

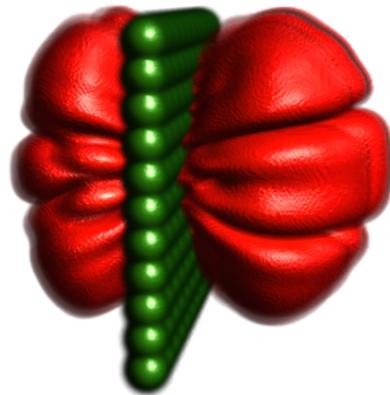
# Time evolution simulation of the quantum mechanical wave function in 3D space

Zoltán Simon

Budapest University of Technology and Economics  
Department of Control Engineering and Information Technology  
Computer Engineering, M.Sc.  
2. semester

Advisors: Dr. Balázs Csébfalvi\* Dr. Péter Vancsó† Dr. Géza István Márk‡

2023. november 1.



---

\*Budapest University of Technology and Economics, Department of Control Engineering and Information Technology

†Institute for Technical Physics and Materials Science, Centre for Energy Research, 1121 Budapest, Hungary

‡Institute for Technical Physics and Materials Science, Centre for Energy Research, 1121 Budapest, Hungary

# Tartalomjegyzék

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Theoretical background</b>	<b>4</b>
2.1	Beginnings of quantum mechanical wave dynamics . . . . .	4
2.2	Wave packet dynamics . . . . .	5
2.3	The Schrödinger equation . . . . .	7
<b>3</b>	<b>Calculation method</b>	<b>8</b>
3.1	Survey and comparison of available methods . . . . .	8
3.2	Detailed explanation of the used Fourier method . . . . .	9
3.3	Defining Gaussian wave packets . . . . .	11
3.4	A simple trick to optimize the algorithm . . . . .	12
3.5	Free space boundary condition . . . . .	12
<b>4</b>	<b>Our implementation</b>	<b>13</b>
4.1	Short summary of the functionality of our simulator . . . . .	13
4.2	Technologies in use . . . . .	13
4.3	GPU parallelization and Just In Time compilation . . . . .	14
4.4	About the program's user interface . . . . .	14
4.5	Generated output . . . . .	15
4.6	Performance test . . . . .	18
<b>5</b>	<b>Results</b>	<b>18</b>
5.1	General approach and used unit system . . . . .	18
5.2	Double-slit experiment . . . . .	19
5.3	Diffraction by optical grating-like potential . . . . .	20
5.4	Simulating 1D particles in 3D configuration space . . . . .	26
5.5	Modeling a flash memory cell . . . . .	27
<b>6</b>	<b>Discussion</b>	<b>29</b>
<b>7</b>	<b>Acknowledgement</b>	<b>31</b>

## 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 [1] to solve the time-dependent Schrödinger equation efficiently using Fourier transformation. In 2020, Géza István Márk published a paper [2] 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. We found it worth stepping out from the two-dimensional plane and investigating these phenomena in three dimensions. We implemented the above method for the three-dimensional case to simulate the time evolution of the wave function. We used our implementation to simulate typical quantum phenomena using wave packet dynamics. First, we tried the method on analytically describable cases, such as the simulation of the double-slit experiment, then we investigated the operation of flash memory. We used raytraced volumetric visualization to render the resulting probability density. In our work, we introduce the basics of wave packet dynamics in quantum mechanics. We describe the used method in detail and showcase our simulation results.

For further information and animations please visite

<https://zoltansimon.info/src/content/research/wavepacketsim.html>

## Kivonat

A kvantummechanikában a hullámfüggvény írja le egy fizikai rendszer állapotát. Nemrelativisztikus esetben a hullámfüggvény időbeli fejlődését az időfüggő Schrödinger-egyenlet határozza meg. 1982-ben D Kosloff és R Kosloff közölt egy módszert [1] az időfüggő Schrödinger-egyenlet hatékony megoldására Fourier-transzformáció felhasználásával. 2020-ban Márk Géza István cikkében [2] bemutatott egy interaktív számítógépes programot az időfüggő és a stacionárius kétdimenziós Schrödinger-egyenlet megoldására. A kvantumos jelenségek bizonyos részletei csak mindenkor térdimenzióval számolva figyelhetők meg. Érdemesnek tartottuk a síkból kilépve, térben is megvizsgálni ezeket a jelenségeket. Munkánkban háromdimenziós esetre implementáltuk az említett módszert, hogy szimuláljuk a hullámfüggvény időbeli fejlődését. Hullámcsomag-dinamikát használva kipróbáltuk több jellegzetes kvantumos jelenség szimulációját. A módszert először analitikusan kezelhető esetekre teszteltük, például a kétréses kísérlet szimulációjára, ezután megvizsgáltuk a flash memória cella működését. A szimuláció kimeneteként előálló valószínűségi sűrűséget sugárkövetéses térfogati megjelenítéssel ábrázoltuk. Dolgozatunkban ismertetjük a kvantummechanikai hullámcsomag-dinamika alapjait. Részletesen leírjuk a használt eljárást, és bemutatjuk a szimulációk eredményeit.

További információ és animációk találhatóak a

<https://zoltansimon.info/src/content/research/wavepacketsim.html> webcímen.

# 1 Introduction

In the first quarter of 20th century Quantum Mechanics (QM) opened a whole new window to understand our universe. Tamás Geszti in his book [3] writes: *"learning QM is part of the process of understanding the world, and the person who masters it, understands the world better"*. It is a common view that while Albert Einstein's General Relativity (GR) provides a model that accurately describes the laws of nature governing large scale phenomena [4], quantum theory excels at atomic scale<sup>1</sup>. Although humanity has not yet accepted a single theory that would be capable of modeling both the small and large thus bridging the gap between QM and GR, there are many use-cases, where technology heavily depends on both theories. QM can be used efficiently to model the behavior of atomic particles. It describes how electrons behave in the orbitals around atomic core and give an explanation for chemical reactions. It can be used to model the structure of molecules. QM made possible for humanity the make use of the nuclear energy in power plants thus creating a new and efficient source of power. Computers –as we know them today– wouldn't be possible without a deep understanding of various quantum phenomena. These systems make use of transistors [5] which can also be thought of as a product of QM.

In nanotechnology it is crucial to make quantum mechanical calculations to predict –and in many cases explain– the behavior of different nanostructures. One interesting field of study is the science of single-layer materials [6]. These are also known as 2D materials. One such carbon structure is called graphene [7, 8, 9]. This single-layered structure conducts heat and electricity very efficiently thus raises high hopes in many when it comes to possible use-cases. Nowadays quantum information science is getting larger and larger audience [10]. Quantum Computing promises newer before seen increase in computation capabilities due to the massively parallel nature of quantum systems leveraging quantum superposition. Quantum Communication on the other hand opens new possibilities when it comes to secure information exchange with efficient channel encoding. Although both of the last two fields mentioned await technological evolution in order to be deployable at scale due to a large amount of technical challenges, they both show that QM has a lot of real life applications and the spectrum of these will only become more colorful as technology evolves.

Inspired by the previously enumerated large amount of various fields of application we set the goal to study the behavior of quantum systems by computer simulation. Such simulations are very useful for scientists. They use such methods to accurately model 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 [11]. In 1982, D Kosloff and R Kosloff proposed a method [1] to solve the time-dependent Schrödinger equation efficiently using Fourier transformation. In 2020, Géza István Márk published a paper [2] describing a computer program for the interactive solution of the time-dependent and stationary two-dimensional (2D) Schrödinger equation. Some details

---

<sup>1</sup>Under extreme circumstances such as in black holes macroscopic quantum phenomena are also present

of quantum phenomena are only observable by calculating with all three spatial dimensions. We found it worth stepping out from the two-dimensional plane and investigating these phenomena in three dimensions. Géza István Márk and his colleagues have already used 3D calculations in their research work [12, 13]. 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 isosurface method. Our contribution mainly lays in leveraging the parallelisation potential of the modern Graphical Processing Unit (GPU) thus significantly boosting the speed of the calculation 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 for analysis of the evolution of probability density in 3D space. We believe that by combining the knowledge available for computer visualization specialists and physicists we can arrive to something greater than the possibilities available if each of us would be working only in his or her own domain of expertise. We write our implementation with future extendibility in mind, since we would like to continue our work and develop a capable simulation platform that can be deployed as a tool for state of the art scientific research.

In Section 2 of this work we start by discussing the theoretical background. Then in Section 3 we proceed to describe the used Fourier method to simulate the time development of the wave function. In Section 4 we provide an overview of the implementation details of our program. After this in Section 5 we showcase our simulation results. Here we compare the result of simple, analytically describable simulation scenarios to the mathematical model. At the end in Section 6 we summarize our results in a short discussion.

## 2 Theoretical background

### 2.1 Beginnings of quantum mechanical wave dynamics

In this section we would like to give a brief introduction to the theory of QM. In this summary of physics history we rely mainly on Tamás Geszti's QM book [3]. For English literature we would like to point the reader's attention towards the critically acclaimed book of Claude Cohen-Tannoudji et al. [14].

in 1924, Louis de Broglie recognized that an electron moving with momentum of  $p = M_e v$  can also be thought of as the propagation of some kind of a matter wave with  $\lambda$  wavelength.

$$\lambda = \frac{h}{p} \tag{1}$$

where  $h$  is called the Planck constant and equals to  $h = 6,6 \times 10^{-34} \text{ Js}$ . This also explains the discrete energy levels in Bohr's model. The allowed orbitals are those where the wave is a single-valued function thus closes into itself after one full rotation of  $2\pi$  radians. If we use polar coordinates this means that the wave has the same complex amplitude for coordinates  $\alpha$  and  $\alpha + n2\pi$  where  $n = 0, 1, 2, \dots$ . Here we can further modify equation 1 and arrive to

$$\vec{p} = \hbar \vec{k} \tag{2}$$

where  $\vec{k}$  is the wave vector and  $\hbar = \frac{h}{2\pi}$  is the reduced Planck constant. The amplitude of this vector equals to  $\|\vec{k}\| = \frac{2\pi}{\lambda}$  and its direction is perpendicular to the wave front.

