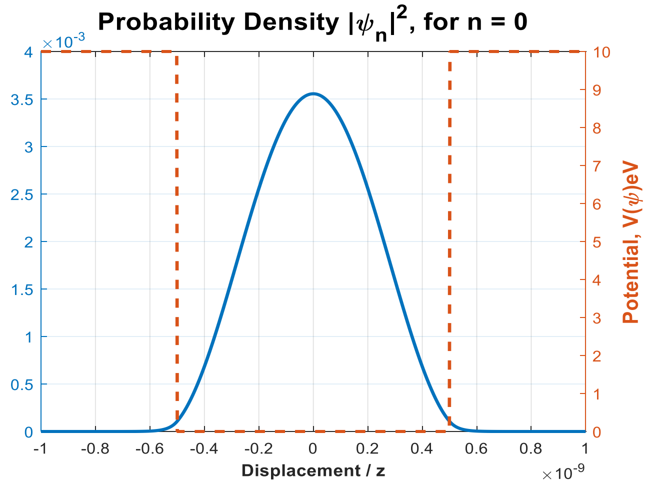




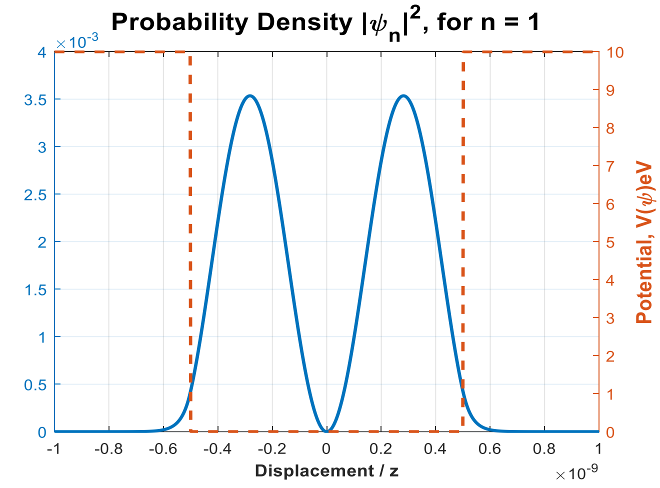




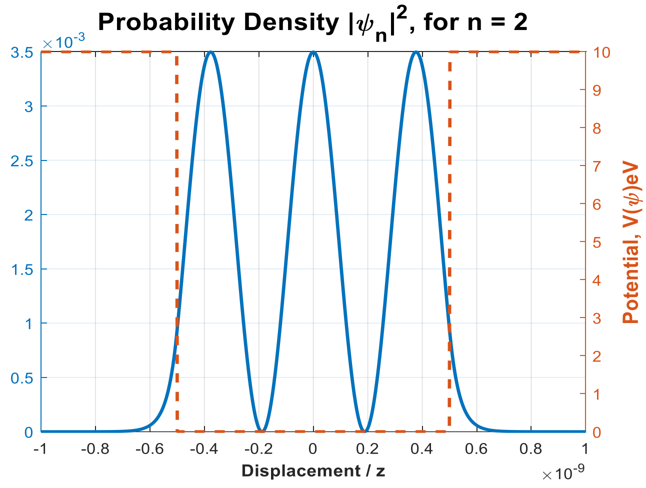
Wavefunction $\psi_n$, for n = 3



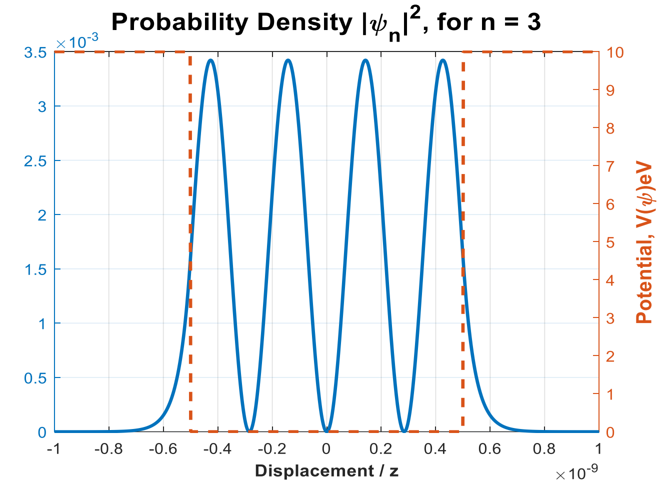
Probability Density $|\psi_n|^2$, for n = 0



Probability Density $|\psi_n|^2$, for n = 1

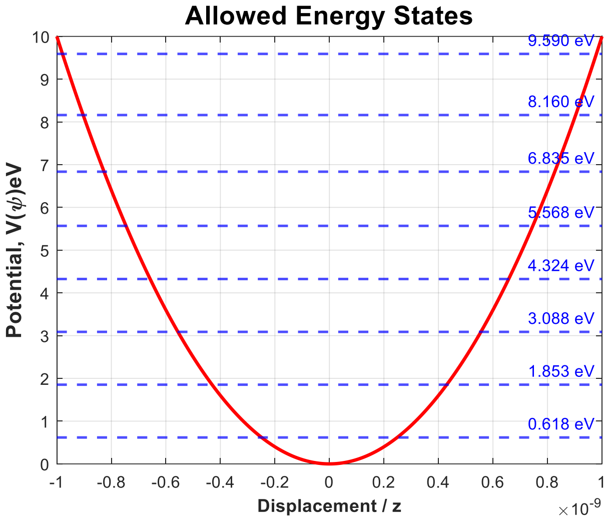


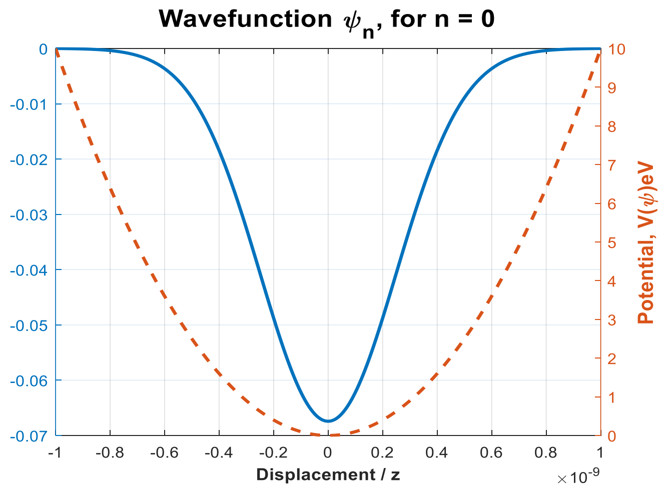
Probability Density $|\psi_n|^2$, for n = 2

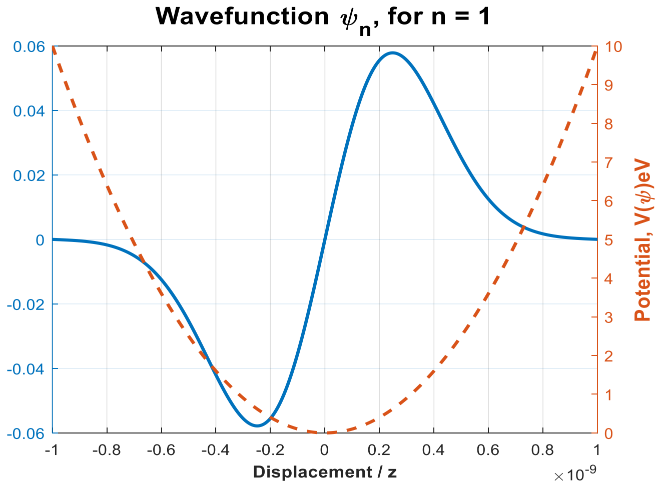


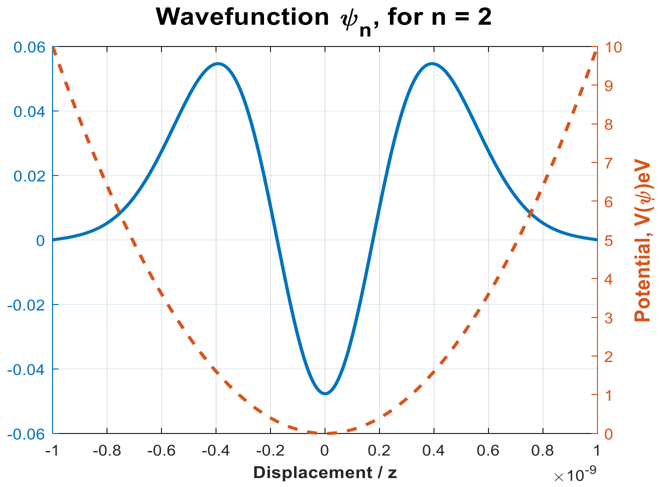
Probability Density $|\psi_n|^2$, for n = 3

