Journal of Theoretical, Experimental, and Applied Physics

PRIME OPEN ACCESS

Volume 1, Issue 1

Research Article

Date of Submission: 28 Nov, 2025
Date of Acceptance: 19 Dec, 2025
Date of Publication: 30 Dec, 2025

Solving the 1D Schrödinger Equation Numerically with Irregular Grid

Eimund Smestad*

Independent Researcher, Norway

***Corresponding Author:**
Eimund Smestad, Independent Researcher, Norway.

Citation: Smestad, E. (2025). Solving the 1D Schrödinger Equation Numerically with Irregular Grid. *J Theor Exp Appl Phys*, 1(1), 01-10.

Abstract

There is a great interest for solving the Schrödinger equation, and here is presented an irregular finite difference scheme to do so. This enables to resolve small details in a larger space with logarithmic grids for instance. Both time-dependent and time-independent Schrödinger equation are formulated with this scheme. Non-zero boundary conditions are also studied for the eigenvalue problem, how changing one boundary affects the other boundary instantly due to normalization.

Introduction

The matrix formulation of quantum mechanics can used to solve the Schrödinger equation computationally, and with the θ-rule, Sec. 8, different schemes can be chosen like the Crank–Nicolson method [1,2]. Sec.4 present an alternative irregular finite-difference scheme, derived by averaging the two-sided limit forms of the derivative, which differs from the standard Taylor-expansion-based non-uniform schemes. New in this paper is matrix formulation, Sec. 9, with irregular grid in the finite difference representation Sec. 4, which enables the use of logarithmic grid to resolve small details Sec. 5. This matrix formulation requires the grid resolution to be included in the state vector, Sec. 6, and Sec. 9 shows that Hamiltonian is self-adjoint [3]. The paper further investigate eigenvalue problem in Sec. 7 with non-zero boundaries and see how changes in one end changes instantly the boundary at the other end, which is due to normalization property of the wave function. A Mathematica script is including in Sec. 11 to show how the irregular grid is implemented.

Related Work

Finite-difference methods on non-uniform meshes have a long history in numerical analysis. The most widely used three-point non-uniform stencil is obtained by Taylor expansion about the grid point, yielding coefficients that depend asymmetrically on the forward and backward spacings. General formulas for such coefficients on arbitrarily spaced grids were given by Fornberg in his seminal paper on generating finite-difference weights [4]. Earlier, Liszka and Orkisz developed what is now called the generalized finite difference method, where local derivative approximations are constructed from scattered node distributions [5]. These schemes, and their modern extensions, have been widely applied to partial differential equations on irregular domains. More recent work has focused on compact or self-adjoint difference operators on stretched meshes to preserve conservation or energy properties in time-dependent problems [6].

In contrast to these standard formulations, the irregular finite difference presented here is derived by averaging the two limit definitions of the derivative from the left and the right. This yields a symmetric three-point expression for the second derivative whose coefficients differ algebraically from the standard Taylor-based scheme, while still collapsing to the familiar central difference in the uniform-mesh limit. To our knowledge, this averaged stencil has not been documented in the numerical analysis literature, and its use in discretizing the Schrödinger operator allows for a naturally Hermitian matrix formulation on logarithmic grids.

The Schrödinger Equation

The time-independent Schrödinger equation is an eigenvalue problem

$$-\frac{\hbar^2}{2\,m}\,\frac{\mathrm{d}^2\Psi}{\mathrm{d}x^2} + V(x)\,\Psi = E\,\Psi \tag{1}$$

We also have the time-dependent Schrödinger equation

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

Irregular Finite Difference

The definition of the derivative is as follows, but to make the derivative unique it should be the same regardless how we approach the point of the derivative

$$\frac{\mathrm{d}\Psi_i}{\mathrm{d}x} = \lim_{x_{i+1}\to x_i} \frac{\Psi_{i+1}-\Psi_i}{x_{i+1}-x_i} = \lim_{x_i \leftarrow x_{i-1}} \frac{\Psi_i-\Psi_{i-1}}{x_i-x_{i-1}} \tag{3}$$

This should also apply to the second derivative

$$\frac{\mathrm{d}^2\Psi_i}{\mathrm{d}x^2} = \lim_{x_{i+1}\to x_i} \frac{\frac{\mathrm{d}\Psi_{i+1}}{\mathrm{d}x}-\frac{\mathrm{d}\Psi_i}{\mathrm{d}x}}{x_{i+1}-x_i} = \lim_{x_i \leftarrow x_{i-1}} \frac{\frac{\mathrm{d}\Psi_i}{\mathrm{d}x}-\frac{\mathrm{d}\Psi_{i-1}}{\mathrm{d}x}}{x_i-x_{i-1}} \tag{4}$$

While the first derivative requires only two points to be defined, the second derivative requires three points. To further discretizate the expression of the second derivative above and only use the same three points on each side of the equality, we find that

$$\begin{aligned}\frac{\mathrm{d}^2\Psi_i}{\mathrm{d}x^2} &= \lim_{x_{i+1}\to x_i \leftarrow x_{i-1}} \left(\frac{\Psi_{i+1}}{\left(x_{i+1}-x_i\right)^2} - \frac{\left(x_{i+1}-x_{i-1}\right)\Psi_i}{\left(x_{i+1}-x_i\right)^2\left(x_i-x_{i-1}\right)} + \frac{\Psi_{i-1}}{\left(x_{i+1}-x_i\right)\left(x_i-x_{i-1}\right)} \right) \\ &= \lim_{x_{i+1}\to x_i \leftarrow x_{i-1}} \left(\frac{\Psi_{i+1}}{\left(x_{i+1}-x_i\right)\left(x_i-x_{i-1}\right)} - \frac{\left(x_{i+1}-x_{i-1}\right)\Psi_i}{\left(x_{i+1}-x_i\right)\left(x_i-x_{i-1}\right)^2} + \frac{\Psi_{i-1}}{\left(x_i-x_{i-1}\right)^2} \right)\end{aligned} \tag{5}$$