Experiments show that matter waves exhibit self interfering behavior similarly to electromagnetic waves<sup>2</sup>. For example the double-slit experiment, where a wave is propagating through a barrier with a pair of two narrow slits and consequently creates an interference pattern on a canvas placed after the barrier can be performed not only with light but a single electron as well. When the experiment is performed with electrons, the canvas is replaced with a measurement device that is able to register the incoming electron. By performing this experiment once the measuring device will only register the electron at a single location. The interference gets noticeable only after repeating the measurement multiple times since the distribution of the measured electron impact locations converges to the interference pattern. This behavior is also referred to as wave-particle duality because the particle interferes with itself but also there is only one single impact at the measuring device for each iteration of the experiment. Another important experimental observation is that if we create the linear combination of multiple matter waves and then the combined wave evolves over time than the resulting wave is precisely the same as if the originally combined waves would have evolved independently and we would only have combined the resulting waves. This latter property was also true for electromagnetic waves and it's called the superposition principle. Please note that although we keep comparing electromagnetic waves with matter waves the two are different phenomena. One very important difference is that interference of electromagnetic waves happen due to different strength of electric and magnetic fields. In QM the wave function has a complex value thus it can interfere even when only the phase is different and the amplitude is the same. Our goal with emphasizing the similarities is only to reinforce a possible intuition in those who are perhaps more familiar with the world of electromagnetism.

## 2.2 Wave packet dynamics

Erwin Schrödinger introduced the concept of the Wave Packet (WP) to show that in limiting cases QM gives back the classical particle model where small bullets are moving around colliding with each other. This particle-like propagation of a WP of the wave is due to the superposition of waves with different frequencies forming peaks. These waves extinguish each other for the most part but at some position and time coordinates the amplitudes add up to larger amplitude. An example of this is demonstrated in figure 1.

$\Psi_k$  denotes the complex amplitude of a matter wave with  $k = \frac{2\pi}{\lambda}$  wave number. The formula of a plain wave propagating in positive  $x$  direction can be described in the form of equation 3.

$$\Psi_k(x, t) = e^{i(kx - \omega t)} \quad (3)$$

let's calculate the superposition of waves with different wave numbers! We assume that the angular velocity  $\omega(k)$  depends on the  $k$  wave number and that the waves with different wave

---

<sup>2</sup>An electron always only interferes with itself while light waves coming from multiple light sources can also interfere with each other.

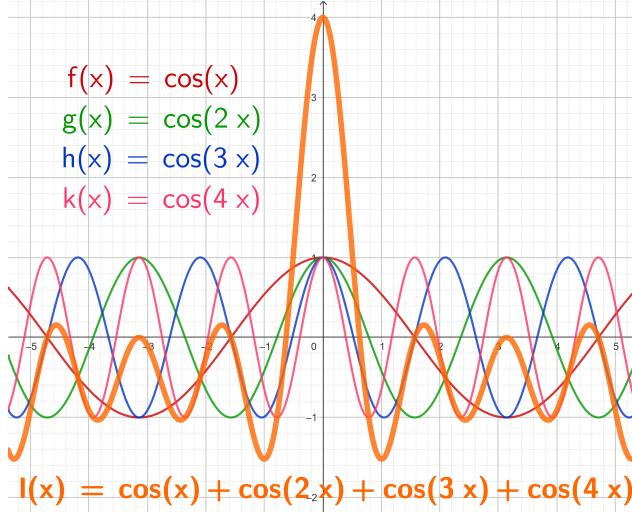


Figure 1: Superposition of cosine waves with different wavelength: the superposition forms a peak at the  $x = 0$  coordinate

numbers have various  $c(k)$  weights in the linear combination. This gives the following Fourier integral:

$$\Psi(x, t) = \int_k c(k) \sin[kx - \omega(k)t] dk \quad (4)$$

Analyzing this formula we can determine the propagation velocity of the amplitude peaks. In equation 4 the peaks form where and when the phase of the sine function is not dependent of the  $k$  integration variable. With other words the derivative of the phase with respect to  $k$  is zero.

$$\frac{d}{dk}[kx - \omega(k)t] = x - \frac{d}{dk}\omega(k)t = 0 \quad (5)$$

By generalizing for all spatial coordinates and rearranging equation 5 we get the group velocity of the wave.

$$\vec{v}_{group} = \frac{\delta\omega(\vec{k})}{\delta\vec{k}} \quad (6)$$

By using the  $E := \hbar\omega$  and  $\hbar\vec{k} = \vec{p}$  substitutions for the energy and momentum we arrive to the classically known formula for a particle of mass  $m$

$$\vec{v}_{group} = \frac{\delta E(\vec{p}, \vec{r})}{\delta \vec{p}} = \frac{\delta}{\delta \vec{p}} \left[ \frac{\|\vec{p}\|^2}{2m} + V(\vec{r}) \right] = \frac{\vec{p}}{m} \quad (7)$$

There are however significant differences between the classical and QM model. The particle-analogy comes with it's own limits. In 1927, Werner Heisenberg formulated his uncertainty principle. This

states that simultaneously the position and the momentum of a particle can not be determined exactly. Only with a specific  $\Delta$  precision. The result can be written in the form of equation 8.

$$\Delta x \cdot \Delta p \geq \frac{\hbar}{2} \quad (8)$$

Similar relation holds between time and energy.

$$\Delta t \cdot \Delta E \geq \frac{\hbar}{2} \quad (9)$$

### 2.3 The Schrödinger equation

So far we know that atomic particles exhibit wave-like behavior and in bounded systems they can absorb or release energy only in discrete quanta. These matter waves can interfere with each other and the superposition principle holds true. Along the way we obtained a seemingly satisfactory energy, momentum and velocity concept. One last step is to instead of real amplitudes we allow complex amplitudes. This solves the problem of energy conservation.

All of these advancements led up to the formalization of the Schrödinger equation in 1926 by Erwin Schrödinger [11]. Already in the 19th century on the field of optics it was a common practice to use linear partial differential equations to calculate light propagation. Linearity is a requirement for matter waves as well, since by the definition of superposition a general equation that aims to properly describe the behavior of matter waves must be satisfied by the simple waves and also by the linear combination of these waves. Let's write the  $\Psi$  plane wave as a complex valued function

$$\Psi(\vec{r}, t) = e^{i(\vec{k} \cdot \vec{r} - \omega t)} = e^{\frac{i}{\hbar}(\vec{p} \cdot \vec{r} - Et)} \quad (10)$$

First let's assume that our wave is propagating in free space. The total energy of the particle consists only of kinetic part.

$$E = \frac{p^2}{2m} \quad (11)$$

We don't know the equation for the wave function yet, but it must incorporate this relation between the energy and momentum. In the wave function both the energy and the momentum are in exponent. To express them we have to take the derivative of the function with respect to position and time. Operation of calculating the partial derivative with respect to position is often referred to as calculating the gradient vector and is denoted by the  $\nabla$  ("nabla") symbol.

$$\begin{aligned} \frac{d}{dt} \Psi(\vec{r}, t) &= -\frac{i}{\hbar} E \Psi(\vec{r}, t) \\ \frac{\delta}{\delta \vec{r}} \Psi(\vec{r}, t) &= \nabla \Psi(\vec{r}, t) = \frac{i}{\hbar} \vec{p} \Psi(\vec{r}, t) \end{aligned} \quad (12)$$

In the kinetic energy equation 11 we actually need the square of the magnitude of momentum. To calculate it we introduce the  $\Delta = \vec{\nabla} \cdot \vec{\nabla}$  Laplace operator. By combining equation 11 and 12 we

can arrive to the following linear differential equation

$$i\hbar \frac{d}{dt} \Psi(\vec{r}, t) = -\frac{\hbar^2}{2m} \Delta \Psi(\vec{r}, t) \quad (13)$$

This is the Schrödinger equation for matter wave propagating in free space. Let's generalize this by getting rid of the previous assumption that the total energy is solely kinetic. Now we allow an  $V(\vec{r})$  localized potential energy to be an arbitrary function of location thus the total energy becomes

$$E = \frac{p^2}{2m} + V(\vec{r}) = \mathcal{H}(\vec{p}, \vec{r}) \quad (14)$$

Here we introduced the  $\mathcal{H}$  notation to express that this is a form of the so called Hamiltonian function named after William Rowan Hamilton who used his formalism to describe classical mechanics in the 19th century [15]. Using equation 14 to modify equation 13 we trivially arrive to the more general form of the Schrödinger equation

$$\begin{aligned} i\hbar \frac{d}{dt} \Psi(\vec{r}, t) &= -\frac{\hbar^2}{2m} \Delta \Psi(\vec{r}, t) + V(\vec{r}) \Psi(\vec{r}, t) \\ \frac{d}{dt} \Psi(\vec{r}, t) &= -\frac{i}{\hbar} \hat{H} \Psi(\vec{r}, t) \end{aligned} \quad (15)$$

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 a  $\hat{H} = -\frac{\hbar^2}{2m} \Delta + V(\vec{r})$  operator affect the wave function. By solving this differential equation and specifying an initial state, we can predict the time development of a quantum mechanical wave function.

The only remaining question also meaningful for our simulation is that how exactly should we interpret the resulting function? What's the actual physical meaning behind the amplitude of the wave? 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 velocity of the particle being found "somewhere" in space equals  $\mathbb{P} = 1$ .

$$\int_{\mathcal{V}} |\Psi(\vec{r}, t)|^2 d^3r = 1 \quad (16)$$

### 3 Calculation method

#### 3.1 Survey and comparison of available methods

In our simulation we numerically solve the time dependent Schrödinger equation. There are multiple available algorithms for this task. A widely used category of solving strategies are the Finite Difference in Time Domain (FDTD) methods. This approach is also widely adopted to simulate the solution of Maxwell equations [16, 17]. FDTD technique for the analysis of quantum devices

(FDTD-Q) however exhibit numerical instability [18]. After a longer sequence of simulation steps the solutions diverge from the ground truth. There are various alterations of the base method. One such method was proposed by Min Zhu et al. [19] in the year 2014. Their algorithm is called the Runge-Kutta High-Order Finite Difference in Time Domain (RK-HO-FDTD). In 2021, a follow up paper [20] demonstrated the superiority of RK-HO-FDTD over the basic FDTD-Q approach. In the same year Frederick Ira Moxley et al. [21] proposed another method based on FDTD-Q. These methods solve the majority of problems related to FDTD-Q.