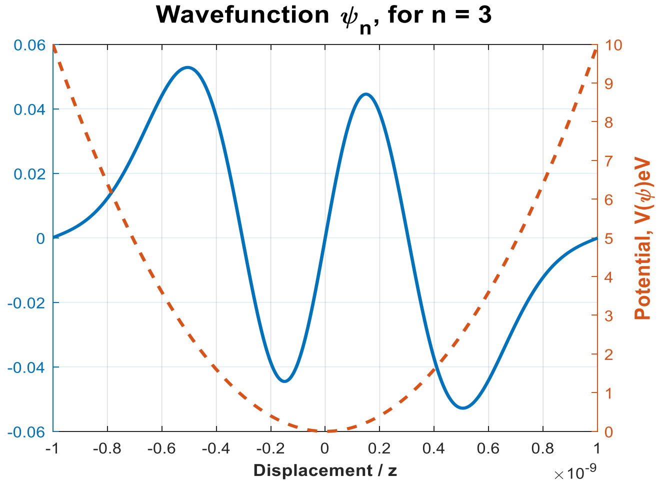
### 3.1.2 Harmonic Potential Well

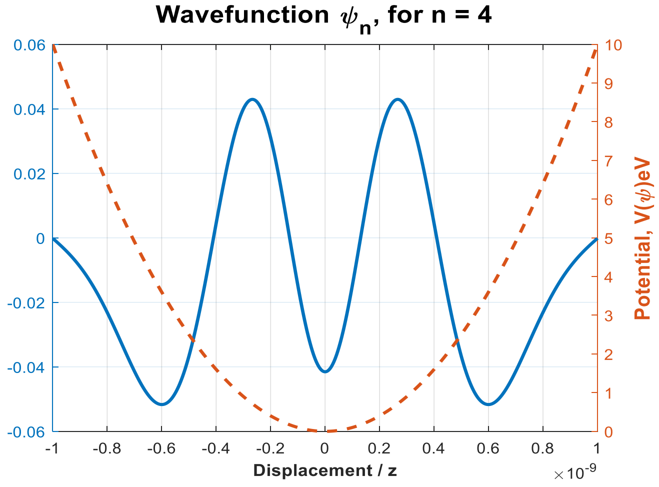


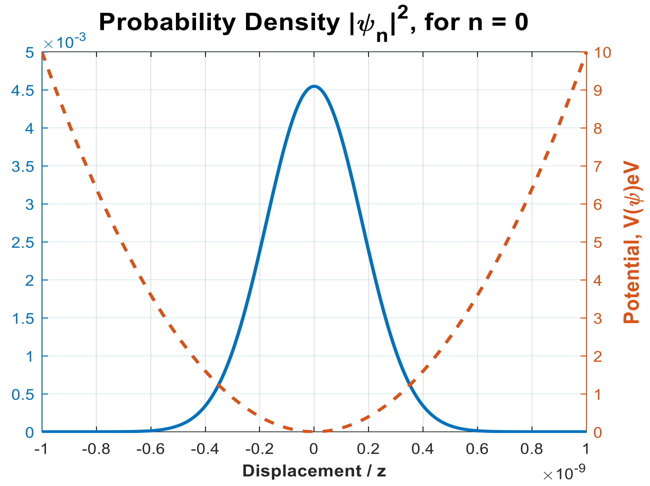


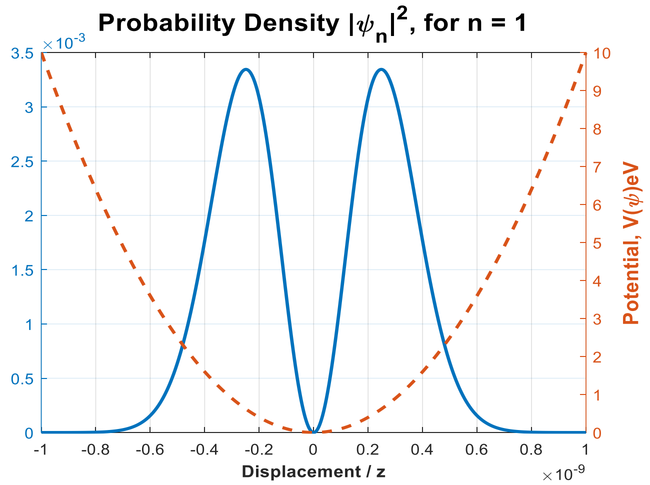


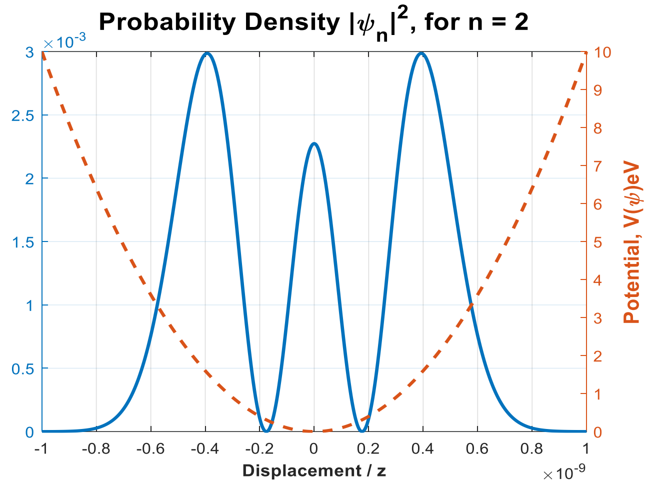


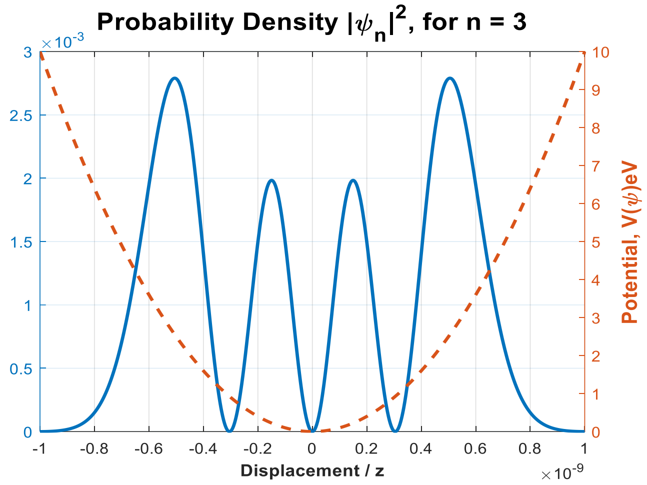


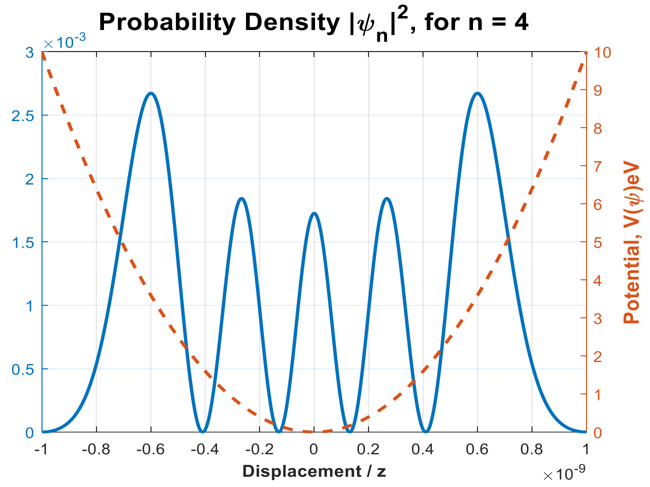



Probability Density $|\psi_n|^2$, for n = 0

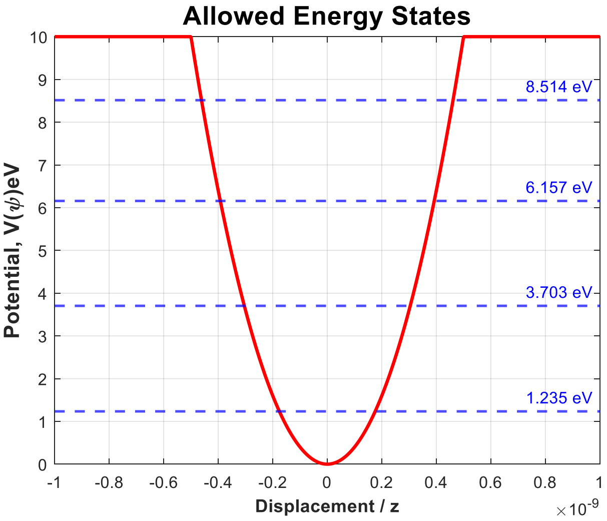

Probability Density $|\psi_n|^2$, for n = 1

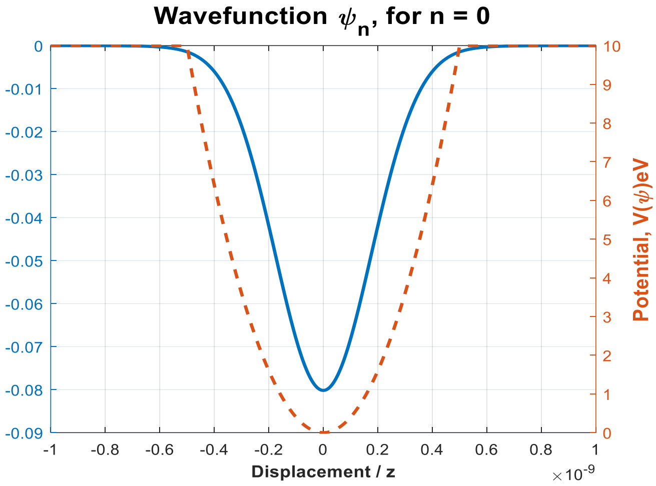

Probability Density $|\psi_n|^2$, for n = 2

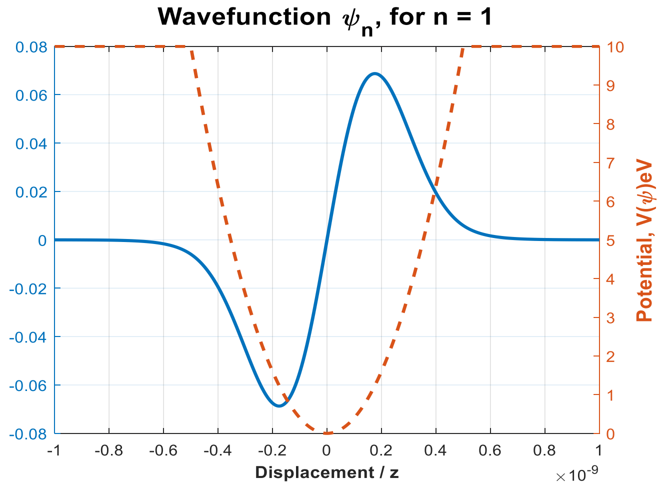

Probability Density $|\psi_n|^2$, for n = 3

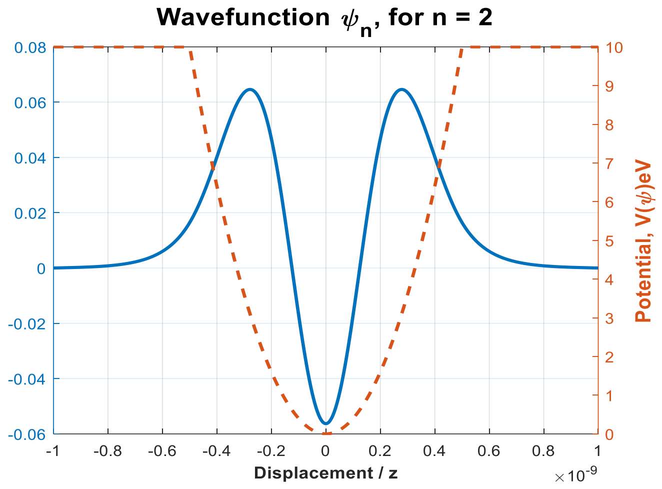

Probability Density $|\psi_n|^2$, for n = 4

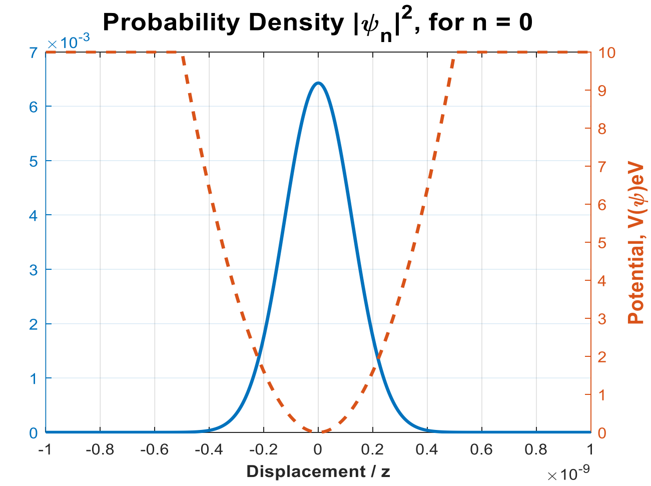
### 3.1.3 Truncated Parabolic Potential Well

