

Numerical Solution of the Time-Independent 1-D Schrödinger Equation

Liam Kavanagh, 17332323

Due 14th October 2019

Abstract

This computational project involves the use of the three-point Numerov algorithm and Shooting Method to find energy eigenvalues and eigenfunctions of the wave equation of a particle in an infinite square well. The accuracy of the numerical results is compared to the analytical results obtained by solving the non-dimensionalised Schrödinger equation. The normalised eigenfunctions are found and the Heisenberg uncertainty relation verified for the position and momentum of the particle. The eigenstates of a harmonic potential bounded by infinite walls are found using a modified version of the same code, and the uncertainty relation verified for this new potential. The differences in energy levels of the harmonic potential were plotted and results drawn.

1 Introduction and Theory

The Schrödinger Equation

In Quantum Mechanics, the Schrödinger equation describes the evolution of state of a system in time in terms of its wavefunction $\Psi(x, t)$. This wavefunction contains all of the possible information that can be known about the system (at least according to the Copenhagen interpretation)¹.

Generally, the Schrödinger equation is given in terms of the wavefunction and the Hamiltonian operator of the state.

$$i\hbar \frac{\partial}{\partial t} |\Psi(x, t)\rangle = \hat{H} |\Psi(x, t)\rangle$$

We will discuss the time-independent, or steady state, Schrödinger equation (TISE). In this case, the equation is given by

$$\hat{H} |\psi_E(x)\rangle = E |\psi_E(x)\rangle$$

This equation relates the Hamiltonian operator on the wave eigenfunction ψ_n to its associated eigenvalue E_n . Writing \hat{H} in position space, we have the equation

$$\left[-\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} + V(x) \right] \psi_n(x) = E_n \psi_n(x)$$

This is a second order linear differential equation, which can be solved analytically with appropriate boundary conditions.

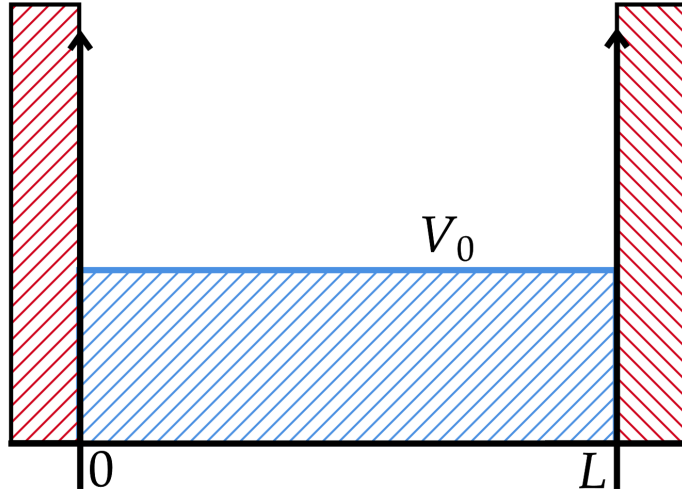


Figure 1: The infinite square well

The Infinite Square Well

We define the infinite square well potential as

$$V(x) = \begin{cases} V_0, & 0 < x < L \\ \infty, & x \leq 0, x \geq L \end{cases}$$

To solve the TISE for this potential, we non-dimensionalise the differential equation.

$$\frac{\partial^2 \psi_n(\tilde{x})}{\partial \tilde{x}^2} + \gamma^2 (\epsilon_n - \nu(\tilde{x})) \psi_n(\tilde{x}) = 0$$

where we have defined $\gamma^2 = \frac{2mL^2V_0}{\hbar^2}$, $\epsilon_n = E_n/V_0$, and $\nu(\tilde{x}) = V(\tilde{x})/V_0$ and rescaled x as $\tilde{x} = x/L$.

Thus, our square well now runs from $0 < \tilde{x} < 1$ and our potential $\nu(\tilde{x})$ takes values between -1 and 1 .

