Time Evolution Simulation of the Quantum Mechanical Wave Function in 3D Space

Zoltán Simon*
Supervised by: Dr. Balázs Csébfalvi, Dr. Péter Vancsó, Dr. Géza István Márk

Department of Control Engineering and Information Technology
Faculty of Electrical Engineering and Informatics
Budapest University of Technology and Economics

Abstract

In quantum mechanics, the wave function describes the state of a physical system. In the non-relativistic case, the time evolution of the wave function is described by the time-dependent Schrödinger equation. In 1982, D Kosloff and R Kosloff proposed a method to solve the time-dependent Schrödinger equation efficiently using Fourier transformation. The computational physics research group, led by Géza I. Márk in the Nanotechnology Department, Institute for Technical Physics and Materials Science, Centre for Energy Research, located in Budapest, in collaboration with Belgian researchers, developed a simulation method based on three-dimensional wave packet dynamics for the study of electron dynamics in nanosystems. A simplified, interactive, two-dimensional version for educational purposes was published in 2020. In this work, we demonstrate two improvements of the wave packet dynamical simulation software: (i) the use of the Graphical Processing Unit (GPU), which results in a vast (up to 50x) increase in simulation speed, and (ii) the introduction of advanced visualization techniques which are helpful to correctly interpret massive 4D space-time wave function data sets obtained from the simulation.

Keywords: Quantum Mechanics, Wave Packet Dynamics, Ray Tracing, Simulation

1 Introduction

In the first quarter of the 20th century Quantum Mechanics (QM) opened a whole new window to understand our universe. Tamás Geszti, in his book [18] writes: *"learning QM is part of the process of understanding the world, and the person who masters it, understands the world better"*. QM can be used efficiently to model the behavior of atomic particles. It describes how electrons behave in the orbitals around the atomic core and explains chemical reactions. It can be used to model the structure of molecules. In nanotechnology, it is crucial to make quantum mechanical calculations to predict –and, in many cases, explain– the behavior of different nanostructures. One exciting field of study is the science of single-layer materials [23]. These are also known as 2D materials. One such carbon structure is called graphene [5, 16, 14]. This single-layered structure conducts heat and electricity very efficiently, thus raising high hopes in many when it comes to possible use-cases. Inspired by the previously enumerated fields of application, we set the goal to study the behavior of quantum systems by computer simulation. Such simulations are beneficial for scientists. They use such methods to accurately model the interaction between particles and various potential fields. In order to accomplish this goal, we choose a method that uses the Fast Fourier Transform (FFT) to efficiently calculate the time development of the quantum mechanical wave function. In QM, the wave function describes the state of a physical system. In the non-relativistic case, the time evolution of the wave function is described by the time-dependent Schrödinger equation [15]. In 1982, D Kosloff and R Kosloff proposed a method [8] to solve the time-dependent Schrödinger equation efficiently using Fourier transformation. In 2020, Géza István Márk published a paper [11] describing a computer program for the interactive solution of the time-dependent and stationary two-dimensional (2D) Schrödinger equation. Some details of quantum phenomena are only observable by calculating with all three spatial dimensions. Géza István Márk and his colleagues have already used 3D calculations in their research work [19, 9]. The difference is that their implementation uses solely the Central Processing Unit (CPU) of a computer. For visualization of the resulting probability density so far, they used the isosurface method. Our contribution mainly lies in leveraging the parallelization potential of the modern Graphical Processing Unit (GPU), thus significantly boosting the calculation speed by approximately a factor of 50 on our test hardware. We also apply state-of-the-art volumetric visualization techniques to create pleasing and comprehensible visuals to analyze the probability density evolution in 3D space.

*zoltan.simon@edu.bme.hu

2 Theoretical background

By examining atomic particles, scientists have observed that such particles exhibit wave-like behavior, and in bounded systems, they can absorb or release energy only in discrete quanta. These matter waves have complex amplitudes, and can interfere with themselves.

Equation 1 is called the Schrödinger equation. It is a linear partial differential equation, and it is the governing equation of QM published by Erwin Schrödinger [15] in 1926. Linearity is a requirement for matter waves since, by the definition of superposition, a general equation that aims to describe the behavior of matter waves must be satisfied not only by simple waves but also by the linear combination of these waves.

$$\frac{d}{dt}\Psi(\vec{r},t) = -\frac{i}{\hbar}\hat{H}\Psi(\vec{r},t) \tag{1}$$

In equation 1, we can see that on the left side, we basically take the first derivative of the wave function with respect to the time and on the right side we let the $\hat{H} = -\frac{\hbar^2}{2m}\Delta + V(\vec{r})$ Hamiltonian operator [6] affect the wave function. $\hbar$ is the reduced Planck constant. By specifying an initial state and solving this differential equation, we can predict the time development of a quantum mechanical wave function.