When this is finite difference the two expressions of the second derivative is not necessarily the same, hence we take the average of the two expressions, which yields the following irregular difference of the second derivative

$$\frac{\mathrm{d}^2\Psi_i}{\mathrm{d}x^2} \approx \frac{x_{i+1}-x_{i-1}}{2\left(x_{i+1}-x_i\right)\left(x_i-x_{i-1}\right)} \left(\frac{\Psi_{i+1}}{x_{i+1}-x_i} - \frac{\left(x_{i+1}-x_{i-1}\right)\Psi_i}{\left(x_{i+1}-x_i\right)\left(x_i-x_{i-1}\right)} + \frac{\Psi_{i-1}}{x_i-x_{i-1}} \right) \tag{6}$$

For uniform grid this expression reduces down to finite difference expression of the second derivative

$$\frac{\mathrm{d}^2\Psi_i}{\mathrm{d}x^2} \approx \frac{\Psi_{i+1}-2\,\Psi_i+\Psi_{i-1}}{\Delta x_i{}^2} \tag{7}$$

Discretizing the second order derivative over n points can then be approximated into a matrix problem

$$\frac{\mathrm{d}\,\mathbf{\Psi}}{\mathrm{d}\,x^2} = \begin{bmatrix} \frac{\mathrm{d}^2\Psi_1}{\mathrm{d}\,x^2} \\ \frac{\mathrm{d}^2\Psi_2}{\mathrm{d}\,x^2} \\ \vdots \\ \frac{\mathrm{d}^2\Psi_n}{\mathrm{d}\,x^2} \end{bmatrix} \approx \begin{bmatrix} d_{1,0} & d_{1,1} & & & & & \\ d_{2,-1} & d_{2,0} & d_{2,1} & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & d_{i,-1} & d_{i,0} & d_{i,1} & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & d_{n-1,-1} & d_{n-1,0} & \end{bmatrix} \begin{bmatrix} \Psi_1 \\ \Psi_2 \\ \vdots \\ \Psi_i \\ \vdots \\ \Psi_n \end{bmatrix} = \mathbf{D_2}\,\mathbf{\Psi} \tag{8}$$

where

$$d_{i,0} = -\frac{b_{i,0}^2}{2}\,, \qquad d_{i,\pm 1} = \frac{b_{i,0}\,b_{i,\pm 1}}{2}\,, \qquad b_{i,\pm 1} = \frac{\pm 1}{x_{i\pm 1}-x_i}\,, \qquad b_{i,0} = b_{i,-1}\,b_{i,1}\left(x_{i+1}-x_{i-1}\right) \tag{9}$$

An alternative generalized finite difference of the second derivative is given in [7]

$$\frac{\mathrm{d}^2\Psi_i}{\mathrm{d}x^2} \approx \frac{2}{x_{i+1}-x_{i-1}}\left(\frac{\Psi_{i+1}-\Psi_i}{x_{i+1}-x_i}-\frac{\Psi_i-\Psi_{i-1}}{x_i-x_{i-1}}\right) \tag{10}$$

Logarithmic Grid

To make an logarithmic grid the step length need to increase exponentially, so a grid function $g(x)$ that starts at xmin would need to look like

$$g(x) = x_{\min}\, a^{x-x_{\min}} \tag{11}$$

where the base a is determined by the end point $x_{\max}$ of the grid

$$g(x_{\max}) = x_{\max} = x_{\min}\, a^{x_{\max}-x_{\min}} \tag{12}$$

which yields the grid function

$$g(x) = x_{\min}\left(\frac{x_{\max}}{x_{\min}}\right)^{\frac{x-x_{\min}}{x_{\max}-x_{\min}}} \tag{13}$$

The discrete grid point i of a total of n points is then calculated as follows

$$g_i = x_{\min}\left(\frac{x_{\max}}{x_{\min}}\right)^{\frac{i}{n}} = {x_{\min}}^{1-\frac{i}{n}}\, {x_{\max}}^{\frac{i}{n}} \tag{14}$$

Discretizing Wave Functions

To discretize an wave functions we must use the properties of linear transformation, which is additivity, homogeneity of degree 1 and commutation relation

$$H(\psi_1(x)+\psi_2(x)) = H(\psi_1(x)) + H(\psi_2(x)) \tag{15}$$
$$H(a\,\psi(x)) = a\,H(\psi(x)) \tag{16}$$
$$H(\psi(x)\ x') = H(\psi(x))\ x' \tag{17}$$

where H is a linear transformation, ψ a function and a a constant. Lets define

$$\vec{\psi} = \sum_i \psi_i\, \Delta\chi_i \tag{18}$$

which approaches an integral when $\Delta\chi_i \to 0$. $\vec{\psi}$ can be made to have the same eigenvalue as $\psi(x)$ with the operator H, such that

$$H(\psi_i)\ \Delta\chi_i = H(\psi_i\,\Delta\chi_i) \tag{19}$$

where $H(\psi_i) = E\,\psi_i$ implies $H(\psi_i\,\Delta\chi_i) = E\,\psi_i\,\Delta\chi_i$. Using additivity in (15)

$$\sum_i H(\psi_i)\ \Delta\chi_i = \sum_i H(\psi_i\,\Delta\chi_i) = H\left(\sum_i \psi_i\,\Delta\chi_i\right) \tag{20}$$

