# Numerical Solution of the Schrödinger Equation in Python

## Beginning of Quantum Mechanics and Ultraviolet Catastrophe

The beginning of quantum mechanics can be attributed to the development of Planck's radiation theory in 1900 [1]. The blackbody radiation curve was known experimentally, but there was no theoretical background that explains this shape [2]. Blackbody is an idealized physical body that emits the maximum amount of heat for its absolute temperature. Moreover, it is a perfect absorber and emitter as it can absorb any electromagnetic radiation regardless of its frequency or angle of incidence [3]. The physical model of a blackbody can be described as electromagnetic waves in a cavity at thermodynamic equilibrium with the cavity walls. In the classical model, the waves continually exchange their energies with the walls. However, the result of this classical model, known as Rayleigh-Jeans law, doesn't match the experimental results. This model's prediction is that the radiation intensity becomes infinitely high as the frequency increases. The name for this problem is *ultraviolet catastrophe* [2]. This would mean that the thermal equilibrium between matter and radiation is impossible to achieve at any temperature because the matter would radiate energy until its temperature reaches the absolute zero [1].

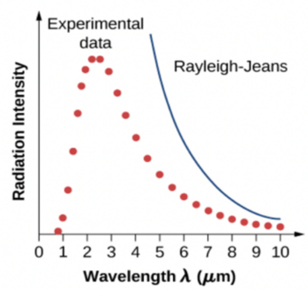


*Figure 1 Rayleigh-Jeans [2]*

Planck solved this problem by assuming that the cavity radiation comes from the atomic oscillations inside the cavity walls and that these oscillations can only have discrete values of energy (quantized energy). The radiation can then exchange energy with the cavity walls only in certain portions – quanta [2]. The word "quanta" is plural of the word quantum, which means the smallest amount of any physical entity that is involved in an interaction [4]. In this case that would describe the smallest amount of energy exchanged between the radiation inside the cavity walls and the walls. Some other quanta are photons which are quanta of light, and phonons which are quanta of the vibration of the lattice.

Quantum mechanics continued to grow with the new developments made by the Einstein's theory behind the photoelectric effect in which he used the quantization of energy and momentum, Compton's effect, de Broglie's hypothesis of wave-particle duality. In 1913, Bohr proposed a model of atom that contradicted some of the classical mechanical concepts. In his model, atom consists of stationary orbits in which electrons don't radiate energy. When an electron moves from one orbit to another it emits or absorbs one photon that has an energy equal to the difference between the two energy levels. However, this theory was unable to explain why this happens and how this would apply to some more complex atoms and molecules. In 1926, Schrödinger constructed wave mechanics, in which the values of physical entities can be found by calculating the eigenvalues of linear differential operators [1].

## Implementation of the Schrödinger's equation solution in Python

Python is a high-level programming language, which makes it a suitable programming language for scientific purposes. The implementation in Python is simple, its syntax is clean and similar to pseudo-code making it intuitive and easy to read. Moreover, there are various library modules in Python that can help with simplifying the computing process [5]. One of these library modules is *Matplotlib* which offers multiple visualization technique that can be done in only a few lines of code. This library module can be used for data visualization which can be useful for any scientific and non-scientific discipline. Furthermore, there is also an interactive Python interpreter, that allows real-time code development and experimentation [5].

Due to these reasons, Python programming language was chosen for the implementation of the Schrödinger's equation solution. This project was done in Jupyter notebook, which not only provides the interactive interpreter, but it also allows the input of markdown language. Therefore, both the text and the code could be in the same file. Jupyter notebook also provides the input of Latex equations, which makes this environment the most suitable for this problem as it offers a visualization of mathematical equations.

## Deriving the Schrödinger's equation

For the purpose of implementing the Schrödinger's equation in Python it is beneficial to understand how this equation is derived. The Schrödinger's equation comes from the Energy conservation laws in which the total energy is equal to the sum of kinetic and potential energy.

$$\frac{p^2}{2m} + V(x,t) = E$$

After that, the momentum $p$ and the total energy $E$ are substituted by their values from Planck Einstein relations in which the energy and momentum are quantized.

$$E = \hbar\omega, \qquad p = k\hbar$$

$$\frac{\hbar^2}{2m}k^2 + V(x,t) = \hbar$$

Then the equation is multiplied by the $\psi(x,t)$ wave function to get the wavefunction as a result of the Schrödinger's equation.

$$\frac{\hbar^2}{2m}k^2\psi(x,t) + V(x,t)\psi(x,t) = \hbar\omega\psi(x,t)$$

Parameter $k$ and $\omega$ can then be replaced by the differential operators acting on the wavefunction $\psi(x,t)$. To substitute $k^2$ and $\omega$ by the derivatives, a complex form of the wavefunction needs to be used. When these values are substituted the result is a one-dimensional Schrödinger's equation.

$$\omega \to i\frac{\partial}{\partial t}$$

$$k^2 \to -\frac{\partial^2}{\partial^2 x}$$

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

As this is only a one-dimensional Schrödinger's equation a second order partial derivative needs to be replaced by the values in the Laplace equation for the three-dimensional coordinates.