The wave function is a complex valued function. Experiments show that the square of the absolute value of the complex amplitude is the probability density associated with the particle being found in a given infinitesimally small portion of space at a given time. For convenience and to be sound with probability theory, we normalize the amplitude of the wave function so that the probability of the particle being found "somewhere" in space equals $\mathbb{P} = 1$.

$$\int_{\mathscr{V}} |\Psi(\vec{r},t)|^2 \, d^3r = 1 \tag{2}$$

3 Calculation method

Back in 1982, D Kosloff and R Kosloff proposed a method [8] to solve the time-dependent Schrödinger equation efficiently using Fourier transformation. The advantage of this algorithm compared to the Finite Difference in Time Domain (FDTD) methods[22, 10] is the high numerical stability of the time evolution step. In the adopted FFT method, no signs of divergence are present even after a large number of simulation steps. The time development step of the algorithm has a time complexity of $\mathscr{O}(N \log N)$ since it only uses six FFT runs ($\mathscr{O}(N \log N)$ each) and three element-wise multiplication between tensors ($\mathscr{O}(N)$ each). The amount of FFT runs and multiplications can be reduced further if we do not want to read the results of the time development in each step. A significant speed-up can be reached by using a parallelized implementation of the FFT algorithm as we did by using an efficient GPU implementation. In the following part, we would like to explain the FFT method in detail. The formal solution of equation 1 can be written in the form of equation 3.

$$\Psi(\vec{r},t) = e^{-\frac{i}{\hbar}\hat{H}(t-t_0)}\Psi(\vec{r},t_0) \tag{3}$$

where $\Psi(\vec{r},t_0)$ is a specified initial state and $\Psi(\vec{r},t)$ is the state after some $\delta t = t - t_0$ time. The problematic part is the Hamiltonian operator in the exponent. The kinetic and potential operators can not be commuted in general. Hence, the exponential can not be factored. We can decompose the exponential by the symmetrical unitary product [4, 3] as shown in the form 4.

$$e^{-\frac{i}{\hbar}\hat{H}\delta t} = e^{-\frac{i}{\hbar}(\hat{K}+\hat{V})\delta t} \approx e^{-\frac{i}{\hbar}\hat{K}\delta t/2}\, e^{-\frac{i}{\hbar}\hat{V}\delta t}\, e^{-\frac{i}{\hbar}\hat{K}\delta t/2} \tag{4}$$

The error of this approximation is $\mathscr{O}(\delta t^3)$; therefore, we have to be careful with selecting a small enough time resolution. When the potential energy is localized, the $\hat{V}$ operator is a simple multiplication by $V(\vec{r})$ function; thus the middle part of the product can be calculated in the form of equation 5.

$$e^{-\frac{i}{\hbar}\hat{V}\delta t}\Psi = e^{-\frac{i}{\hbar}V(\vec{r})\delta t}\Psi \tag{5}$$

The $\hat{K}$ kinetic operator involves calculating the spatial derivative of the wave function. We can use the Fourier transform, to make the conversion between real space and momentum space. Relation 6 holds for the derivative of an arbitrary f function and its Fourier transform.

$$ik\mathscr{F}\{f\} = \mathscr{F}\{f'\} \tag{6}$$

Taking the derivative in real space means multiplication by ik imaginary wave number in momentum space. We work with the $\Delta = \nabla \cdot \nabla$ Laplace operator, so we have to multiply by $(ik)^2 = -k^2$. By exploiting the linearity of the Fourier transform, we arrive at formula 7 for the kinetic energy part of the Hamiltonian function.

$$\hat{K}\Psi = \frac{p^2}{2m}\Psi = -\frac{\hbar^2}{2m}\Delta\Psi = -\frac{\hbar^2}{2m}\mathscr{F}^{-1}\{-k^2\mathscr{F}\{\Psi\}\} \tag{7}$$

where $\mathscr{F}^{-1}$ is the inverse Fourier transform. In momentum space, the k wave number is trivially given as it can be thought of as the very coordinate the function is parameterized with.

Actually, in equation 4, the $\hat{K}$ kinetic energy operator is in the exponent multiplied by $-\frac{i}{\hbar}\delta t/2$. Using the knowledge gathered from equation 7, we can now write equation 8.

$$e^{-\frac{i}{\hbar}\hat{K}\delta t/2}\Psi = \mathscr{F}^{-1}\left[e^{-\frac{ik^2\hbar\delta t}{4m}}\mathscr{F}\left[\Psi\right]\right] \tag{8}$$

Having a discrete data set, Discrete Fourier transform (DFT) can be efficiently implemented using the Fast Fourier Transform (FFT) algorithm. The output of the simulation is the wave function. The probability density can be obtained by calculating the square of the absolute value of the wave function for each grid cell. Making use of formulas 4, 5, and 8 and plugging them into the formal solution of the Schrödinger equation we can create algorithm 1 for the time development of the wave function.