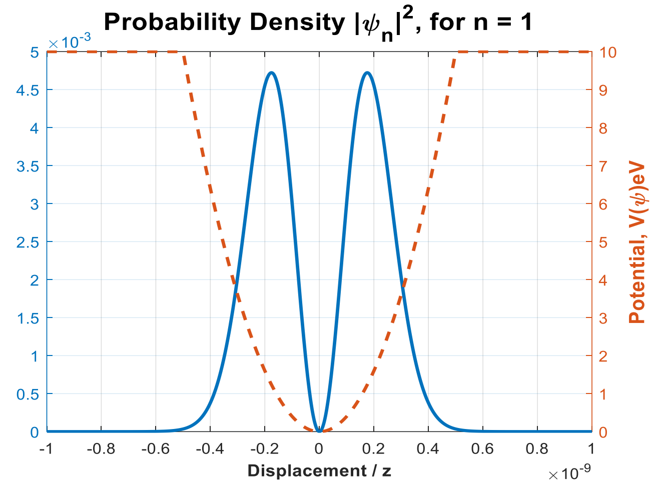

Allowed Energy States

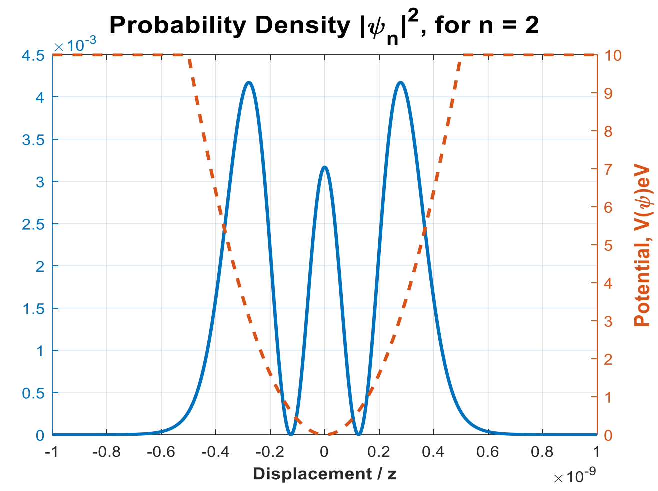

Wavefunction $\psi_n$, for n = 0


Wavefunction $\psi_n$, for n = 1

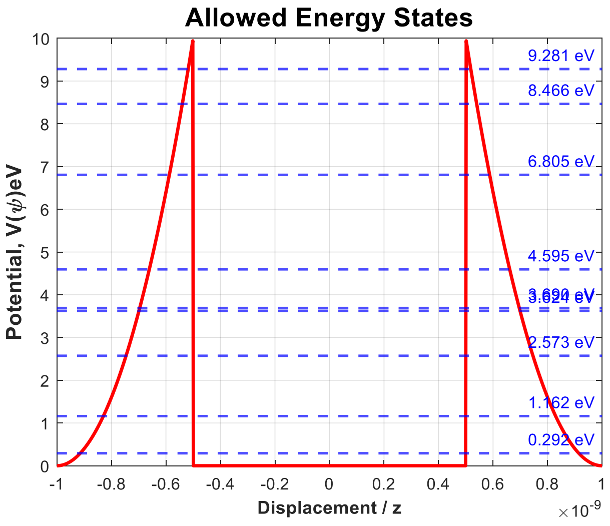

Wavefunction $\psi_n$, for n = 2

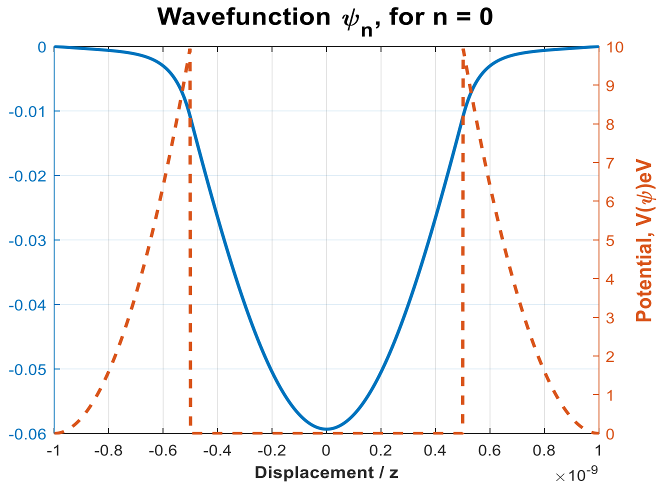

Probability Density $|\psi_n|^2$, for n = 0

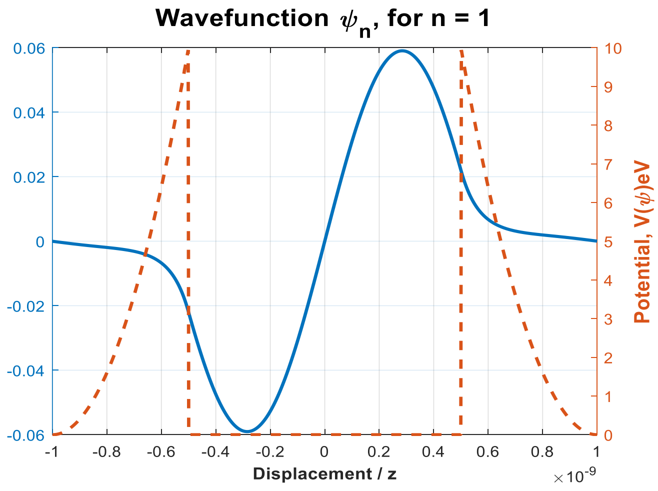

Probability Density $|\psi_n|^2$, for n = 1

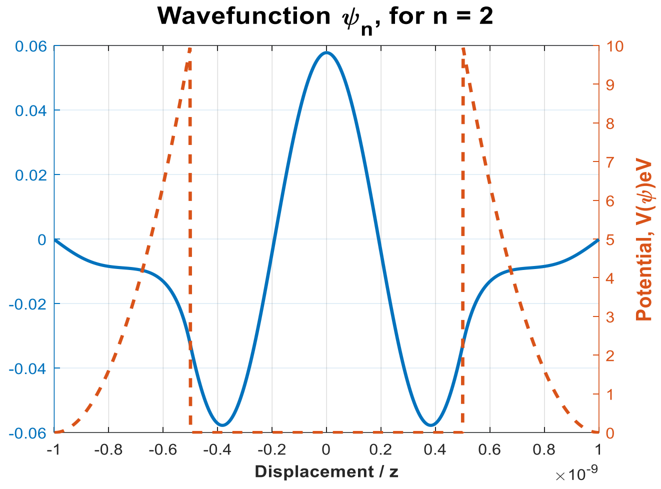

Probability Density $|\psi_n|^2$, for n = 2

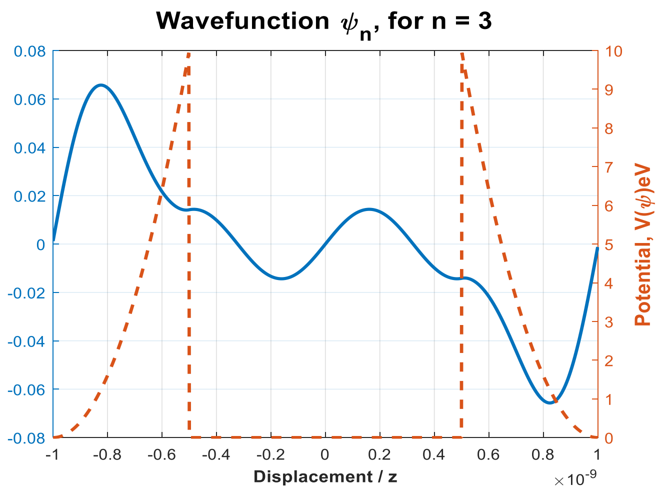
### 3.1.4 Quadratic Potential Well

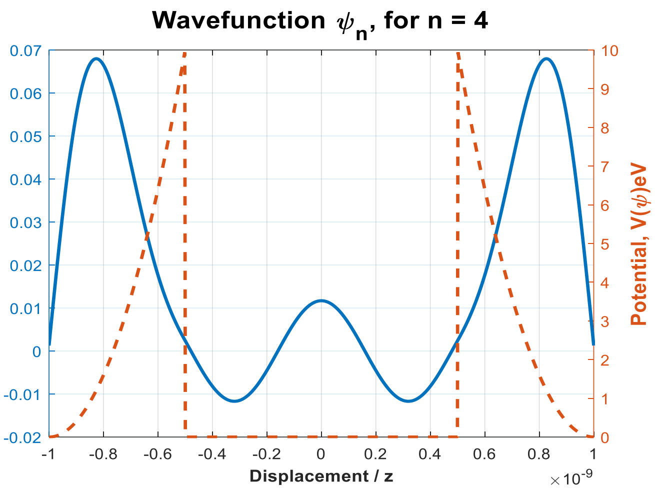

Allowed Energy States

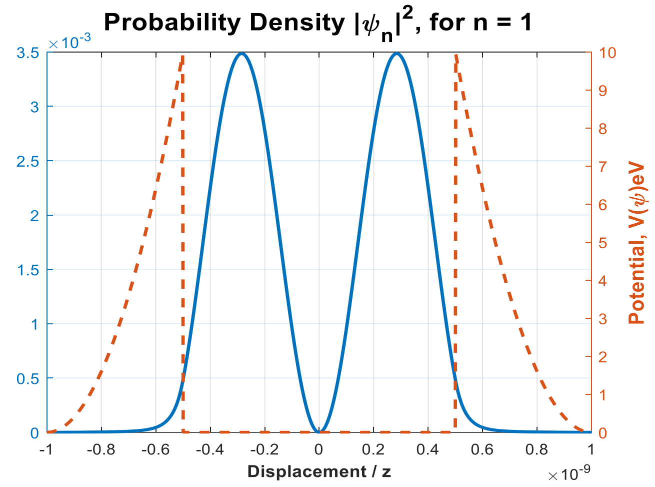

Wavefunction $\psi_n$, for n = 0

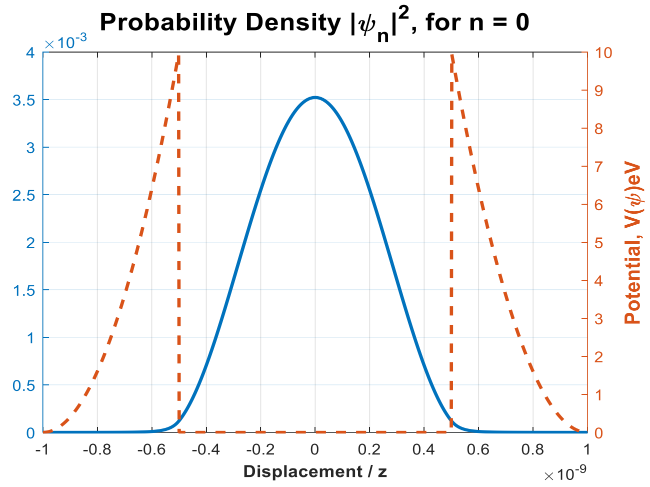

Wavefunction $\psi_n$, for n = 1

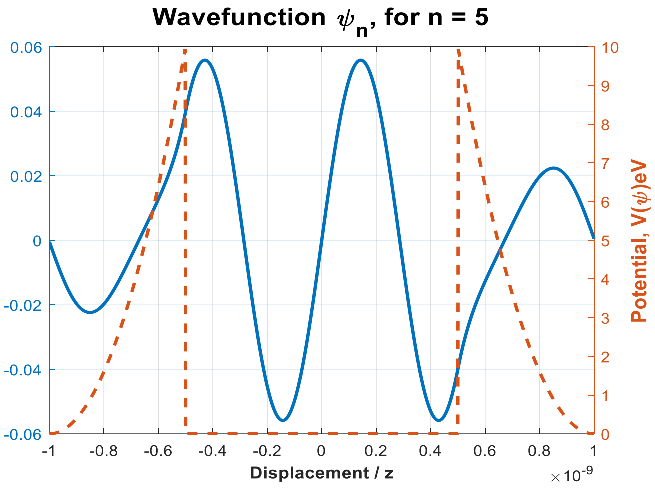

Wavefunction $\psi_n$, for n = 2

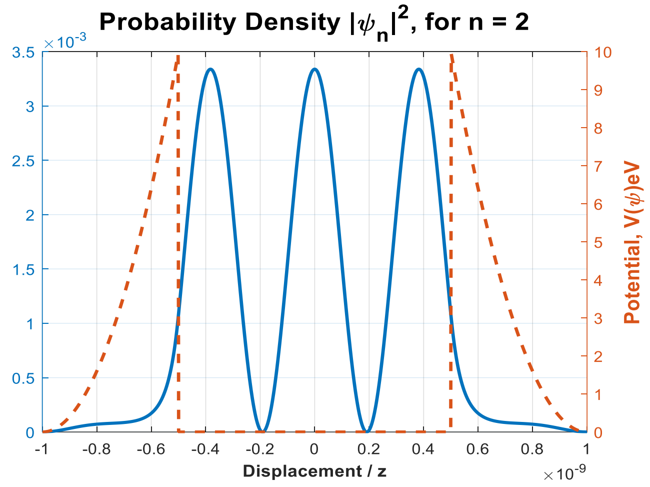

Wavefunction $\psi_n$, for n = 3
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.08
10
9
8
7
6
5
4
3
2
1
0
Potential, V($\psi$)eV
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
Displacement / z
$\times 10^{-9}$