This results in the following equality for a linear combination

$$H\left(\sum_{k=1}^{n} a_k\,\psi_k(x)\right) = \sum_{k=1}^{n} a_k\, H(\psi_k(x)) \tag{21}$$

Using the fact that a vector can be written as a linear combination, means that we can use an integral to represent an infinite dimensional vector

$$\vec{\psi}(x) = \int^x \psi(x)\ \mathrm{d}x \tag{22}$$

Now let the dx be a non-constant step length which we want to change to a constant step length dy, then the linear transformation of the above vector can be rewritten to

$$H\left(\vec{\psi}\left(x\left(y\right)\right)\right) = H\left(\int^{x(y)} \psi\left(x\right)\,\mathrm{d}x\right) = H\left(\int^{y} \psi\left(x\left(y\right)\right)\, x'\left(y\right)\,\mathrm{d}y\right) = \int^{y} H\left(\psi\left(x\left(y\right)\right)\, x'\left(y\right)\right)\mathrm{d}y = \int^{y} H\left(\psi\left(x\left(y\right)\right)\right)\, x'\left(y\right)\,\mathrm{d}y \tag{23}$$

The inner product of the linear transformation is then given by

$$\begin{aligned}
\langle H\rangle &= \int^{x(y)} \psi\left(x\left(y\right)\right)^{*}\cdot H\left(\psi\left(x\left(y\right)\right)\right)\mathrm{d}x = \int^{y} \psi\left(x\left(y\right)\right)^{*}\, H\left(\psi\left(x\left(y\right)\right)\right)\, x'\left(y\right)\,\mathrm{d}y \\
&= \psi\left(x\left(y\right)\right)^{*} \int^{y} H\left(\psi\left(x\left(y\right)\right)\right)\, x'\left(y\right)\,\mathrm{d}y - \int^{y} \psi\left(x\left(y\right)\right)^{*\prime} \int^{y} H\left(\psi\left(x\left(y\right)\right)\right)\, x'\left(y\right)\,\mathrm{d}y\,\mathrm{d}y \\
&= \psi\left(x\left(y\right)\right)^{*} \int^{y} H\left(\psi\left(x\left(y\right)\right)\, x'\left(y\right)\right)\mathrm{d}y - \int^{y} \psi\left(x\left(y\right)\right)^{*\prime} \int^{y} H\left(\psi\left(x\left(y\right)\right)\, x'\left(y\right)\right)\mathrm{d}y\,\mathrm{d}y \\
&= \int^{y} \psi\left(x\left(y\right)\right)^{*}\, H\left(\psi\left(x\left(y\right)\right)\, x'\left(y\right)\right)\mathrm{d}y
\end{aligned} \tag{24}$$

This means that the discretization of Hamiltonian in Schrödinger equation is done like this on matrix form

$$\mathbf{H}\,\mathbf{\Psi} = \frac{\mathbf{H}}{\Delta y}\begin{bmatrix}\psi_0\,\Delta x_0\\ \psi_1\,\Delta x_1\\ \vdots\\ \psi_n\,\Delta x_n\end{bmatrix} \tag{25}$$

where Δx_i forms an irregular grid.

Numerically Find Eigenfunctions with Non-Zero Boundaries

The Schrödinger equation in (1) can be discretized with (6) to form a tridiagonal matrix

$$\begin{bmatrix}
E & & & & & & & \\
a_{10} & a_{11} & a_{12} & & & & & \\
 & a_{21} & a_{22} & a_{23} & & & & \\
 & & \ddots & \ddots & \ddots & & & \\
 & & & a_{i(i-1)} & a_{ii} & a_{i(i+1)} & & \\
 & & & & \ddots & \ddots & \ddots & \\
 & & & & & a_{n(n-1)} & a_{nn} & a_{n(n+1)} \\
 & & & & & & & E
\end{bmatrix}
\begin{bmatrix}\psi_0\,\Delta x_0\\ \psi_1\,\Delta x_1\\ \psi_2\,\Delta x_2\\ \vdots\\ \psi_i\,\Delta x_i\\ \vdots\\ \psi_n\,\Delta x_n\\ \psi_{n+1}\,\Delta x_{n+1}\end{bmatrix}
= E\begin{bmatrix}\psi_0\,\Delta x_0\\ \psi_1\,\Delta x_1\\ \psi_2\,\Delta x_2\\ \vdots\\ \psi_i\,\Delta x_i\\ \vdots\\ \psi_n\,\Delta x_n\\ \psi_{n+1}\,\Delta x_{n+1}\end{bmatrix} \tag{26}$$

where ψ_0 and ψ_{n+1} are the boundary conditions, and we have

$$a_{i(i-1)} = -\frac{\hbar^2}{2\,m}\,\frac{x_{i+1}-x_{i-1}}{\left(x_{i+1}-x_i\right)\left(x_i-x_{i-1}\right)^2} \tag{27}$$

$$a_{ii} = \frac{\hbar^2}{2\,m}\,\frac{\left(x_{i+1}-x_{i-1}\right)^2}{\left(x_{i+1}-x_i\right)^2\left(x_i-x_{i-1}\right)^2} + V_i \tag{28}$$

$$a_{i(i+1)} = -\frac{\hbar^2}{2\,m}\,\frac{x_{i+1}-x_{i-1}}{\left(x_{i+1}-x_i\right)^2\left(x_i-x_{i-1}\right)} \tag{29}$$