Algorithm 1 Time advance algorithm

$\Psi \leftarrow$ initial state of the wave function
$V \leftarrow$ localized potential
$\delta t \leftarrow$ time resolution
$N_t \leftarrow$ number of time steps
$P_V \leftarrow e^{-\frac{i}{\hbar}V(\vec{r})\delta t}$
$P_K \leftarrow e^{-\frac{ik^2\hbar\delta t}{4m}}$
for $i \in [0, N_t)$ **do**
  $\Psi^{(1)} \leftarrow FFT^{-1}\left[P_K\, FFT\left[\Psi\right]\right]$
  $\Psi^{(2)} \leftarrow P_V\, \Psi^{(1)}$
  $\Psi \leftarrow FFT^{-1}\left[P_K\, FFT\left[\Psi^{(2)}\right]\right]$
  Visualize $|\Psi|^2$
end for

3.1 Defining Gaussian wave packets

In the algorithm, first, we have to specify an initial state of the wave function. Erwin Schrödinger introduced the concept of the Wave Packet (WP). A WP is a wavefront that propagates and reflects as a classical particle would do and also exhibits all the wave-like behavior described by QM. It bridges the gap between classical and quantum physics. The term Wave Packet Dynamics (WPD) refers to the process of modeling QM systems by initializing WPs and observing the propagation, reflection, scattering, and interference of the WP. In our work, we use Gaussian WPs. In this case the probability density of the WP has Gaussian distribution [21], hence the name. The definition of such wave function can be written in the form of equation 9.

$$\psi(\vec{r}) = \left[\frac{2}{\pi a^2}\right]^{\frac{D}{4}} exp\left[i\vec{k_0}\cdot\vec{r}\right] exp\left[-\frac{|\vec{r}-\vec{r_0}|^2}{a^2}\right] \quad (9)$$

where $\vec{r_0}$ is the initial position (with the highest probability density), $\vec{k_0}$ is the initial wave vector, and D is the dimension, which is $D := 3$ in our simulation. We can obtain the width of the Gaussian WP as $\Delta r = \frac{a}{2}$.

If we do not want to visualize the probability density in each iteration, we can further optimize the calculation by merging the first step of the nth iteration and the last step of the $(n-1)$th iteration. If we omit the visualization step, we can do one forward FFT then perform a multiplication between the moment space wave tensor and the P_K^2 kinetic propagator calculated for a whole δt interval, instead of the one used in Algorithm 1 calculated for $\delta t/2$ interval.

4 Our implementation

Using the Fourier method, we created a Python application simulating the time development of the quantum mechanical wave function. We use ray tracing to visualize the resulting volumetric probability density. The visualization requires the sampling of a 3D data set on a discretized grid. This makes it impossible to fully reconstruct the wave function that we simulated using only a finite resolution to begin with. In order to fight sampling artifacts, we deploy a state-of-the-art triquadratic reconstruction filter recently proposed by Balázs Csébfalvi [2].

4.1 GPU parallelization and Just-In-Time compilation

The Fourier method described in section 3 opens up the possibility to implement the simulation on the GPU. Using GPU acceleration is one of our contributions to the already existing implementation used at the Nanotechnology Departement, Institute for Technical Physics and Materials Science, Centre for Energy Research. The Compute Unified Device Architecture (CUDA) toolkit is often used for parallel computational tasks implemented on the GPU [12]. It comes with a powerful GPU based FFT implementation. To use CUDA with Python, we selected the CuPy wrapping library [13] that provides abstraction over CUDA. We have used Numba to access Just-In-Time compilation (JIT) features. JIT means that for some parts of the otherwise interpreted source code, the compiler performs a runtime translation to native code. This feature is especially useful when iterating over large arrays.

4.2 Performance test

We measured the performance of our application. We used a personal laptop to run and test the program. The system specification of our computer is summarized in figure 1. First, we tried a CPU-only version of our simulator to compare the results with the GPU accelerated implementation. The results of the comparison can be found in figure 2. Here, we tested three different configurations with varying resolutions. We measured the average iteration count per second. The test shows that by using GPU acceleration, we obtained significant speed up over the CPU-only implementation.

CPU	AMD Ryzen 5 6600H 3.30 GHz
GPU	NVIDIA GeForce RTX 3050 Ti Laptop GPU
RAM	16 GB
OS	MS Windows 11 64-bit

Figure 1: System specification of the used test hardware

Input size	CPU only [iter/s]	GPU accel. [iter/s]
128^3	1.1	11.5
256^3	0.09	6.5
512^3	0.01	0.5

Figure 2: Results of a performance test using a CPU-only and the GPU accelerated version

5 Results

5.1 General approach and the system of units

We used our software to perform various WPD simulations. In this section, we present the results of some of these simulations. Our simulator uses Hartree atomic units [7]. Every quantity in the upcoming part should be interpreted as such. This unit system makes it convenient to deal with quantities at atomic scale.