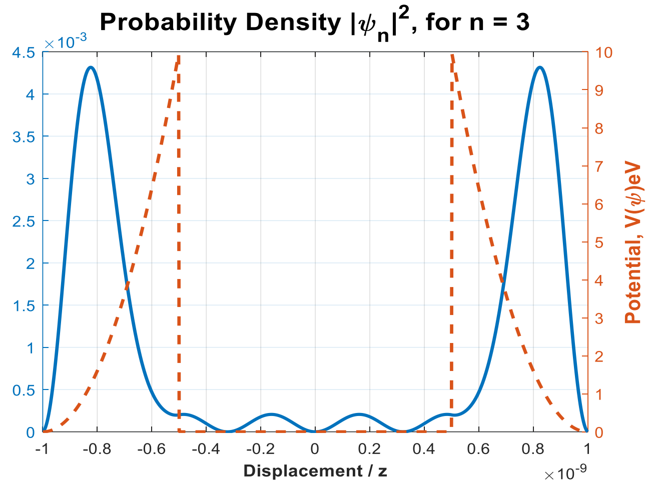

Wavefunction $\psi_n$, for n = 4
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
-0.01
-0.02
10
9
8
7
6
5
4
3
2
1
0
Potential, V($\psi$)eV
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
Displacement / z
$\times 10^{-9}$

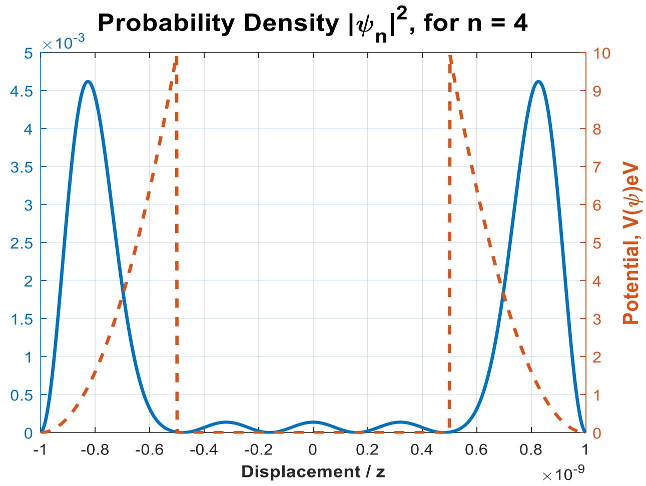

Probability Density $|\psi_n|^2$, for n = 1
$\times 10^{-3}$
3.5
3
2.5
2
1.5
1
0.5
0
10
9
8
7
6
5
4
3
2
1
0
Potential, V($\psi$)eV
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
Displacement / z
$\times 10^{-9}$

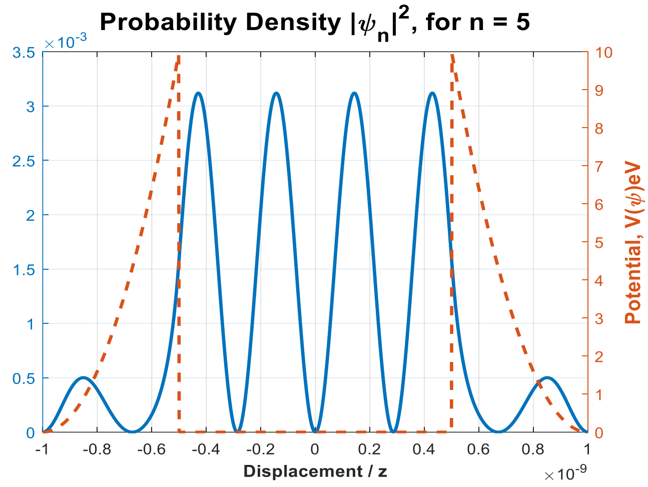

Probability Density $|\psi_n|^2$, for n = 0
$\times 10^{-3}$
4
3.5
3
2.5
2
1.5
1
0.5
0
10
9
8
7
6
5
4
3
2
1
0
Potential, V($\psi$)eV
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
Displacement / z
$\times 10^{-9}$


Wavefunction $\psi_n$, for n = 5
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
10
9
8
7
6
5
4
3
2
1
0
Potential, V($\psi$)eV
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
Displacement / z
$\times 10^{-9}$


Probability Density $|\psi_n|^2$, for n = 2
$\times 10^{-3}$
3.5
3
2.5
2
1.5
1
0.5
0
10
9
8
7
6
5
4
3
2
1
0
Potential, V($\psi$)eV
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
Displacement / z
$\times 10^{-9}$

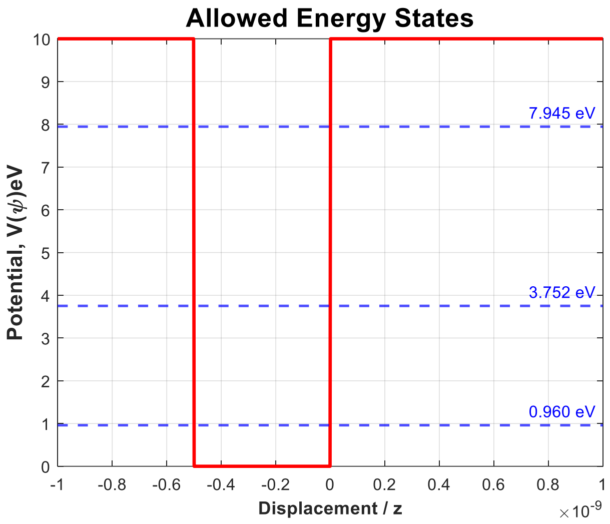

Probability Density $|\psi_n|^2$, for n = 3
$\times 10^{-3}$
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
Displacement / z
$\times 10^{-9}$
10
9
8
7
6
5
4
3
2
1
0
Potential, V($\psi$)eV

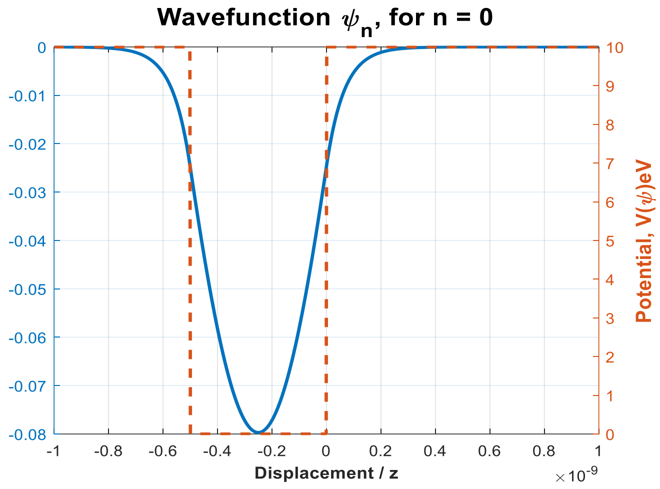

Probability Density $|\psi_n|^2$, for n = 4
$\times 10^{-3}$
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
Displacement / z
$\times 10^{-9}$
10
9
8
7
6
5
4
3
2
1
0
Potential, V($\psi$)eV

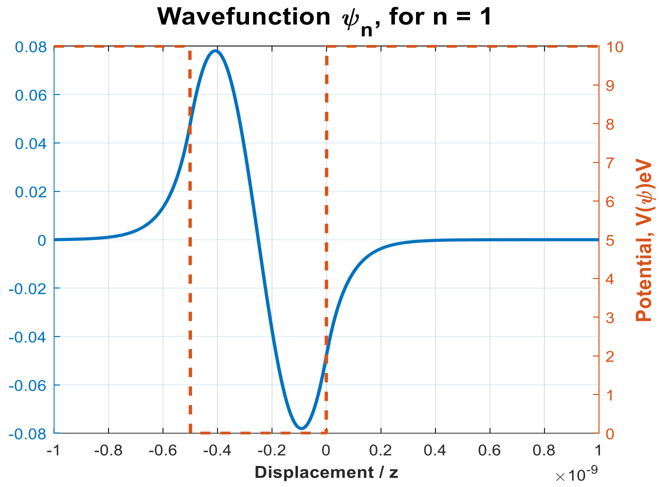

Probability Density $|\psi_n|^2$, for n = 5
$\times 10^{-3}$
3.5
3
2.5
2
1.5
1
0.5
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
Displacement / z
$\times 10^{-9}$
10
9
8
7
6
5
4
3
2
1
0
Potential, V($\psi$)eV