The Numerov Algorithm

We can integrate equations such as the one above using the Numerov algorithm and discretising the function ψ into $h = 1/(N-1)$, where h is the step size and N is the number of steps. The Numerov algorithm yields

$$\psi_{j+1} = \frac{2(1 - (5/12)h^2k_j^2)\psi_j - (1 + (1/12)h^2k_{j-1}^2)\psi_{j-1}}{1 + (1/12)h^2k_{j+1}^2}$$

where $\psi_j = \psi_n(\frac{j}{N-1})$ and $k_j^2 = \gamma^2(\epsilon_n - \nu(\tilde{x}_j))$.

Analytical Solutions to the Non-Dimensionalised TISE

We wish to solve an equation of the form

$$\frac{d^2 \psi(\tilde{x})}{d\tilde{x}^2} + k^2 \psi(\tilde{x}) = 0$$

where $k^2 = \gamma^2(\epsilon + 1)$. The characteristic equation of this ODE is $\lambda^2 + k^2 = 0$. Thus, $\lambda = \pm ik$, and our solutions are

$$\psi(\tilde{x}) = A \sin(k\tilde{x}) + B \cos(k\tilde{x})$$

The constants A and B can be fixed using the boundary conditions $\psi(0) = 0$ and $\psi(1) = 0$, since we require the wavefunction to vanish on the well boundaries. Thus, we find

$$0 = A \sin(0) + B \cos(0) \implies B = 0$$

$$0 = A \sin(k) \implies k = n\pi$$

$$\implies \psi(x) = A \sin(n\pi\tilde{x})$$

We require that the wavefunction satisfies the normalisation condition

$$\int_0^1 |\psi(\tilde{x})|^2 d\tilde{x} = 1$$

since this integral gives the probability of finding the particle anywhere in the allowed region, which should of course be 1. We can thus solve for A , the normalisation constant.

$$\begin{aligned} \int_0^1 |\psi(\tilde{x})|^2 d\tilde{x} &= 1 \\ \int_0^1 |A \sin(n\pi\tilde{x})|^2 d\tilde{x} &= 1 \\ A^2 &= \frac{1}{\int_0^1 \sin^2(n\pi\tilde{x}) d\tilde{x}} \\ A &= \sqrt{2} \end{aligned}$$

Thus, our wavefunction $\psi_n(\tilde{x}) = \sqrt{2} \sin(n\pi\tilde{x})$. Using our equation $k_n^2 = \gamma^2(\epsilon_n + 1)$ and the fact that $k_n = n\pi$, we have our allowed non-dimensional energies for $n > 0$:

$$\epsilon_n = \frac{\pi^2 n^2}{\gamma^2} - 1$$

We can see that ϵ must be strictly greater than -1 .

The first 10 allowed energies, calculated analytically, are

Table 1: Allowed energies calculated analytically

n	ϵ_n (analytical)
1	-0.95065197799
2	-0.80260791197
3	-0.55586780195
4	-0.21043164791
5	0.23370055013
6	0.77652879219
7	1.41805307826
8	2.15827340834
9	2.99718978244
10	3.9348022005

The Heisenberg Uncertainty Relation

The Heisenberg uncertainty relation gives a condition on the uncertainties of the position and momentum of a particle, $\Delta x \Delta p \geq \frac{\hbar}{2}$. These values are given by $\Delta x = \sqrt{\langle x^2 \rangle - \langle x \rangle^2}$ and $\Delta p = \sqrt{\langle p^2 \rangle - \langle p \rangle^2}$. A function is defined to perform these calculations, using the fact that $\langle \tilde{x} \rangle = 0.5$ for any \tilde{x} . In non-dimensionalised units, $\Delta \tilde{x} \Delta \tilde{p} \geq \frac{1}{2}$. We also have $\langle \tilde{p} \rangle = 0$ for stationary states, and

$$\langle \tilde{x}^2 \rangle = \int_0^1 \tilde{x}^2 |\psi(\tilde{x})|^2 d\tilde{x}$$

$$\Delta \tilde{p} = \sqrt{-\int_0^1 \psi(\tilde{x}) \frac{d^2 \psi}{d\tilde{x}^2} d\tilde{x}}$$

The second derivative was calculated using the finite difference scheme.