5.2 Double-slit experiment

First, we simulated electron scattering experiments. Scattering of a particle happens when the WP of the particle passes through some kind of a barrier with holes in it. In our simulation, we can model the barrier as a localized potential. The WP arrives from one side of the barrier. While passing through this barrier, it scatters, and some of the WP gets reflected. The portion of the WP that passed through –suffering scattering– continues forward and consequently arrives to a measuring device[1]. In our simulation, our measuring device is a virtual canvas where we measure the probability density. A simple scattering scenario is the double-slit experiment. Here the barrier is a potential wall with two narrow parallel slits. The WP passes through these slits.

We performed the simulation using a distance between the barrier and the measuring canvas of $L = 30$ Bohr radii, a distance between the two slits of $d = 4.0$ Bohr radii, and a WP wavelength of $\lambda = \frac{2\pi}{3} \simeq 2.1$ Bohr radii. The width of each slit was a small enough value of 1.0 Bohr radii. Snapshots of the double-slit simulation can be seen in figure 3, where we used ray tracing to visualize the probability density and the potential. The corresponding interference pattern is visualized in figure 4 on a canvas of size 60×60 Bohr radius.

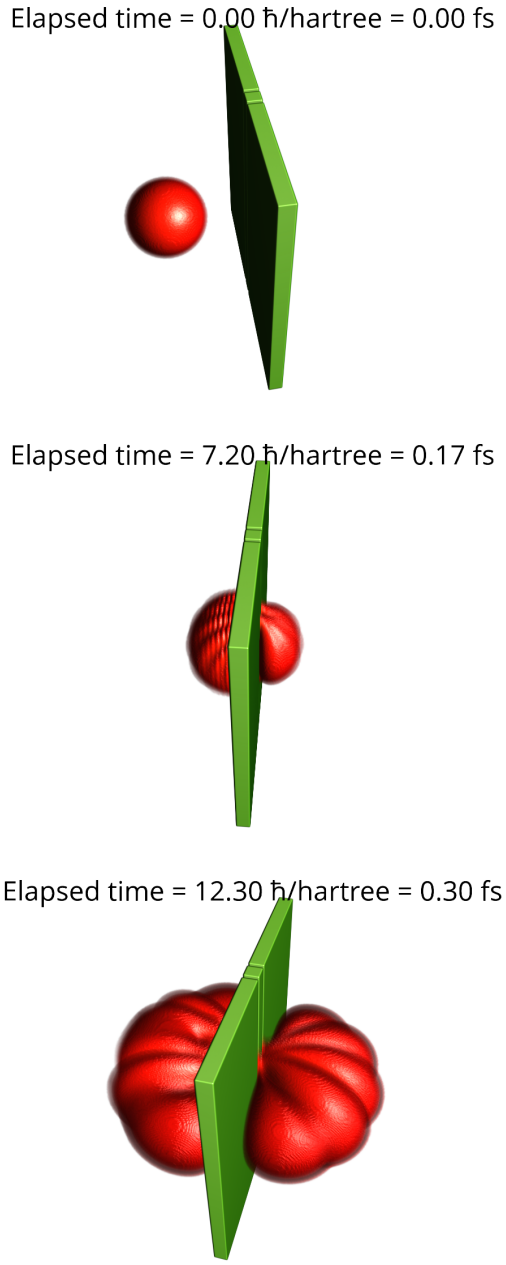

Figure 3: Double-slit experiment: the wave packet passes through the slits in the potential barrier

5.3 Diffraction by optical grating-like potential

Many different forms of diffraction can be explained using QM. The scale at which diffraction happens ranges from the scale of subatomic particles to larger molecules. Measuring diffraction patterns is a useful tool in the hands of scientists. It provides information about the object that caused the diffraction. The previously presented double-slit experiment is a 2D phenomenon because the localized potential is independent of the z coordinate. In the third dimension, there is free propagation. To make use of all three simulated dimensions, we also modeled diffraction on diffraction gratings. In optics, a diffraction grating is a periodic 2D structure that diffracts light [17]. In QM, similar gratings can also be utilized to diffract wave packets. The holes between the potential nodes behave like the holes in the double-slit experiment. We put 11 nodes in each direction, forming a rectangular grid. Each node has a Gaussian potential distribution and a maximal potential of $V_{max} = 8$ Hartree. The distance between adjacent grid points is $d = 4$ Bohr radii. The canvas distance is $L = 30$ Bohr radii, and the wavelength of the WP is $\lambda \simeq 2.1$ Bohr radii. Note that the kinetic energy of the WP $E = \frac{p^2}{2m} = \frac{h^2}{2\lambda^2} \simeq 4.5$ Hartree is less than V_{max}. Oth-

[1] In scattering and diffraction experiments we can make the distinction between a near field and far field solution.