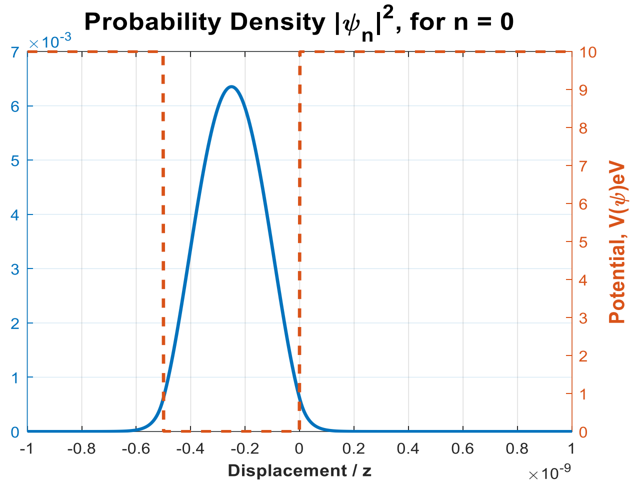
## 3.2 Asymmetric Potentials

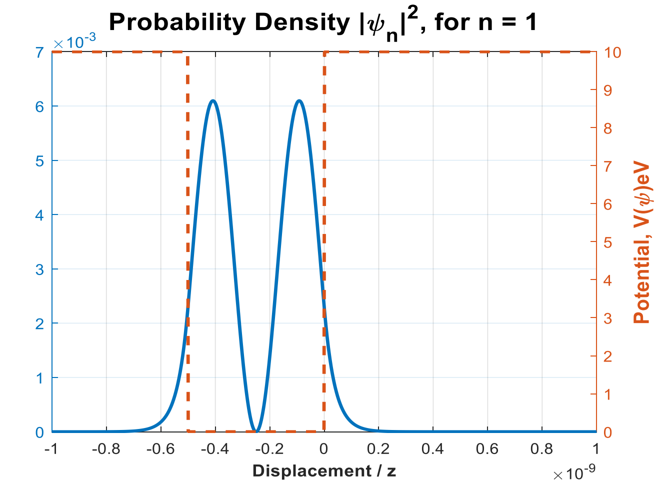
### 3.2.1 Offset Square Potential Well


Allowed Energy States

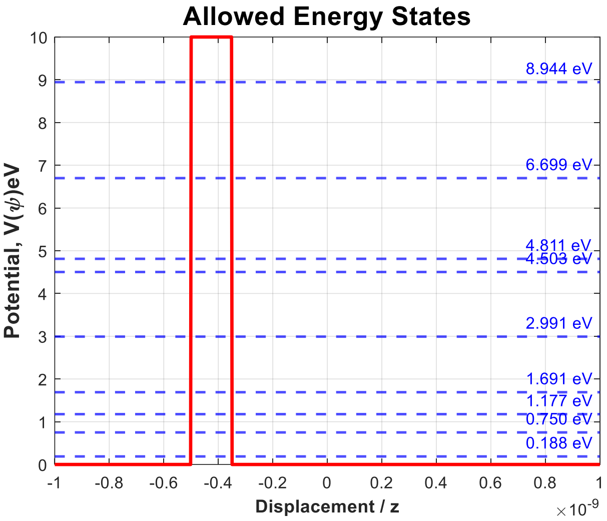

Wavefunction $\psi_n$, for n = 0

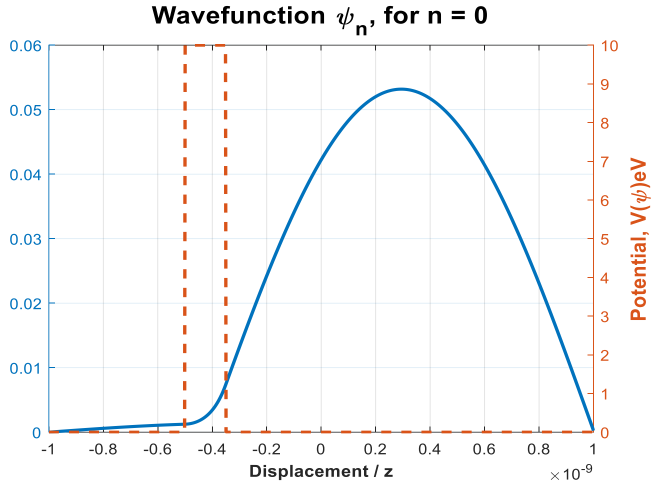

Wavefunction $\psi_n$, for n = 1

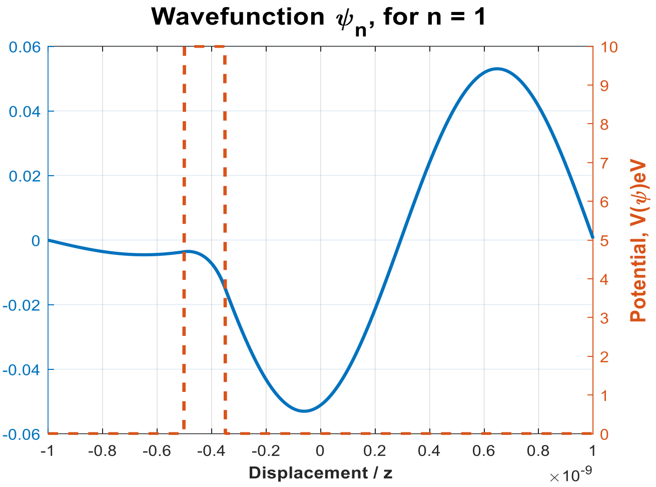

Probability Density $|\psi_n|^2$, for n = 0

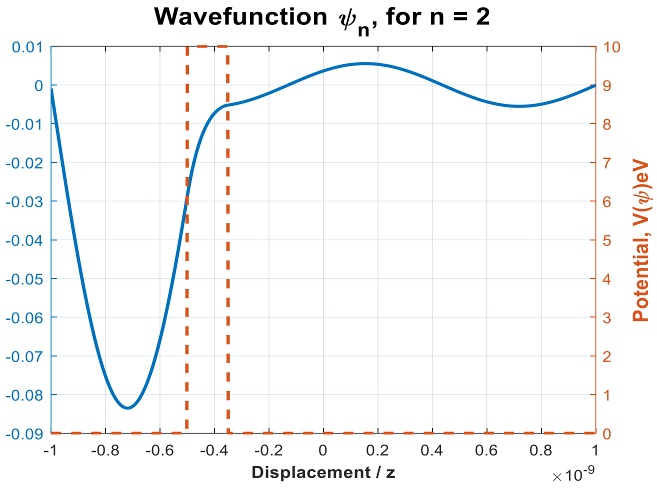

Probability Density $|\psi_n|^2$, for n = 1

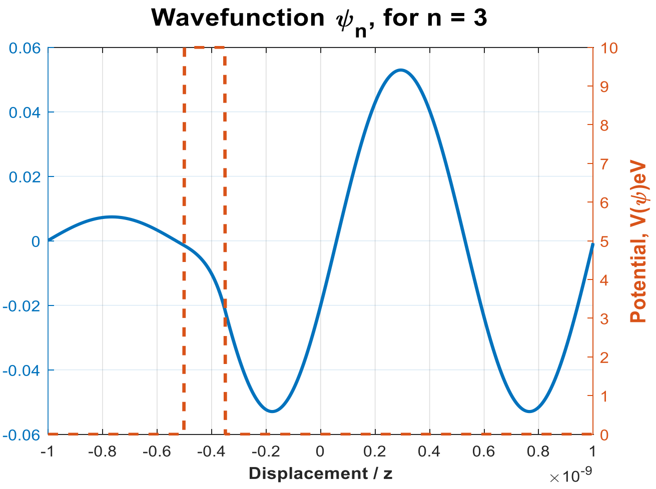
### 3.2.2 Offset Potential Tunnel

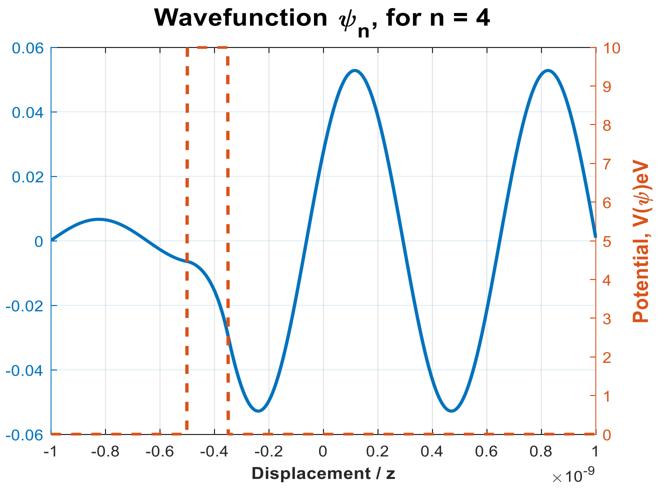




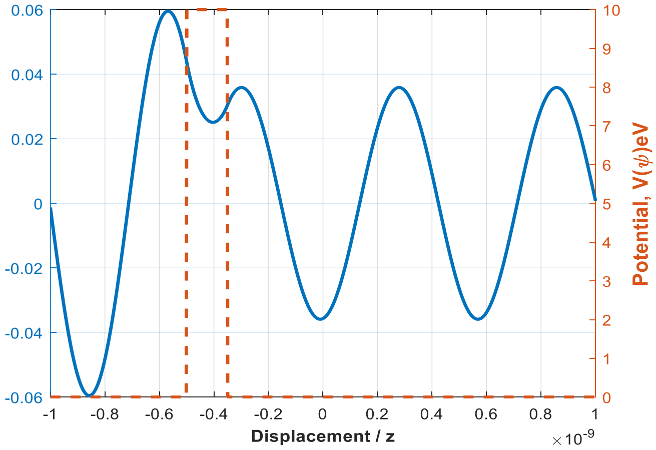




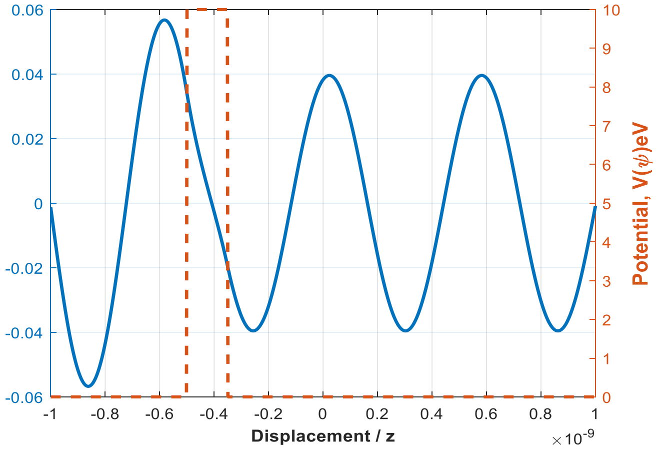




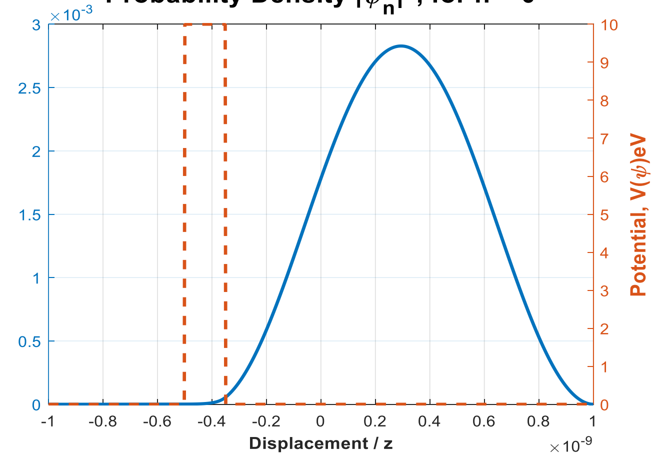
Wavefunction $\psi_n$, for n = 5



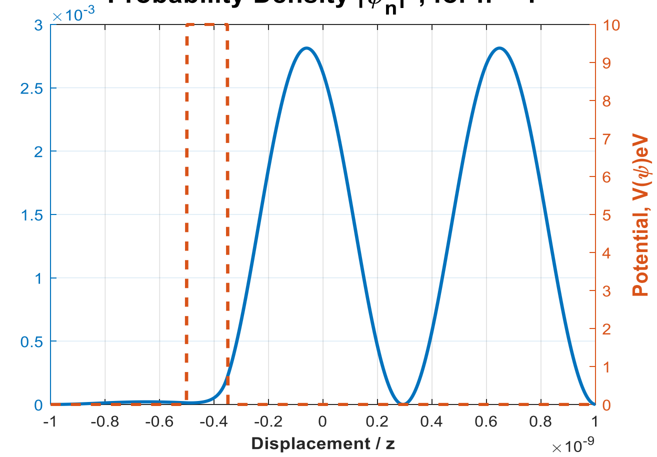
Wavefunction $\psi_n$, for n = 6



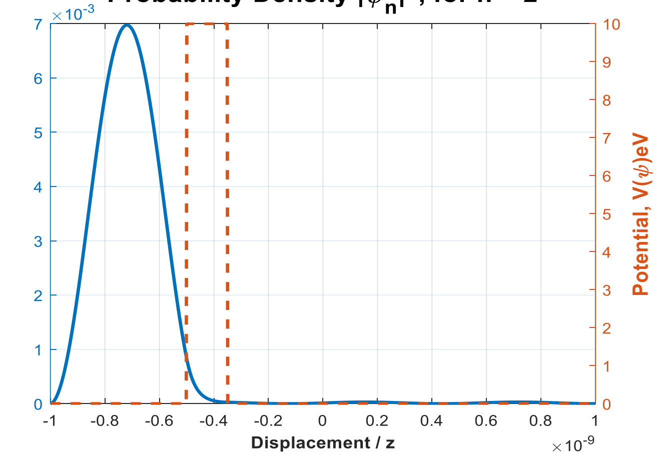
Probability Density $|\psi_n|^2$, for n = 0