2 Computation and Results

We can numerically solve for these energy eigenvalues using the Numerov algorithm. Suitable values for N and γ^2 are 1000 and 200 respectively. We define a function (Fig. 2) which takes a trial energy E and computes the values of the wavefunction up to the boundary at $\tilde{x} = 1$. We use another function (Fig. 3) to increase the trial energy by an amount $dE = 0.01$ and calculate the wavefunction at the boundary again. We do this until the sign of one is different to the sign of the other, in which case we let $dE = -dE/2.0$ and continue until dE is below a certain tolerance (in this case, 10^{-10}). This is called the 'Shooting' method.

```
def wavefunction(E, gamma_sq, N):
    psi = np.zeros(N)
    psi[0] = 0
    psi[1] = 10**(-4)
    steps = 1.0/(N-1)
    pot = potential(N)
    i = 1
    while i < N-1:
        part_a = 2*(1 - (5.0/12.0)*(steps**2)*(k_squared(gamma_sq, E, pot[i])))
        part_b = 1 + (1.0/12.0)*(steps**2)*k_squared(gamma_sq, E, pot[i-1])
        part_c = 1 + (1.0/12.0)*(steps**2)*(k_squared(gamma_sq, E, pot[i+1]))
        psi[i+1] = (part_a*psi[i] - part_b*psi[i-1])/(part_c)

        i = i + 1

    return psi
```

Figure 2: The function `wavefunction` which returns an array `psi` using the Numerov algorithm.

A suitable trial energy would be $E = -1$, since we know we must have solutions of $\epsilon_n > -1$ and will thus definitely hit the first allowed energy level with this starting point.

Running these functions (within some reasonable boilerplate code) results in the first allowed energy level of $\epsilon_1 = -0.950651978030801$. The wavefunction corresponding to this energy was plotted (Fig. 4). This result agrees strongly with our analytical solution. The first 10 numerically-computed energies are shown in Table 2.

```

def eigenstate(E, dE, gamma_sq, N):
    #wavefunction(E, N)
    psi_old = wavefunction(E, gamma_sq, N)[N-1]

    E = E + dE
    while np.fabs(dE) > 10*(-10):
        psi_new = wavefunction(E, gamma_sq, N)[N-1]

        if psi_old*psi_new < 0:
            dE = -dE/2.0

    E = E + dE
    psi_old = psi_new

    return E

```

Figure 3: The function `eigenstate` which returns an allowed energy E for the wavefunction `psi`.

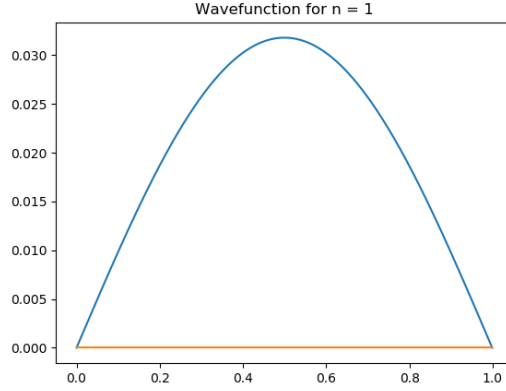


Figure 4: The non-normalised wavefunction for the first energy level ϵ_1 .

Table 2: Allowed energies calculated numerically. The square brackets indicate where the value begins to deviate from the analytical results of Table 1. The error is of order $\leq 10^{-7}$.

n	ϵ_n (numerical)
1	-0.95065197[8030801]
2	-0.80260791[2049]
3	-0.5558678019[79]
4	-0.2104316479[71]
5	0.2337005[49826]
6	0.77652879[1264]
7	1.41805307[589]
8	2.15827340[312]
9	2.9971897[7176]
10	3.934802[18045]

The wavefunctions must still be normalised. We use a simple function which integrates the normalisation condition in order to calculate the normalising coefficient. We use the `simps` function from SciPy to approximate the integral using Simpson's Rule. Each energy level yields a normalising coefficient of approximately 1.4142, which agrees with our analytical solution of $\sqrt{2}$. We can also read the normalisation from the plots in the Appendix.

The uncertainty relations for these energy levels was computed and shown in Table 3. All of the energy eigenstates have an uncertainty of greater than one half, verifying the uncertainty relation.