However in our work we use a different approach. Back in 1982, D Kosloff and R Kosloff proposed a method [1] to solve the time-dependent Schrödinger equation efficiently using Fourier transformation. The advantage of this algorithm 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  $\mathcal{O}(N \log N)$  since it only uses six FFT runs ( $\mathcal{O}(N \log N)$  each) and three element wise multiplication between tensors ( $\mathcal{O}(N)$  each). The amount of FFT runs and multiplications can be reduced further if we don't want to read the results of the time development in each step. A significant speed up can be reached by using parallelized implementation of the FFT algorithm as we did by using an efficient GPU implementation.

### 3.2 Detailed explanation of the used Fourier method

In the following part we would like to explain the FFT method in detail. Let's start by deriving the formal solution of the 15 differential equation.

$$\begin{aligned} \int_{t_0}^t \frac{d}{d\tau} \Psi(\vec{r}, \tau) d\tau &= \int_{t_0}^t -\frac{i}{\hbar} \hat{H} \Psi(\vec{r}, \tau) d\tau \\ \Psi(\vec{r}, t) - \Psi(\vec{r}, t_0) &= \int_{t_0}^t -\frac{i}{\hbar} \hat{H} \Psi(\vec{r}, \tau) d\tau \\ &\vdots \\ \Psi(\vec{r}, t) &= e^{-\frac{i}{\hbar} \hat{H}(t-t_0)} \Psi(\vec{r}, t_0) \end{aligned} \quad (17)$$

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 exponential can not be factored. Géza István Márk in his Web-Schrödinger simulator [2] decomposes the exponential by the symmetrical unitary product as shown in form 18.

$$e^{-\frac{i}{\hbar} \hat{H} \delta t} = e^{-\frac{i}{\hbar} (\hat{K} + \hat{V}) \delta t} \approx e^{-\frac{i}{\hbar} \hat{K} \delta t / 2} e^{-\frac{i}{\hbar} \hat{V} \delta t} e^{-\frac{i}{\hbar} \hat{K} \delta t / 2} \quad (18)$$

The error of this approximation is  $\mathcal{O}(\delta t^3)$  therefore we have to be careful with the selection of small enough time resolution. When the potential energy is localized the  $\hat{V}$  operator is a simple multiplication with  $V(\vec{r})$  function thus the middle part of the product is as follows

$$e^{-\frac{i}{\hbar} \hat{V} \delta t} \Psi = e^{-\frac{i}{\hbar} V(\vec{r}) \delta t} \Psi \quad (19)$$

The  $\hat{K}$  kinetic operator involves calculating the spatial derivative of the wave function.

Previously in equation 12 we have seen that calculating the spatial gradient  $\nabla\Psi(\vec{r}, t)$  yields multiplication by  $\frac{i}{\hbar}\vec{p} = i\vec{k}$ . We only need to get the value of the wave vector. This is when the Fourier transform comes into play which transforms between real space and momentum space. Following relation holds for the derivative of an arbitrary  $f$  function and it's Fourier transform

$$ik\mathcal{F}\{f\} = \mathcal{F}\{f'\} \quad (20)$$

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 Fourier transform we arrive to the following formula for the kinetic energy part of the Hamiltonian function

$$\hat{K}\Psi = \frac{p^2}{2m}\Psi = -\frac{\hbar^2}{2m}\Delta\Psi = -\frac{\hbar^2}{2m}\mathcal{F}^{-1}\{-k^2\mathcal{F}\{\Psi\}\} \quad (21)$$

where  $\mathcal{F}^{-1}$  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 functions are parameterized with. Having that said, it is important to distinguish between  $\nu$  frequency,  $\lambda$  wavelength,  $k$  wave number and even  $\omega$  angular velocity. These relate as follows

$$k = \frac{2\pi}{\lambda} = \frac{2\pi\nu}{v_{group}} = \frac{\omega}{v_{group}} \quad (22)$$

where  $v_{group}$  denotes the group velocity of the wave packet.complex amplitude of a matter wave

Actually in equation 18 the  $\hat{K}$  kinetic energy operator is in the exponent multiplied by  $-\frac{i}{\hbar}\delta t/2$ . Using the knowledge gathered from equation 21 we can now write

$$e^{-\frac{i}{\hbar}\hat{K}\delta t/2}\Psi = \mathcal{F}^{-1}\left[e^{-\frac{i}{\hbar}\left(-\frac{\hbar^2}{2m}\right)(-k^2)\delta t/2}\mathcal{F}\{\Psi\}\right] = \mathcal{F}^{-1}\left[e^{-\frac{ik^2\hbar\delta t}{4m}}\mathcal{F}\{\Psi\}\right] \quad (23)$$

Before writing the pseudo code let's write the wave function as a function of discrete  $i_j \in \{0, 1, \dots, N_j - 1\}$  coordinates where  $j \in \{x, y, z, t\}$  and  $N_j$  are the number of discrete grid points along each axis.

Now  $\vec{r}_x = i_x\delta x$ ,  $\vec{r}_y = i_y\delta y$ ,  $\vec{r}_z = i_z\delta z$  and  $t = i_t\delta t$  are the original spatial and time coordinates and  $\delta j$  is the resolution of the different dimensions respectively. With the new discretized coordinates the Schrödinger equation 15 can be written in the following form

$$i\hbar\frac{d}{dt}\Psi(i_x, i_y, i_z, i_t) = -\frac{\hbar^2}{2m}\Delta\Psi(i_x, i_y, i_z, i_t) + V(\vec{r})\Psi(i_x, i_y, i_z, i_t) \quad (24)$$

Having a discrete data set, Discrete Fourier transform (DFT) can be efficiently implemented using the Fast Fourier Transform (FFT) algorithm. The output of 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 which is the same as calculating the standard scalar product between  $\langle \Psi |$  and  $|\Psi\rangle$  vectors where  $\langle \Psi |$  is the transposed complex conjugate of  $|\Psi\rangle$  as seen in equation 25.

$$p(\vec{r}, t) = |\Psi(\vec{r}, t)|^2 = \langle \Psi(\vec{r}, t) | \Psi(\vec{r}, t) \rangle \quad (25)$$

Making use of formulas 18, 19 and 23 and plugging them into the formal solution of the Schrödinger equation we can create an algorithm for the time development of the wave function. We will use the  $\Psi$  to represent a complex valued tensor where the elements of this tensor can be obtained by using the  $i_j$  discrete indices. The algorithm can be written in the form of the following pseudo code

---

**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
for  $i \in [0, N_t)$  do
     $\Psi^{(1)} \leftarrow FFT^{-1} [P_K FFT [\Psi]]$ 
     $\Psi^{(2)} \leftarrow P_V \Psi^{(1)}$ 
     $\Psi \leftarrow FFT^{-1} [P_K FFT [\Psi^{(2)}]]$ 
    Visualize  $|\Psi|^2$ 
end for

```

---

Here  $P_K(i_x, i_y, i_z) \sim e^{-\frac{i\hbar\delta t}{4m}[(2\pi i_x/N_x/\delta x)^2 + (2\pi i_y/N_y/\delta y)^2 + (2\pi i_z/N_z/\delta z)^2]}$  is the kinetic energy propagator in momentum space and  $P_V(i_x, i_y, i_z) = e^{-\frac{i}{\hbar}V(i_x, i_y, i_z)\delta t}$  is the potential energy propagator in real space. We used the discretized coordinates to express the same operations as in 19 and 23. Please note that in the kinetic propagator's case it is important to follow the convention used by the specific FFT implementation. The formula written here is not entirely correct. The FFT used in our implementation puts the amplitude associated with the largest representable frequency of  $\frac{1}{2N}$  in the middle of the tensor. In the second half of the tensor for each axis the amplitudes are for the negative frequencies in ascending absolute value so that the largest index represents the  $-\frac{1}{N}$  frequency. This means that we had to modify the definition of  $P_K$  accordingly. One way to make the correction is to check whether  $i_j/N_j > \frac{1}{2}$  is true. If it is than modify it like  $i_j := N - i_j$ .

### 3.3 Defining Gaussian wave packets

In the algorithm first we have to specify an initial state for the wave function. Erwin Schrödinger introduced the concept of the Wave Packet (WP). A WP is a wave front 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) refer 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. This probability

density of this WP has Gaussian distribution [22], hence the name. The definition of such packet can be written in the following form

$$\psi(\vec{r}; a, \vec{r}_0, \vec{k}_0) = \left( \frac{2}{\pi a^2} \right)^{\frac{D}{4}} \cdot \exp(i \vec{k}_0 \cdot \vec{r}) \cdot \exp\left(-\frac{|\vec{r} - \vec{r}_0|^2}{a^2}\right) \quad (26)$$

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  $\Delta r = \frac{a}{2}$ .

### 3.4 A simple trick to optimize the algorithm

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  $n$ th iteration and 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  $\delta t$  interval –instead of the one used in Algorithm 1 calculated for  $\delta t/2$  interval– than do the backward FFT. The pseudo code of this optimized algorithm is written below

---

**Algorithm 2** Optimized time advance algorithm without visualization

---