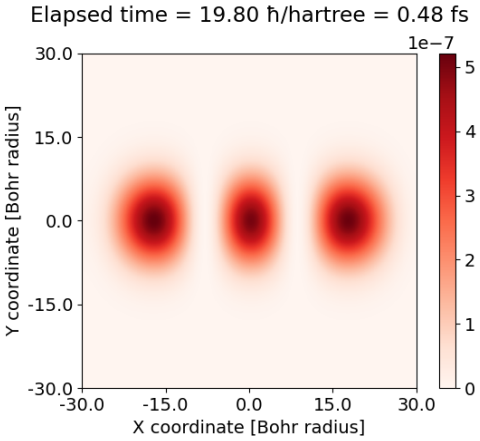

Figure 4: Simulated interference pattern of double-slit experiment

erwise, the grating would not impact the propagation of the WP sufficiently. In figure 5, we visualize subsequent stages of the scattering.

During the simulation, many interesting interference patterns arise. We show some of these for the 4 Bohr radii lattice constant case in figure 6, and for the 8 Bohr radii lattice constant case in figure 7.

5.4 Many-body interactions

One interesting use case of a higher-dimensional WPD simulator is that the higher-dimensional space can be used to model the interactions between multiple lower-dimensional particles. For example, our 3D simulation is capable of the simulation of three 1D particles. To do this, we have to define a special interaction potential. To create such potential, we have to think about the coordinates in the higher-dimensional configuration space as the coordinates of multiple lower-dimensional particles. If the potential energy affecting all particles can be expressed as a $V(x_a, x_b, x_c)$ function of the location of particle A and B and C, then we can reinterpret this function as the $V(\vec{r})$ localized potential function used in the potential propagator in equation 5. Note that here $\vec{r}$ becomes (x_a, x_b, x_c). To model the interaction between the three 1D particles, we initialized a hard interaction potential that takes its maximum inside an ε radius around each particle otherwise it is constant zero. To prevent blotting of the Gaussian WP we also added a harmonic oscillator potential. This helps because the Gaussian WP is the eigenstate of the harmonic oscillator. The potential for a harmonic oscillator is given in the form of equation 10.

$$V(x) = \frac{m\omega^2}{2}x^2 \tag{10}$$

Here m is the oscillating mass and ω is the angular frequency of the oscillation. We created a scenario where particle A starts at 25 Bohr radii away from the center of

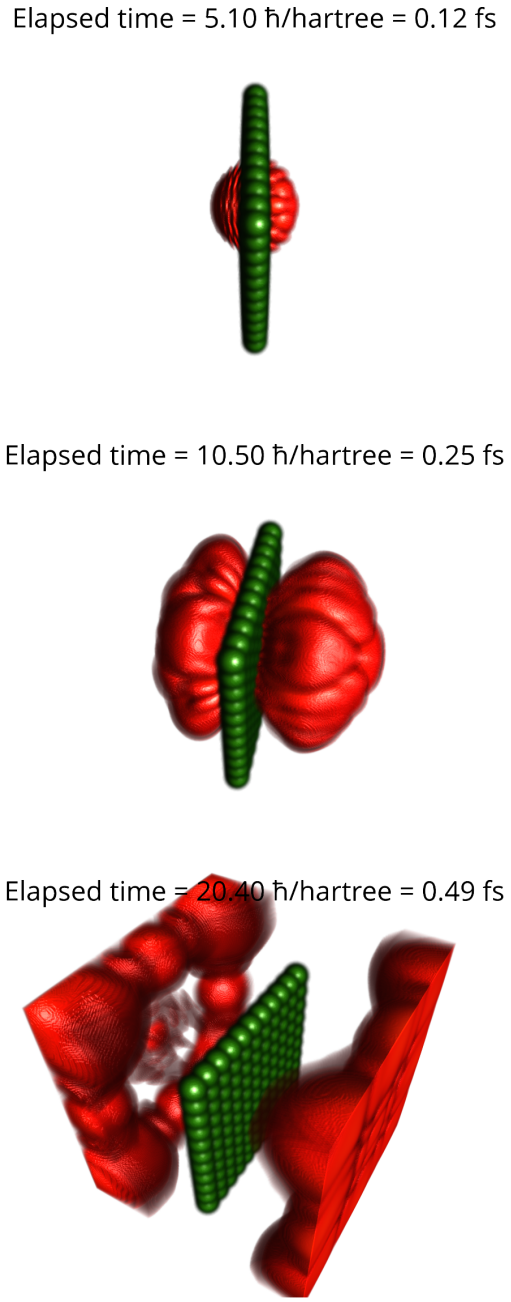

Figure 5: Diffraction grating experiment: the grating has a lattice constants of 4 Bohr radii

the oscillator where the potential energy is maximal; thus, it accelerates towards the other two particles (B and C), consequently transferring the momentum to particle C on the far right. The angular frequency of the oscillator was selected to be $\omega = \frac{2\pi}{40} \simeq 0.1571 \ \frac{\text{rad·Hartree}}{\hbar}$.