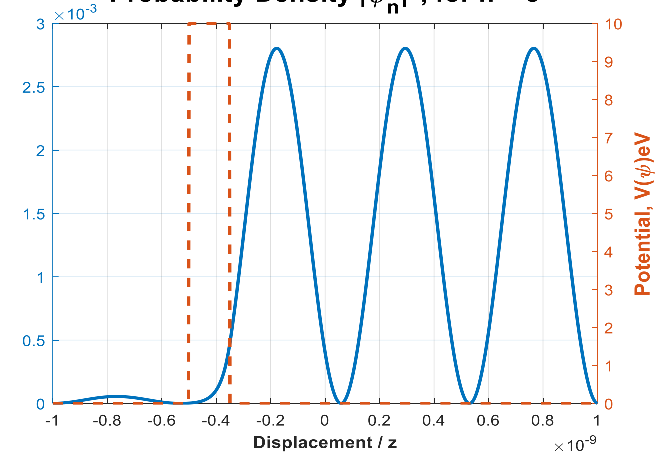
Probability Density $|\psi_n|^2$, for n = 1

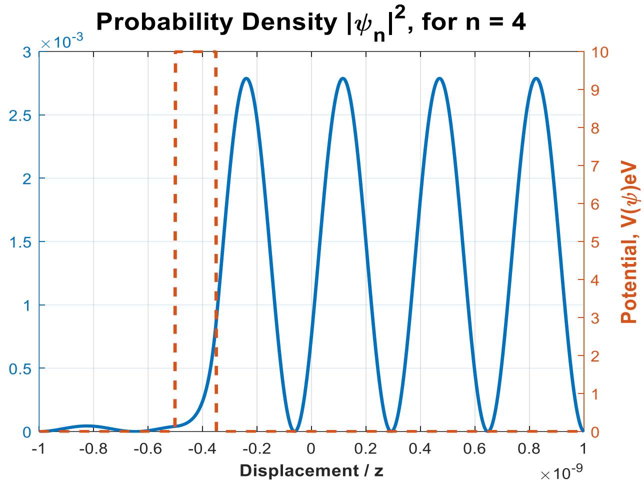


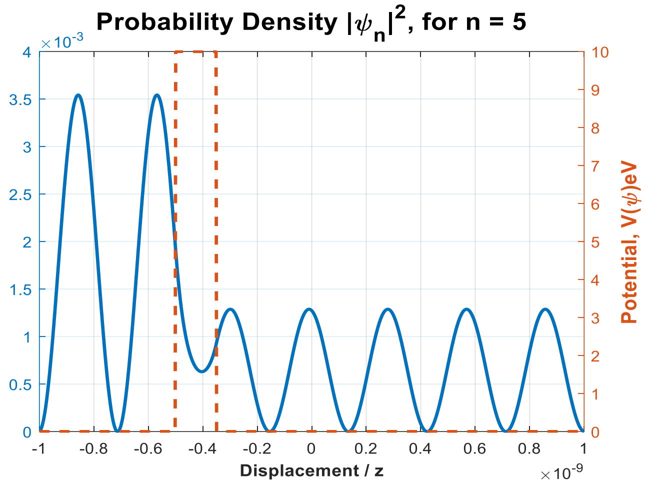
Probability Density $|\psi_n|^2$, for n = 2