```

 $\Psi \leftarrow$  initial state of the wave function
 $V \leftarrow$  localized potential
 $\delta t \leftarrow$  time resolution
 $N_t \leftarrow$  number of time steps
for  $i \in [0, N_t)$  do
     $\Psi^{(1)} \leftarrow FFT^{-1} [P_K^2 FFT [\Psi]]$ 
     $\Psi^{(2)} \leftarrow P_V \Psi^{(1)}$ 
end for

```

---

### 3.5 Free space boundary condition

When working with DFT it is important to take into account the boundary conditions of the transformed region. The DFT implies periodic boundary condition in which value at  $i_j = 0$  connects smoothly to values at  $i_j = 0$  for each of the respective  $j \in \{x, y, z\}$  dimensions. If the data transform does not satisfy this condition unrealistic artifacts arise. One technique to battle this requirement is to define a high potential wall on all the edges of the simulated volume so that the wave packet bounces back and never reaches the problematic edge. This however limits the possible simulation cases since we can not simulate scenarios where the wave packet would leave the volume. The wave packet reflected by the boundary potential interferes with itself thus potentially disturbing the observation of the object of simulation. In our implementation we used another approach called free space boundary condition. This fixes the issues arising both with the periodic boundary

condition and the potential box. We extend the simulated volume beyond the observed part and outside the visible box we create a draining potential. This draining potential gradually increases towards the edge of the simulated volume. As the name suggests it's effect on the wave packet is that it diminishes it's amplitude. By initializing a high enough draining potential the unrealistic artifacts on the edge of the simulated volume are practically negligible. In equation 19 we have seen that the potential propagator is a multiplication by a unit length complex number with phase proportional to  $V(\vec{r})$  localized potential. This normally only introduces a rotation on the complex plane. We can however incorporate the desired draining effect by introducing complex potential. Let's examine what happens if a complex number has a complex phase!

$$e^{i|z|e^{i\gamma}} = e^{i|z|[\cos(\gamma) + i\sin(\gamma)]} = e^{i|z|\cos(\gamma) - |z|\sin(\gamma)} = e^{i|z|\cos(\gamma)} e^{-|z|\sin(\gamma)} \quad (27)$$

In equation 27 We showed that by defining an imaginary potential we can reduce the magnitude of the wave function if we then apply the potential propagator. In our simulation we initialized the draining potential to be zero in the observed region and right from the corners of this box it begins to increase as a quadratic function of distance from the center of the simulated volume. It reaches it's maximum in the four corners of the simulated volume.

## 4 Our implementation

### 4.1 Short summary of the functionality of our simulator

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 resulting volumetric of probability density. The visualization requires the sampling of 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 [23].

### 4.2 Technologies in use

We choose Python [24] as our programming language because there is a waste amount of helpful tools implemented to use with it that specifically aim towards leveraging the difficulty of writing mathematical and physics related simulation. The majority of these tools are written in a hardware friendly languages such as Fortran, C or C++. However they come with an easy-to-use Application Programming Interface (API) that can be accessed from high level programming languages. In these modern languages such as Python one works on a higher abstraction level usually not dealing with memory management ore pointer arithmetic. Out implementation heavily depends on the NumPy library [25]. NumPy is a library that can be utilized to work efficiently with large arrays and tensors performing computationally intensive operations. A similar library is SciPy [26] also used in our program. For 3D visualization we choose to use VisPy [27]. This library provides a nice basic set of features to handle virtual camera and create scenes but also enables us to go deeper and write

our own GPU shader code. We made use of this to modify the default volume visualization code to fit our needs.

### 4.3 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 that to the already existing implementation used by Géza István Márk and Péter Vancsó. The Compute Unified Device Architecture (CUDA) toolkit is often used for parallel computational tasks implemented on the GPU [28]. It comes with a powerful GPU based FFT implementation. To use CUDA with Python we selected the CuPy wrapping library [29] 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 code than runs efficiently on the CPU. This feature is especially useful when iterating over large arrays. We utilize JIT when initializing the WP, localized potential and propagators. For other purposes we use multiple other libraries such as Imageio, Matplotlib, toml, Pillow, Keyboard, Colorama, Tqdm, PyQt5, PyOpenGL. The sources for these libraries can be found on the Internet. We have listed the required versions in a requirements.txt file in the projects GitHub repository.

### 4.4 About the program's user interface

In its current state our application has only a console interface and it saves the resulting images and videos into files. The reason that we so far haven't prioritized the development of a graphical user interface is that we think that terminal interfaces can still have their benefits even in 2023. An application that only requires a terminal to function is simpler thus more effort can be made to improve the core functionality. The audience of this software are scientists and engineers in the first place. This is especially true in the early stage of development that we are currently in. It already has however a snapshot system that makes it possible to interrupt a longer simulation and later resume from the exact state where it was previously halted. This turned out to be a very convenient feature although since we use GPU acceleration the simulation times are generally much shorter. Our design philosophy dictated to communicate as much information about the simulated wave function towards the user as possible. This direction can be thought of as controversial since too much data can obfuscate the important details especially in a plain console print. We try to battle this effect by using colorful prints and by saving the text into a log file so that the parameters can be found even after the simulation has finished.

To specify the parameters of a simulation we use a configuration file in Tom's Obvious Minimal Language (TOML) format [30]. The flexibility and universality of this data-format made it an overall good choice. We can use a single TOML file to set the resolution and dimensions of the simulated volume, specify the parameters of the wave packet, add an arbitrary number of various potential barriers. In this file we have parameters to configure the 3D visualization and others.

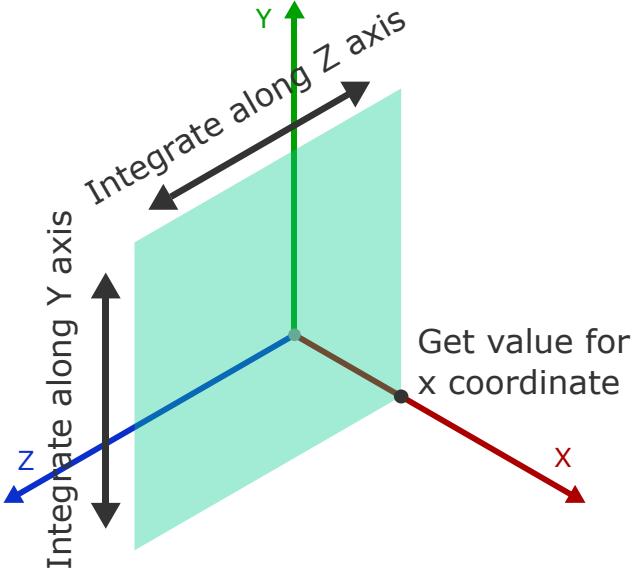


Figure 2: Calculation method of the *Per axis probability density* for a given  $x$  coordinate

#### 4.5 Generated output

As we have mentioned earlier the output of the program gets saved in files. The program creates images and videos and text files. The images can be thought of as higher resolution snapshots from the videos but we also create images that are not corresponding to any of the videos. Currently we generate five types of different images. We call these *Canvas probability*, *Canvas dwell time*, *Per axis probability density*, *Probability density 3D* and finally, *Probability evolution*. In each of these categories a sequence of images is being generated for each run of the simulation. The interval at which the state of the simulation is captured can be specified in the above described configuration file. *Canvas probability* is the  $\rho(\vec{r}, t)$  probability density in a specified plane intersecting the volume.  $\tau(\vec{r})$  *canvas dwell time* is the integral of the  $\rho(\vec{r}, t)$  probability density over  $t$  elapsed time in a specified plane intersecting the simulated volume. This can be formalized as

$$\tau(\vec{r}, t) = \int_0^t \rho(\vec{r}, \delta) d\delta \quad (28)$$

For now the measurement plane used in the above described methods has to be perpendicular to one of the three main axes to simplify the calculation. Both *dwell time* and *probability density* can be used to visualize interference patterns after a wave packet passes through some kind of a potential barrier such as a double-slit or an optical grid as we will show in Section 5. *Per axis probability density* gives a plot of integrated probability densities for each axis. To obtain the plot of  $\rho_X(x, t)$  per axis probability density we integrate the  $\rho(\vec{r}, t)$  where  $r_x = x$  and the other two coordinates run