We placed a finite potential barrier in the middle of the oscillator. We chose the thickness of this barrier so that approximately half of the wave packet of particle B tunnels through the barrier, giving its momentum to particle C on the next side of the wall. This causes C to start moving

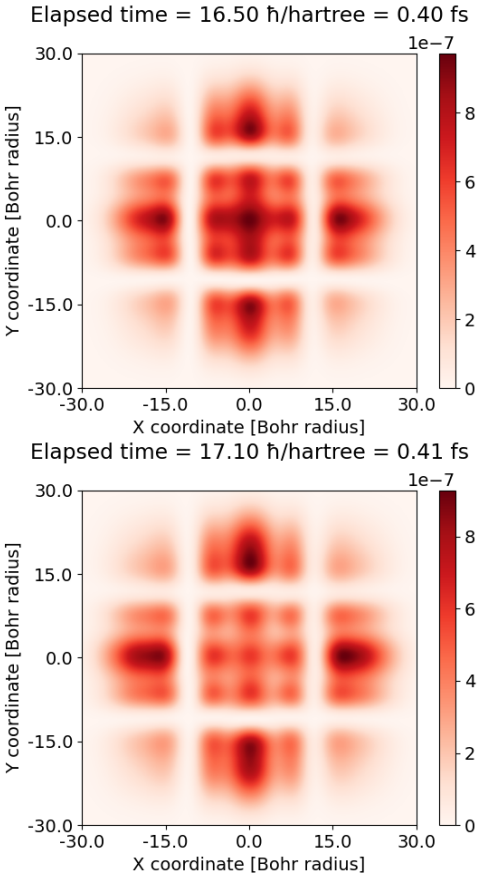

Figure 6: Interference pattern forming on the measurement canvas during diffraction grating simulation using lattice constant of 4 Bohr radii

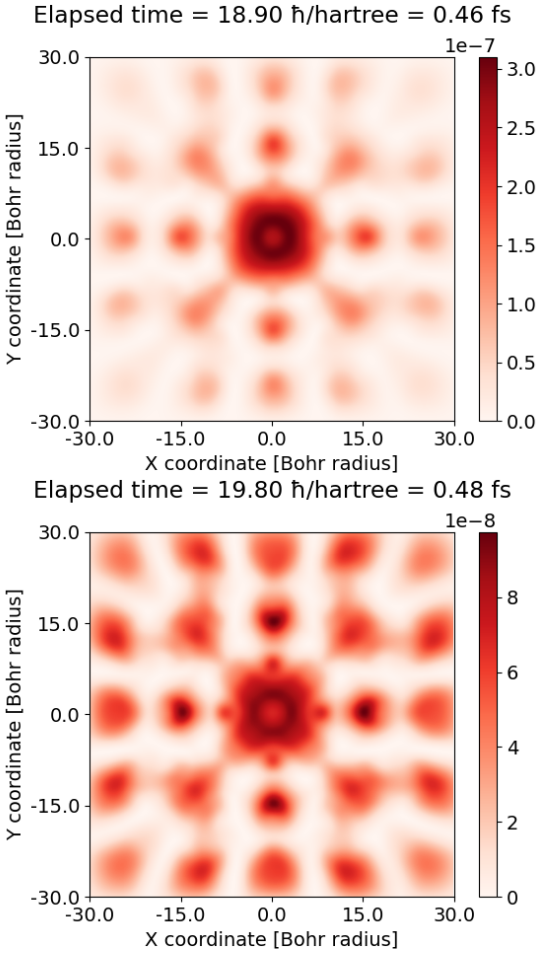

Figure 7: Interference pattern forming on the measurement canvas during diffraction grating simulation using lattice constant of 8 Bohr radii

with a probability of approximately $\frac{1}{2}$. What we just described is called the entanglement of the states of particles A, B, and C. Let's perform a measurement to determine the location of particles A, B, and C right after the previously described sequence of interactions! If we would measure particle A to be located in the middle of the harmonic oscillator, that means that it gave its momentum to particle B and B has tunneled through the finite potential barrier. If B tunneled, that also means that beyond the barrier, it collided with particle C, consequently transferring all of its kinetic energy to C. On the contrary, if the result of the measurement determining the location of particle A would have shown that particle A bounced back from B, that means that B did not tunnel through the barrier. This also means that particle C did not receive any kinetic energy and stayed stationary right beyond the barrier. The measurement of the state of one entangled particle determines the outcome of the measurement of the other entangled particles. Real-life experiments are sound with this thought experiment [20]. The probability density plot can be observed in figure 8.