Table 3: Uncertainty relations for energy eigenvalues $n = 1, \dots, 10$.

n	$\Delta\tilde{x}\Delta\tilde{p}$
1	0.56729589372
2	1.66861978032
3	2.62457184359
4	3.55443955355
5	4.47450836799
6	5.38979208084
7	6.3023688071
8	7.21325877622
9	8.12302201572
10	9.03199142948

The constant potential was then replaced with a harmonic potential bounded by infinite walls.

$$\nu(\tilde{x}) = 8(\tilde{x} - 0.5)^2 - 1$$

The value γ^2 was changed to 1000 to better represent the new potential. The energy eigenstates and uncertainties are shown in Table 4. The uncertainty for $n = 1$ is almost exactly 0.5 (within 1%), which corresponds to equality of the uncertainty relation for that energy eigenvalue. The difference in energy level is shown in the Appendix. This plot is linear, which corresponds to constant differences in energy levels for the harmonic potential. Indeed, analytically, the energy levels of the harmonic oscillator are given by

$$E_n = \hbar\omega \left(n + \frac{1}{2} \right)$$

However, at larger energies the plot deviates, showing a non-linear difference in energy levels. This corresponds to the harmonic potential behaving similarly to the square well for larger energies.

Table 4: Energy eigenvalues $n = 1, \dots, 10$ for the harmonic potential and associated uncertainty.

n	ϵ_n	$\Delta\tilde{x}\Delta\tilde{p}$
1	-0.9105572807043794	0.499508394604
2	-0.731671834961	1.49849724555
3	-0.55278624855	2.49747569139
4	-0.373898976371	3.4964523599
5	-0.194997625425	4.49549617597
6	-0.0160098636895	5.49500189924
7	0.163377251104	6.49655779419
8	0.344165238962	7.50418245912
9	0.528691549078	8.52217194992
10	0.720875510648	9.54539288941

3 Conclusions and Closing Remarks

The time-independent Schrödinger equation in one dimension was solved analytically for the square well potential and a Python script written to calculate the energy eigenvalues numerically. The two methods correlated strongly, to within an error of order 10^{-7} . The accuracy could not be improved due to the introduction of floating point errors for smaller tolerances.

The Heisenberg uncertainty relation was verified for the wavefunctions of these energies.

The square well potential was then replaced with a harmonic potential and the energy eigenvalues computed. The uncertainty relation was also verified for the corresponding wavefunctions.