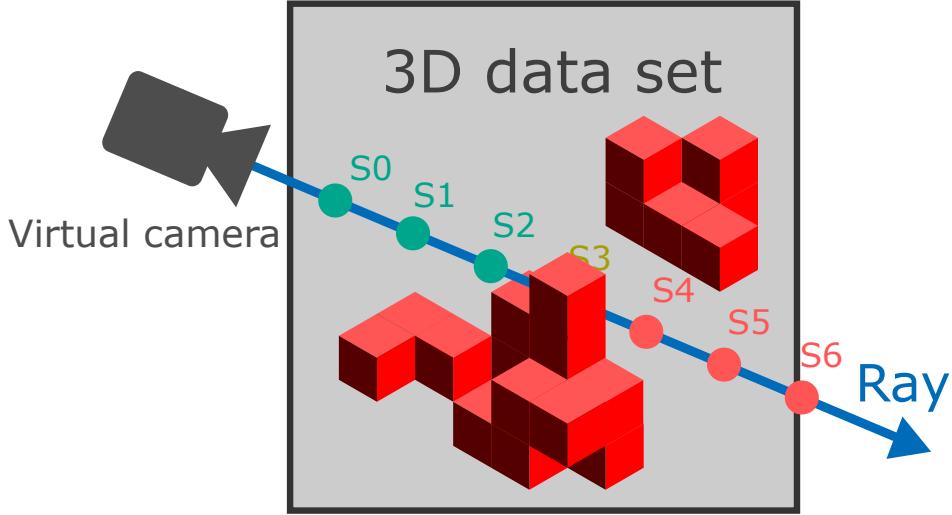


Figure 3: Visual explanation of the ray tracing method: we sample the 3D data set along rays shot from the virtual camera

across their domain. Equation 29 gives the formula for calculating this measurable for the  $X$  axis.

$$\rho_X(x, t) = \int_Y \int_Z \rho(\vec{r}, t) dy dz \quad (29)$$

Figure 2 attempts to provide further intuition. We also overlap the plot of the potential barriers on the same graph to understand the changes in propagation of the wave packet caused by the localized potential. This type of plot can provide useful information for most of the simulation cases, but it is especially useful when we want to simulate the behavior of three one dimensional particles in a three dimensional configuration space. In this case the projected probability density along each axis represents the wave packet of a different 1D particle. It is possible to initialize a potential that models the collision between these three particles. This creates the effect that the particles interact. In the configuration file there is a possibility to set whether a potential barrier should appear in the visualization. By disabling the visualization of the interaction potential we can get rid of any hardly conceptualizable elements of the resulting potential and maybe focus on a wall or a harmonic oscillator instead. Probability density 3D is the most self explanatory output of the system. Here we take the probability density as a 3D volumetric data set and visualize it using ray tracing. Ray tracing is the method where we conceptually shoot rays of light from the virtual camera through the visualized volume. Figure 3 explains the concept of ray tracing. For each pixel of the image there is exactly one ray. We progress along the ray in discrete steps and at each step we sample the data set. The data read from the volume has usually a single intensity value. We interpret the data using a transfer function that determines the color and opacity that should be associated with the intensity. The color and opacity is then combined along the ray resulting in the final color and opacity value for the pixel of the image. The operator that recursively combines the

color and opacity values is called under operator. For recursive calculation of the  $A_n$  opacity in  $n$ th step along the ray it can be written in the form of

$$A_n = A_{n-1} + \alpha_n(1 - A_{n-1}) \quad (30)$$

where  $A_{n-1}$  is the opacity calculated at the previous step and  $\alpha_n$  is the local opacity at the  $n$ th step. A similar equation could be written for red, green and blue (RGB) color channels. If we would only use the colors obtained by simply stepping through the sample points the look of the probability density would be rather dull. We also utilize approximated gradient vectors. One can calculate these vectors by taking the central difference using six additional samples around the currently sampled point. (We however deployed a better method described later.) The gradient vector points towards in the direction in which the probability density increases the fastest. If we make the analogy that a solid body objects has also got gradient vectors right on the edge of it's surface so that these vectors point towards the inside of the object perpendicular to the surface we can make the connection between gradient vectors and surface normals. Normal vectors of a surface are unit length vectors pointing towards the outside world from the surface and are also perpendicular to the surface. Normal vectors can be expressed as negated and normalized gradients. Normal vectors are often used in various surface shading models, since they give a good description how surfaces reflect light. Even tho our probability density has a much less clearly defined gradient this technique can still be used to obtain an approximated normal vector. If this vector is sufficiently defined (the magnitude of the gradient is not too close to zero) than we can use simple shading model such as the Blinn-Phong reflection model [31] to enhance our visuals. We even use the length information of the gradient to fade between the shaded and unshaded look. We would like to mention that in our previous works we have experimented with more physically plausible shading techniques such as applying the Cook-Torrance Bidirectional Reflectance Distribution Function (BRDF) [32]. This however has not yielded good results for volumetric visualization since surface imperfections are calculated into the model thus the generally fussy nature of volumetric datasets result in an overall dark look. The simple Blinn-Phong model however is cheap to compute, and provides sufficiently strong specular highlights while it's still easy to parameterize. For most of our simulations we use a total simulated volume of  $512 \times 512 \times 512$  data points. We leave the outer half of the total volume to accommodate the draining potential described in the last part of section 3. This means that so far we only visualize 50% of the stored data set:  $256 \times 256 \times 256$ . In the future we may experiment with reducing the space used up by the draining potential. Until than we rely heavily on good reconstruction filters to get rid of the aliasing effects visible by sampling the data set of relatively low resolution. We utilize Balázs Csébfalvi's reconstruction filter [23]. This method provides consistent gradient estimation that is the analytic gradient of the reconstructed function. It has an approximation order of three and it's also  $C^1$  continuous. This method evaluates eight trilinear samples to calculate the intensity and the gradient for a given point. In current generation of GPUs trilinear filtering can be efficiently evaluated using the built in sampling routines. The eight is only one more sample compared to the seven used in the less advanced intensity plus central difference technique and we also get better result as shown in the referred paper.

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

Figure 4: System specification of the used test hardware

The last type of the five different image outputs of our program we call *Probability evolution*. To create this output we divided the visible region of the simulated volume to separate partitions. On these images of this type we display a plot where we integrate the probability density over these regions following equation 31.

$$\text{IP}(t) = \int_V \rho(\vec{r}, t) d^3r \quad (31)$$

Conceptually this gives us the probability of the event of the particle being found in these spatial regions. It is also allowed to specify overlapping regions. By doing so we only get the probability of non excluding events. We prefer to create a measurement region for the entirety of the visible volume. This is very informative when the particle leaves the visible parts, hence the probability of the particle being found somewhere inside the visible part drops below one. Note that modeling such scenarios is made possible by using the draining potential.

Besides the sequence of images our program also renders two animations in MP4 format. One is created from a sequence of images rendered as in the *Probability density 3D* sequence. The other is the animated version of the *Per axis probability density* sequence. The rate at which the state of the simulation is captured as an animation frame and the playback frame rate can be configured separately.

## 4.6 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 4. 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 5. Here we tested three different configurations with varying resolution. We measured the average iteration count per second. The test shows that by using GPU acceleration we obtained significant speed up over the the CPU-only implementation.

## 5 Results

### 5.1 General approach and used unit system

We used our software to perform various WPD simulations. In this section we present the results of some of these simulations. We will shortly introduce the simulated scenarios. We also want

Input size	CPU-only Avg. [iter/s]	GPU accelerated Avg. [iter/s]	Avg. speed up
$128 \times 128 \times 128$	1.1	11.5	$\times 10.45$
$256 \times 256 \times 256$	0.09	6.5	$\times 72.22$
$512 \times 512 \times 512$	0.01	0.5	$\times 50.00$

Figure 5: Results of a performance test using a CPU-only and a GPU accelerated version of the application averaged