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

$$\nabla^2 = \frac{\partial^2}{\partial^2 x} + \frac{\partial^2}{\partial^2 y} + \frac{\partial^2}{\partial^2 z}$$

From that the probability density function can be derived. The probability density function needs to be normalized in this way for the sum of all probabilities to be equal to one.

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}|\psi(x,y,z;t)|^2 2dx\,dy\,dz = 1$$

## Particle in a one-dimensional box

The first solution of the Schrödinger's equation that was implemented in Python was particle in a box model. This is a hypothetical situation in which multiple conditions that are impossible in realistic situations must be met. In this model a particle is trapped in a one-dimensional "box". This "box" can be imagined as a one axis on a coordinate system on which a particle can move. However, this axis is not infinite as it is bounded by two infinitely high potential walls. Since the potential energy of these walls is infinitely high there can be no quantum tunnelling and the particle can only exist in space between these walls. Inside the walls the potential energy is equal to zero as this is an energetically favorable position, while the potential energy in the walls and outside them is infinite. [6]

With these conditions the Schrödinger's equation is significantly simplified to the following equation.

$$\frac{d^2\psi}{dx^2} + \left(\frac{2mE}{\hbar}\right)\psi = 0$$

This is then an ordered differential equation eigen value problem. By solving this problem, a final wavefunction used in the implementation can be reached. Where particle is bounded between the walls at positions 0 and L.

$$\psi_n(x) = \sqrt{\frac{2}{L} sin\frac{xn\pi}{L}}$$

By using the visualization tools in *Matplotlib*, the following graphs of the $\psi$ function and the probability density function $|\psi_n(x)|^2$ for the first three values of n are achieved.

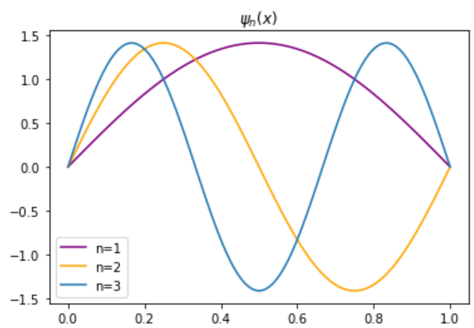


*Figure 3 Particle in a box psi function*

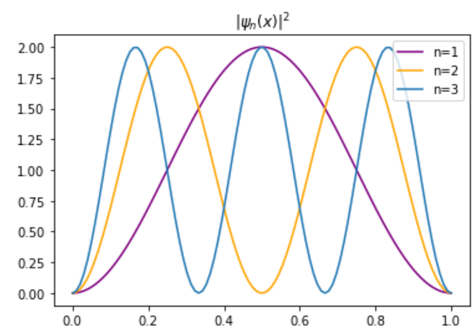


*Figure 2 Particle in a box probability density function*

## Particle in a two-dimensional box

In this model the same constraints are set as in the one-dimensional box model. The only difference is the dimensionality of the "box" due to which the $\psi$ function differs slightly. For this situation the value of the psi function can be seen from the following equation.

$$\psi(x,y) = \frac{2}{\sqrt{LxLy}} \sin\left(\frac{x\pi nx}{Lx}\right) \sin\left(\frac{y\pi ny}{Ly}\right)$$

Using the visualization tools in *Matplotlib*, the following graphs of the $\psi$ function and the probability density function $|\psi_n(x,y)|^2$ for the first three values of $ny$ and the first two values of $nx$.