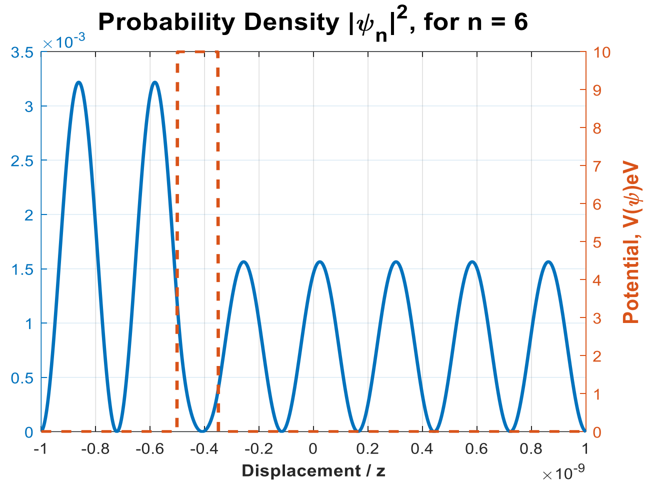


Probability Density $|\psi_n|^2$, for n = 3

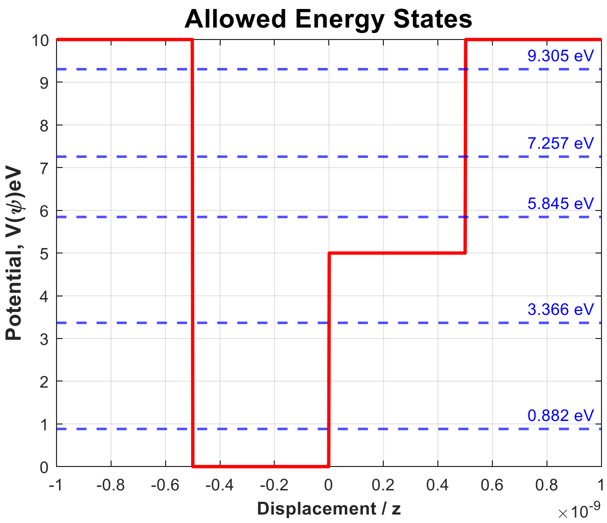


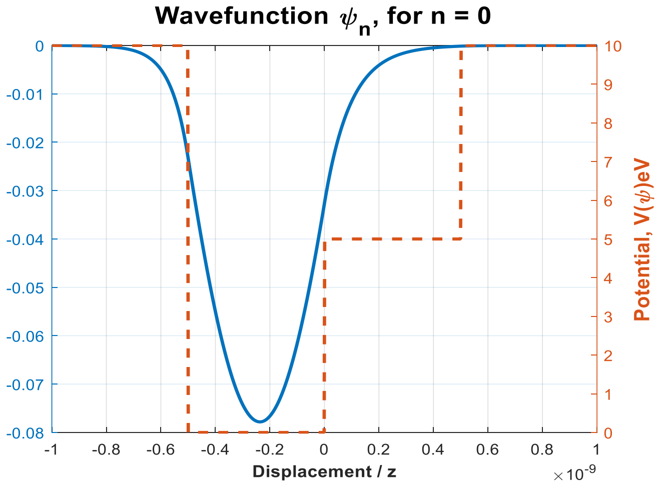


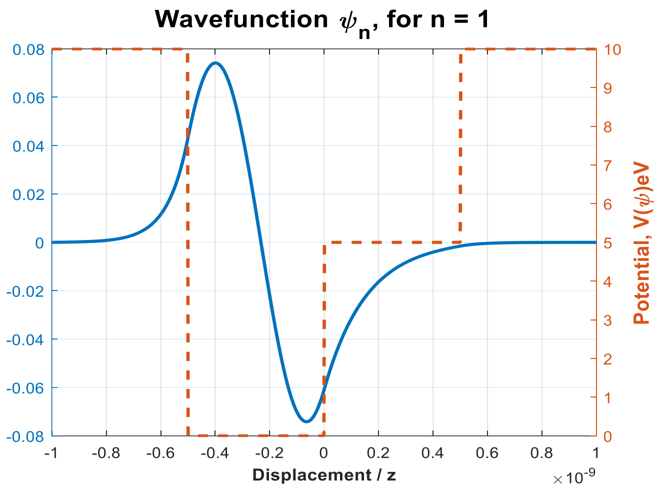


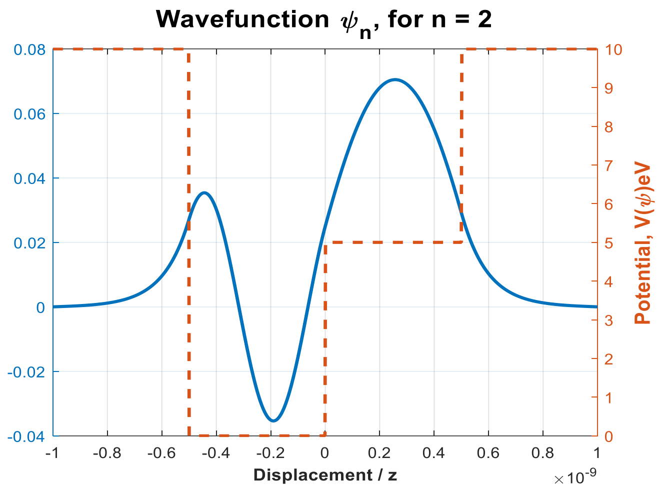


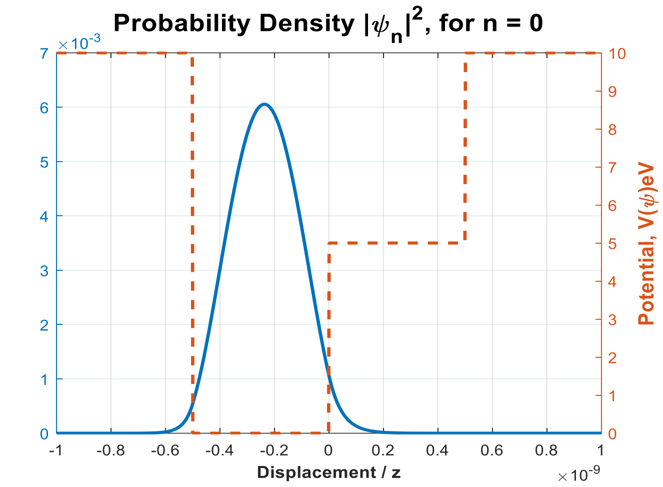
### 3.2.3 Stepped Potential Well

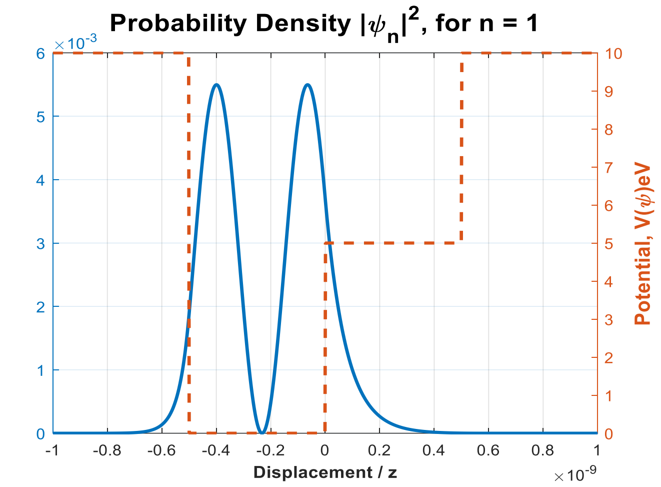


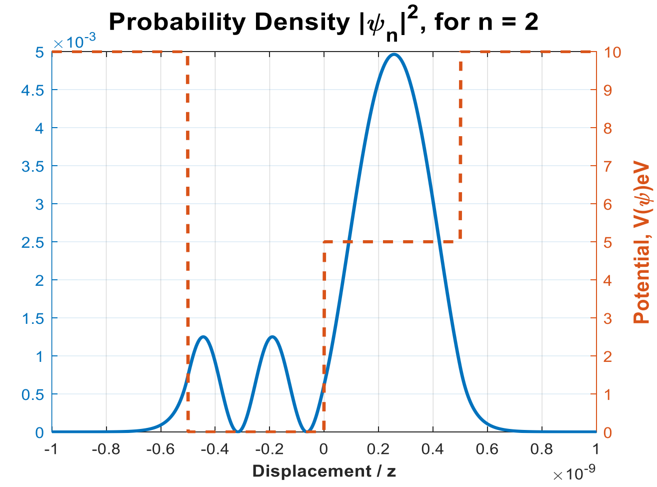



Wavefunction $\psi_n$, for n = 1

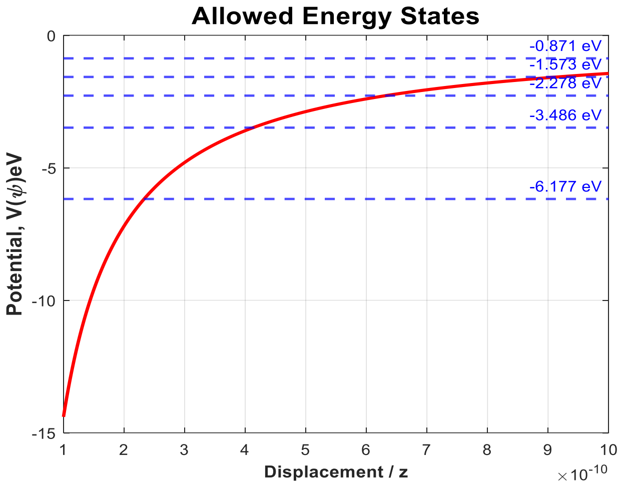

Wavefunction $\psi_n$, for n = 2

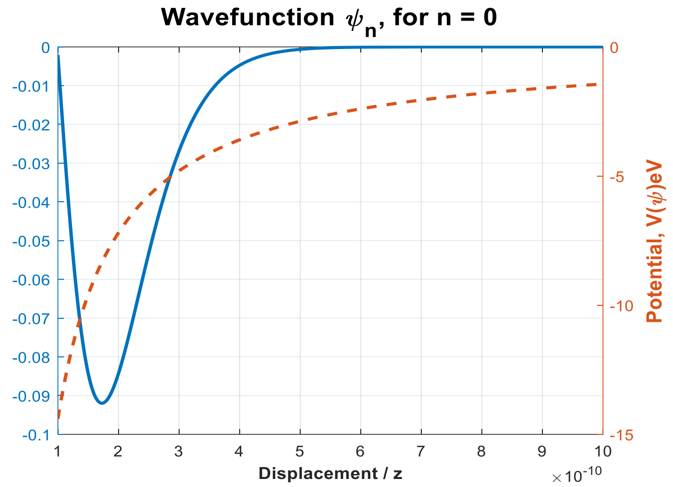

Probability Density $|\psi_n|^2$, for n = 0

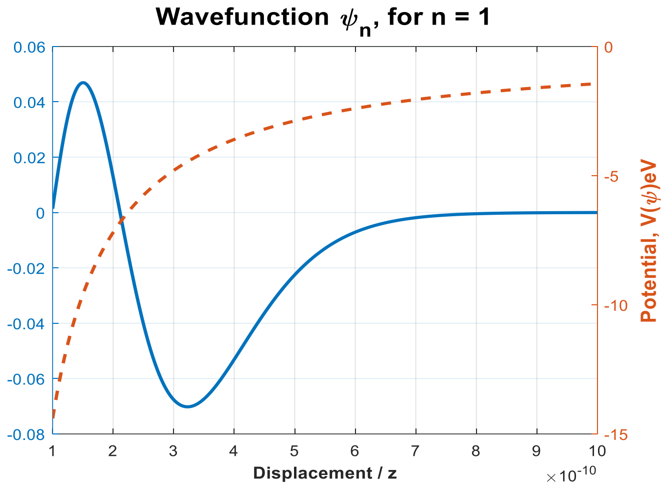

Probability Density $|\psi_n|^2$, for n = 1

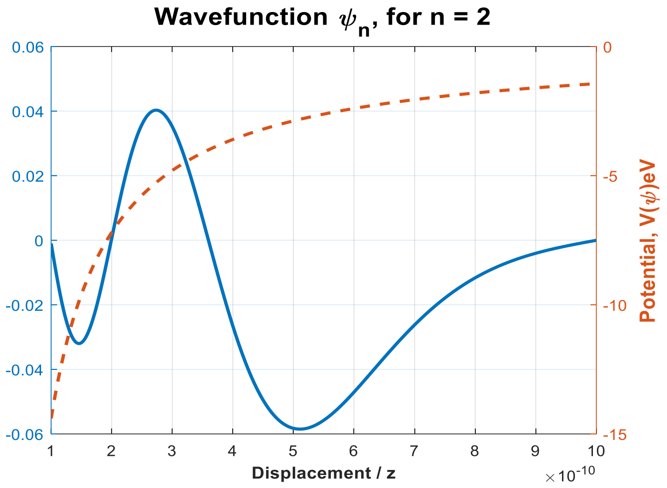

Probability Density $|\psi_n|^2$, for n = 2

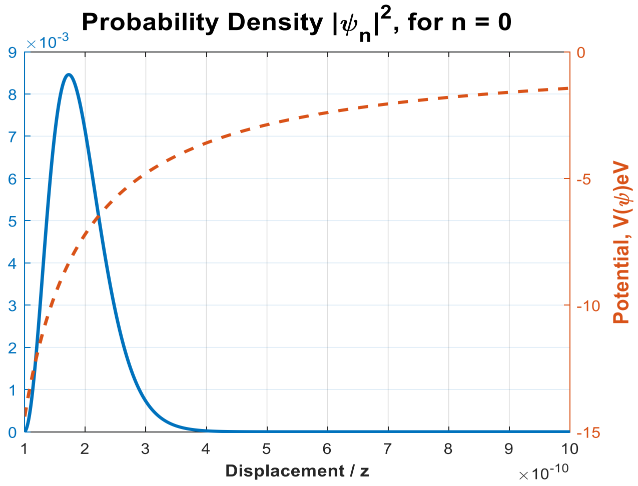
## 3.3 One-Electron Atom Potential

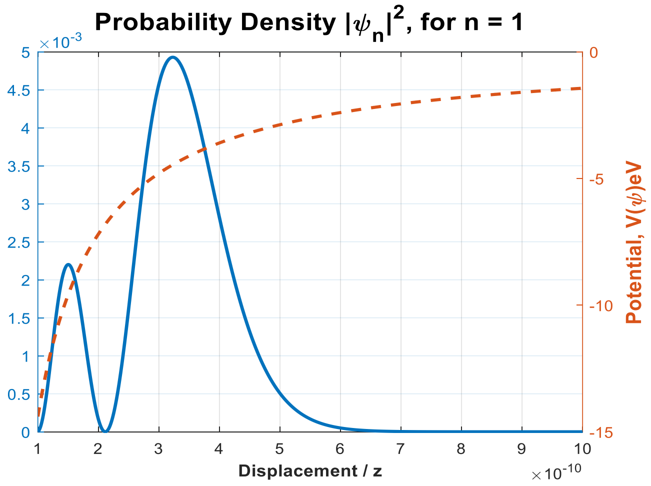

Allowed Energy States


Wavefunction $\psi_n$, for n = 0

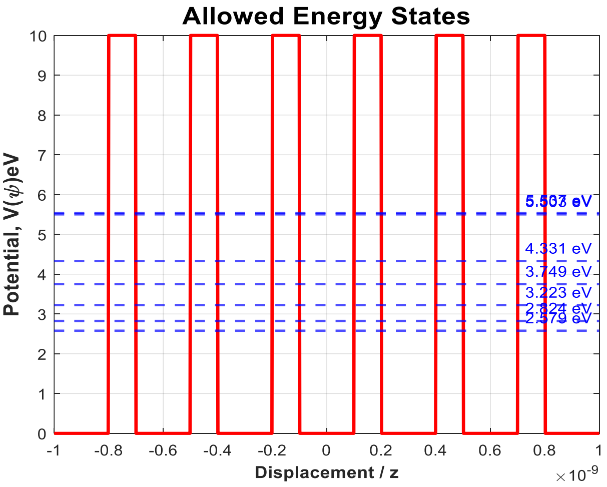

Wavefunction $\psi_n$, for n = 1

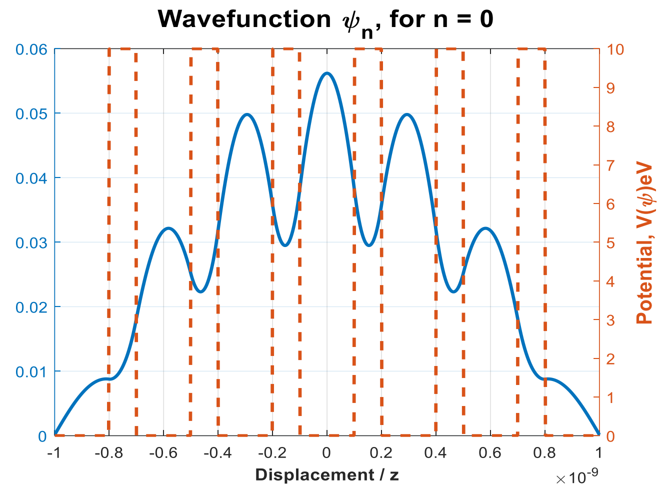

Wavefunction $\psi_n$, for n = 2

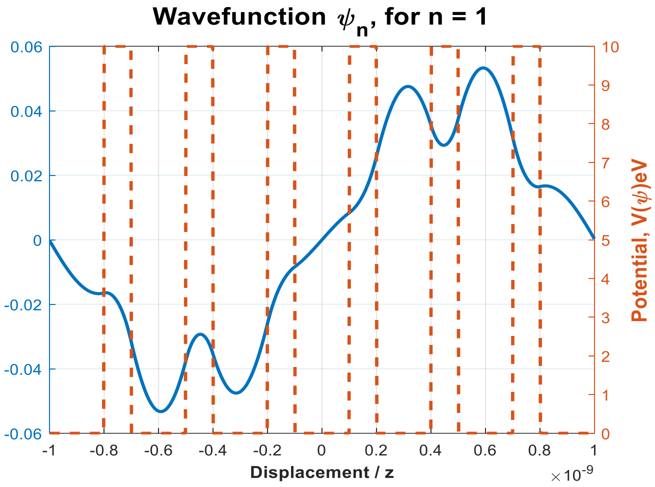

Probability Density $|\psi_n|^2$, for n = 0

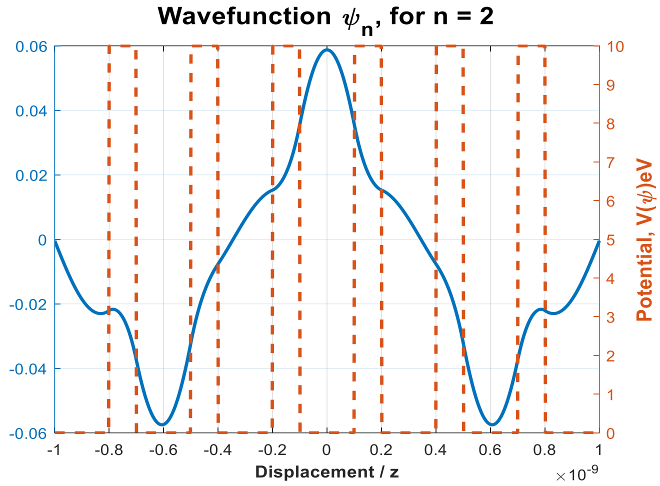

Probability Density $|\psi_n|^2$, for n = 1

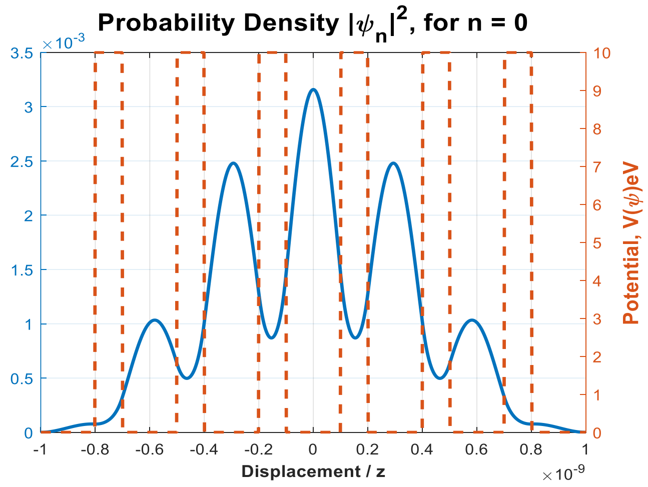
## 3.4 Finite Series of Square Potential Wells

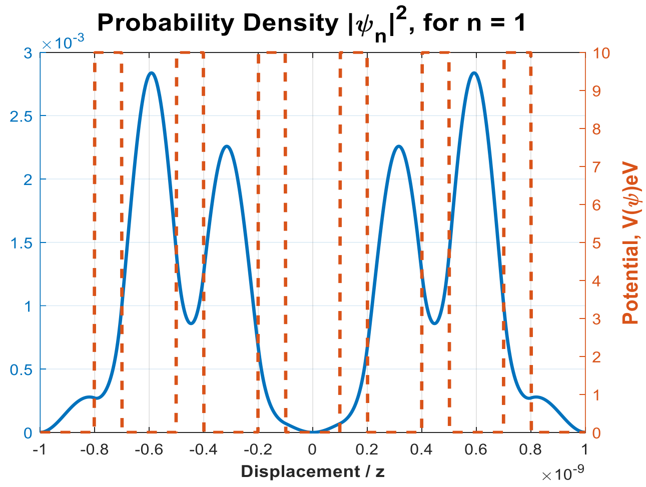

Allowed Energy States


Wavefunction $\psi_n$, for n = 0


Wavefunction $\psi_n$, for n = 1


Wavefunction $\psi_n$, for n = 2


Probability Density $|\psi_n|^2$, for n = 0


Probability Density $|\psi_n|^2$, for n = 1

## 3.5 Observations and Inference

From the various potential functions that were simulated, key observations were made regarding the one-dimensional simulation accuracy of the algorithm.

From the simulation of the symmetric wells, in the case of the square potential well, the simulated results closely match the closed form solutions. In the regions where a finite potential exists, small portions of the probability density and wave function exist and is in line with what is expected for a finite potential well. Increasing the potential barrier to large numbers results in a significant decrease in the region of the density function in the potential regions, hence the wavefunction becomes constricted to the region with no potential function. This behavior is observed throughout in all the types of potential functions that were simulated and is in line with rational expectation. Furthermore, for the more complex potential functions the energy eigen-values and wavefunctions obtained were compared against data on similar simulations performed with alternative numerical methods: good correlation between the simulated results was observed. Hence, it can be assumed that the designed algorithm is able to simulate the TISE and provide solutions with a reasonable margin of accuracy.

However, the results obtained with the simulation of the one-electron atom potential show large variances in the obtained energy eigen values to that of closed form solutions. These anomalous results need to be further investigated to identify the limitations in the model.

# 4 References

Evans, M., 2013. *Solving the Schrödinger Equation with Numerov's Algorithm.* [Online]
Available at: https://mtdevans.com/2013/07/solving-the-schrodinger-equation-with-numerovs-algorithm/index.html#mjx-eqn-eqsch

Kalnay, E., 2012. *https://www2.atmos.umd.edu/~ekalnay/syllabi/AOSC614/.* [Online]
Available at: https://www2.atmos.umd.edu/~ekalnay/syllabi/AOSC614/NWP-CH03-2-2.pdf

Kesserwan, H., 2011. *Ultrafast Magnetization dynamics of magnetic nanostructures,* s.l.: Universite de Strasbourg.

Robertson, D. G., 2009. *Solving the Time-Independent Schrodinger Equation.* [Online]
Available at: http://faculty.otterbein.edu/DRobertson/compsci/tise-stud.pdf

Young, P., 2007. *Physics 115/242, Eigenvalues of the Schr¨odinger Equation: the Shooting Method.* [Online]
Available at: http://physics.ucsc.edu/~peter/242/shoot.pdf

Young, P., 2009. [Online]
Available at: http://physics.ucsc.edu/~peter/242/numerov.pdf

# 5 Appendix A

MATLAB Code for 1D TISE

```
% 208643N - Kaushan G. Ranasinghe - 25 MAY 2021