The eigenvalue problem in (26) can be written as $\mathbf{A}\,\vec{\psi} = E\,\vec{\psi}$, and we find the eigenvalues by diagonalizing the matrix $\mathbf{A} - E\,\mathbf{I}$ ($\mathbf{I}$ is the identity matrix).

$$\begin{bmatrix}
0 & & & & & & & \\
a_{10} & a_{11}-E & a_{12} & & & & & \\
 & a_{21} & a_{22}-E & a_{23} & & & & \\
 & & \ddots & \ddots & \ddots & & & \\
 & & & a_{i(i-1)} & a_{ii}-E & a_{i(i+1)} & & \\
 & & & & \ddots & \ddots & \ddots & \\
 & & & & & a_{n(n-1)} & a_{nn}-E & a_{n(n+1)} \\
 & & & & & & & 0
\end{bmatrix} \tag{30}$$

The inverse iteration method can be used to find an approximate eigenvector $\vec{\psi}$ to a given eigenvalue E. We introduce an approximate eigenvalue $\alpha\ E$ where we choose α sufficiently close to 1 such the that the inverse iteration method converge to the approximate eigenvector to E, but α should also be chosen sufficiently far away from 1 such that the magnitude of $E - \alpha\ E$ is sufficiently bigger than 0 to avoid numerical issues in the inverse iteration method. The foundation of the inverse iteration method is given by

$$\mathbf{A}\,\vec{\psi} = E\,\vec{\psi} \qquad \Leftrightarrow \qquad (\mathbf{A} - \alpha\,E\,\mathbf{I})\,\vec{\psi} = (1-\alpha)\;E\,\vec{\psi} \tag{31}$$

Using the power iteration method we arrive at the inverse iterative method

$$\vec{\psi}_{k+1} = (1-\alpha)\;E\;(\mathbf{A} - \alpha\,E\,\mathbf{I})^{-1}\;\vec{\psi}_k = (1-\alpha)^k\;E^k\;(\mathbf{A} - \alpha\,E\,\mathbf{I})^{-k}\;\vec{\psi}_1 \tag{32}$$

Let $\vec{\psi}_1 = c\,\vec{\psi}_0$, then the norm of $\vec{\psi}_{k+1}$ are given by

$$||\vec{\psi}_{k+1}|| = c\;(1-\alpha)^k\;E^k\;\left\|(\mathbf{A} - \alpha\,E\,\mathbf{I})^{-k}\;\vec{\psi}_0\right\| \tag{33}$$

which implies that $\vec{\psi}_{k+1}$ is normalizable when

$$c = \frac{1}{(1-\alpha)^k\;E^k\;\left\|(\mathbf{A} - \alpha\,E\,\mathbf{I})^{-k}\;\vec{\psi}_0\right\|} \tag{34}$$

and we get

$$\vec{\psi}_{k+1} = \frac{(\mathbf{A} - \alpha\,E\,\mathbf{I})^{-k}\;\vec{\psi}_0}{\left\|(\mathbf{A} - \alpha\,E\,\mathbf{I})^{-k}\;\vec{\psi}_0\right\|} = \frac{\mathbf{B}\,\vec{\psi}_0}{\left\|\mathbf{B}\,\vec{\psi}_0\right\|} \tag{35}$$

where $\mathbf{B} = (1-\alpha)^k\;E^k\;(\mathbf{A} - \alpha\,E\,\mathbf{I})^{-k}$. Following the form of **A** in (26) we have the following form of **B**

$$\mathbf{B} = \begin{bmatrix} b_{00} & b_{01} & \cdots & b_{0n} & b_{0(n+1)} \\ b_{10} & b_{11} & \cdots & b_{1n} & b_{1(n+1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{n0} & b_{n1} & \cdots & b_{nn} & b_{n(n+1)} \\ b_{(n+1)0} & b_{(n+1)1} & \cdots & b_{(n+1)n} & b_{(n+1)(n+1)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ b_{10} & b_{11} & \cdots & b_{1n} & b_{1(n+1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{n0} & b_{n1} & \cdots & b_{nn} & b_{n(n+1)} \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \tag{36}$$

Note the simplification of row 1 and $n+1$, which is due to the fact that $\mathbf{A} - \alpha\ E\ \mathbf{I}$ has only one none-zero element "$(1-\alpha)\ E$" in each of the rows 1 and $n+1$. Let $\vec{\psi}_{k+1} = \left[\psi_0^{k+1}, \psi_1^{k+1}, \ldots, \psi_{n+1}^{k+1}\right]$ and $\vec{\psi}_0 = [a, 0, \ldots, 0, b]$ where a and b is none-zero, then we can use (35) to setup

$$\left\|\mathbf{B}\,\vec{\psi}_0\right\| = \frac{a}{\psi_0^{k+1}} = \frac{b}{\psi_{n+1}^{k+1}} \tag{37}$$