to provide some formulas to analytically validate the correctness of some of the simpler scenarios. Our simulator uses the Hartree atomic units [33]. Every quantity in the upcoming part should be interpreted as such. This unit system makes it convenient to deal with quantities at atomic scale. You can find animations showing the time evolution of WPs simulated by our software on <https://zoltansimon.info/src/content/research/wavepacketsim.html>.

## 5.2 Double-slit experiment

First we would like to present the simulation results of an electron scattering experiment. 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 specifically initialized 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 back. The portion of the WP that passed through –suffering scattering– continues forward and consequently arrives to a measuring device<sup>3</sup>. 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. According to the Huygens–Fresnel principle in scattering experiments we can approximate the wave function after the barrier as concentric waves emitted from the slits or holes of the barrier. This gives a simple enough model to predict the shape of interference pattern forming on the canvas. We would like to validate the distance between the stripes in the interference pattern. To calculate the distance between the probability density peaks first we have to get a formula for the intensity on the canvas.

$$I(\vec{r}) = I_0 \frac{\sin^2\left(\pi N \frac{dy}{\lambda L}\right)}{\sin^2\left(\pi \frac{dy}{\lambda L}\right)} \quad (32)$$

where  $N$  is the number of holes on the barrier (in double-slit experiment 2),  $d$  is the distance between the holes,  $L$  is the distance between the canvas and the barrier  $\lambda$  is the wavelength of the wave and  $y$  is the location on the canvas. We performed the simulation using  $L = 30$  Bohr radii,  $d = 2.5$  Bohr

---

<sup>3</sup>In scattering and diffraction experiments we can make the distinction between a near field and far field solution.

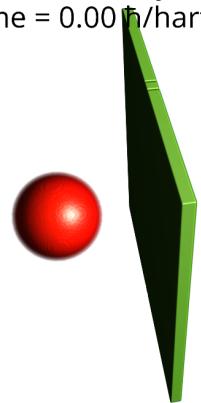
radii,  $\lambda = \frac{2\pi}{3} \simeq 2.1$  Bohr radii with two slits. The width of each slit was a small enough value of 0.5 Bohr radii. Different stages of double-slit simulation can be seen in figure 6 where we have used ray tracing to visualize the probability density and the potential. The interference pattern is visualized in figure 7 on a canvas of size  $60 \times 60$  Bohr radius. We compared the simulated pattern and the pattern predicted by the Huygens–Fresnel principle. It can be seen that that the two patterns are very similar.

### 5.3 Diffraction by optical grating-like potential

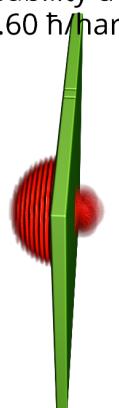
Many different forms of diffraction can be explained using QM. The scale at which diffraction happens ranges from the scale of subatomic particles to larger molecules. Measuring diffraction patterns is a very useful tool in the hand of scientists. It provides information about the object that caused the diffraction. At this point it is probably important to emphasize that the diffraction of light is a different phenomenon from the diffraction of matter waves since light is an electromagnetic wave. With that said, the fact that particles are not electromagnetic waves but still exhibit wave-like behavior as discussed in section 2 makes this experiment even more interesting. Note that in equation 32 the  $y$  is a one dimensional coordinate. Indeed the double-slit experiment can be fully described using only two spatial dimensions and the diffraction pattern forms in a single dimension perpendicular to the propagation direction of the wave. This is because the localized potential in this scenario is independent of the  $z$  coordinate. The make use of all three simulated dimensions we also modeled diffraction on diffraction grating. In optics, a diffraction grating is a periodic 2D structure that diffracts light [34]. In QM such 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. We experimented with various lattice constants (distance between adjacent grid points). Here we present two simulations. In the first simulation 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. In the second simulation each node has a Gaussian potential distribution and a maximal potential of  $V_{max} = 25$  hartree. The distance between adjacent grid points is  $d = 8$  Bohr radii. The canvas distance is  $L = 30$  Bohr radii and the wavelength of the WP is  $\lambda \simeq 2.1$  Bohr radii for both simulations. Note in both cases the kinetic energy of the WP  $E = \frac{p^2}{2m} = \frac{\hbar^2}{2\lambda^2} \simeq 4.5$  hartree is less than  $V_{max}$ . Otherwise the grating wouldn't impact sufficiently the propagation of the WP. In figure 8 we visualized different stages of the time development of the first simulation and 9 showcases stages of the second simulation. Both sequences of rendered images make use of the ray tracing visualization method.

During the simulation many interesting interference patterns arise. We showcase some of these for the 4 Bohr radius lattice constant case in figure 10 and for the 8 Bohr radius lattice constant case in figure 11.

Probability density  
(Elapsed time = 0.00 h/hartree = 0.00E+00 ns)



Probability density  
(Elapsed time = 6.60 h/hartree = 1.60E-07 ns)



Probability density  
(Elapsed time = 11.40 h/hartree = 2.76E-07 ns)

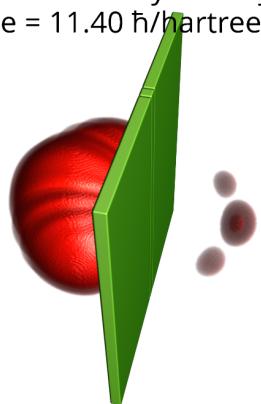


Figure 6: Stages of the double-slit experiment: before, during and after passing through the double-slit

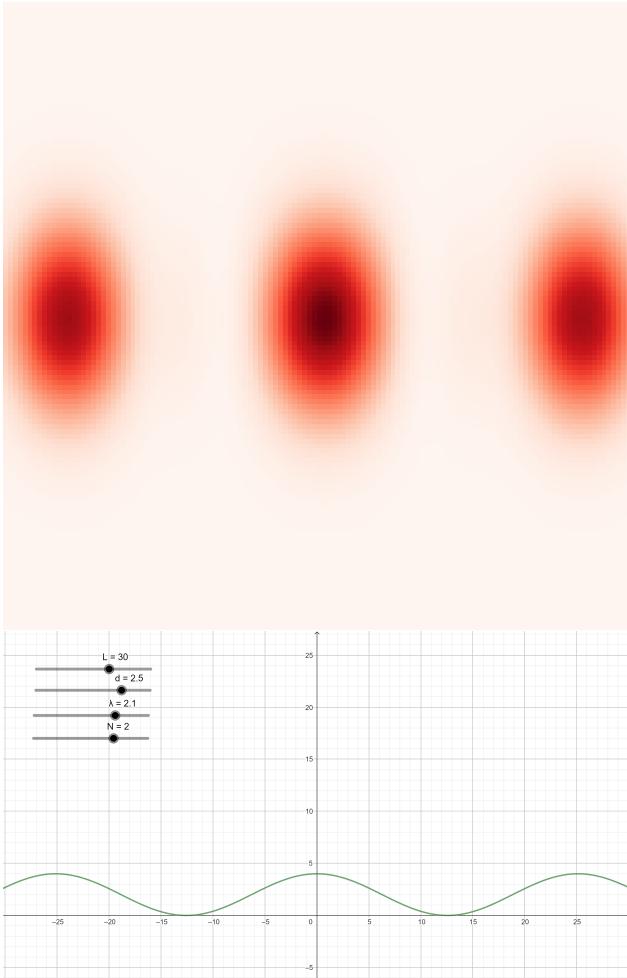
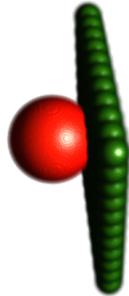


Figure 7: Comparison of the simulated (above) and the analytically predicted (below) interference pattern during double-slit experiment

Probability density  
(Elapsed time = 1.80  $\hbar/\text{hartree}$  = 4.35E-08 ns)



Probability density  
(Elapsed time = 6.00  $\hbar/\text{hartree}$  = 1.45E-07 ns)



Probability density  
(Elapsed time = 10.50  $\hbar/\text{hartree}$  = 2.54E-07 ns)

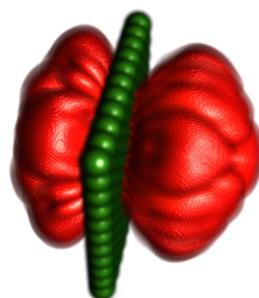
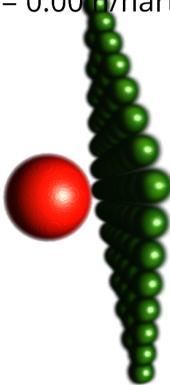


Figure 8: Stages of the optical grid experiment using a grid with lattice constants of 4 Bohr radii: during, right after and later after passing through the grid

Probability density  
(Elapsed time = 0.00  $\hbar/\text{hartree}$  = 0.00E+00 ns)



Probability density  
(Elapsed time = 5.40  $\hbar/\text{hartree}$  = 1.31E-07 ns)



Probability density  
(Elapsed time = 12.30  $\hbar/\text{hartree}$  = 2.98E-07 ns)

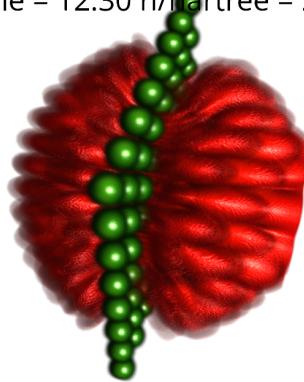


Figure 9: Stages of the optical grid experiment using a grid with lattice constants of 8 Bohr radii: during, right after and later after passing through the grid

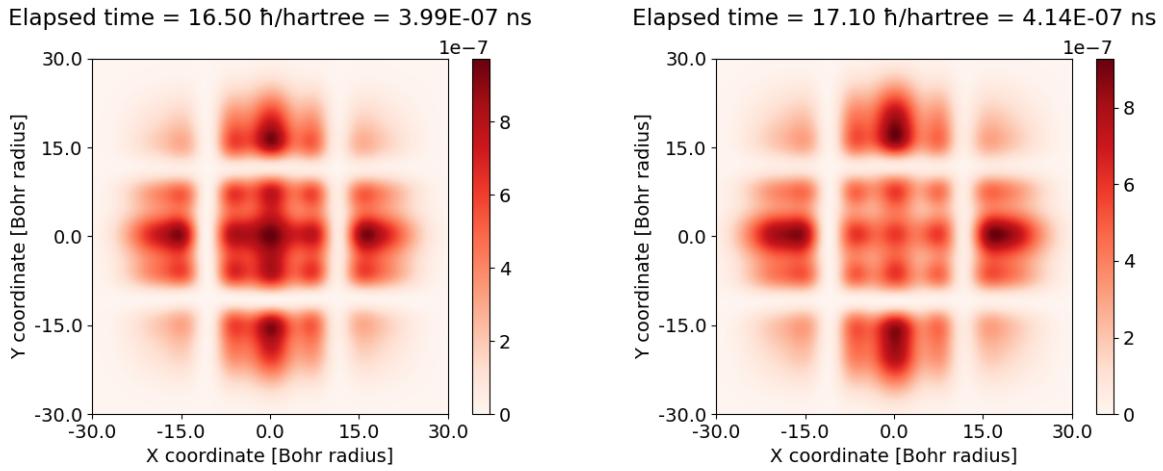


Figure 10: Two stages of interference patterns forming on the measurement canvas during optical grid simulation using lattice constant of 4 Bohr radii

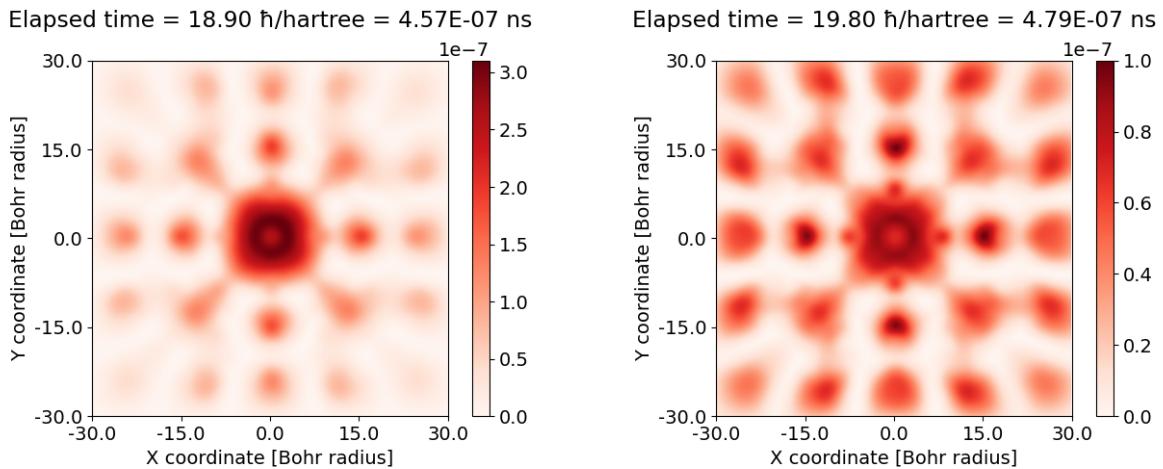


Figure 11: Two stages of interference patterns forming on the measurement canvas during optical grid simulation using lattice constant of 8 Bohr radii

## 5.4 Simulating 1D particles in 3D configuration space

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 special interaction potential. To create such potential we have to think about the coordinates in the higher dimensional configuration space as the coordinates of the lower dimensional space describing the location of the lower dimensional particle. 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  $C$  than we can reinterpret this function as the  $V(\vec{r})$  localized potential function used in the potential propagator in equation 19. Note that here  $\vec{r}$  becomes  $(x_a, x_b, x_c)$ . To model the interaction between the three 1D particles we initialized a linear combination of two types of interaction potentials. One is proportional to  $\frac{1}{|x_i - x_j|}$  where  $i, j \in \{a, b, c\}$  and  $i \neq j$  and the other is a hard interaction that takes it's maximum if the particles approach each other inside an  $\epsilon$  radius otherwise it's zero. To prevent blotting of the Gaussian WP we also initialized 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 following form

$$V(x) = \frac{m\omega^2}{2}x^2 \quad (33)$$

where  $m$  is the oscillating weight and  $\omega$  is the angular velocity of the oscillation. First we created a scenario where particle  $A$  starts at 25 Bohr radii away from the center of the oscillator where the potential energy is maximal thus it accelerates towards the other two particles consequently transferring the momentum to the particle on the far right. For this experiment the interaction potential consists only from the hard interaction. The angular velocity of the oscillator is  $\omega = \frac{2\pi}{44} \simeq 0.1415 \frac{\text{rad-hartree}}{\hbar}$ . After the collision particle  $C$  propagates towards the maximal potential of the oscillator. Particle  $B$  stays in place since it served only as a medium in the momentum transfer. The momentum gained from particle  $A$  was immediately given to particle  $C$ . When all of it's kinetic energy is transferred to potential energy it turns back. The oscillation with the momentum transfer between the particles continues. Different phases of this simulation can be observed on the per axis visualization in figure 12. Please note that particle  $C$  converts all of its kinetic energy to potential energy right at the half of the first period of the oscillation  $t = 44/2\frac{\hbar}{\text{hartree}} \simeq 22\frac{\hbar}{\text{hartree}}$ . This means that the harmonic oscillator functions correctly.