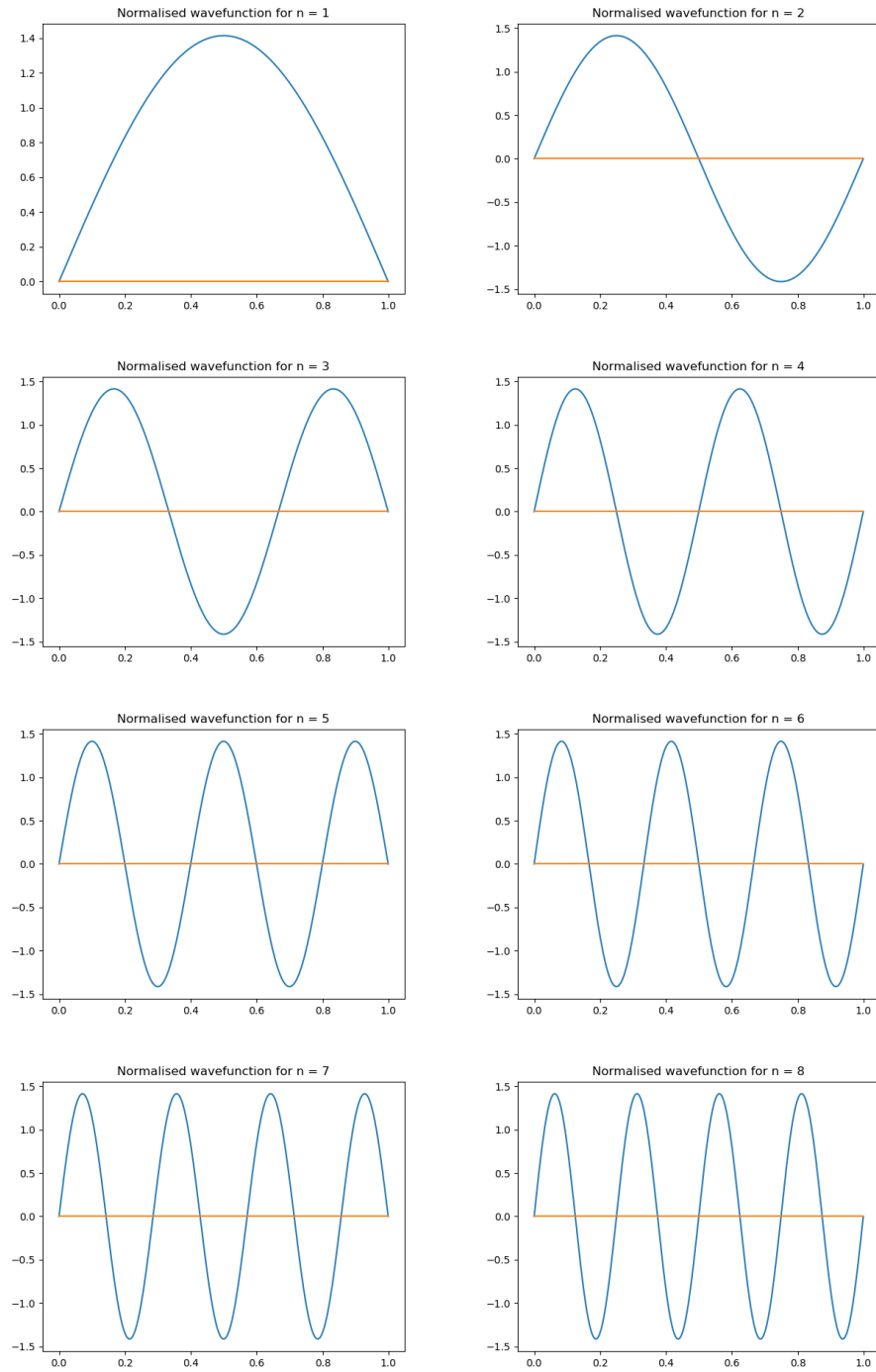
A log-log plot of the difference in energy levels of the harmonic potential was made for each n , showing a linear relationship for small n and a change back to the square well regime for larger energies.

References

- 1: Griffiths, David J., *An Introduction to Quantum Mechanics*

Appendix

Figure 5: Normalised wavefunctions for $n = 1, \dots, 10$ for the square well.