%============================================================================
% TISE Simulation using FDM with Central Difference - 1D
%============================================================================

clear all
close all

% Simulation of a electron in various potential functions

%% ------Initiating the constants------------------------------------------

hbar = 1.055e-34;
m = 9.11e-31; % Rest mass of electron
eV = 1.602e-19;

%% ------Initiating the stepped lattice points------------------------------

L = 10e-10; % Potential well width
z_min = -L; % -ve Z range
z_max = L; % +ve Z range
n = 1000; % Number of lattice points
z = linspace(z_min,z_max,n); % Vector of 'z' lattice points
d = z(2) - z(1); % Successive lattice spacing

%% ------Generating the potential energy matrix----------------------------

Wh = 10; % Well Height
Vo = Wh*eV; % Max finite potential
B1 = -L/2; % -ve well boundary
B2 = L/2; % +ve well boundary

% -------Generation of potential function----------------------------------

%Select potential well type

%Symmetric Wells

%V = ones(1,n)*Vo; V(z>= B1 & z<=B2) = 0; % Square Well
%V = Vo/(L^2)*(z.^2); % Harmonic Well
%V = ones(1,n)*Vo; V(z>= B1 & z<=B2) = Vo/(B2^2)*(z((z>= B1 & z<=B2)).^2);
% Truncated Parabolic Well
%V((z<=B1)) = Vo/(B2-L)^2*((z(z<=B1)) + L).^2; V(z>= B1 & z<=B2) =
0;V((z>=B2)) = Vo/(L-B2)^2*((z(z>=B2)) - L).^2; % Quadratic Well

%Asymmetric Wells

%V = ones(1,n)*Vo; V(z>= B1 & z<=0) = 0; % Offset Square Well
%V = zeros(1,n)*Vo; V(z>= -0.5e-9 & z<=-0.35e-9) = Vo; % Offset Tunnel
%V = ones(1,n)*Vo; V(z>= B1 & z<=0) = 0; V(z>=0 & z<=B2) = 0.5*Vo; %
Stepped Well
```

```
%One-Electron Atom
%
% Wh = 0; % Well Height
% epsi=8.854e-12; % Permittivity of free space
% z_min = 1e-10; % Minimum electron radius
% z = linspace(z_min,z_max,n);
%
% V = -((eV^2)./(4*pi*epsi*z)); % Potential function

% Series of Square Potential Wells

Wd = 0.2e-9; % Well width
Ws = 0.1e-9; % Well spacing
V=ones(1,n)*Vo;
q = 7; % Number of Series Wells
z0 = z(:,1);
for ser = 1:q

   z1 = z0;
   z2 = z1+Wd;
   V(z1<=z & z<=z2) = 0;
   z0 = z2+Ws

end

%----------------------------------------------------------------------------

Vn = eye(n,n);
Vp = V'.*Vn;

%% ------Generating the kinetic energy matrix using FDM--------------------

K = eye(n,n); % Initiating the kinetic energy matrix
K = K*(-2); % Setting the main diagonal
for t = 1:n-1 % Setting the two flanking diagonals
   K(t,t+1) = 1;
   K(t+1,t) = 1;
end

%% ------Generating the Hamiltonian----------------------------------------

H = ((-(hbar^2)/(2*m*(d^2)))*K) + Vp ;

%% ------Generating the energy eigen values/states-------------------------

[U,EV] = eig(H); % Obtaining the wavefunctions and the eigen energies

P = U.^2; % Probability density of wavefunctions

E = diag(EV); % Accumulating the energy states to a column vector

Ec = E./(eV);

Ve = (1/eV).*V;

disp(E(1))
disp(E(2))
```

```
%% -------Plotting the Functions-------------------------------------------

%---------Plotting the allowable energy states-----------------------------

Ea = Ec(Ec<Wh)

figure
plot(z',Ve,'r','Linewidth',2);
hold on
title('Allowed Energy States','Fontsize',16)
ylabel('Potential, V(\psi)eV','Fontsize',13,'FontWeight','bold')
xlabel('Displacement / z','FontWeight','bold')

for r = 1:length(Ea)

    yline(Ea(r),'b--',sprintf('%.3f eV',Ea(r)),'Linewidth',1.5);

end
grid on

%---------Plotting the Wavefunction / Probabilities------------------------

for w = 1:length(Ea)

    count = w-1;
    figure
    title("Wavefunction \psi_n, for n = " +count,'Fontsize',16)
    xlabel('Displacement / z','FontWeight','bold')
    yyaxis right % Plot the potential function

    plot(z',Ve,'--','Linewidth',2) ;
    ylabel('Potential, V(\psi)eV','Fontsize',13,'FontWeight','bold')
    hold on
    grid on
    yyaxis left
    plot(z',(U(:,w)),'Linewidth',2);

end

for b = 1:length(Ea)

    count = b-1;
    figure
    title("Probability Density |\psi_n|^2, for n = " +count,'Fontsize',16)
    xlabel('Displacement / z','FontWeight','bold')
    yyaxis right % Plot the potential function

    plot(z',Ve,'--','Linewidth',2) ;
    ylabel('Potential, V(\psi)eV','Fontsize',13,'FontWeight','bold')
    hold on
    grid on
    yyaxis left
    plot(z',(P(:,b)),'Linewidth',2);

end
```