Now let's repeat this experiment but now we place a finite potential barrier in the middle of the oscillator. The probability of a particle with  $E$  energy tunneling through a  $d$  thick barrier of  $V_0$  potential can be expressed in the following form

$$\text{IP}_{tunnel} = \frac{1}{1 + \frac{V_0^2}{4E(V_0-E)} \sinh^2(\kappa d)} \quad (34)$$

where  $\kappa = \sqrt{2m(V_0 - E)}/\hbar$  is determined by the energy of the incoming WP. Energy is conserved so  $E = V(x_A)$ . We solve for  $\text{IP}_{tunnel} = \frac{1}{2}$ . One possible solution is  $V_0 \simeq 8.5$  hartree and  $d \simeq 0.3$

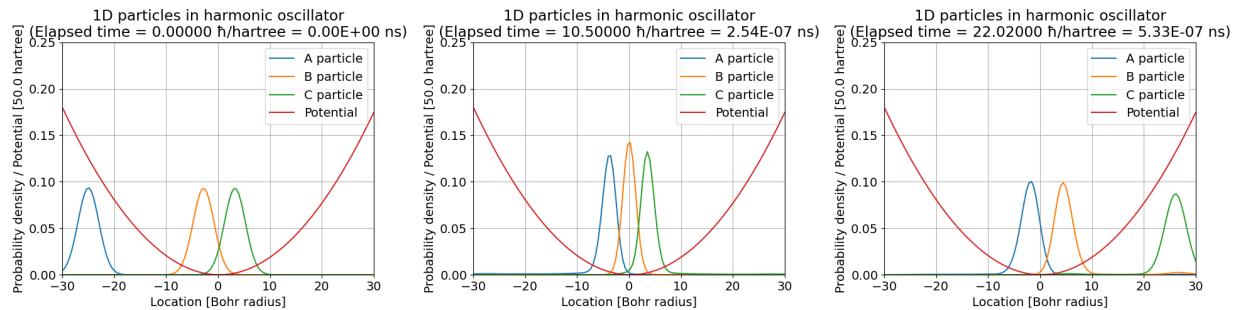


Figure 12: Stages of the interactions between 1D particles in harmonic oscillator: initial state, particle A giving momentum to particle C, particle C reaching maximal potential and turning back

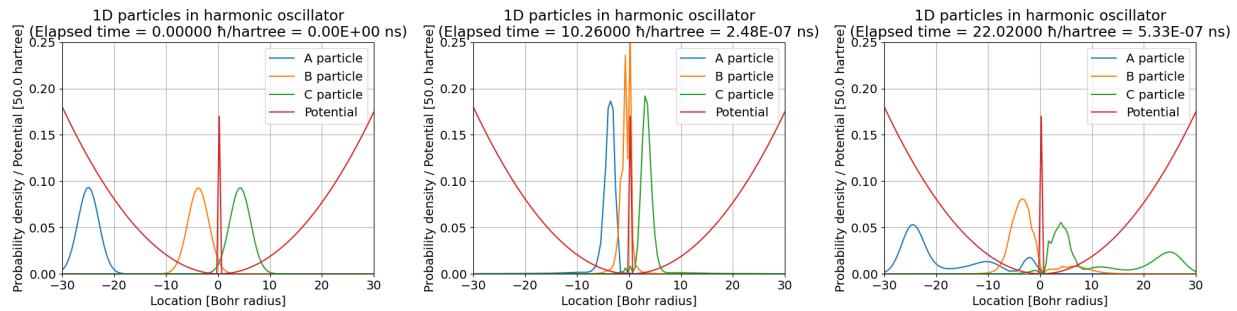


Figure 13: Stages of the interactions between 1D particles in harmonic oscillator: initial state, particle A giving momentum to particle C, particle C reaching maximal potential and turning back

Bohr radius. Particle  $A$  gives it's momentum to  $B$  as before. With these parameters approximately the half of the probability density of particle  $B$  tunnels through the barrier giving it's momentum to particle  $B$  on the next side of the wall. This causes the  $C$  to start moving with probability of approximately  $\frac{1}{2}$ . This can be observed in figure 13.

## 5.5 Modeling a flash memory cell

Our last group of simulations try to model the working of a flash memory cell. The flash memory is used extensively in many modern electronic devices such as smart phones, laptops or tablets. The general structure of a cell is visualized in figure 14. A flash memory stores information by trapping electrons on a floating gate. The writing and flushing of the a flash cell is performed by inducing a strong enough electric field that the electron can tunnel through the potential barrier between the floating gate and the channel. The reading of a memory cell is done by inducing current through the the channel between the source and the drain. If there are electrons trapped on the floating gate they create a Coulomb potential shifting the potential in the substrate allowing or denying the

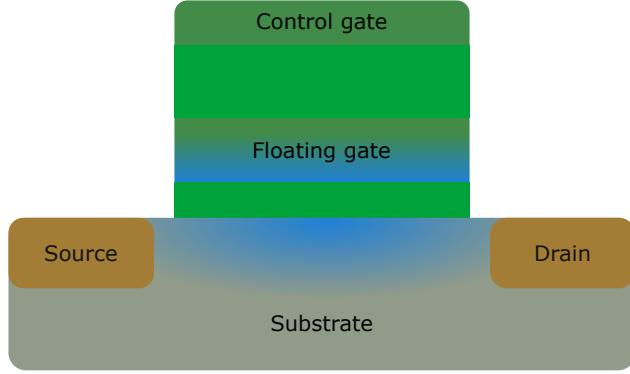


Figure 14: Structure of a flash memory cell

flow of electrons between the drain and the source thus the resistance changes and the information about the state of the floating gate was successfully retrieved. In figure 14 we used brighter green for the potential barriers. It can be seen that between the control gate and the floating gate there is a thicker potential barrier. This prevents the flow of electrons between these two parts. Only the Coulomb potential induced by the electrons on the control gate must be felt by the electrons on the floating gate. In order to the writing and flushing to be possible the thickness and height of the potential wall between the floating gate and the substrate must be chosen carefully. This barrier must be thick enough to prevent unwanted tunneling when the control gate is empty. After the control gate fills with electrons the Coulomb potential created by those electrons makes an overall slope in the potential. This slope adds to the potential wall thus changing its shape. The previously rectangular barrier now acquired a triangular outline. More importantly the  $d$  length of the barrier gets reduced. This increases the probability of the electron in floating gate to tunnel out from the gate. This is sound with equation 34 for the probability of tunneling. This phenomena is called Fowler–Nordheim tunneling [35]. In our first flash memory related simulation we have initialized a wall that represents the barrier between the floating gate and the substrate. Our simulation deals with only one electron. This initialize the location of this electron to be inside the floating gate. When we want to simulate the flushing of the cell, we add a Coulomb potential the decreases towards the barrier. Create initialized the barrier to have a height of  $V_0 = 10$  hartree and width of  $d = 4.0$  Bohr radius. We surrounded the floating gate area with thick potential barriers from the remaining five directions to prevent tunneling in these directions but excluded these from the visualization to make the electron's probability density visible from outside. Coulomb potential induced by a negatively charged point charge  $Q$  acting on an electron can be written like

$$V_E = \frac{1}{4\pi\epsilon_0} \frac{Q}{r} \quad (35)$$

To make the simulation more stable we approximated this potential with a linear gradient decreasing as we move farther from the control gate. This way we were able to precisely configure the gradient necessary for Fowler–Nordheim tunneling without creating extremely high potential near the control

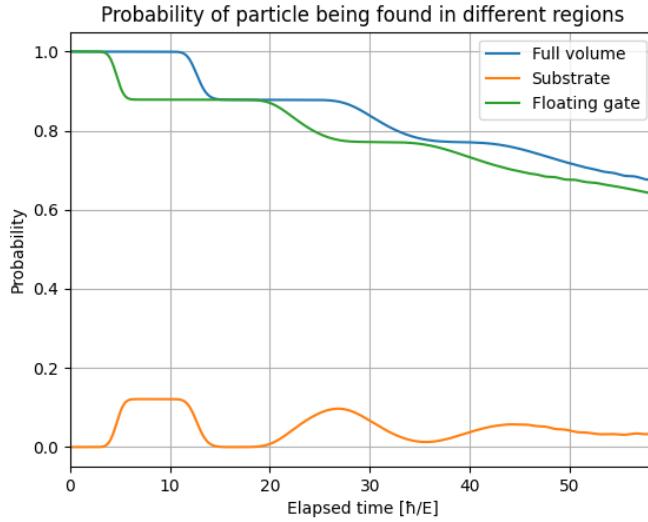


Figure 15: Flushing of the floating gate

gate. We initialized this gradient potential to be zero at the control gate and decrease to negative values with a constant slope of  $\frac{dV}{dx} = -1 \frac{\text{hartree}}{\text{Bohr radius}}$ . We run the simulation with the Coulomb potential turned on and off and compared the probability evolution plots. Figure 15 shows the flushing of the floating gate with the Coulomb potential enabled. The probability of the electron being found on the gate is rapidly decreasing. In figure 16 however the the electron stays much longer on the gate since here we disabled the potential slope. This shows that by changing the electric field we can control the writing and flushing of the floating gate.

## 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 Márks 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 implemented the draining potential technique to allow longer simulation scenarios without the forming of unrealistic reflections and interference patterns.

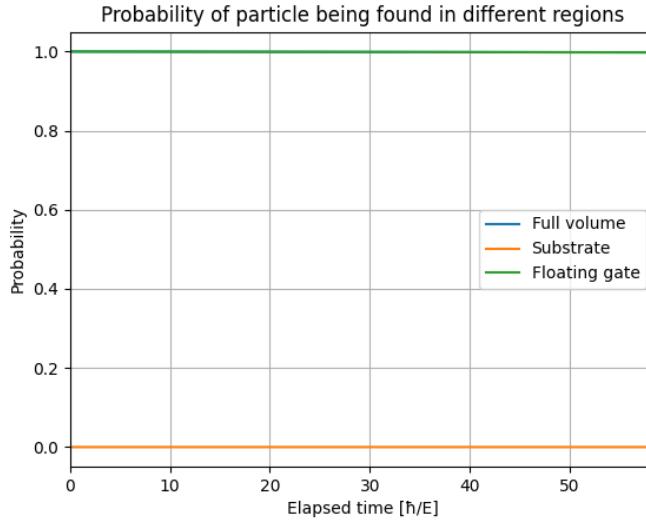


Figure 16: Storing the electron on the floating gate over a longer period of time

- We combined state of the art volume visualization techniques to enhance the visual quality of the resulting probability density images.
- We used multiple alternatives to visualize the probability densities including *canvas probability density* good for visualizing diffraction the *per axis* approach that is handy when we do configuration space simulations or the *probability evolution* used in the flash memory simulation.
- We used our simulator software to run various Wave Packet Dynamical simulations ranging from basic diffraction scenarios, through simulation of lower dimensional particles in configuration space up to a more advanced case of simulating the writing and flushing sequence of a flash memory cell. For some of these scenarios we provided analytical validation methods such as the one for the interference pattern of the double-slit experiment or the easily testable periodic time of the harmonic oscillator or even the probability of the wave packet tunneling through barriers.

We see our work as a successful entry into the world of quantum mechanical wave packet dynamics and a definitely good starting point for further research. We believe that the vast amount of application of quantum mechanics speaks for itself when the question is whether this research is important or not. Another motivation to do wave packet dynamical simulations is the Nobel prize winning research of Dr. Ferenc Krausz. Attosecond physics opens a new frontier in understanding our universe. Doing computer simulations and comparing the results with laboratory experiments can be a powerful strategy to make new discoveries. We already have multiple ideas how to improve and append the current method. One obvious direction for development is to improve the