Expanding the expression for the norm $\left\|\mathbf{B}\,\vec{\psi}_0\right\|$, we can then setup

$$\sqrt{\sum_{j=0}^{n+1}\left(b_{j0}\,a + b_{j(n+1)}\,b\right)^2} = \frac{a}{\psi_0^{k+1}} \tag{38}$$

and using the relation between a and b in (37) we get that

$$\sum_{j=0}^{n+1}\left(b_{j0}\,\psi_0^{k+1} + b_{j(n+1)}\,\psi_{n+1}^{k+1}\right)^2 = 1 \tag{39}$$

and solving this equation with regard to ψ_{n+1}^{k+1} yields

$$\psi_{n+1}^{k+1} = \frac{-\psi_0^{k+1}\sum_{j=1}^{n} b_{j0}\, b_{j(n+1)} \pm \sqrt{\left(\psi_0^{k+1}\sum_{j=1}^{n} b_{j0}\, b_{j(n+1)}\right)^2 - \sum_{j=1}^{n+1} b_{j(n+1)}{}^2 \left(\left(\psi_0^{k+1}\right)^2 \sum_{j=0}^{n} b_{j0}{}^2 - 1\right)}}{\sum_{j=1}^{n+1} b_{j(n+1)}{}^2} \tag{40}$$

or with regard to

$$\psi_0^{k+1} = \frac{-\psi_{n+1}^{k+1}\sum_{j=1}^{n} b_{j0}\, b_{j(n+1)} \pm \sqrt{\left(\psi_{n+1}^{k+1}\sum_{j=1}^{n} b_{j0}\, b_{j(n+1)}\right)^2 - \sum_{j=0}^{n} b_{j0}{}^2 \left(\left(\psi_{n+1}^{k+1}\right)^2 \sum_{j=1}^{n+1} b_{j(n+1)}{}^2 - 1\right)}}{\sum_{j=0}^{n} b_{j0}{}^2} \tag{41}$$

When $k \to \infty$ these converge to the boundary conditions $\psi_0\, \Delta x_0 = \lim_{k\to\infty} \psi_0^k$ and $\psi_{n+1}\, \Delta x_{n+1} = \lim_{k\to\infty} \psi_{n+1}^k$. As we see in (40) and (41) the two boundary conditions are dependent on each other. We also see that given one of the boundary conditions then there are two solutions that is possible for the other boundary condition, making two path ways possible for transition between boundary conditions of the wave function.

The θ-Rule and the Time-Dependent Schrödinger Equation

Assuming

$$t_\theta \approx \theta\, t + \left(1-\theta\right) t_0 \tag{42}$$

then the first order approximation can be written

$$\begin{aligned} \Psi(x,t_\theta) &\approx \Psi(x,t_0) + \frac{\partial \Psi(x,t_0)}{\partial t}\left(t_\theta - t_0\right) = \Psi(x,t_0) + \theta\,\frac{\partial \Psi(x,t_0)}{\partial t}\left(t - t_0\right) \\ &\approx \Psi(x,t_0) + \theta\left(t-t_0\right)\frac{\Psi(x,t) - \Psi(x,t_0)}{t-t_0} = \theta\,\Psi(x,t) + \left(1-\theta\right)\Psi(x,t_0) \end{aligned} \tag{43}$$

which is known as the θ-rule. The θ-rule can be used to approximate the time-dependent Schrödinger equation in (2)

$$\begin{aligned} \mathrm{i}\,\hbar\,\frac{\Psi(x,t) - \Psi(x,t_0)}{t-t_0} &= \mathrm{i}\,\hbar\,\frac{\Psi(x,t_\theta) - \Psi(x,t_0)}{t_\theta - t_0} \approx H\left(\Psi(x,t_\theta)\right) \\ &= H\left(\theta\,\Psi(x,t) + \left(1-\theta\right)\Psi(x,t_0)\right) + F\left(\Psi(x,t_\theta)\right) = \theta\, H\left(\Psi(x,t)\right) + \left(1-\theta\right) H\left(\Psi(x,t_0)\right) \end{aligned} \tag{44}$$

where $\theta = 0$ is Forward Euler scheme (explicit), $\theta = 1$ is Backward Euler scheme (implicit), and $\theta = 1/2$ is Crank-Nicolson scheme. Crank-Nicolson have the benefit of second order local truncation error. Therefore the numerical scheme for the θ-rule can be approximated

$$\mathrm{i}\,\hbar\,\frac{\Psi(x,t) - \Psi(x,t_0)}{t-t_0} \approx \theta\, H\left(\Psi(x,t)\right) + \left(1-\theta\right) H\left(\Psi(x,t_0)\right) \tag{45}$$

Time-Dependent Schrödinger Equation and Irregular Finite Difference

Combining the θ-rule of the time-dependent Schrödinger equation in (45) and the irregular finite difference of the second derivative in (6), yields the following difference equation with i-indices are spacial steps and j-indices are time steps

$$\alpha_j\left(\Psi_{i,j+1} - \Psi_{i,j}\right) = \theta \sum_{k=-1}^{1} h_{i,k}\,\Psi_{i+k,j+1} + \left(1-\theta\right)\sum_{k=-1}^{1} h_{i,k}\,\Psi_{i+k,j} \tag{46}$$

where

$$\alpha_j = \frac{\mathrm{i}\,\hbar}{t_{j+1} - t_j}, \qquad h_{i,\pm1} = -\frac{\hbar^2\, d_{i,0}\, \mathrm{d}_{i,\pm1}}{2\,m}, \qquad h_{i,0} = -\frac{\hbar^2\, d_{i,0}{}^2}{2\,m} + V_i \tag{47}$$

and the $h_{i,k}$ are elements in the Hamilton operator

$$\mathbf{H} = -\frac{\hbar^2}{2\,m}\,\mathbf{D_2} + \mathbf{V} \tag{48}$$

The equation in (46) can be written as a matrix equation with all the spatial step, where $\mathbf{\Psi}_{j+1}$ is the only unknown giving the wave function in the next time step

$$\left(\theta\,\mathbf{H} - \alpha_j\,\mathbf{I}\right)\mathbf{\Psi_{j+1}} = \left(\left(\theta - 1\right)\mathbf{H} - \alpha_j\,\mathbf{I}\right)\mathbf{\Psi_j} \tag{49}$$