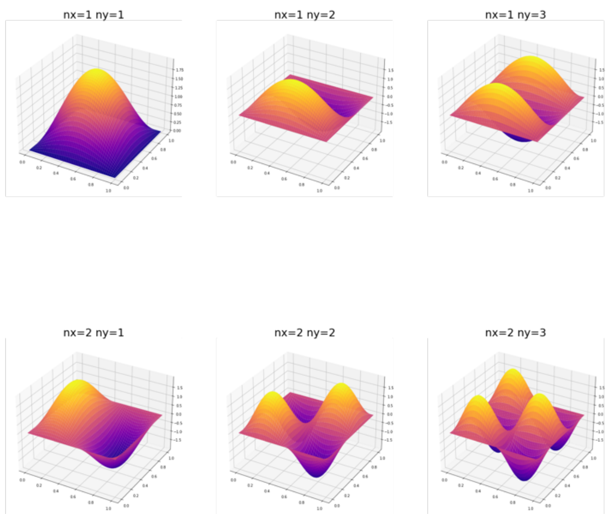


*Figure 4 Particle in a 2D box psi function*

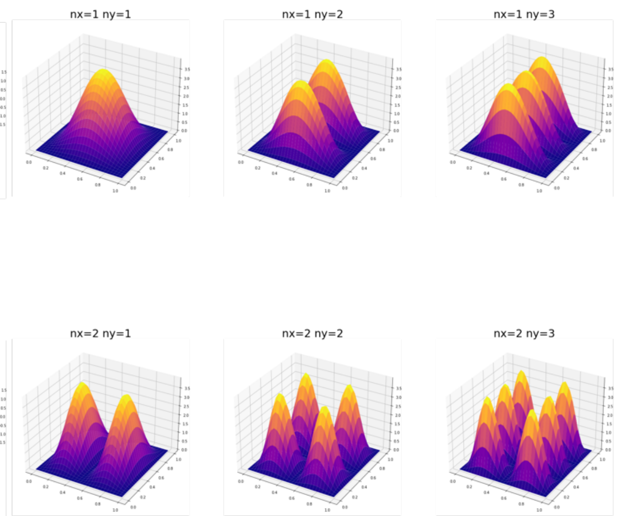


*Figure 5 Particle in a 2D box probability density*

## Hydrogen 1s density

Based on the following 1s wave equation the probability of finding an electron in a 1s orbital can be found.

$$\varphi 1,0,0 = \frac{e^{-r}}{\sqrt{\pi}}$$

From this equation it can be concluded that as r increases the probability would decrease, which means that the probability of finding an electron decreases when its distance from the nucleus increases. From the visualization, it can be seen that this is true as the distribution of the electrons is the densest in the center.

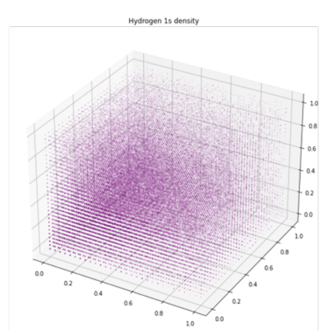


*Figure 6 Hydrogen 1s density*

## Conclusion

The main goal for this paper was to implement the solution of the Schrödinger equation in Python. This was achieved for a few simplest problems including the particle box and the 1s orbital of the Hydrogen atom. Some of the future additions to this project could include the animation of the time-dependent Schrödinger's equation and the visualization of other Hydrogen's orbitals. Besides the implementation of the Schrödinger equation, this paper also provides a brief introduction to quantum mechanics by summarizing its beginnings, as well as an explanation of the derivation of the Schrödinger's equation.

## Bibliography


[1] F. A. a. M. S. Berezin, The Schrödinger Equation, vol. 66, Springer Science & Business Media, 2012.

[2] "Blackbody Radiation," Openstax, [Online]. Available: https://openstax.org/books/university-physics-volume-3/pages/6-1-blackbody-radiation. [Accessed 11 December 2022].

[3] N. Connor, "What is Blackbody Radiation – Definition," 22 May 2019. [Online]. Available: https://www.thermal-engineering.org/what-is-blackbody-radiation-definition/. [Accessed 11 December 2022].

[4] "Quantum," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Quantum. [Accessed 11 December 2022].

[5] T. E. Oliphant, " Python for Scientific Computing," *Computing in Science & Engineering ,* vol. 9, no. 3, pp. 10-20, 18 June 2007.

[6] K. Haas, "Particle in a Box," 13 08 2022. [Online] [Accessed 06 12 2022].

[7] J. R. Chasnov, "The Schrödinger Equation," 18 Nov 2021. [Online] [Accessed 06 12 2022].

[8] M. T. a. M. Fowler, "Particle in a 2-Dimensional Box," 29 Nov 2021. [Online]. [Accessed 6 Dec 2022].

[9] "Quantum Physics Visualization with Python: Particle in a box, Visualising the Orbitals," 18 Apr 2018. [Online]. Available: https://towardsdatascience.com/quantum-physics-visualization-with-python-35df8b365ff . [Accessed 3 Dec 2022].


## Appendix

The Jupyter notebook can be accessed via this hyperlink:

https://github.com/sarakojic/Schrodinger_Equation/blob/main/Schrodinger's%20Equation.ipynb