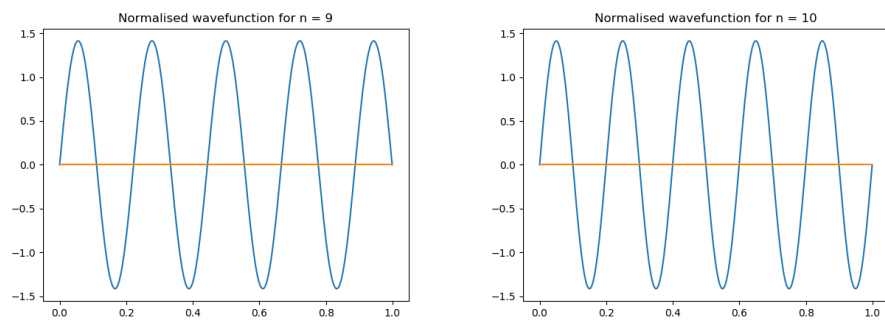
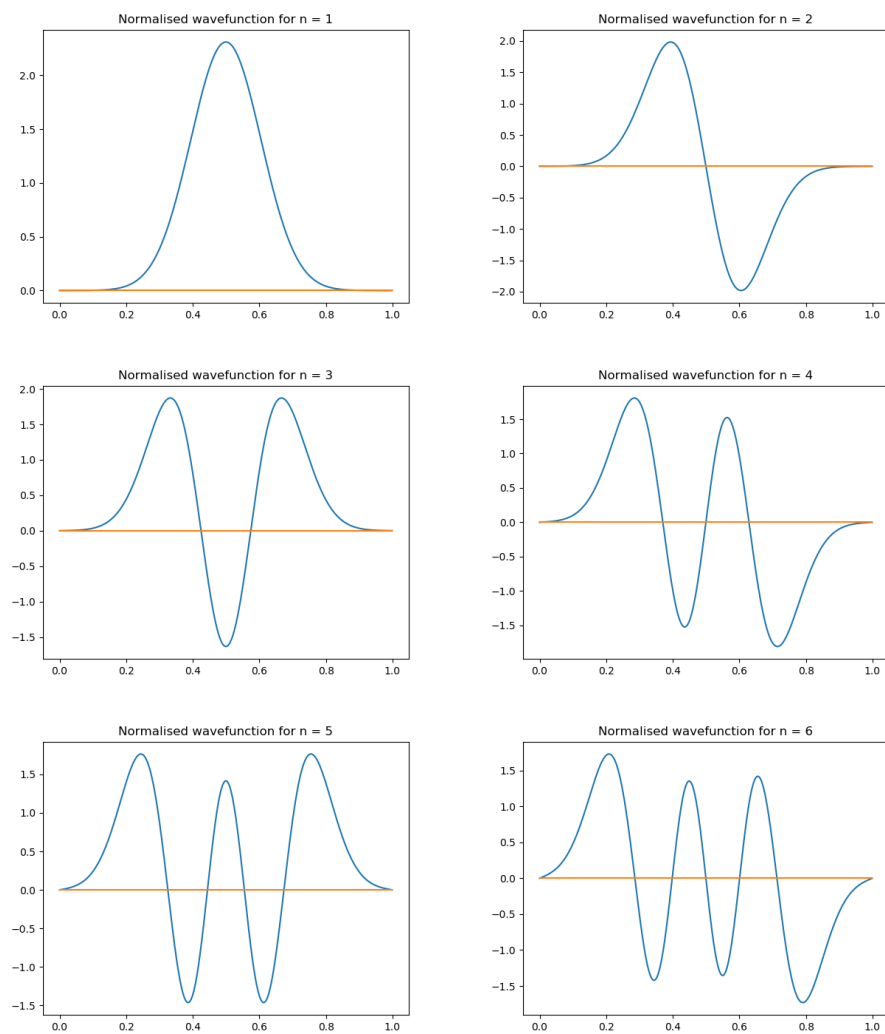


Figure 6: Normalised wavefunctions for $n = 1, \dots, 10$ for the harmonic potential.



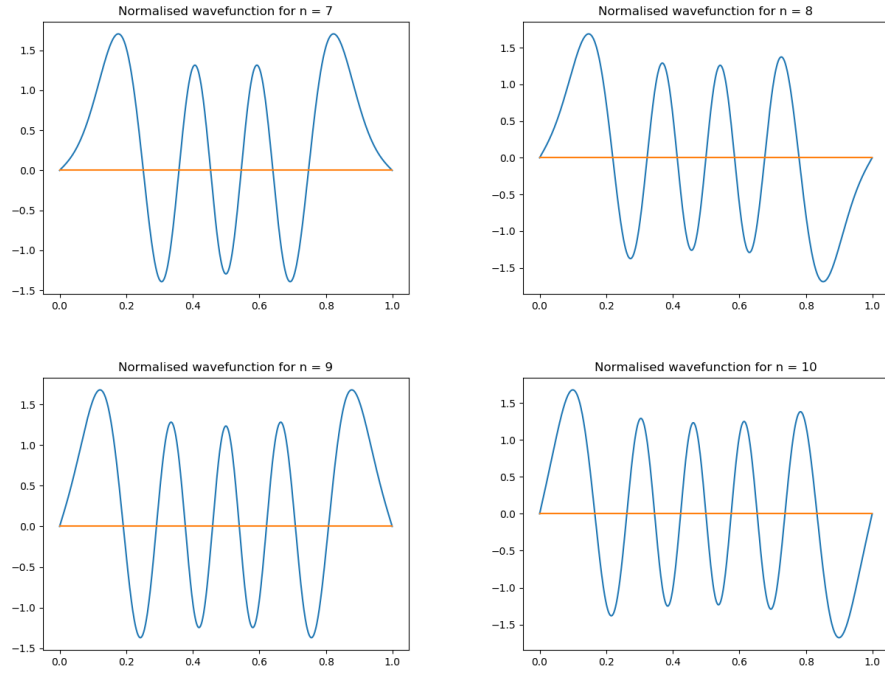


Figure 7: The energy level difference for the square well and harmonic potential on a log-log plot.