Hermitian/Self-Adjont

Looking at the inner product

$$\langle \mathbf{u}, \mathbf{v}\rangle = \sum_i u_i^*\, v_i \tag{50}$$

we can show that the discretization used in this paper is self-adjont

$$\begin{aligned}\langle \mathbf{u}, \mathbf{H}\,\mathbf{v}\rangle &= \sum_{i,j} u_j^*\, H_{j,i}\, v_i = \sum_i \left(h_{i,0}\, u_i^*\, v_i + h_{i,-1}\, u_{i-1}^*\, v_i + h_{i,1}\, u_{i+1}^*\, v_i\right) = \sum_i \left(h_{i,0}\, u_i^* + h_{i,1}\left(u_{i-1}^* + u_{i+1}^*\right)\right) v_i \\ &= \sum_i \left(h_{i,0}^*\, u_i^* + h_{i,1}^*\left(u_{i-1}^* + u_{i+1}^*\right)\right) v_i = \sum_i \left(h_{i,0}^*\, u_i^*\, v_i + h_{i,1}^*\, u_{i-1}^*\, v_i + h_{i,-1}^*\, u_{i+1}^*\, v_i\right) = \sum_{i,j} H_{i,j}^*\, u_j^*\, v_i = \langle \mathbf{H}\,\mathbf{u}, \mathbf{v}\rangle\end{aligned} \tag{51}$$

which shows that the eigenvalues are real and the corresponding eigenvectors are orthogonal.

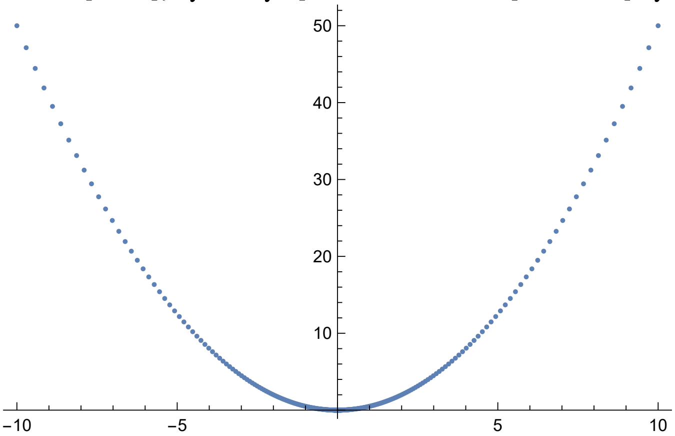
Mathematica Code

Algorithm 1 Mathematica code to solve Schrödinger equation for harmonic oscillator with irregular grid where $\theta = 1$