user interface and to create a graphical interface besides the current terminal based interface. In the future we want to make possible to calculate the eigenstates of the localized potential. This would require to calculate the Fourier transform in time domain to obtain the energy state of the system. Then iteratively converge towards the eigenstate. There is also a possibility to incorporate electromagnetism into the Hamiltonian operator. Making this work would be also a very interesting project. From visualization point of view there are also many possibilities to improve. As we have seen in this work there is room for even better reconstruction filters and clever ways to make use of the limited resolution. We are very hopeful about the future research potential of this topic and are very eager to continue the fruitful work.

## 7 Acknowledgement

I would like to thank my advisor, Dr. Balázs Csébfalvi, who has always been very supportive and open towards my silly ideas. One such ideas of mine was to begin to work on wave packet simulation. I am immensely grateful for the work of Dr. Géza István Márk and Dr. Péter Vancsó as my supervisors and as first hand sources about wave packet dynamics. Without their help this work wouldn't exist. I would like to mention Dr. János Asbóth who recommended to look into the work of Dr. Géza I. Márk's work. Basically he gave the first quantitative impulse to this research.

## Acronyms

**API** Application Programming Interface. 13

**BRDF** Bidirectional Reflectance Distribution Function. 15

**CPU** Central Processing Unit. 4, 13, 16

**CUDA** Compute Unified Device Architecture. 13

**DFT** Discrete Fourier transform. 11, 12

**FDTD** Finite Difference in Time Domain. 9

**FDTD-Q** FDTD technique for the analysis of quantum devices. 9

**FFT** Fast Fourier Transform. 3, 9, 11, 13

**GPU** Graphical Processing Unit. 4, 9, 13, 14, 16

**GR** General Relativity. 3

**JIT** Just In Time compilation. 13

**QM** Quantum Mechanics. 3, 4, 7, 12, 17

**RK-HO-FDTD** Runge-Kutta High-Order Finite Difference in Time Domain. 9

**TOML** Tom's Obvious Minimal Language. 14

**WP** Wave Packet. 12, 13

**WPD** Wave Packet Dynamics. 12, 16

## References

- [1] 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*, vol. 52, no. 1, pp. 35–53, 1983. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0021999183900153>
- [2] G. I. Márk, “Web-schrödinger: Program for the interactive solution of the time dependent and stationary two dimensional (2d) schrödinger equation,” 2020. [Online]. Available: <https://doi.org/10.48550/arXiv.2004.10046>
- [3] G. Tamás, *Kvantummechanika*, ser. Elméleti fizika. Typotex, 2007. [Online]. Available: [https://www.typotex.hu/book/343/geszti\\_tamas\\_kvantummechanika](https://www.typotex.hu/book/343/geszti_tamas_kvantummechanika)
- [4] R. M. Wald, *General relativity*. University of Chicago press, 2010.
- [5] I. Ross, “The invention of the transistor,” *Proceedings of the IEEE*, vol. 86, no. 1, pp. 7–28, 1998.
- [6] H. L. Zhuang and R. G. Hennig, “Computational discovery, characterization, and design of single-layer materials,” *JOM*, vol. 66, no. 3, pp. 366–374, Mar 2014. [Online]. Available: <https://doi.org/10.1007/s11837-014-0885-3>
- [7] A. K. Geim, “Graphene: Status and prospects,” *Science*, vol. 324, no. 5934, pp. 1530–1534, 2009. [Online]. Available: <https://www.science.org/doi/abs/10.1126/science.1158877>
- [8] M. D. Stoller, S. Park, Y. Zhu, J. An, and R. S. Ruoff, “Graphene-based ultracapacitors,” *Nano Letters*, vol. 8, no. 10, pp. 3498–3502, 2008, pMID: 18788793. [Online]. Available: <https://doi.org/10.1021/nl802558y>

- [9] C. N. R. Rao, K. Biswas, K. S. Subrahmanyam, and A. Govindaraj, “Graphene, the new nanocarbon,” *J. Mater. Chem.*, vol. 19, pp. 2457–2469, 2009. [Online]. Available: <http://dx.doi.org/10.1039/B815239J>
- [10] S. Imre, *Quantum Computing and Communications: An Engineering Approach*. John Wiley Sons, Ltd, 2004. [Online]. Available: <https://onlinelibrary.wiley.com/doi/book/10.1002/9780470869048>
- [11] E. Schrödinger, “An undulatory theory of the mechanics of atoms and molecules,” *Phys. Rev.*, vol. 28, pp. 1049–1070, Dec 1926. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRev.28.1049>
- [12] P. Vancsó, G. I. Márk, P. Lambin, A. Mayer, Y.-S. Kim, C. Hwang, and L. P. Biró, “Electronic transport through ordered and disordered graphene grain boundaries,” *Carbon*, vol. 64, pp. 101–110, 2013. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0008622313006751>
- [13] G. Mark, P. Vancsó, L. Biro, D. Kvashnin, L. Chernozatonskii, A. Chaves, K. Rakhimov, and P. Lambin, *Wave Packet Dynamical Calculations for Carbon Nanostructures*, 01 2016, pp. 89–102.
- [14] C. Cohen-Tannoudji, B. Diu, and F. Laloë, *Quantum Mechanics, Volume 1: Basic Concepts, Tools, and Applications*. John Wiley Sons, Ltd, 2019. [Online]. Available: <https://www.wiley.com/en-ie/Quantum+Mechanics%2C+Volume+1%3A+Basic+Concepts%2C+Tools%2C+and+Applications%2C+2nd+Edition-p-9783527822713>
- [15] W. R. Hamilton, *On a general method of expressing the paths of light, & of the planets, by the coefficients of a characteristic function*. Dublin: Printed by P.D. Hardy Dublin, 1833.
- [16] J. C. Maxwell, “A dynamical theory of the electromagnetic field,” *Philosophical Transactions of the Royal Society of London*, vol. 155, pp. 459–512, 1865. [Online]. Available: <http://www.jstor.org/stable/108892>
- [17] U. Andersson, “Time-domain methods for the maxwell equations,” 04 2001.
- [18] A. Soriano, E. A. Navarro, J. A. Portí, and V. Such, “Analysis of the finite difference time domain technique to solve the Schrödinger equation for quantum devices,” *Journal of Applied Physics*, vol. 95, no. 12, pp. 8011–8018, 06 2004. [Online]. Available: <https://doi.org/10.1063/1.1753661>
- [19] M. Zhu, Q. Cao, and L. Zhao, “Study and analysis of a novel runge–kutta high-order finite-difference time-domain method,” *IET Microwaves, Antennas & Propagation*, vol. 8, no. 12, pp. 951–958, 2014. [Online]. Available: <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-map.2013.0650>

- [20] M. Zhu and Y. Wang, “Rk-ho-fdtd scheme for solving time-dependent schrodinger equation,” *The Applied Computational Electromagnetics Society Journal (ACES)*, vol. 36, no. 08, p. 968–972, Oct. 2021. [Online]. Available: <https://journals.riverpublishers.com/index.php/ACES/article/view/11753>
- [21] F. I. Moxley, T. Byrnes, F. Fujiwara, and W. Dai, “A generalized finite-difference time-domain quantum method for the n-body interacting hamiltonian,” *Computer Physics Communications*, vol. 183, no. 11, pp. 2434–2440, 2012. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S001046551200224X>
- [22] X. Zhang, *Gaussian Distribution*. Boston, MA: Springer US, 2010, pp. 425–428. [Online]. Available: [https://doi.org/10.1007/978-0-387-30164-8\\_323](https://doi.org/10.1007/978-0-387-30164-8_323)
- [23] B. 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*, vol. 42, no. 2, pp. 191–200, 2023. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14753>
- [24] G. Van Rossum and F. L. Drake Jr, *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.
- [25] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, “Array programming with NumPy,” *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. [Online]. Available: <https://doi.org/10.1038/s41586-020-2649-2>
- [26] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python,” *Nature Methods*, vol. 17, pp. 261–272, 2020.
- [27] (2023) Vispy. [Online]. Available: <https://zenodo.org/record/4321173>
- [28] J. Nickolls, I. Buck, M. Garland, and K. Skadron, “Scalable parallel programming with cuda,” in *ACM SIGGRAPH 2008 Classes*, ser. SIGGRAPH ’08. New York, NY, USA: Association for Computing Machinery, 2008. [Online]. Available: <https://doi.org/10.1145/1401132.1401152>
- [29] R. Okuta, Y. Unno, D. Nishino, S. Hido, and C. 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. [Online]. Available: [http://learningsys.org/nips17/assets/papers/paper\\_16.pdf](http://learningsys.org/nips17/assets/papers/paper_16.pdf)
- [30] T. Preston-Werner. Tom’s obvious minimal language. [Online]. Available: <https://toml.io/en/>
  - [31] J. F. Blinn, “Models of light reflection for computer synthesized pictures,” in *Proceedings of the 4th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH ’77. New York, NY, USA: Association for Computing Machinery, 1977, p. 192–198. [Online]. Available: <https://doi.org/10.1145/563858.563893>
  - [32] R. L. Cook and K. E. Torrance, “A reflectance model for computer graphics,” *ACM Trans. Graph.*, vol. 1, no. 1, p. 7–24, jan 1982. [Online]. Available: <https://doi.org/10.1145/357290.357293>
  - [33] 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*, vol. 24, no. 1, p. 89–110, 1928.
  - [34] G. W. Stroke, *Diffraction Gratings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1967, pp. 426–754. [Online]. Available: [https://doi.org/10.1007/978-3-642-46077-7\\_8](https://doi.org/10.1007/978-3-642-46077-7_8)
  - [35] R. H. Fowler and L. Nordheim, “Electron emission in intense electric fields,” *Proc. Roy. Soc. Lond. A*, vol. 119, pp. 173–181, 1928.