6 Discussion

In our work, we wrote about simulating the time development of the quantum mechanical wave function in 3D space. Our accomplishments are the following

- We adopted a simulation method that uses the Fourier transform as a subroutine to efficiently calculate the solution of the time-dependent Schrödinger equation.
- As an improvement over Géza István Márk's implementation, we ported the Fast Fourier Transform to the Graphical Programming Unit, thus reaching a major speed-up of a factor of 50 for some cases.
- We combined state-of-the-art volume visualization techniques to enhance the visual quality of the resulting probability density images.
- We used our simulator software to run various Wave Packet Dynamical simulations ranging from basic diffraction scenarios up to simulation of lower-dimensional particles in configuration space.

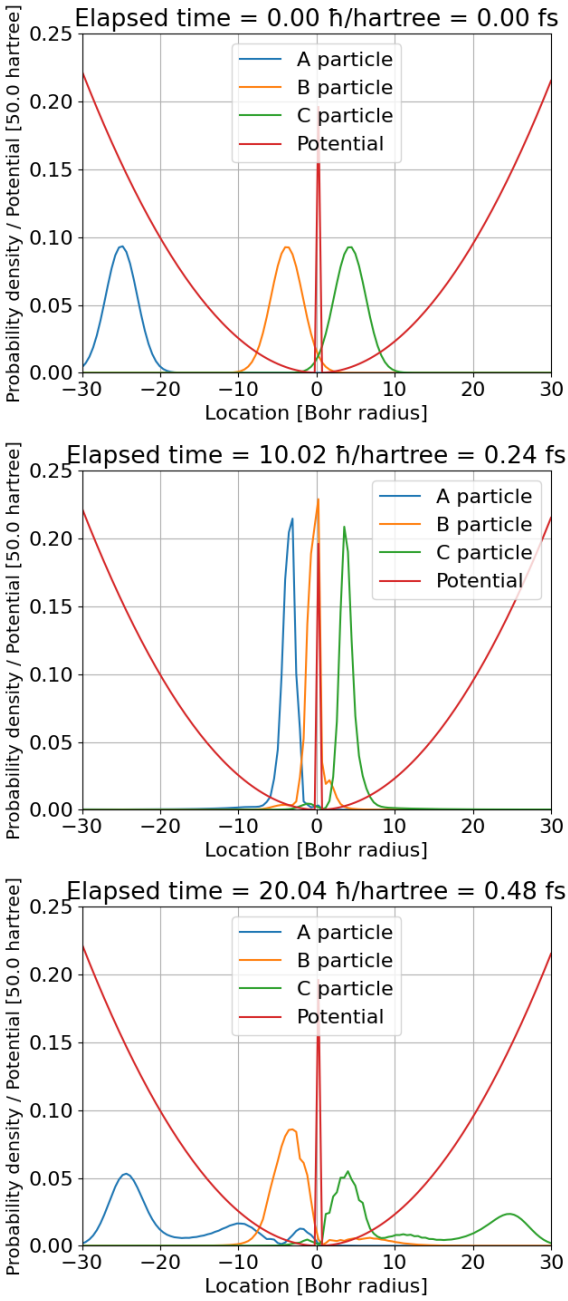

Figure 8: Stages of interactions between 1D particles in harmonic oscillator with finite potential barrier: initial state; particle A colliding with particle B; particle C reaching maximal potential on the right side of the oscillator with approximately 0.5 probability while it's in superposition with the state of staying stationary next to the central barrier

We see our work as a successful entry into the quantum mechanical wave packet dynamics world and a good starting point for further research. In the future, we want to make it possible to calculate the eigenstates of the localized potential. This would require the calculation of the Fourier transform in the time domain to obtain the energy state of the system. Then, iteratively converge towards the eigenstate. There is also a possibility of incorporating electromagnetism into the Hamiltonian operator. As we have simulated 1D particles in configuration space, we could use a higher-dimensional space to model the interaction between multiple multidimensional particles. One interesting path to go down on is to build a machine learning solution that is able to initialize a localized potential field that guides the wave function into a desired state. This particular idea is inspired by the marvelous work of Barnabás Börcsök, who presented his paper about controlling 2D laplacian eigenfluids [1] at the Central European Seminar on Computer Graphics in 2023. From a visualization point of view, there are also many possibilities to improve. There is room for even better reconstruction filters. We are very hopeful about the future research potential of this topic and are very eager to continue the fruitful work.