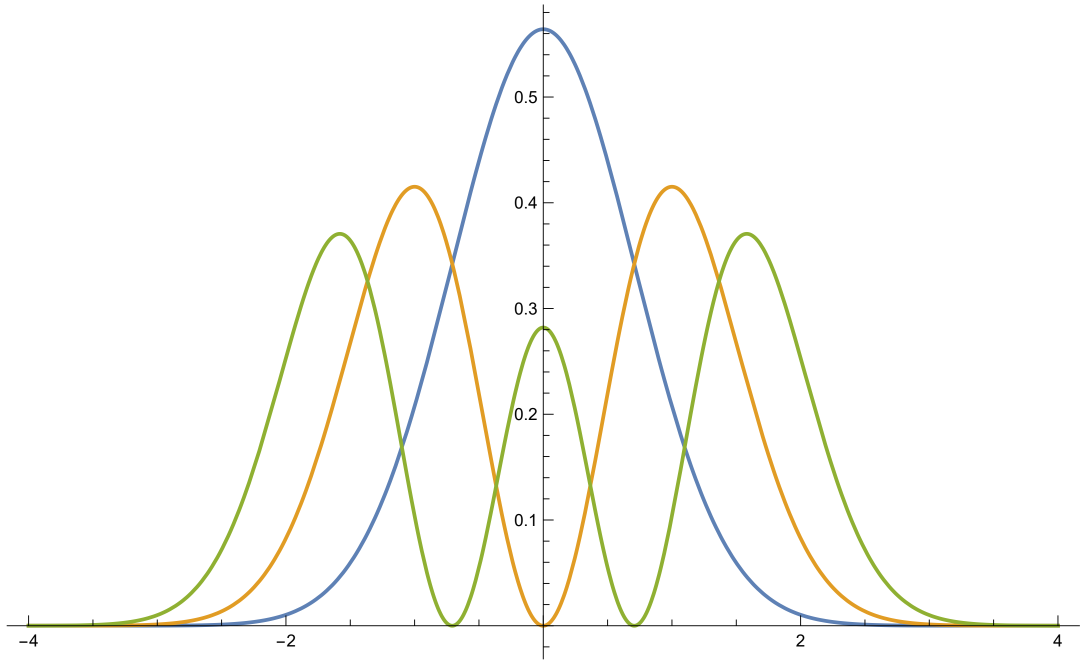
```
Potential[x__]:=m ∗ x ∗ x/2
UniformGrid[min__, max__, i__, n__]:= min +(max − min) ∗ i/n
LogGrid[min__, max__, i__, n__]:=If[min ∗ max ≥ 0, min ∗(max / min)^(i/n), 2∗min − min ∗((min − max)/ min)^(i/n)]
CoreGrid[min__, mid__, max__, i__, m__, n__]:=
  If[Abs[i]<=m,
    UniformGrid[min, mid, i, m],
    If[i > 0,
      LogGrid[mid, max, i − m, n − m],
      2 ∗ min −LogGrid[mid, max, −i − m, n − m]
    ]
  ]
Tri[n__, min__, max__, a__, V__, G__]:=Block[
  {i, A = SparseArray[{Band[{1, 2}] → 0, Band[{1, 1}] → 0, Band[{2, 1}] → 0}, n],
  x0 = Apply[G, {min, max, 0, n + 1}], x1 = Apply[G, {min, max, 1, n + 1}], x2, γm1, γ0, γ1, b},
  For[i = 1, i ≤ n, i++,
    x2 = Apply[G, {min, max, i + 1, n + 1}];
    γm1 = 1/(x1 − x0);
    γ1 = 1/(x2 − x1);
    γ0 = γm1 ∗ γ1(x2 − x0);
    b = a ∗ γ0/2;
    If[i > 1, A[[i − 1, i]] = γm1 ∗ b];
    A[[i, i]] = −γ0 ∗ b + Apply[V, {x1}];
    If[i < n, A[[i + 1, i]] = γ1 ∗ b];
    x0 = x1;
    x1 = x2;


  ];
  A
]
EigenFunction[vec__, min__, max__, G__, order__]:=Block[
  {fsum, sum, v, vv = Block[{nmax = Length[vec] + 1},
    Block[{x0, x1 = Apply[G, {min, max, 0, nmax}], x2 = Apply[G, {min, max, 1, nmax}]},
      Join[
        {{min, 0}},
        Table[x0 = x1; x1 = x2; x2 = Apply[G, {min, max, i + 1, nmax}]; {x1, vec[[i]] ∗ (x2 − x0)/((x2 − x1) ∗ (x1 − x0))}, {i, 1, nmax − 1}],
        {{max, 0}}
      ]
    ]]
  }
  fsum = Interpolation[vv, InterpolationOrder → order];
  sum = Sqrt[NIntegrate[Conjugate[fsum[x]] ∗ fsum[x], {x, min, max}]];
  v = Table[{vv[[i]][[1]], vv[[i]][[2]]/sum}, {i, 1, Length[vv]}];
  Interpolation[v, InterpolationOrder → order]
]
TimeSolve[c__, V__, ψ__, xmin__, xmax__, X__, Xsave__, tmin__, tmax__, nt__, T__, Tsave__, order__]:=
  Block[{nx = Length[ψ] − 2}, Block[
    {B = Tri[nx, xmin, xmax, c, V, X], x = Table[Apply[X, {xmin, xmax, i, nx + 1}], {i, 0, nx + 1}],
```

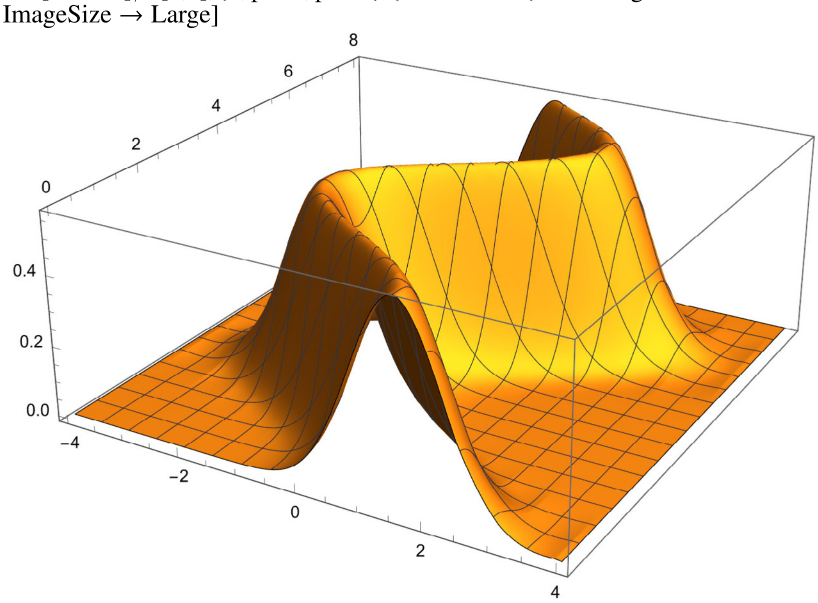
```
        t0 = Apply[T, {tmin, tmax, 0, nt}], t1, ψ0 = ψ, fsum},
      Block[{j, k = 2, b, ψ1, w = Join[{1}, Table[(x[[i + 1]] − x[[i − 1]])/(2 ∗ (x[[i + 1]] − x[[i]]) ∗ (x[[i]] − x[[i −
1]])), {i, 2, nx + 1}], {1}]},
      If[Tsave[[1]] == 0, ψ1 = Table[{{x[[Xsave[[i]]]], t0}, Conjugate[ψ0[[Xsave[[i]]]]] ∗ ψ0[[Xsave[[i]]]]},
        {i, 1, Length[Xsave]}], ψ1 = {}];
      ψ0 = Normalize[ψ0/w];
      For[j = 1, j ≤ nt, j++,
        t1 = Apply[T, {tmin, tmax, j, nt}];
        b = ConstantArray[i ∗ ħ/(t0 − t1), nx];
        t0 = t1;
        ψ0 = Join[{0}, LinearSolve[B + DiagonalMatrix[b], b ∗ ψ0[[2;;nx + 1]]], {0}];
        If[Tsave[[k]] == j,
          b = w ∗ ψ0;
          fsum = Interpolation[Table[{x[[Xsave[[i]]]], b[[Xsave[[i]]]]}, {i, 1, Length[Xsave]}], InterpolationOrder →
order];
          b/=Sqrt[NIntegrate[Conjugate[fsum[y]] ∗ fsum[y], {y, xmin, xmax}]];
          ψ1 = Join[ψ1, Table[{{x[[Xsave[[i]]]], t0}, Re[Conjugate[b[[Xsave[[i]]]]]∗b[[Xsave[[i]]]]]}, {i, 1, Length[Xsave]}]];
          k++;
        ];
      ];
      Interpolation[ψ1, InterpolationOrder → order]
    ]]]
GaussianWave[pos__, λ__, __, min__, max__, n__, G__, order__]:=Block[
    {i = 1, j = 1, x = Table[{Apply[G, {min, max, k, n+1}], 0}, {k, 0, n+1}], ψ = ConstantArray[0, n+2], fsum, sum},
    For[j = 1, j ≤ Length[pos], j++,
      For[i = 1, i ≤ n + 2, i++,
        x[[i, 2]]+=c[[j]] ∗ Exp[−λ[[j]] ∗ (x[[i, 1]] − pos[[j]])^2];
        ψ[[i]] = x[[i, 2]];
      ];
    ];
    fsum = Interpolation[x, InterpolationOrder → order];
    ψ/=Sqrt[NIntegrate[Conjugate[fsum[y]] ∗ fsum[y], {y, xmin, xmax}]]
]
intorder = 3;
m = 1;
ħ = 1;
pos = {1.5};
λ = {0.5};
a = {1};
ncore = 0.1;
xcore = 0.005;
xmin = −10.0;
xmax = 10.0;
nx = 399;
Xsave = Range[1, nx + 2, Floor[(nx + 2)/50]];
tmin = 0;
tmax = 8;
nt = 10000;
Tsave = Range[0, nt, Floor[nt/50]];
pmin = −4.0;
pmax = 4.0;
pp = 2;
MyGridSym[min__, max__, i__, n_]:=
    Block[{xc = (max + min)/2}, CoreGrid[xc, xc + xcore ∗ (max −xc), max, i − n/2, ncore ∗ n/2, n/2]];
ListPlot[Table[{MyGridSym[xmin, xmax, i, nx+1], Potential[MyGridSym[xmin, xmax, i, nx+1]]}, {i, 0, nx+1}], PlotRange->All]
```




```
sys1 = Eigensystem[Tri[nx, xmin, xmax, −ħ^2/(2 ∗ m), Potential, MyGridSym], −3];
Show[Plot[Evaluate[Table[EigenFunction[sys1[[2]][[Length[sys1[[2]]]−i]], xmin, xmax, MyGridSym, intorder][x]^2,
    {i, 0, pp}]], {x, pmin, pmax}, PlotRange → All], ImageSize → Large]
```




```
102: ψ1 = TimeSolve[−ℏ^2/(2 ∗ m), Potential, GaussianWave[pos, λ, a, xmin, xmax, nx, MyGridSym, intorder],
       xmin, xmax, MyGridSym, Xsave, tmin, tmax, nt, UniformGrid, Tsave, intorder];
103: Show[Plot3D[ψ1[x, t], {x, pmin, pmax}, {t, tmin, tmax}, PlotRange → All, PlotPoints → {200, 200}],
       ImageSize → Large]
```



Conclusion

This paper shows a successful implementation of irregular grid in the Schrödinger equation, which can be used to resolve smaller detail like the core-potential of atoms. This approach with the irregular grid can be used for other differential equations.

Declaration of Generative AI and AI-Assisted Technologies in the Writing Process

During the preparation of this work the author used Chat-gpt for literature study. After using this tool/service, the author reviewed and edited the content as needed and takes full responsibility for the content of the publication.

References

1. D. Griffiths, Introduction to Quantum Mechanics, 2nd Edition, Pearson, 2004.
2. H. Wong, J. Zhao, An artificial boundary method for the Hull–White model of American interest rate derivatives, Applied Mathematics and Computation 217 (9) (2011) 4627–4643.
3. A. Kitson, R. McLachlan, N. Robidoux, Skew-adjoint finite difference methods on non-uniform grids.
4. Fornberg, B. (1988). Generation of finite difference formulas on arbitrarily spaced grids. Mathematics of computation, 51(184), 699-706.
5. Liszka, T., & Orkisz, J. (1980). The finite difference method at arbitrary irregular grids and its application in applied mechanics. Computers & Structures, 11(1-2), 83-95.
6. T. Sengupta, S. Bhaumik, U. Shameem, A new compact difference scheme for second derivative in nonuniform grid expressed in self-adjoint form, Journal of Computational Physics 230 (5) (2011) 1822–1848.

7. Veldman, A. E. P., & Rinzema, K. (1992). Playing with nonuniform grids. Journal of engineering mathematics, 26(1), 119-130.