7 Acknowledgements

I am immensely grateful for the support of my supervisor, Dr. Balázs Csébfalvi, and for all the valuable insights and advice of the researchers at the Centre for Energy Research: Dr. Géza István Márk and Dr. Péter Vancsó. This work has been supported by the OTKA K-145970 project.

References

[1] Barnabás Börcsök. Controlling 2d laplacian eigenfluids. *Central European Seminar on Computer Graphics*, 2023.

[2] Balázs Csébfalvi. One step further beyond trilinear interpolation and central differences: Triquadratic reconstruction and its analytic derivatives at the cost of one additional texture fetch. *Computer Graphics Forum*, 42(2):191–200, 2023.

[3] M.D Feit, J.A Fleck, and A Steiger. Solution of the schrödinger equation by a spectral method. *Journal of Computational Physics*, 47(3):412–433, 1982.

[4] J. A. Fleck, J. R. Morris, and M. D. Feit. Time-dependent propagation of high energy laser beams through the atmosphere. *Applied physics*, 10(2):129–160, Jun 1976.

[5] A. K. Geim. Graphene: Status and prospects. *Science*, 324(5934):1530–1534, 2009.

[6] William Rowan Hamilton. *On a general method of expressing the paths of light, & of the planets, by the coefficients of a characteristic function*. Printed by P.D. Hardy Dublin, Dublin, 1833.

[7] D. R. Hartree. The wave mechanics of an atom with a non-coulomb central field. part i. theory and methods. *Mathematical Proceedings of the Cambridge Philosophical Society*, 24(1):89–110, 1928.

[8] D Kosloff and R Kosloff. A fourier method solution for the time dependent schrödinger equation as a tool in molecular dynamics. *Journal of Computational Physics*, 52(1):35–53, 1983.

[9] Geza Mark, Péter Vancsó, Laszlo Biro, Dmitry Kvashnin, Leonid Chernozatonskii, Andrey Chaves, Khamdam Rakhimov, and Philippe Lambin. *Wave Packet Dynamical Calculations for Carbon Nanostructures*, pages 89–102. 01 2016.

[10] Frederick Ira Moxley, Tim Byrnes, Fumitaka Fujiwara, and Weizhong Dai. A generalized finite-difference time-domain quantum method for the n-body interacting hamiltonian. *Computer Physics Communications*, 183(11):2434–2440, 2012.

[11] Géza I. Márk. Web-schrödinger: Program for the interactive solution of the time dependent and stationary two dimensional (2d) schrödinger equation, 2020.

[12] John Nickolls, Ian Buck, Michael Garland, and Kevin Skadron. Scalable parallel programming with cuda. In *ACM SIGGRAPH 2008 Classes*, SIGGRAPH '08, New York, NY, USA, 2008. Association for Computing Machinery.

[13] Ryosuke Okuta, Yuya Unno, Daisuke Nishino, Shohei Hido, and Crissman Loomis. Cupy: A numpy-compatible library for nvidia gpu calculations. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS)*, 2017.

[14] C. N. R. Rao, Kanishka Biswas, K. S. Subrahmanyam, and A. Govindaraj. Graphene, the new nanocarbon. *J. Mater. Chem.*, 19:2457–2469, 2009.

[15] E. Schrödinger. An undulatory theory of the mechanics of atoms and molecules. *Phys. Rev.*, 28:1049–1070, Dec 1926.

[16] Meryl D. Stoller, Sungjin Park, Yanwu Zhu, Jinho An, and Rodney S. Ruoff. Graphene-based ultracapacitors. *Nano Letters*, 8(10):3498–3502, 2008. PMID: 18788793.

[17] George W. Stroke. *Diffraction Gratings*, pages 426–754. Springer Berlin Heidelberg, Berlin, Heidelberg, 1967.

[18] G. Tamás. *Kvantummechanika*. Elméleti fizika. Typotex, 2007.

[19] Péter Vancsó, Géza I. Márk, Philippe Lambin, Alexandre Mayer, Yong-Sung Kim, Chanyong Hwang, and László P. Biró. Electronic transport through ordered and disordered graphene grain boundaries. *Carbon*, 64:101–110, 2013.

[20] D. Vasilyev, F. O. Schumann, F. Giebels, H. Gollisch, J. Kirschner, and R. Feder. Spin-entanglement between two freely propagating electrons: Experiment and theory. *Phys. Rev. B*, 95:115134, Mar 2017.

[21] Xinhua Zhang. *Gaussian Distribution*, pages 425–428. Springer US, Boston, MA, 2010.

[22] Min Zhu and Yi Wang. Rk-ho-fdtd scheme for solving time-dependent schrodinger equation. *The Applied Computational Electromagnetics Society Journal (ACES)*, 36(08):968–972, Oct. 2021.

[23] Houlong L. Zhuang and Richard G. Hennig. Computational discovery, characterization, and design of single-layer materials. *JOM*, 66(3):366–374, Mar 2014.